PKI Uncovered

Certificate-Based Security Solutions for Next-Generation Networks

Andre Karamanian
Srinivas Tenneti
Francois Dessart

ciscopress.com

PKI Uncovered

Andre Karamanian

Srinivas Tenneti

Francois Dessart

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing February 2011

Library of Congress Cataloging-in-Publication Data:

Karamanian, Andre.

PKI uncovered / Andre Karamanian, Srinivas Tenneti, Francois Dessart.

p. cm.

Includes index.

ISBN-13: 978-1-58705-916-2 (pbk.)

ISBN-10: 1-58705-916-9 (pbk.)

1. Public key infrastructure (Computer security) 2. Computers—Access control. 3. Computer networks—Security measures. I. Tenneti, Srinivas. II. Dessart, Francois. III. Title.

QA76.9.A25K346 2011

005.8—dc22

2011002835

ISBN-13: 978-1-58705-916-2

ISBN-10: 1-58705-916-9

Warning and Disclaimer

This book is designed to provide information about public key infrastructure. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: U.S. **Corporate and Government Sales** 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States, please contact: **International Sales** international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Project Editor: Seth Kerney

Editorial Assistant: Vanessa Evans

Book Designer: Louisa Adair

Indexer: Tim Wright

Business Operation Manager, Cisco Press: Anand Sundaram

Manager Global Certification: Erik Ullanderson

Development Editor: Kimberley Debus

Copy Editor: Apostrophe Editing Services

Technical Editor: Alex Teichmann

Proofreader: Sheri Cain

Composition: Mark Shirar

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Authors

Andre Karamanian, CCIE R/S No. 10228, attended Capitol College where he received his master's degree in network security and where he is currently a doctoral student in information assurance. He is currently a security consultant at Cisco. He has worked in the field of security for approximately 11 years. Before he came to Cisco, Andre worked as a security leader at a large service provider for its large custom clients. He is highly credentialed with many industry certifications and has been a presenter at Networkers at Cisco Live for two years.

Srinivas Tenneti, CCIE R/S, Security, No. 10483, is currently working as an Enterprise systems engineer at Cisco. He has published design guides, white papers, and presentations on end-to-end solutions for enterprise and commercial customers. He also worked with several service providers to validate their network designs and architectures. Before he came to Cisco, he worked as a network specialist for a large service provider where he designed WANs for enterprise customers.

Francois Dessart, CCIE Security No. 15962, is currently a security consultant at Cisco. Before joining the European Advanced Services organization, he spent 4 years in the Security TAC in Brussels, solving complex PKI and VPN issues for Cisco customers. Francois has a master's degree in electrical engineering from Université Catholique de Louvain and recently received his master's degree in management from the Louvain School of Management.

About the Technical Reviewers

Alex Teichmann is a consultant for Cisco. He has helped developed leading practices for PKI and has personally worked on several IPsec and PKI deployments with great success and accolades. Alex Teichmann has an unmatched knowledge of PKI and is a leader in the field.

Piotr Jarzynka, CCIE R/S, Security, No.4737, is a Solutions Architect at Cisco. He is currently focusing on the security of Unified Communications (UC) for which he has developed a complete services portfolio, helping organizations to secure their UC environment. He has also created leading practices for the application of PKI within UC and has worked on several large customer implementations.

Dedications

Andre Karamanian: To my wife and family.

Srinivas Tenneti: To my wife, children, and my parents.

Francois Dessart: To my wife Anne-Sophie and my son Grégoire.

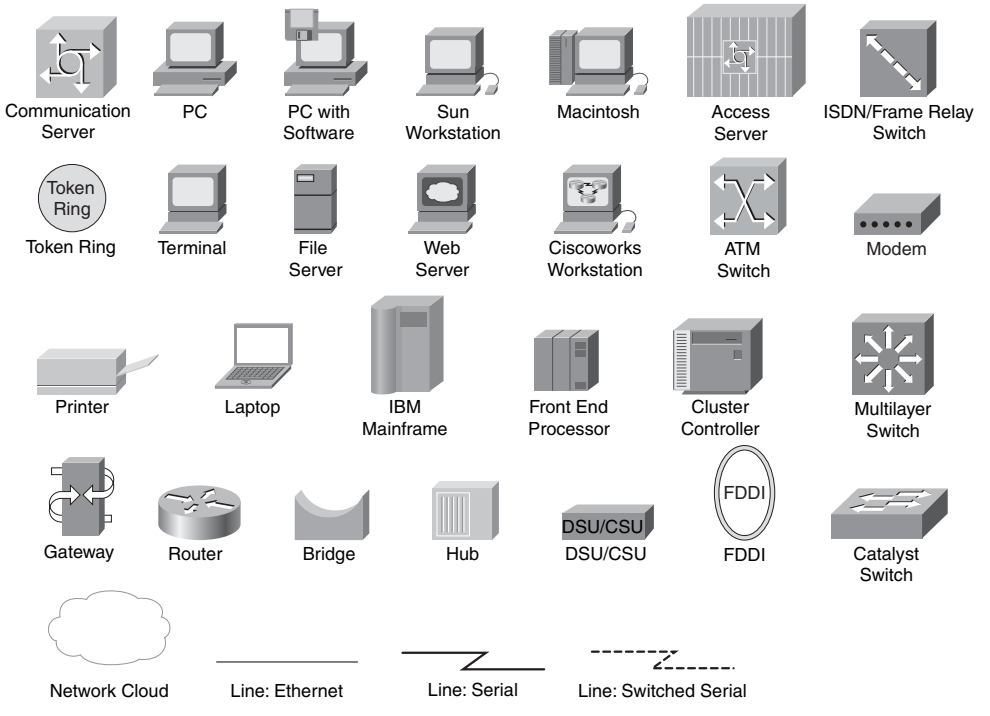
Acknowledgments

We'd like to give special recognition to Alex Teichmann for providing his expert technical knowledge in editing this book. He's also been as good a colleague as anyone could hope to have.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands manually input by the user (such as a show command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Introduction

With the increasing focus on IT Security comes a higher demand for identity management in the modern business. This requires a flexible, scalable, and secure authentication method. Identity control is made mandatory by many public standards, such as PCI, and PKI is an essential component to set up authentication in many technologies, such as VPN. Public Key Infrastructure (PKI) plays a key role in achieving the required degree of security and scalability. Other approaches have been either scalable but not secure, or secure but not scalable. Not only does PKI provide the framework for security and scalability, it also is a standard adaptable for the coming years. This book's unique approach illustrates the techniques to practically apply PKI into solutions while developing the foundational concepts of the technology. Consequently, this book makes deploying this complex and essential technology simple.

