



SECURITY

IKEv2 IPsec Virtual Private Networks

Understanding and Deploying IKEv2,
IPsec VPNs, and FlexVPN in Cisco IOS

ciscopress.com

Graham Bartlett, CCIE No. 26709
Amjad Inamdar, CISSP No. 460898

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



IKEv2 IPsec Virtual Private Networks

Understanding and Deploying IKEv2, IPsec VPNs, and FlexVPN in Cisco IOS

Graham Bartlett, CCIE No. 26709

Amjad Inamdar, CISSP No. 460898

Cisco Press

800 East 96th Street

Indianapolis, Indiana, 46240 USA

IKEv2 IPsec Virtual Private Networks

Understanding and Deploying IKEv2, IPsec VPNs, and FlexVPN in Cisco IOS

Graham Bartlett, CCIE No. 26709

Amjad Inamdar, CISSP No. 460898

Copyright © 2017 Cisco Systems, Inc.

Cisco Press logo is a trademark of Cisco Systems, Inc.

Published by:

Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing August 2016

Library of Congress Cataloging-in-Publication Number: 2016947704

ISBN-13: 978-1-58714-460-8

ISBN-10: 1-58714-460-3

Warning and Disclaimer

This book is designed to provide information about IKEv2 and IPsec VPNs on Cisco IOS. 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.

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.

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.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.



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.



CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CQVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, Gigastorage, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Editor in Chief: Mark Taub

Product Line Manager: Brett Bartow

Acquisitions Editor: Brett Bartow

Business Operation Manager, Cisco Press:
Ronald Fligge

Managing Editor: Sandra Schroeder

Development Editor: Ellie Bru

Project Editor: Mandie Frank

Copy Editor: Lori Martinsek

Technical Editors: Alex Honore, Olivier Pelerin

Team Coordinator: Tammi Barnett

Cover Designer: Chuti Prasertsith

Composition: codeMantra

Indexer: Lisa Stumpf

About the Authors

Graham Bartlett, CCIE No. 26709, has designed a number of large scale Virtual Private Networks within the UK and worked with customers throughout the world using IKEv2 and Next Generation Encryption. Graham's interests include Security and Virtual Private Networks. Within this space he has discovered zero-day vulnerabilities, including the highest severity security advisory in the March 2015 Cisco IOS software and IOS XE software security advisory bundled publication. He has contributed to numerous IETF RFCs, and has intellectual property published as prior art. He is a CiscoLive speaker and has developed Cisco Security exam content (CCIE/CCNP). He is a CCP (Senior) IA Architect, CCP (Practitioner) Security & Information Risk Advisor, CCNP, CISSP, Cisco Security Ninja and holds a BSc(Hons) in Computer Systems and Networks.

Amjad Inamdar CISSP 460898, is a Senior Technical Leader with Cisco IOS Security Engineering, India. He has primarily worked on design, development and deployment of Cisco IOS secure connectivity solutions including the industry leading FlexVPN, DMVPN, GETVPN and EzVPN solutions and is currently working on the Cisco next generation SD-WAN solution. He has contributed to IETF drafts, holds a Cisco patent and has prior art publications. He holds many industry certifications including CISSP, CCSK, CCNP Security, CCDP, CCNP R/S, CCNA (SP, Data Center, Wireless, Voice), Cisco Security Ninja and has presented security at conferences, internal forums and to Cisco customers and partners. He holds a degree (B.E) in Electronics and Communication Engineering.

Note from the Authors

Alex Honore was originally an author of this book, but due to commitments opted to become a Technical Reviewer. Alex has been a fundamental member of the team, a number of chapters were originally written by Alex in addition to some of the diagrams.

About the Technical Reviewers

Alex Honore, (CCIE Security No. 19553) has been with Cisco since 2005 and currently works as a Technical Leader in Cisco's Security Business Group, specializing in leading-edge network and threat analytics. He was a senior engineer in Cisco Technical Services for 9 years, focusing on advanced troubleshooting and escalation support for VPN, Security, and Content Networking solutions in the Technical Assistance Center (TAC), as well as consulting for Cisco Advanced Services and speaking regularly at Cisco Live on the topic of IPsec VPNs. Alex holds a M.Sc. degree of Electrical Engineering and Telecommunications from the Faculty of Engineering in Mons, Belgium.

Olivier Pelerin (CCIE Security No. 20306) has more than 16 years of experience in networking and security. He joined Cisco TAC (Cisco Technical Assistance Center) as a customer support engineer back in 2005. He is still working for TAC as a Technical Leader, focusing on escalations around VPN solutions. He has been involved in FlexVPN since the start and is co-leading the development of the packet tracer on ISR-NG/ASR1000. Olivier is a distinguished speaker at CiscoLive. He holds a degree in Geography from Université Catholique de Louvain (Belgium).

Dedications

Graham Bartlett:

I dedicate this book to my loving family: Lorna, Edward, Rose, and my parents.

Amjad Inamdar:

I dedicate this book to my parents, Sayedrasul and Ayesha Inamdar, who have been my role models and inspiration, and to my entire family.

Acknowledgments

Graham Bartlett:

Alex Honore deserves a very special mention. Alex originally was an author, but due to work and family commitments became a reviewer. Alex has been a diligent reviewer due to his ability to break complex topics into many simple layers. I want to thank Alex for his patience and the help he has provided. We are honored to have him as a reviewer and he is a great asset to Cisco. I know if Alex ever writes a book himself, it will be, for me, the perfect book.

Olivier Pelerin has been a fantastic reviewer and brought a lot of issues and resolutions to our attention that otherwise we would have missed. Olivier is 100% focused on customers and many of the tips, tricks, and guidance contained within need to be credited to him. Many times I've had late-night chats regarding IKEv2 and our implementation. Olivier should also be credited for improving Cisco's IKEv2 usability, stability, and serviceability, which ultimately has a positive effect on you—the customer.

Thank you both—I treat you as good friends and could not have achieved this without your knowledge and skills.