Goals and Methods

This book is tailored to enable you to deploy PKI-based solutions in a simple, efficient, and manageable way. The book achieves this goal by taking a layered approach. First, it presents the foundations of PKI to ensure that you have the required theoretical background to properly understand the mechanisms. Then the book modularly takes those foundations into generic design considerations: The goal is to help you to perform the choices most suitable for the targeted environment; guidance is provided through sharing best practices and experiences acquired in production customer deployments. Those design modules are pieced together into hierarchical models, which are then applied to comprehensive solutions. Through the book, troubleshooting sections are included to ensure smooth implementations and enable you to gain a deep understanding of the internals.

Who Should Read This Book?

This book has been written primarily for enterprise network security designers, planners, architects, operators, and support personnel. These are the people responsible for the design, deployment, and support; and they can find the topic, scope, and level of detail beneficial. The book's structure is layered, starting from foundational topics, moving toward high-level architectures, and finally into detailed designs. This layered and modular approach can benefit both the intermediate reader and the advanced reader or individuals seeking a practical view of PKI. They can read the modules of interest or start from the beginning and learn the solutions throughout.

This book is also of interest to the user and purchaser of enterprise networks, including IT directors and CIOs or CTOs in small, medium-sized, and large enterprises and network engineers and support staff. Technical sales personnel both at network vendors and their integration partners can also greatly benefit from this book.

How This Book Is Organized

Although this book could be read cover-to-cover, it is designed to be flexible and enable you to easily move between chapters and sections of chapters to cover just the material that you need more work with. Chapter 1, "Crypto Refresh," provides an overview of the encryption-related technologies to provide a foundation and review of core concepts. Chapters 2 through 11 can be covered out of order; however, they are designed to build on each other. If you intend to read them all, the order; in the book is an excellent sequence to use.

The book is broken out into three major sections. The first section provides theoretical knowledge and background. The second section covers design principals and solutions. The third section discusses case studies for PKI and specific use cases. Chapters 2 through 11 cover the following topics:

- **Chapter 2, "Understanding PKI Building Blocks"**—Discusses analyzing criteria for placing the foundational pieces used to build a PKI and certificates and certificate authorities.
- **Chapter 3, "PKI Processes and Procedures"**—Discusses the basic processes required for a PKI to function, including enrollment, expiration, renewal, verification, and enforcement.
- **Chapter 4, "Troubleshooting"**—Covers how to troubleshoot basic PKI deployments, specifically key generation problems, enrollment problems and certificate verification problems.

Part II: Design and Solutions

- **Chapter 5, "Generic PKI Designs"**—Starts by covering a basic, small-style PKI design. It then covers a more involved hierarchical design, which is common among complex and larger deployments.
- **Chapter 6, "Integration in Large-Scale Site-to-Site VPN Solutions"**—Covers the two most popular large scale VPN deployments using certificates and examines how to deploy GET-VPN and DMVPN using PKI.
- **Chapter 7, "Integration in Remote Access VPN Solutions"**—Covers remote access VPN solutions. It covers ASA-based IPsec VPN remote access connections, ASA SSL VPN, and the Cisco VPN client.
- **Chapter 8, "Using 802.1x Certificates in Identity-Based Networking"**—Covers the basics of how to deploy certificates to control access at the switchport level.
- **Chapter 9, "PKI in Unified Communications"**—Covers the use of certificates in IPT-based systems to drive identity. This chapter covers Call Manager and IP phones' implementation of certificates.

Part III: Case Studies

- **Chapter 10, "Understanding Cisco Virtual Office Overview"**—Builds upon previous chapters' topics and weaves together a variety of certificate-based solutions. This topic uses 802.1x, DMVPN, and PKI architecture to build a cohesive virtual office solution.
- **Chapter 11, "Deploying VPNs with PKI Using Cisco Security Manager"**—Covers the use of Cisco Security Manager for PKI-based systems. It also covers how to migrate from preshared keys for IKE authentication to PKI.

This page intentionally left blank

Chapter 3

PKI Processes and Procedures

Several processes need to occur in a PKI network for a deployment to function smoothly. To address these processes, this chapter covers the following topics:

- Enrollment
- Certificate Expiration and Renewal
- Certificate Verification and Enforcement
- PKI Resiliency

Understanding the basics of cryptography and the building blocks of public key infrastructures provides a foundation for exploring the core processes and practical application of PKI. These processes govern how to get a certificate, how to keep a certificate that is current, how to revoke a certificate, and how to keep a PKI up and running if an outage occurs.

Enrollment

Enrollment is the process to obtain a certificate. The two process of enrollment are manual enrollment and a network SCEP-based enrollment. Network-based SCEP is discussed later in this chapter. Simple Certificate Enrollment Protocol (SCEP) is an IETF draft, draft-nourse-scep-20. Whereas both processes follow the same principles, the procedure for implementation varies. The common events for both scenarios are as follows:

- An end host generates an RSA (Rivest, Shamir and Adleman) key pair.
- A certificate request containing the end host's public key is delivered to a certificate authority (CA).
- The CA signs the request with the CA's private key and generates the end host's certificate.
- The certificate is delivered back to the end host.

Manual Enrollment

Sometimes a network connection may not be possible or secure between an endpoint and a certificate server. In this situation a non-network-based approach might be preferred. This approach requires an administrator to manually copy and paste a certificate into the local router.

Manual copy-and-paste enrollment has several steps. The high-level steps are presented here, followed by a detailed example. Example 3-1 through Example 3-6, which illustrates the execution of the following steps:

1. The spoke is configured to use terminal enrollment.
2. The certificate authority exports its certificate to the screen.
3. The spoke authenticates the certificate authority certificate and verifies the fingerprint.
4. The spoke makes an enrollment request.
5. The certificate authority grants the request.
6. The spoke certificate is pasted into the terminal.

Note In the following example, the name of the sub-ca is ra, which refers to Raleigh, not RA (registration authority).

Step 1. Configure the spoke to use terminal enrollment, as illustrated in Example 3-1.

Example 3-1 *Configure Spoke to Use Terminal Enrollment*

```
r35-4-1023(config)# crypto pki trustpoint ra
r35-4-1023(ca-trustpoint)# enrollment terminal
```

Step 2. The certificate authority exports its certificate to the screen, as shown in Example 3-2.

Example 3-2 *CA Exports Certificate*

```
Device: SUB-CA

S-3845-ra-subca(config)# crypto pki export ra-subca pem terminal
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICMDCCAZmgAwIBAgIBCDANBgkqhkiG9w0BAQQFADASMRAwDgYDVQQDEwdyb290
LWNhMB4XDTA5MDEyODE2MjExOVoXDTExMDEyODE2MjExOVowEzERMA8GA1UEAxMI
cmEtc3ViY2EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAPoXSGDFGRqPiVQt
cRscN6uGG+nY1exDTzY18AUaP83laS6ylbHek1P9nzwKNZysO9Ya8+ObhG9SEHCh
XUJd4Y2DovwWnxzFEhqvWI7hVP8vkWmRFZx7EooiWlW/lTxgqrnjdg4/N9OTej0E
```

```
pmExbQfL3TN+ZAckHrVbWl8w7OH7AgMBAAGjgZQwgZEwMQYDVR0fBCowKDAmoCSg
IoYgaHR0cDovLzE3Mi4yNi4xODUuOTkvcm9vdC1jYS5jcmwwDwYDVR0TAQH/BAUw
AwEB/zALBgNVHQ8EBAMCB4AwHwYDVR0jBBgwFoAUDkMCSiWkFtEXEC4a0UrEnEV/
QdAwHQYDVR0OBBYEFOOEC8szKHCxiv4yrUtP+fgFjhTtMA0GCSqGSIb3DQEBBAUA
A4GBAF1IN0RnKRKmj2SwrygZcYdgmMPkzaXFW+9c7xEq8UWO25bG3MqKLEwEURgU
DcZ1jMgJeciGiQMO6N0kpWwYwVI1w0dJZ5Ab2Nby9ew892viw/vFWjeTdJvTkrd7
KjLtRgnnslm26gsFhA1X9uvKpXfFsDp4kLnMxZxRIPQUc8m7
-----END CERTIFICATE-----
```

Step 3. The spoke authenticates the CA certificate and verifies the fingerprint, as shown in Example 3-3.

Example 3-3 *Authentication of CA Certificate and Verification of Fingerprint*

```
Device: SPOKE

r35-4-1023(config)# crypto pki authenticate ra

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIICMDCCAZmgAwIBAgIBCDANBgkqhkiG9w0BAQQFADASMRAwDgYDVQQDEwdyb290
LWNhMB4XDTA5MDEyODE2MjExOVoXDTExMDEyODE2MjExOVowEzERMA8GA1UEAxMI
cmEtc3ViY2EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAPoXSGDFGRqPiVQt
cRscN6uGG+nY1exDTzY18AUaP83laS6ylbHek1P9nzwKNZysO9Ya8+ObhG9SEHCh
XUJd4Y2DovwWnxzFEhqvWI7hVP8vkWmRFZx7EooiWlW/lTxgqrnjdg4/N9OTej0E
pmExbQfL3TN+ZAckHrVbWl8w7OH7AgMBAAGjgZQwgZEwMQYDVR0fBCowKDAmoCSg
IoYgaHR0cDovLzE3Mi4yNi4xODUuOTkvcm9vdC1jYS5jcmwwDwYDVR0TAQH/BAUw
AwEB/zALBgNVHQ8EBAMCB4AwHwYDVR0jBBgwFoAUDkMCSiWkFtEXEC4a0UrEnEV/
QdAwHQYDVR0OBBYEFOOEC8szKHCxiv4yrUtP+fgFjhTtMA0GCSqGSIb3DQEBBAUA
A4GBAF1IN0RnKRKmj2SwrygZcYdgmMPkzaXFW+9c7xEq8UWO25bG3MqKLEwEURgU
DcZ1jMgJeciGiQMO6N0kpWwYwVI1w0dJZ5Ab2Nby9ew892viw/vFWjeTdJvTkrd7
KjLtRgnnslm26gsFhA1X9uvKpXfFsDp4kLnMxZxRIPQUc8m7
-----END CERTIFICATE-----

Trustpoint 'ra' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
        Fingerprint MD5: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
       Fingerprint SHA1: 0A86F03E 077E587B 2DB4644A 5BA55F0F FC57D2EF

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

```
Device: SUB-CA verify fingerprint
S-3845-ra-subca#show crypto pki certificates verbose
Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 0D
  Certificate Usage: General Purpose
  Issuer:
    cn=ra-subca
  Subject:
    Name: ra-subca.cisco.com
    IP Address: 192.168.159.243
    Serial Number: FTX1111A468
    serialNumber=FTX1111A468+ipaddress=192.168.159.243+hostname=ra-subca.cisco.com
  CRL Distribution Points:
    http://172.26.185.99/ra-subca.crl
  Validity Date:
    start date: 15:26:27 EST Jul 13 2009
    end   date: 15:26:27 EST Jan 9 2010
    renew date: 15:26:27 EST Dec 4 2009
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (512 bit)
  Signature Algorithm: MD5 with RSA Encryption
  Fingerprint MD5: 542CDC69 10C8D510 65DF5E3C 66CEF438
  Fingerprint SHA1: 5C4C6F15 E1F5E184 C4681535 3CC61012 F5D694EC
  X509v3 extensions:
    X509v3 Key Usage: A0000000
      Digital Signature
      Key Encipherment
    X509v3 Subject Key ID: 5A1CBE8B A043B0A3 651D50C7 AFB04761 B92A8862
    X509v3 Authority Key ID: E3840BCB 332870B1 8AFE32AD 4B4FF9F8 058E14ED
    Authority Info Access:
  Associated Trustpoints: ra
  Storage: nvram:ra-subca#D.cer
  Key Label: ra
  Key storage device: private config