Frederic Detienne: Fred—thank you for leading the creation of DMVPN, Flex, and IKEv2 on Cisco IOS. FlexVPN was my main motivation for joining Cisco, and it's now apparent that as VPN Architect you have developed a technology that is unique within the industry. It's a great pleasure and honor to work with you.

Raffaele Brancaleoni: Raff, thanks for all the help you provided with the development of this book. You've been a great mentor and a good friend.

The many great VPN engineers in Cisco TAC: Piotr and the fantastic Krakow crew, Wen, Atri, Jay, Frank, Marcin, Istvan, Nitin, Tom. Xavier and PSIRT. All of the IKEv2 developers that have helped: Sairam, Manish, Balaji, Tapes. Crypto team: Anthony, David, Scott and Panos.

Richard Gallagher: A great engineer and friend.

All the customers that I work with, who are not just customers of Cisco, but friends. You know who you are.

Thanks for the support of my management team, Francois, Khalid, David and Ashley.

Thanks to all my good friends and colleagues within Cisco and within the industry; you know who you are.

Also thanks for the support and patience of the Cisco Press team, namely Ellie and Brett.

Amjad Inamdar:

First, I would like to thank God for this opportunity to write a book which I have always wanted to do. Special thanks to my wife, Huzefiya, for being a constant companion in writing this book and her encouragement and support; my children, Sobia, Aleena, Arfa, and Ammar, for their patience and understanding while I spent many of the weekends and holidays writing this book. Thanks to my brother, sisters and family members for their support.

Graham Bartlett—Thank you for launching me onto this journey by offering me the chance to be the co-author. I look forward to continuing this journey with you. Thanks for your patience and all your help, you are a great friend.

Frederic Detienne—Thank you for being my mentor. I have truly enjoyed working with you in designing and implementing the Cisco IOS IKEv2 and FlexVPN solution.

Alex Honore—Thank you for your excellent review. You have been an inspiration and a great friend and guide. I will always cherish our work together on IKEv2 and FlexVPN.

Olivier Pelerin—Thank you for your detailed review comments, most of which were customer focused. I value our work together on FlexVPN scalability.

Thanks to Raffaele Brancaleoni for his review comments and to Yeleshwarapu Sairam, Tapes Maheshwari, Kalyani Garigipati from the IKEv2 and FlexVPN development team for their input.

Thanks to my management, colleagues, and friends at Cisco for their support.

Contents at a Glance

Foreword xxvii

Introduction xxxiii

Part I Understanding IPsec VPNs

Chapter 1 Introduction to IPsec VPNs 1

Part II Understanding IKEv2

Chapter 2 IKEv2: The Protocol 23

Chapter 3 Comparison of IKEv1 and IKEv2 67

Part III IPsec VPNs on Cisco IOS

Chapter 4 IOS IPsec Implementation 79

Part IV IKEv2 Implementation

Chapter 5 IKEv2 Configuration 105

Chapter 6 Advanced IKEv2 Features 171

Chapter 7 IKEv2 Deployments 189

Part V FlexVPN

Chapter 8 Introduction to FlexVPN 211

Chapter 9 FlexVPN Server 269

Chapter 10 FlexVPN Client 331

Chapter 11 FlexVPN Load Balancer 363

Chapter 12 FlexVPN Deployments 381

Part VI IPsec VPN Maintenance

Chapter 13 Monitoring IPsec VPNs 417

Chapter 14 Troubleshooting IPsec VPNs 445

Part VII IPsec Overhead

Chapter 15 IPsec Overhead and Fragmentation 503

Part VIII Migration to IKEv2

Chapter 16 Migration Strategies 539

Index 567

Contents

Foreword	xxvii
Introduction	xxxiii

Part I Understanding IPsec VPNs

Chapter 1	Introduction to IPsec VPNs	1
	The Need and Purpose of IPsec VPNs	2
	Building Blocks of IPsec	2
	Security Protocols	2
	Security Associations	3
	Key Management Protocol	3
	IPsec Security Services	3
	Access Control	4
	Anti-replay Services	4
	Confidentiality	4
	Connectionless Integrity	4
	Data Origin Authentication	4
	Traffic Flow Confidentiality	4
	Components of IPsec	5
	Security Parameter Index	5
	Security Policy Database	5
	Security Association Database	6
	Peer Authorization Database	6
	Lifetime	7
	Cryptography Used in IPsec VPNs	7
	Symmetric Cryptography	7
	Asymmetric Cryptography	8
	The Diffie-Hellman Exchange	8
	Public Key Infrastructure	11
	Public Key Cryptography	11
	Certificate Authorities	12
	Digital Certificates	12
	Digital Signatures Used in IKEv2	12
	Pre-Shared-Keys, or Shared Secret	13
	Encryption and Authentication	14
	IP Authentication Header	15
	<i>Anti-Replay</i>	16

IP Encapsulating Security Payload (ESP)	17
<i>Authentication</i>	18
<i>Encryption</i>	18
<i>Anti-Replay</i>	18
<i>Encapsulation Security Payload Datagram Format</i>	18
Encapsulating Security Payload Version 3	19
<i>Extended Sequence Numbers</i>	19
<i>Traffic Flow Confidentiality</i>	20
<i>Dummy Packets</i>	20
Modes of IPsec	20
IPsec Transport Mode	20
IPsec Tunnel Mode	21
Summary	22
References	22

Part II Understanding IKEv2

Chapter 2 IKEv2: The Protocol 23

IKEv2 Overview	23
The IKEv2 Exchange	24
IKE_SA_INIT	25
Diffie-Hellman Key Exchange	26
Security Association Proposals	29
Security Parameter Index (SPI)	34
Nonce	35
Cookie Notification	36
Certificate Request	38
HTTP_CERT_LOOKUP_SUPPORTED	39
Key Material Generation	39
IKE_AUTH	42
Encrypted and Authenticated Payload	42
Encrypted Payload Structure	43
Identity	44
Authentication	45
<i>Signature-Based Authentication</i>	46
<i>(Pre) Shared-Key-Based Authentication</i>	47
EAP	48
Traffic Selectors	50
Initial Contact	52