CA Certificate (subordinate CA certificate)
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 08
  Certificate Usage: Signature
```

```
  Issuer:
    cn=root-ca
  Subject:
    cn=ra-subca
  CRL Distribution Points:
    http://172.26.185.99/root-ca.crl
  Validity Date:
    start date: 12:21:19 EST Jan 28 2009
    end   date: 12:21:19 EST Jan 28 2011
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
  Signature Algorithm: MD5 with RSA Encryption
  Fingerprint MD5: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
  Fingerprint SHA1: 0A86F03E 077E587B 2DB4644A 5BA55F0F FC57D2EF
  X509v3 extensions:
    X509v3 Key Usage: 80000000
      Digital Signature
    X509v3 Subject Key ID: E3840BCB 332870B1 8AFE32AD 4B4FF9F8 058E14ED
    X509v3 Basic Constraints:
        CA: TRUE
    X509v3 Authority Key ID: 0E43024A 25A416D1 17102E1A D14AC49C 457F41D0
    Authority Info Access:
  Associated Trustpoints: ra ra-subca
  Storage: nvram:root-ca#8CA.cer
```

Step 4. The spoke makes an enrollment request, as shown in Example 3-4.

Example 3-4 *Spoke Makes Enrollment Request*

```
Device: SPOKE Generate enrollment request

r35-4-1023(config)# crypto pki enroll ra
% Start certificate enrollment ..

% The subject name in the certificate will include: r35-4-1023
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: FTX1048A6EJ
% Include an IP address in the subject name? [no]:
Display Certificate Request to terminal? [yes/no]: yes
Certificate Request follows:

-----BEGIN CERTIFICATE REQUEST-----
MIIBCjCBtQIBADAvMS0wEgYDVQQFEwtGVFgxMDQ4QTZFSjAXBgkqhkiG9w0BCQIW
```

```
CnIzNS00LTEwMjMwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAxcrafPm39Mmk51I+
dhnuVtkU9cYPOSHhS694b1taJG42esxtSUV8AwP4TcnQC/omIaIM1k5qIwnPe7FI
7Vic8QIDAQABoCEwHwYJKoZIhvcNAQkOMRIwEDAOBgNVHQ8BAf8EBAMCBaAwDQYJ
KoZIhvcNAQEEBQADQQBXw6esEMhzh9Jig0M3COwpX/wWMxUYQryYJK+uNDQf/PqH
n7zzC6Ii3UmfxlJKoK+Dgc6K3X87TVY6JRgMnlos
-----END CERTIFICATE REQUEST-----


Device: SUB-CA paste request generated from spoke

S-3845-ra-subca#crypto pki server ra-subca request pkcs10 terminal pem
% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE REQUEST-----
MIIBCjCBtQIBADAvMS0wEgYDVQQFEwtGVFgxMDQ4QTZFSjAXBgkqhkiG9w0BCQIW
CnIzNS00LTEwMjMwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAxcrafPm39Mmk51I+
dhnuVtkU9cYPOSHhS694b1taJG42esxtSUV8AwP4TcnQC/omIaIM1k5qIwnPe7FI
7Vic8QIDAQABoCEwHwYJKoZIhvcNAQkOMRIwEDAOBgNVHQ8BAf8EBAMCBaAwDQYJ
KoZIhvcNAQEEBQADQQBXw6esEMhzh9Jig0M3COwpX/wWMxUYQryYJK+uNDQf/PqH
n7zzC6Ii3UmfxlJKoK+Dgc6K3X87TVY6JRgMnlos
-----END CERTIFICATE REQUEST-----

% Enrollment request pending, reqId=2
```

Step 5. The certificate authority grants the request, as shown in Example 3-5.

Example 3-5 *CA Grants Request*

```
Device: SUB-CA

S-3845-ra-subca# crypto pki server ra-subca grant 2
Writing 2.crt !
Writing 2.cnm !
Writing ra-subca.ser !
% Granted certificate:
-----BEGIN CERTIFICATE-----
MIIB/DCCAWWgAwIBAgIBAjANBgkqhkiG9w0BAQQFADATMREwDwYDVQQDEwhyYS1z
dWJjYTAeFw0wOTA3MjkxNTI1MzZaFw0xMDAxMjUxNTI1MzZaMC8xLTASBgNVBAUT
C0ZUWDEwNDhBNkVKMBcGCSqGSIb3DQEJAhYKcjM1LTQtMTAyMzBcMA0GCSqGSIb3
DQEBAQUAA0sAMEgCQQDFytp8+bf0yaTnUj52Ge5W2RT1xg85IeFLr3hvW1okbjZ6
zG1JRXwDA/hNydAL+iYhogzWTmojCc97sUjtWJzxAgMBAAGjgYcwgYQwMgYDVR0f
BCswKTAnoCWgI4YhaHR0cDovLzE3Mi4yNi4xODUuOTkvcmEtc3ViY2EuY3JsMA4G
A1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBTjhAvLMyhwsYr+Mq1LT/n4BY4U7TAd
BgNVHQ4EFgQULA4QQvFQjDDe2ZwgmND9L1MYhJIwDQYJKoZIhvcNAQEEBQADgYEA
```

```
4eyutNSNdNA2uKgqatQGT66Nxx2s6DF4fLPJY7wLMHJv+pXwrmzYzJpKqQrzf0ZL
WbaVHu6RdRvq35PFSdIm72l/whuATZSEdnHUsEU9GnGDjpvJCmAw73IAa8LDnfaZ
3N2NaAxY4CXAsxHWWtD1ea7A7utdS0R29d2aqNkvaXM=
-----END CERTIFICATE-----
```

Step 6. Paste the spoke certificate into the terminal, as shown in Example 3-6.

Example 3-6 *Spoke Certificate Pasted into Terminal*

```
SPOKE import certificate from SUB-CA

r35-4-1023(config)# crypto pki import ra certificate

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIB/DCCAWWgAwIBAgIBAjANBgkqhkiG9w0BAQQFADATMREwDwYDVQQDEwhyYS1z
dWJjYTAeFw0wOTA3MjkxNTI1MzZaFw0xMDAxMjUxNTI1MzZaMC8xLTASBgNVBAUT
C0ZUWDEwNDhBNkVKMBcGCSqGSIb3DQEJAhYKcjM1LTQtMTAyMzBcMA0GCSqGSIb3
DQEBAQUAA0sAMEgCQQDFytp8+bf0yaTnUj52Ge5W2RT1xg85IeFLr3hvW1okbjZ6
zG1JRXwDA/hNydAL+iYhogzWTmojCc97sUjtWJzxAgMBAAGjgYcwgYQwMgYDVR0f
BCswKTAnoCWgI4YhaHR0cDovLzE3Mi4yNi4xODUuOTkvcmEtc3ViY2EuY3JsMA4G
A1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBTjhAvLMyhwsYr+Mq1LT/n4BY4U7TAd
BgNVHQ4EFgQULA4QQvFQjDDe2ZwgmND9L1MYhJIwDQYJKoZIhvcNAQEEBQADgYEA
4eyutNSNdNA2uKgqatQGT66Nxx2s6DF4fLPJY7wLMHJv+pXwrmzYzJpKqQrzf0ZL
WbaVHu6RdRvq35PFSdIm72l/whuATZSEdnHUsEU9GnGDjpvJCmAw73IAa8LDnfaZ
3N2NaAxY4CXAsxHWWtD1ea7A7utdS0R29d2aqNkvaXM=
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

The entire process is conducted by use of terminal access. Consequently, no packet exchanges are required between the certificate authority and the end spoke.

SCEP-Based Enrollment

Adding a large number of routers in an enterprise and going through the steps for each of those would be a painful exercise for the network engineer. Consider a thousand routers. Often, engineers prefer to have a templated configuration that can be set up one time, enabling automation for subsequent certificates upon certificate expiration.

When network connections are possible between an endpoint and a certificate server, a network-based approach might be preferred because it provides the opportunity to

templatize the approach, and in the future with features mentioned later, automatic addressing of certificate expiry issues. This approach is easier to implement and requires significantly less labor. Whenever possible, SCEP enrollment is the preferred solution. This approach requires minimal configuration on the router endpoints.

The use of the network-based approach has the chief benefit of improving scalability and limiting operational overhead. SCEP enables an endpoint to request a certificate or other certificate-related functions (revocation checking, and so on) remotely. SCEP runs on TCP port 80; however, it can also run on a nonstandard TCP port.

When an end device has an RSA key pair, it can make a request to the certificate authority using SCEP. That certificate request includes the public key. The CA responds with the new certificate, which is encrypted with the requestor's public key. This way, only the person making the request can decrypt it.

SCEP-based enrollment is configured in trustpoint mode. TCP port 80 is the default port used for SCEP and is configurable using the enrollment command. If a nonstandard port is used, make sure the http server configuration on the CA matches the nonstandard port. As shown in Example 3-7, the spoke is configured to use the CA or sub-CA URL for enrollment.

Example 3-7 *SCE- Based Enrollment Configuration Example*

```
r35-4-1023(config)#crypto pki trustpoint ra
r35-4-1023(ca-trustpoint)# enrollment url http://192.168.159.243:80
```

Certificate Expiration and Renewal

Certificates have a fixed lifetime. Eventually, both the root's certificate and the spoke's certificate expire. When a certificate expires, widespread connectivity issues might result so that in large scale VPN solutions, authentication in IKE would fail and connectivity could not be established. To prevent this type of failure, two mechanisms should be deployed for certificate renewal: auto-enrollment and rollover for end spokes and servers.

Auto-Enrollment

When a certificate on an end device is going to expire, auto-enrollment obtains a new certificate without disruption. By configuring auto-enrollment, the end host can request a new certificate at X time before its local certificate expires. This feature is used with SCEP, and together this provides an automated mechanism for enrollment requests prior to end node certificate expiration.

In Example 3-8, a spoke is configured to request a new certificate at 50 percent of the life time expiration, or 15 minutes into its assigned 30-minute lifetime. In the show crypto pki certificate output, notice the renew date is exactly 50 percent between the start date and end date (15 minutes).

Example 3-8 *Auto-Enrollment Example with* **show** *Command*

```
crypto pki trustpoint ra
 enrollment url http://192.168.159.243:80
 auto-enroll 50

S-3845-gm4-s-134# sh crypto pki cert
Certificate
  Status: Available
  Certificate Serial Number: 0x0DD
  Certificate Usage: General Purpose
...
  Validity Date:
    start date: 15:57:54 EST Mar 28 2008
    end   date: 16:27:54 EST Mar 28 2009
    renew date: 16:12:54 EST Sep 28 2008
  Associated Trustpoints: ra
```

The certificate authority has the option to grant requests manually or use grant auto, which is a feature that automatically grants certificate requests. This raises a classic problem in network security: availability versus security. Using grant auto makes the entire granting process more highly available and easier. However, grant auto on the CA makes it easy for any device to request and get a certificate.

Grant auto should be used with great care. Some circumstances where it might be all right are in closed systems, such as staging areas. Another situation would be in which policy controls are in place, such as a firewall, which enables only specific end hosts to access the CA, and only during windows when auto-enrollment requests occur. Also, the feature **grant auto trustpoint xxx** will only auto-grant requests signed by trustpoint xxx. Normally, xxx is the server trustpoint. Renewal requests are signed by the existing certificate. In that way, only renewal requests from clients with a valid certificate from your CA will be auto-granted.

Example 3-9 *Grant Auto to Facilitate Auto-Enrollment*

```
crypto pki server root-ca
 grant auto
```

Rollover

When a certificate on the CA server is going to expire, rollover enables the root CA to obtain a new certificate without disruption. By configuring rollover, the CA can generate a new certificate at X time before its local certificate expires. The new certificate, which is called the shadow certificate, becomes active at the precise moment the current CA certificate expires.

Notice in Example 3-10, the end date of the current certificate is exactly the same as the start date of the rollover shadow certificate.