CREATE_CHILD_SA	53
IPsec Security Association Creation	53
IPsec Security Association Rekey	54
IKEv2 Security Association Rekey	54
IKEv2 Packet Structure Overview	55
The INFORMATIONAL Exchange	56
Notification	56
Deleting Security Associations	57
Configuration Payload Exchange	58
Dead Peer Detection/Keepalive/NAT Keepalive	59
IKEv2 Request – Response	61
IKEv2 and Network Address Translation	61
NAT Detection	64
Additions to RFC 7296	65
RFC 5998 An Extension for EAP-Only Authentication in IKEv2	65
RFC 5685 Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)	65
RFC 6989 Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2)	65
RFC 6023 A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)	66
Summary	66
References	66

Chapter 3 Comparison of IKEv1 and IKEv2 67

Brief History of IKEv1	67
Exchange Modes	69
IKEv1	70
IKEv2	71
Anti-Denial of Service	72
Lifetime	72
Authentication	73
High Availability	74
Traffic Selectors	74
Use of Identities	74
Network Address Translation	74
Configuration Payload	75
Mobility & Multi-homing	75

Matching on Identity	75
Reliability	77
Cryptographic Exchange Bloat	77
Combined Mode Ciphers	77
Continuous Channel Mode	77
Summary	77
References	78

Part III IPsec VPNs on Cisco IOS

Chapter 4 IOS IPsec Implementation 79

Modes of Encapsulation	82
GRE Encapsulation	82
GRE over IPsec	83
IPsec Transport Mode with GRE over IPsec	83
IPsec Tunnel mode with GRE over IPsec	84
Traffic	85
<i>Multicast Traffic</i>	85
<i>Non-IP Protocols</i>	86
The Demise of Crypto Maps	86
Interface Types	87
Virtual Interfaces: VTI and GRE/IPsec	87
Traffic Selection by Routing	88
Static Tunnel Interfaces	90
Dynamic Tunnel Interfaces	91
sVTI and dVTI	92
Multipoint GRE	92
Tunnel Protection and Crypto Sockets	94
Implementation Modes	96
Dual Stack	96
Mixed Mode	96
Auto Tunnel Mode	99
VRF-Aware IPsec	99
VRF in Brief	99
VRF-Aware GRE and VRF-Aware IPsec	101
VRF-Aware GRE over IPsec	102
Summary	103
Reference	104

Part IV IKEv2 Implementation

Chapter 5 IKEv2 Configuration 105

IKEv2 Configuration Overview	105
The Guiding Principle	106
Scope of IKEv2 Configuration	106
IKEv2 Configuration Constructs	106
IKEv2 Proposal	107
Configuring the IKEv2 Proposal	108
Configuring IKEv2 Encryption	111
Configuring IKEv2 Integrity	113
Configuring IKEv2 Diffie-Hellman	113
Configuring IKEv2 Pseudorandom Function	115
Default IKEv2 Proposal	115
IKEv2 Policy	117
Configuring an IKEv2 Policy	118
<i>Configuring IKEv2 Proposals under IKEv2 Policy</i>	119
<i>Configuring Match Statements under IKEv2 Policy</i>	120
Default IKEv2 Policy	121
IKEv2 Policy Selection on the Initiator	122
IKEv2 Policy Selection on Responder	124
IKEv2 Policy Configuration Examples	125
<i>Per-peer IKEv2 Policy</i>	125
<i>IKEv2 Policy with Multiple Proposals</i>	126
IKEv2 Keyring	128
Configuring IKEv2 Keyring	129
<i>Configuring a Peer Block in Keyring</i>	130
Key Lookup on Initiator	132
Key Lookup on Responder	133
IKEv2 Keyring Configuration Example	134
IKEv2 Keyring Key Points	136
IKEv2 Profile	136
IKEv2 Profile as Peer Authorization Database	137
Configuring IKEv2 Profile	138
<i>Configuring Match Statements in IKEv2 Profile</i>	139
<i>Matching any Peer Identity</i>	142
<i>Defining the Scope of IKEv2 Profile</i>	143
<i>Defining the Local IKE Identity</i>	143

<i>Defining Local and Remote Authentication Methods</i>	145
<i>IKEv2 Dead Peer Detection</i>	149
<i>IKEv2 Initial Contact</i>	151
<i>IKEv2 SA Lifetime</i>	151
<i>NAT Keepalives</i>	152
<i>IVRF (inside VRF)</i>	152
<i>Virtual Template Interface</i>	153
<i>Disabling IKEv2 Profile</i>	153
<i>Displaying IKEv2 Profiles</i>	153
IKEv2 Profile Selection on Initiator and Responder	154
IKEv2 Profile Key Points	154
IKEv2 Global Configuration	155
HTTP URL-based Certificate Lookup	156
IKEv2 Cookie Challenge	156
IKEv2 Call Admission Control	157
IKEv2 Window Size	158
Dead Peer Detection	158
NAT Keepalive	159
IKEv2 Diagnostics	159
PKI Configuration	159
Certificate Authority	160
Public-Private Key Pair	162
PKI Trustpoint	163
PKI Example	164
IPsec Configuration	166
IPsec Profile	167
IPsec Configuration Example	168
Smart Defaults	168
Summary	169
Chapter 6	Advanced IKEv2 Features 171
Introduction to IKEv2 Fragmentation	171
IP Fragmentation Overview	172
IKEv2 and Fragmentation	173
IKEv2 SGT Capability Negotiation	178
IKEv2 Session Authentication	181
IKEv2 Session Deletion on Certificate Revocation	182
IKEv2 Session Deletion on Certificate Expiry	184