Example 3-10 *Rollover Example on the Root CA*

```
crypto pki server root-ca
 grant auto
 auto-rollover 0 1
 database url ftp://172.26.129.252
 S-3825-root-ca# show crypto pki certificates
CA Certificate (Rollover)
  Status: Available
  Certificate Serial Number: 0x4
  Certificate Usage: Signature
  Issuer:
    cn=root L\=RTP ST\=NC C\=US
  Subject:
    Name: root L=RTP ST=NC C=US
   cn=root L\=RTP ST\=NC C\=US
  Validity Date:
    start date: 15:14:48 EST Feb 28 2008
    end   date: 15:14:48 EST Mar 1 2008
  Associated Trustpoints: root-ca

CA Certificate
  Status: Available
  Certificate Serial Number: 0x3
  Certificate Usage: Signature
  Issuer:
    cn=root L\=RTP ST\=NC C\=US
  Subject:
    cn=root L\=RTP ST\=NC C\=US
  Validity Date:
    start date: 15:14:48 EST Feb 26 2008
    end   date: 15:14:48 EST Feb 28 2008
  Associated Trustpoints: root-ca
```

Certificate Verification and Enforcement

Certificates expire. Network administrators might simply wait for a certificate to expire or use another method to remove a certificate. For example, if a router is stolen, there needs to be a way to revoke its certificate so that it can no longer participate in the network. In the case of IPsec deployments, for example, a revoked certificate would result in failure during IKE.

There are three significant approaches that use certificates. The first approach uses certificate revocation lists (CRL), which are periodically downloaded to a router and thus require lower overhead. The second approach uses OCSP, which provides real-time updates and makes a network call for each certificate that is presented. The third approach uses an AAA server and certificates together, which involves the end user performing authentication. The differences in these approaches are outlined in Table 3-1.

Table 3-1 *Certificate Verification Approaches*

	Advantages	Disadvantages
CRL	Low network profile, CRL server supported in IOS	Periodic, hours can pass between the time revocation occurs and CRL update takes effect. If lists grow long, processing time becomes a problem.
OCSP	Real-time revocations	Server feature is not available in IOS. IOS CA is not supported with OCSP. Only client checking is supported.
AAA	Real-time authorization enforcement and optional granular authorization controls	Specific certificate credentials must be entered into the AAA server. Depending on the selection criteria, this could be labor-intensive for an administrator.

Certificate Revocation Lists

Certificate revocation lists (CRL) enable devices to determine if a certificate has been revoked prior to expiration. A certificate revocation list is composed of the certificate's serial number (issued by the granting authority) and the date of revocation.

The CRL database is located on an external server (recommended) or on the CA. The CA will, by default, store the CRL locally. If the recommended practice of housing the CRL on an external server is used, the command **database url crl** points to the location where the CRL database file is stored. This is configured under cs-server sub configuration mode.

The location of the database file and where end devices or users go to access the CRL might be the same. The location can also be different (see Figure 3-1 for an example CRL stored on Windows). As a recommended practice, housing the CRL for retrieval for end devices should be in a different location than the database file actively used by the CA. This insures that end users do not have access to the source CRL database file that might pose a security risk. The command to configure the location to direct end devices and users to retrieve the CRL information is the **cdp-url** command, which is also configured in cs-server sub configuration mode. The cdp url information is given to certificate users as part of the certificate they receive. Consequently, the decision regarding the url for end user retrieval of the CRL needs to be made before certificates are issued.

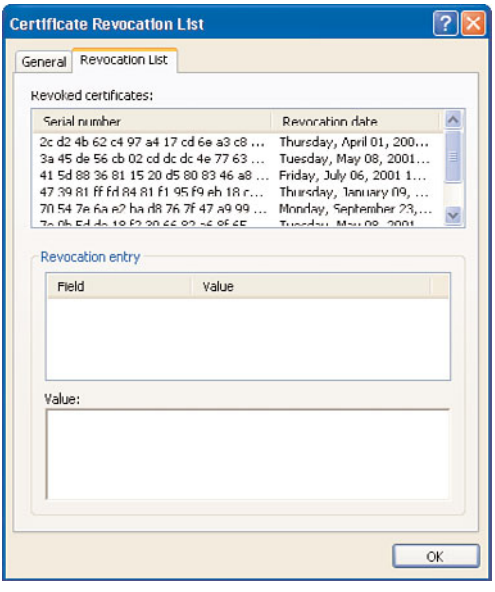


Figure 3-1 *A CRL Stored in Windows*

CRLs also have a lifetime. At a given time a CRL will expire and is valid only for an interval. When the interval is complete, a new CRL is downloaded by IOS via http. The CRL is then cached locally on the router. Consequently CRLs are not in real time. A certificate is revoked and then that information is propagated at a periodic interval.

There are two significant drawbacks to using CRLs in some environments. The first drawback is that CRLs are downloaded periodically, which means that a revoked certificate can still be authenticated before a new CRL is downloaded. The second drawback involves scalability of CRLs. If CRLs are deployed, the choice to revoke a certificate should be done with great care (that is, not add entries for administration or testing purposes). The lookup routers do against the CRL when verifying a peers certificate is linear; that is, it is line by line. As lists become longer, this takes up that much more CPU resources. Consequently, this can slow down and even timeout during IKE negotiations.

Example 3-11 shows a certificate being revoked.

Example 3-11 *Revoking a Certificate*

```
s-3845-ra-subca# crypto pki server ra-subca revoke 0x50
Writing ra-subca.crl !
% Certificate 0x50 successfully revoked.
```

The **Crypto pki server {name} revoke {serial number}** is executed on the granting certificate authority. Serial numbers are used to track certificates. After the certificate is revoked, the information will not be updated until the CRL expires, which might be many hours from the time of expiration. The CR lifetime can be changed. Example 3-12 illustrates shortening the CRL lifetime from the default of 24 hours.

Example 3-12 *CRL Lifetime Configuration*

```
3845-root-ca# Show run
...
crypto pki server root-ca
 database archive pkcs12 password 7 843595F
 grant auto rollover ca-cert
 grant auto
 lifetime crl 0 10
 cdp-url http://www.crl.cisco.com/ca.crl
 database url crl ftp://172.26.129.252
```

Figure 3-2 illustrates a possible design for handling CRLs.

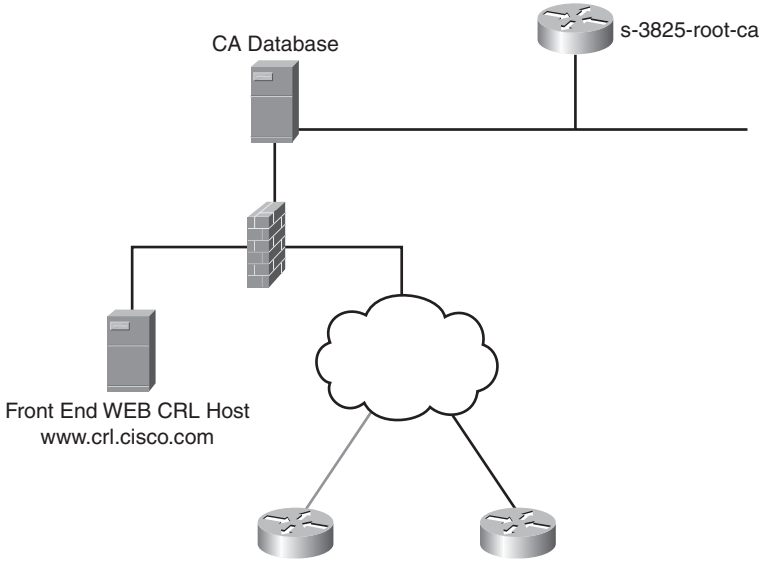


Figure 3-2 *CRL Server Architecture*

As shown in the figure, the end routers would have the frontend web server's URL included in their certificates for the CRL distribution point. The frontend server can get data from the backend server's database. This can be done via ftp and crontab or other methods. The firewall can provide a separation between the vulnerable frontend server and

backend database by enabling only the minimal traffic to pass between the frontend service layer and backend server in the datacenter's access layer.

Online Certificate Status Protocol

A major disadvantage of CRL checking is the timeliness of updates for end hosts. The chief advantage of Online Certificate Status Protocol (OCSP) is that it provides a real-time update to end users. OCSP's disadvantage is that it relies on third-party software. A router cannot act as an OCSP server. Also, IOS CA is not officially supported with OCSP servers at the time of this writing. OCSP as a method of revocation checking is supported for end spokes.

An OCSP server has two methods to obtain information about the validity of a certificate. It can receive periodic updates from a CA by means of a "push" from the CA, or it can periodically poll a CRL distribution point (see Figure 3-3). This approach is still periodic in nature. The periods are much smaller than with a traditional CRL approach, and simple exchanges occur between a CRL distribution point and the OCSP server.

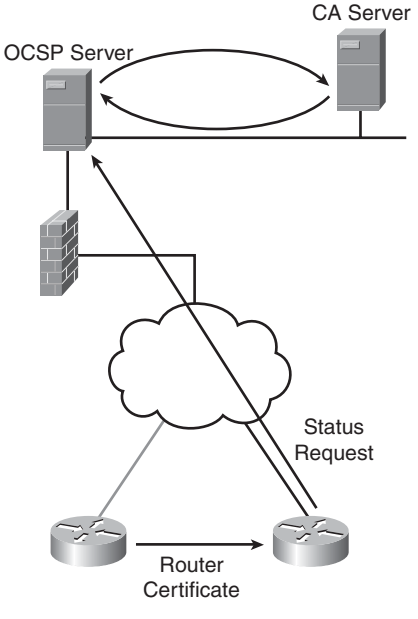


Figure 3-3 *OCSP Devices*

When an end host requests the validity of a certificate, it submits a query to the OCSP server, which contains the certificate's serial number. The OCSP server can provide a response to the query with a status for that certificate. The status response can be good, unknown, or revoked. The response from the OCSP server can be used immediately and consequently does not require local storage space on the router. Example 3-13 shows how to configure OCSP in IOS.

Example 3-13 *OCSP Configuration*

```
Router(ca-trustpoint)# ocsp url http://ocspserver.cisco.com:80
Router(ca-trustpoint)# revocation-check ocsp
```

OCSP service can function like a "cloud" service, using a push model between the CA server and the OCSP server. Also the OCSP server can have a certificate issued by the CA to verify its identity to others who make requests.

PKI Integration with AAA

Authentication, authorization, and accounting (AAA) servers are common in enterprise infrastructures. The Cisco AAA server is Cisco Secure Access Control System (ACS). AAA integration provides a mechanism for authorization. A certificate can provide authentication; when combined with an AAA server, the AAA server can provide authorization for the end host.

Fields in the certificate (such as **subject** and **serial number**) can be passed back to a RADIUS server or TACACS server. The server can check the credentials provided to it by the authorizing router to determine if the device is authorized for network access.

The advantage of using AAA as a solution is that it enables authorization in addition to authentication. The moment an administrator decides a certificate is no longer authorized, the administrator can make the change in the AAA server, and it is immediately effective. The disadvantage of the solution is that it requires manual entry of certificate credentials and authorization in the AAA server.

The leading practice for this approach uses an ACS RADIUS server. The credentials recommended to pass back are several Cisco AV pairs. The Cisco AV pairs recommended are **avpair=pki:cert-application=all**, which announces this is a certificate, and **cisco-avpair=pki:cert-trustpoint={trust point name}**, which announces the trustpoint associated with the certificate. Lastly, user level credentials are passed back. The recommended credential is the subject name as it appears in the certificate, which is the FQDN provided to the CA by the router requesting a certificate.

The ACS server would reside local to the server performing the authorization. Often, the authorizing router can be a central or hub gateway to a central location. Cisco AV pairs that are commonly passed to a RADIUS server are **cisco-avpair=pki:cert-application=all**, **cisco-avpair=pki:cert-trustpoint={trust point name}**, and **cisco-avpair=pki:cert-serial={serial number}.**

Although these AV pairs are often used, the drawback is every time a new certificate is issued the serial number and potentially other information would need to be re-entered. A simpler approach would be to use the Fully Qualified Domain Name (FQDN) of the router, which would be included in the certificate. Then the only AV pair should be associated with the CA at the group level, as will be shown in the example. The AV pair associated with the CA is combined with the FQDN taken from the certificate's subject name field will provide all the credentials for authorization.

Upon disabling authorization for that router, the fully qualified domain name of that router can be removed as a user on the AAA server to deny authorization. This reduces the overall administrative overhead in keeping up with the changing fields in certificates (such as serial number). Example 3-14 illustrates how to configure a router to use AAA.