IKEv2 Session Lifetime 185

Summary 187

References 188

Chapter 7 IKEv2 Deployments 189

Pre-shared-key Authentication with Smart Defaults 189

Elliptic Curve Digital Signature Algorithm Authentication 194

RSA Authentication Using HTTP URL Lookup 200

IKEv2 Cookie Challenge and Call Admission Control 207

Summary 210

Part V FlexVPN

Chapter 8 Introduction to FlexVPN 211

FlexVPN Overview 211

The Rationale 212

FlexVPN Value Proposition 213

FlexVPN Building Blocks 213

IKEv2 213

Cisco IOS Point-to-Point Tunnel Interfaces 214

Configuring Static P2P Tunnel Interfaces 214

Configuring Virtual-Template Interfaces 216

Auto-Detection of Tunnel Encapsulation and Transport 219

Benefits of Per-Peer P2P Tunnel Interfaces 221

Cisco IOS AAA Infrastructure 221

Configuring AAA for FlexVPN 222

IKEv2 Name Mangler 223

Configuring IKEv2 Name Mangler 224

Extracting Name from FQDN Identity 225

Extracting Name from Email Identity 226

Extracting Name from DN Identity 226

Extracting Name from EAP Identity 227

IKEv2 Authorization Policy 228

Default IKEv2 Authorization Policy 229

FlexVPN Authorization 231

Configuring FlexVPN Authorization 233

FlexVPN User Authorization 235

*FlexVPN User Authorization, Using an External
AAA Server* 235

FlexVPN Group Authorization 237

	<i>FlexVPN Group Authorization, Using a Local AAA Database</i>	238
	<i>FlexVPN Group Authorization, Using an External AAA Server</i>	239
	FlexVPN Implicit Authorization	242
	<i>FlexVPN Implicit Authorization Example</i>	243
	FlexVPN Authorization Types: Co-existence and Precedence	245
	<i>User Authorization Taking Higher Precedence</i>	247
	<i>Group Authorization Taking Higher Precedence</i>	249
	FlexVPN Configuration Exchange	250
	Enabling Configuration Exchange	250
	FlexVPN Usage of Configuration Payloads	251
	Configuration Attributes and Authorization	253
	Configuration Exchange Examples	259
	FlexVPN Routing	264
	Learning Remote Subnets Locally	265
	Learning Remote Subnets from Peer	266
	Summary	268
Chapter 9	FlexVPN Server	269
	Sequence of Events	270
	EAP Authentication	271
	EAP Methods	272
	EAP Message Flow	273
	EAP Identity	273
	EAP Timeout	275
	EAP Authentication Steps	275
	Configuring EAP	277
	EAP Configuration Example	278
	AAA-based Pre-shared Keys	283
	Configuring AAA-based Pre-Shared Keys	284
	RADIUS Attributes for AAA-Based Pre-Shared Keys	285
	AAA-Based Pre-Shared Keys Example	285
	Accounting	287
	Per-Session Interface	290
	Deriving Virtual-Access Configuration from a Virtual Template	291
	Deriving Virtual-Access Configuration from AAA Authorization	293
	<i>The interface-config AAA Attribute</i>	293

Deriving Virtual-Access Configuration from an Incoming Session	294
Virtual-Access Cloning Example	295
Auto Detection of Tunnel Transport and Encapsulation	297
RADIUS Packet of Disconnect	299
Configuring RADIUS Packet of Disconnect	300
RADIUS Packet of Disconnect Example	301
RADIUS Change of Authorization (CoA)	303
Configuring RADIUS CoA	304
RADIUS CoA Examples	305
<i>Updating Session QoS Policy, Using CoA</i>	305
<i>Updating the Session ACL, Using CoA</i>	307
IKEv2 Auto-Reconnect	309
Auto-Reconnect Configuration Attributes	310
Smart DPD	311
Configuring IKEv2 Auto-Reconnect	313
User Authentication, Using AnyConnect-EAP	315
AnyConnect-EAP	315
<i>AnyConnect-EAP XML Messages for User Authentication</i>	316
Configuring User Authentication, Using AnyConnect-EAP	318
AnyConnect Configuration for Aggregate Authentication	320
Dual-factor Authentication, Using AnyConnect-EAP	320
<i>AnyConnect-EAP XML Messages for dual-factor authentication</i>	322
Configuring Dual-factor Authentication, Using AnyConnect-EAP	324
RADIUS Attributes Supported by the FlexVPN Server	325
Remote Access Clients Supported by FlexVPN Server	329
<i>FlexVPN Remote Access Client</i>	329
<i>Microsoft Windows7 IKEv2 Client</i>	329
<i>Cisco IKEv2 AnyConnect Client</i>	330
Summary	330
Reference	330

Chapter 10 FlexVPN Client 331

Introduction	331
FlexVPN Client Overview	332
FlexVPN Client Building Blocks	333
<i>IKEv2 Configuration Exchange</i>	334

<i>Static Point-to-Point Tunnel Interface</i>	334
<i>FlexVPN Client Profile</i>	334
<i>Object Tracking</i>	334
NAT	335
FlexVPN Client Features	335
<i>Dual Stack Support</i>	335
<i>EAP Authentication</i>	335
<i>Dynamic Routing</i>	335
<i>Support for EzVPN Client and Network Extension Modes</i>	336
<i>Advanced Features</i>	336
Setting up the FlexVPN Server	336
EAP Authentication	337
Split-DNS	338
Components of Split-DNS	340
Windows Internet Naming Service (WINS)	343
Domain Name	344
FlexVPN Client Profile	345
Backup Gateways	346
Resolution of Fully Qualified Domain Names	346
Reactivating Peers	346
Backup Gateway List	347
Tunnel Interface	347
Tunnel Source	348
Tunnel Destination	349
Tunnel Initiation	350
Automatic Mode	350
Manual Mode	350
Track Mode	350
<i>Tracking a List of Objects, Using a Boolean Expression</i>	350
Dial Backup	352
Backup Group	353
Network Address Translation	354
Design Considerations	356
Use of Public Key Infrastructure and Pre-Shared Keys	356
The Power of Tracking	356
<i>Tracked Object Based on Embedded Event Manager</i>	356

Troubleshooting FlexVPN Client	358
Useful Show Commands	358
Debugging FlexVPN Client	360
Clearing IKEv2 FlexVPN Client Sessions	360
Summary	361