Example 3-14 *Configuring the Authorizing Router for AAA Using RADIUS*

```
aaa authentication login no-auth none
aaa authorization exec dmvpn-pki group radius
aaa authorization network dmvpn-pki group radius
!
crypto pki trustpoint ra
 enrollment url http://192.168.159.243:12345
 serial-number
 ip-address 192.168.159.242
 revocation-check crl
 rsakeypair hub-keys
 auto-enroll 70 regenerate
 authorization list dmvpn-pki
 authorization username subjectname unstructuredname
! above line will not appear in show run since it is a default !
```

On the ACS configuration, screen captures can be found in Figure 3-4 and Figure 3-5. The PKI group is created with the appropriate AV pairs. Then a user with the FQDN is named and added to that group.

Figure 3-4 *ACS AAA Server Configuration for PKI Integration*

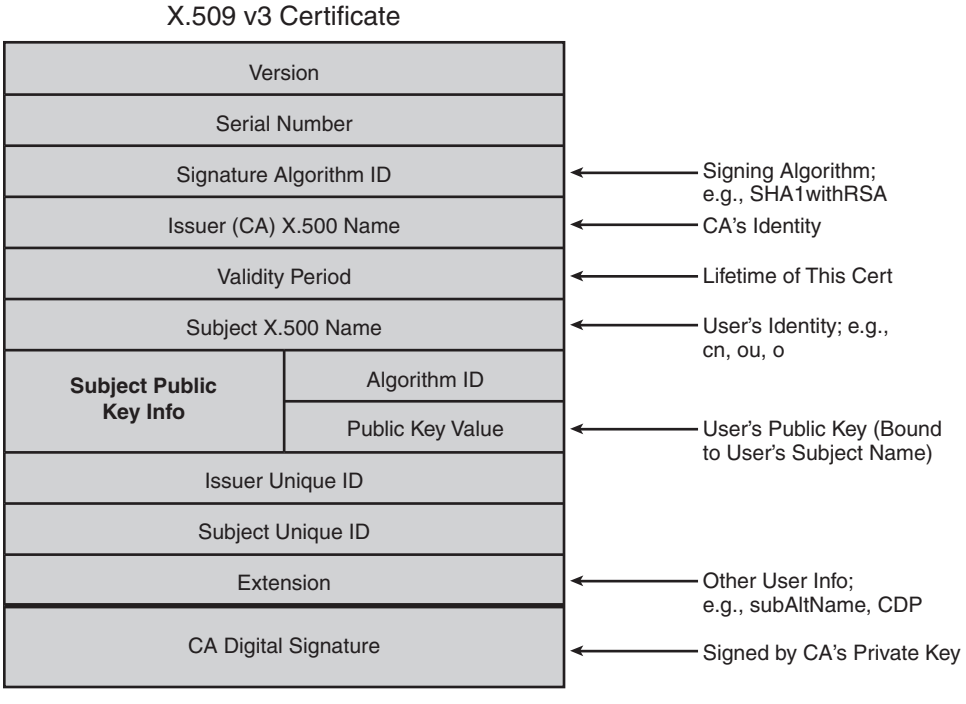


Figure 3-5 *X.509 Certificate Structure*

PKI Resiliency

Sometimes, routers experience hardware failures, or an administrator might accidentally lose information on a router. If this router is the certificate authority, a key part of the network infrastructure is compromised. Consequently, a method should exist to recover from such events without resulting in a catastrophic failure.

Certificate Authority Resiliency

The certificate authority is the key piece to consider for a resilient PKI. There are several files on a CA server to consider, including the following:

- **Database** file contains the RSA keys and local certificate.
- The **.Ser** file has the last serial number issued by the CA.
- The **.CRL** file contains the list certificates that have been revoked.

The default location for file storage is on the local NVRAM. For maximum resiliency, it is considered best practice to use an external FTP server to store these files. This external server should not be used for anything else and should have reachability only from the CA servers. Resiliency practices for mission critical servers should be applied to this

server. Example 3-15 shows optionally placing the CRL file in a different location than the URL file. The CRL file by default would be stored in the same location as the database file.

Example 3-15 *Configuring an External FTP Server*

```
3845-root-ca# show run
crypto pki server root-ca
 database archive pkcs12 password {password}
 database url ftp://172.26.129.252
 database url crl ftp://172.26.129.252
```

If a router fails, a new router should be available to become the new CA. The steps to restore are simple:

Step 1. Import the database file using the command **crypto pki import** {*root-ca name*} **pkcs12 ftp://**{*x.y.z.w*} {*password*}.

Step 2. Paste the configuration that is a common and recommended standard practice to be backed up regularly. Using this method the restoration process is simple and straight forward.

Summary

Many processes need to occur in the background of a PKI for things to run smoothly. Some considerations are enrollment, certificate renewal, certificate verification and enforcement, and resiliency. This chapter discussed manual enrollment and SCEP, which is a network-based enrollment process and is preferred where ever possible because enrollment over the network is much simpler to implement.

For certificate renewal, consider two elements: the CA certificate expiring and the spoke certificate expiring. To renew the CA certificate, the IOS feature rollover is used that creates a shadow certificate on the CA server that is valid at the moment of the current certificate's expiration. For the spoke, an auto-enrollment certificate renewal feature is used. At a time in which is a certain percentage "X" of the lifetime has passed, the spoke requests a new certificate.

Certificate verification and enforcement is required to make sure certificates presented during authentication are valid. Two principal methods are used for this enforcement, plus a third authorization-based method that is adapted to provide similar functionality. The approaches are CRL, OCSP, and AAA integration. CRL lists provide a list of revoked certificates and is supported by IOS CA. However, CRLs are not real time and may take many hours for information to be propagated about the expiration of a certificate. OCSP is real time, however, is not supported on IOS CA and requires third-party servers. Integration with AAA provides a method of authorization that is real time.

Authentication occurs as usual, and authorization enforcement can determine if network access is permitted. The disadvantage of this approach is that the AAA server needs to have information for all certificates in the network.

Another important process for any network device is what to do if a device must be restored. If you follow leading practices that dictate using an external FTP server to store the database, restoring an IOS CA is straightforward. The steps involved in restoration are twofold; import the database file and copy-paste the old configuration on to the new IOS CA server.

This page intentionally left blank

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

V

X