Chapter 11 FlexVPN Load Balancer 363

Introduction	363
Components of the FlexVPN Load Balancer	363
IKEv2 Redirect	363
Hot Standby Routing Protocol	366
FlexVPN IKEv2 Load Balancer	367
Cluster Load	369
IKEv2 Redirect	372
Redirect Loops	373
FlexVPN Client	374
Troubleshooting IKEv2 Load Balancing	374
IKEv2 Load Balancer Example	376
Summary	379

Chapter 12 FlexVPN Deployments 381

Introduction	381
FlexVPN AAA-Based Pre-Shared Keys	381
Configuration on the Branch-1 Router	382
Configuration on the Branch-2 Router	383
Configuration on the Hub Router	383
Configuration on the RADIUS Server	384
FlexVPN User and Group Authorization	386
FlexVPN Client Configuration at Branch 1	386
FlexVPN Client Configuration at Branch 2	387
Configuration on the FlexVPN Server	387
Configuration on the RADIUS Server	388
Logs Specific to FlexVPN Client-1	389
Logs Specific to FlexVPN Client-2	390
FlexVPN Routing, Dual Stack, and Tunnel Mode Auto	391
FlexVPN Spoke Configuration at Branch-1	392
FlexVPN Spoke Configuration at Branch-2	394
FlexVPN Hub Configuration at the HQ	395
Verification on FlexVPN Spoke at Branch-1	397

Verification on FlexVPN Spoke at Branch-2	399
Verification on the FlexVPN Hub at HQ	401
FlexVPN Client NAT to the Server-Assigned IP Address	404
Configuration on the FlexVPN Client	404
Verification on the FlexVPN Client	405
FlexVPN WAN Resiliency, Using Dynamic Tunnel Source	407
FlexVPN Client Configuration on the Dual-Homed Branch Router	408
Verification on the FlexVPN Client	409
FlexVPN Hub Resiliency, Using Backup Peers	411
FlexVPN Client Configuration on the Branch Router	411
Verification on the FlexVPN Client	412
FlexVPN Backup Tunnel, Using Track-Based Tunnel Activation	414
Verification on the FlexVPN Client	415
Summary	416

Part VI IPsec VPN Maintenance

Chapter 13 Monitoring IPsec VPNs 417

Introduction to Monitoring	417
Authentication, Authorization, and Accounting (AAA)	418
NetFlow	418
Simple Network Management Protocol	419
<i>VRF-Aware SNMP</i>	420
Syslog	421
Monitoring Methodology	422
IP Connectivity	423
VPN Tunnel Establishment	425
<i>Cisco IPsec Flow Monitor MIB</i>	425
<i>SNMP with IKEv2</i>	425
<i>Syslog</i>	428
Pre-Shared Key Authentication	429
PKI Authentication	431
EAP Authentication	434
Authorization Using RADIUS-Based AAA	436
Data Encryption: SNMP with IPsec	437
Overlay Routing	439
Data Usage	440
Summary	443
References	443

Chapter 14 Troubleshooting IPsec VPNs 445

Introduction	445
Tools of Troubleshooting	446
Show Commands	447
Syslog Messages	447
Event-Trace Monitoring	447
Debugging	449
<i>IKEv2 Debugging</i>	449
<i>IPsec Debugging</i>	453
Key Management Interface Debugging	453
PKI Debugging	456
Conditional Debugging	456
IP Connectivity	457
VPN Tunnel Establishment	460
IKEv2 Diagnose Error	460
Troubleshooting the IKE_SA_INIT Exchange	461
<i>Troubleshooting the IKE_AUTH Exchange</i>	464
Authentication	464
Troubleshooting RSA or ECDSA Authentication	465
Certificate Attributes	469
Debugging Authentication Using PKI	470
Certificate Expiry	470
Matching Peer Using Certificate Maps	472
Certificate Revocation	473
Trustpoint Configuration	476
Trustpoint Selection	476
Pre-Shared Key	478
Extensible Authentication Protocol (EAP)	480
Authorization	485
Data Encryption	488
Debugging IPsec	488
IPsec Anti-Replay	491
Data Encapsulation	495
Mismatching GRE Tunnel Keys	495
Overlay Routing	495

Static Routing	496
IKEv2 Routing	496
Dynamic Routing Protocols	498
Summary	499
References	502

Part VII IPsec Overhead

Chapter 15 IPsec Overhead and Fragmentation 503

Introduction	503
Computing the IPsec Overhead	504
General Considerations	504
IPsec Mode Overhead (without GRE)	505
GRE Overhead	505
Encapsulating Security Payload Overhead	507
Authentication Header Overhead	509
Encryption Overhead	510
Integrity Overhead	511
Combined-mode Algorithm Overhead	512
Plaintext MTU	513
Maximum Overhead	514
<i>Maximum Encapsulation Security Payload Overhead</i>	515
<i>Maximum Authentication Header Overhead</i>	516
<i>Extra Overhead</i>	516
IPsec and Fragmentation	518
Maximum Transmission Unit	518
Fragmentation in IPv4	519
Fragmentation in IPv6	522
Path MTU Discovery	523
TCP MSS Clamping	525
<i>MSS Refresher</i>	525
<i>MSS Adjustment</i>	526
IPsec Fragmentation and PMTUD	527
Fragmentation on Tunnels	531
<i>IPsec Only (VTI)</i>	531
<i>GRE Only</i>	532
<i>GRE over IPsec</i>	534
<i>Tunnel PMTUD</i>	534
The Impact of Fragmentation	535

Summary 536

References 536

Part VIII Migration to IKEv2

Chapter 16 Migration Strategies 539

Introduction to Migrating to IKEv2 and FlexVPN 539

Consideration when Migrating to IKEv2 539

Hardware Limitations 540

Current VPN Technology 540

Routing Protocol Selection 541

Restrictions When Running IKEv1 and IKEv2 Simultaneously 541

Current Capacity 542

IP Addresses 543

Software 543

Amending the VPN Gateway 543

Global IKE and IPsec Commands 543

FlexVPN Features 544

Familiarization 545

Client Awareness 545

Public Key Infrastructure 545

Internet Protocol Version 6 546

Authentication 546

High Availability 547

Asymmetric Routing 547

Migration Strategies 548

Hard Migration 548

Soft Migration 549

Soft Migration Example 550

Migration Verification 559

Consideration for Topologies 561

Site-to-Site 561

Hub and Spoke 562

Remote Access 565

Summary 566

Index 567

Foreword

Dear reader,

My name is Frederic Detienne and I had the chance to participate during the early days and eventually to lead cryptographic product development in Cisco, acting as Architect of DMVPN and FlexVPN. I started as a TAC Engineer and have evolved into network designer and consultant, inside Cisco and toward our customers. I work across functions with our Engineering, Advanced Services, Support, and Marketing departments.

It seems like yesterday or many eons ago ... In the beginning were crypto maps.

A few dinosaurs (like myself) started their journey in cryptographic protocols and algorithms when Cisco released CET (Cisco Encryption Technology) on IOS 11.2 in August 2003.

My only exposure to cryptography had been strictly theoretical, as a student at the University of Liège 7 years before. I suppose I was lucky to have had such a background as around me, nobody seemed to have received any crypto education nor felt inclined toward that very obscure technology. Before that, cryptography was managed through very complex systems, mostly reserved to governments and militaries.

CET was commercial grade in the sense that it was a major simplification over the former systems. It allowed a mere mortal to configure a very regular and relatively cheap router (Cisco 2500) to encrypt data across a public Layer 3 network. The cryptographic algorithms were very good: DES, then 3-DES, Diffie-Hellman key exchange. At 160 Kbps, the throughput was acceptable in those days.

In the aggregation services, technology goes through 4 steps: make it work, make it work reliably, make it work at speed, make it work at scale. There are other timelines of interest, but this one mattered particularly for cryptographic VPNs.

CET evolved into IKE/ISAKMP + IPsec as the drafts matured into standards under the leadership of Dan Harkins.

Nobody really knew what we were going toward. The initial code inherited from CET which we still had to support for our early adopters. It was also modular and ready to accommodate future enhancements, optimizations and hardware architectures. In a word: it was messy.

The data-plane vs control-plane separation outlined into RFC2408 was both a blessing and a curse. On one hand it brought complexity, on the other, it brought good OSI and code separation without which we may not have survived.

At the control plane level, IKE itself and its rekey complexity, the differences in behavior between IKE SA rekeys and IPsec SA rekeys triggered many race conditions. Overall, we managed to stabilize the system and we “made it to work reliably.” Step 2 was complete.

In the data plane things were less rosy. Crypto maps quickly showed their limits:

- the combinatory explosion of source/destination pairs on ever larger and complex networks

- code complexity due to packets being stolen in OSI layer 2 and re-encapsulated into a new IP header (OSI layer 3)

The security policy size explosion made mesh networks totally unmanageable. Besides, the security policy was mostly a static transcription of information we already had in the dynamic routing table, which led to customer frustration.

A few site-to-site and hub-and-spoke configurations were possible, but manageability and scalability suffered badly. The TAC was recommending the use of GRE protected by IPsec in order to run routing protocols on top of the tunnels. This quickly became the preferred way to deploy complex meshes. Scalability was relatively limited due to hardware performance, but it really made everyone's life easier and the support of those network became very pleasant with more and more satisfied customers.

Meanwhile, EasyVPN had appeared and was offering a remote access solution. The clients were either PC software or small branch routers. The big advantage was that the hub configuration was very compact—a few lines would allow hundreds of remote branches to connect. Unfortunately, the underlying implementation relied on crypto maps and suffered from quality and supportability issues. While EasyVPN was very good, it was not stable enough compared to the GRE/IPsec solution we used in mesh.

Customers had to choose between easy of configuration for large but simple hub-and-spoke networks and a more complex configuration for mesh networks.

One day, someone showed me NHRP: a protocol to establish circuits on demand. The code was very crude and incomplete but the developer (who had left Cisco by then) had provisioned for GRE tunnels, very likely in order to test his code without expensive equipment. I had this light bulb moment and hacked together a prototype to encrypt those GRE tunnels as they were created.

DMVPN was born in a TAC lab in Brussels, demonstrated to our colleagues in San Jose, California, and developed into a product.

We now had something that worked well, was satisfactorily stable despite being a fresh feature, and offered an easy configuration for complex networks. It started as DMVPN phase 1 with hub-and-spoke only and followed quickly with DMVPN phase 2 allowing dynamic branch-to-branch tunnel creation.

Scale was not there yet though. The IGP (OSPF and EIGRP mostly) caused significant burdens and deploying more than 350 nodes networks was still a burden. It may seem small today, but the bulk of the network sizes grew as technology permitted. A mesh network of 350 nodes was fantastic back then. Just that the market quickly got used to it and demand for more appeared.

The market demanded that we scale up both the tunnel density (the number of tunnels on a single given platform) and horizontally (the ability of a cluster of DMVPN hubs to collaborate to service). DMVPN phase 2 daisy chaining was a dreaded system to design and troubleshoot. Besides scalability problems, it also suffered from reconvergence time and convoluted configuration.

The workload reduced dramatically while market shares and revenue took off; we started work on scaling DMVPN before it became too stringent.

The semantic of the NHRP redirect and NHRP resolution forwarding appeared and helped us scale almost limitlessly across hubs. You could literally have dozens of hubs working in cluster mode. Also importantly, we could finally get away from the traditional IGP and investigate lighter protocols such as RIP, OTV and even BGP (which is feature-rich and complex at large but out of which we only needed the simplest elements). We could now scale to about 1500 peers per hub and a virtually limitless number of hubs. Each additional hub would linearly add its capability to the cluster. This was an important step forward in network design.

The biggest challenge was now to educate our customers and sales team about the various options and design. When you work on Crypto VPNs all day, every day, it is easy to forget that very few people actually understand the ins and outs of every feature and design. Also, many customers were satisfied with what they had and had no reason to investigate for more—or even suspect that something better could exist.

A metric of complexity could be seen in our 8 hours CiscoLive session going over our multiple Crypto VPN solutions and their use case:

- Crypto maps
- Easy VPN (client mode and network extension mode)
- Enhanced Easy VPN
- GRE/IPsec
- GET VPN
- DMVPN phase 1, 2 and 3

The pros and cons panned out as below:

- Crypto maps were still as limited and terrible as before but are necessary for third-party integration as they offer minimal compatibility with devices that have minimal functionality.
- EasyVPN supported remote access (especially the software client) compact but the feature had grown organically and the UI was terrible; it was also crypto map based, and its quality was poor.
- Enhanced Easy VPN solved the crypto map problem and was a major improvement over Easy VPN, but it did not enjoy proper marketing and remained poorly adopted. The UI was the same and hence difficult.

- GRE/IPsec was slowly disappearing at the benefit of DMVPN and tunnel protection in the site-to-site scenarios.
- GET VPN has lower security and limited scalability, but it is lighter on resources when used properly, if the use case is adequate. Notably, it allows native multicast.
- DMVPN was growing in both the hub-and-spoke and partial mesh cases, but the routing protocol was a deterrent for Security Operations who preferred using EasyVPN.

This really meant 8 hours during which we barely had the time to describe how a solution worked and what use case it was best for.

Customers who were successively shopping for a remote access solution, then a site-to-site, then a dynamic mesh ... had to study and learn new ways of designing and troubleshooting for each feature, over and again.

The complexity we were witnessing in TAC on our fresh recruits was impacting our customers, partners, Advanced Services, and sales teams.

At the same time, as all things so far, after a few years, market demand slowly started to outgrow DMVPN. Tunnel density still had to increase and the routing protocols were not scaling anymore.

We decided to merge EasyVPN and DMVPN features into a single feature that would offer us the advantages of both under a single feature set: one time learning, applicable always. The characteristics had to be the following:

- clear, consistent, compact, and powerful CLI: simple things ought to be simple to configure, complex things ought to be possible.
- using routing protocols should be a customer choice, not mandatory.
- NHRP usage could decrease except for spoke-spoke tunnel creation
- increased scale to 10,000 tunnels per hub at least.
- all the remote access management features had to be applicable to site-to-site and hub-and-spoke (AAA authorization in particular to apply per user QoS, ACLs, and so on.)
- reduce the reliance on PKI and make pre-shared keys more manageable. Both had to be possible, at least for hub-and-spoke
- backup and load balancing scenario
- third-party interoperability
- high serviceability/troubleshootability
- reduced learning time by using consistent protocol and data flows
- state of the art security at the cryptographic and network level

Because we could not take the risk to break IKEv1 stability nor invest in a protocol that was slated to disappear, we used IKEv2 as an inflection point to do things right. Clean implementation, clean user interface.

Today, we are capable of offering combined training, including hands-on experience, covering remote-access, hub-and-spoke, dynamic mesh, AAA management, and some troubleshooting in 4 (fours) hours. The total training time has decreased by an order of magnitude.

FlexVPN is not perfect and is not the end of the road, but in terms of applicability and total cost of ownership, taking in account training time and supportability, this is the best we have ever had.

I hope you will have as much pleasure discovering FlexVPN in this book as we had developing those features, thinking about you, our users, our customers, our sponsors.

None of this would have happened without great individuals who went beyond the basic market analysis that a typical Product Management team performs and took it on themselves to listen to our customers' real demands.

Namely, it took the courage of one Senior Manager, Pratima Sethi, to sponsor and execute on the development of FlexVPN. She also made DMVPN and EasyVPN successful; she understood deeply the need of post-deployment capabilities such as monitoring and troubleshooting and made it all possible.

The authors of this book, Amjad Inamdar and Graham Bartlett, are long-time collaborators who also deeply impacted all our VPN solutions, and I am very proud to work with them.

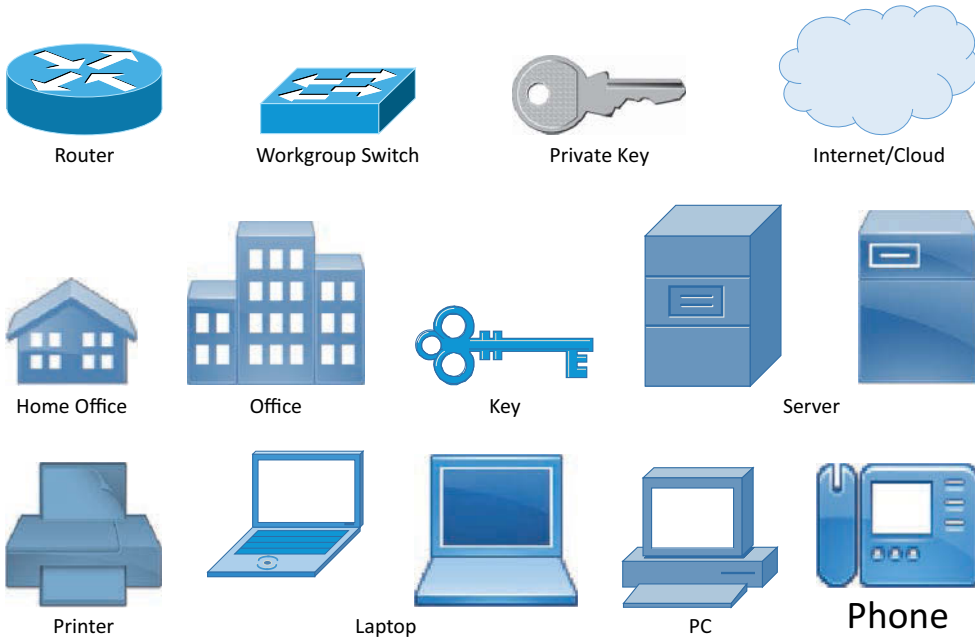
The teams prominent members included Alexandre Honore, Olivier Pelerin, Wen Zhang, Raffaele Brancaleoni, Sairam Yeleshwarapu, Saikrishna Adoni, Tapesh Maheshwari, Raghunandan P., and many others to whom I apologize for not citing.

Frederic Detienne

Distinguished Services Engineer

Cisco

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

Introduction

The motivation for writing this book was to educate users and customers about the benefits that FlexVPN and IKEv2 bring and provide an in-depth coverage of the building blocks and topics related to IPsec VPNs in general, in an easy-to-understand manner. FlexVPN was a breath of fresh air with regards to VPN technologies; for the first time all VPN technologies could be configured under a single CLI construct. We want to educate users so that secure, efficient VPN technologies can be implemented not only using Cisco IOS, but with third-party equipment also.

FlexVPN has allowed IKEv2 and IPsec VPNs on Cisco IOS to become a lot more user friendly; IKE, IPsec, concepts of cryptography, and VPNs can be hard subjects to understand. This book is intended to explain these topics and allow the reader to not only grasp the concepts, but master them.

The book explains how IPsec VPNs deal with NAT traversal, fragmentation, segmentation, IP dual stack, multicast, non-IP protocols and so on.

When VPNs are configured, there are a plethora of options; this book is intended to clarify these with ample illustrations and configuration examples so technologies are implemented in a secure and streamlined fashion.

When we talk to customers, many are unaware of what's happening under the hood and what impact a certain command will have. This book tries to clarify these points.

Goals and Methods

Provide a guide that will take the reader from knowing very little about VPN technologies to having an in-depth understanding.

Prevent customers from making mistakes that lead to network down scenarios or put the overall architecture at risk.

Give architects an understanding of the technology, allowing them to design VPN systems that meet business needs.

Give designers the knowledge where to position certain features.

Give implementors the understanding of how various features work along with end-to-end configurations examples that can be used as reference.

Give support staff an understanding of the protocols, configuring them, and how features integrate. This will result in a deep understanding, enabling timely debugging and troubleshooting.

Provide Security Operation Center guidance on telemetry that can be gained when an IKE and IPsec SA are created. This provides a methodology to perform monitoring and troubleshooting.

Provide advice and guidance on how to migrate existing IKEv1 architecture to using IKEv2.

Allow accreditors to understand technologies, resulting in assurance that the architecture presented will meet the intended security requirements.

Give project managers an understanding of the components required to perform migrations from IKEv1 to IKEv2.

Who Should Read This Book?

Anyone that is involved with the lifecycle of deploying an IPsec VPN. This includes architects, designers, security engineers, support engineers, accreditors, and members of a SOC/NOC.

This book tries to explain the protocols at an RFC level, so it will provide the reader with an understanding that is not just specific to Cisco, but is applicable to any standards-based implementation.

For any individual that is developing services that are consumed by an IPsec VPN architecture (RADIUS, PKI, and similar ones), this book allows the reader to understand the protocol flows and the interaction between IKEv2/IPsec and their services.

VPN technologies are an integral part of the many of the Cisco certification tracks. This book would serve as a valuable study aid providing an in-depth coverage of the IPsec VPN foundational topics in an easy to understand manner. Some of such certification tracks are

- Security-CCNA, CCNP and CCIE
- Routing and Switching-CCNA, CCNP and CCIE
- Design-CCDA, CCDP, CCDE
- Service provider-CCIE

Simply put—if you want to understand IPsec VPN building blocks and architectures, and deploying IPsec VPNs when using IKEv2 this book is for you.

How this book is organized

This book contains a structured approach to VPN technologies.

Chapter 1 Introduction to IPsec VPNs

This chapter describes the purpose of VPNs and the types of cryptography (symmetric and asymmetric). We cover cryptographic protocols used in the generation of IPsec VPNs.

We explain how confidentiality and integrity are achieved using Encapsulation Security Payload (ESP) and how integrity is achieved using Authentication Header (AH).

We introduce IKE and IPsec and the relationship that these have.

This chapter describes the components that make up IPsec, including the Security Parameter Index (SPI), Security Policy Database (SPD), Security Association Database (SADB), Peer Authorization Database (PAD), lifetime, and sequence numbers. We explain how these are interlinked and what relationship exists.

The two modes of IPsec, tunnel and transport, are described. We explain the benefits of each. The benefits of ESP version 3 are described.

Chapter 2 IKEv2 the Protocol

The Internet Key Exchange (IKE) protocol is described in detail. The format of the IKE header and the various packet exchanges (SA_INIT, IKE_AUTH, INFORMATIONAL, CREATE_CHILD_SA) are described. You will understand how the IKE SA is created and the components that are used to construct this, such as key material generation.

Features of IKEv2, such as anti-replay, the anti-DDoS cookie, configuration payload, and acknowledged responses, are described along with the protocols used by IKE; encryption, integrity, PRF, and Diffie-Hellman are listed.

This chapter details how IKEv2 operates when NAT is used on the transport network. The various keepalive mechanisms are covered, including IKE and NAT keepalives. This chapter covers a number of additional IKEv2 related RFCs.

Chapter 3 Comparison of IKEv1 and IKEv2

Within this chapter the history of IPsec and IKEv1 is covered, including all the RFCs (2401 to 2412) that were created to define the implementation of IKEv1-based IPsec VPNs. The key similarities and the key differences of IKEv2 compared to IKEv1 are covered, including exchange modes, authentication, use of identities, anti-DDoS, lifetimes, and many more topics.

Chapter 4 IOS IPsec implementation

The specific types of VPN implementation of Cisco IOS and IOS-XE are introduced. This chapter describes how to implement tunnel or transport mode. The two encapsulation types, GRE and VTI, are described, along with their benefits and limitations. The various implementation modes (dual stack, mixed mode, and auto) are covered. We also introduce VRF-aware IPsec.

Chapter 5 IKEv2 Configuration

This chapter contains an overview of the IKEv2 configuration features and how these interoperate. The various components of IKEv2 are covered: IKEv2 proposal, IKEv2 policy, IKEv2 profile, IKEv2 keyring, and the IKEv2 global configuration. We also cover other components that are critical to configuring IPsec VPNs, such as PKI and IPsec. The powerful pre-configured attributes are introduced, and their benefits are explained.

Chapter 6 Advanced IKEv2 features

This chapter covers IKEv2 advanced features, including some that are not part of the standard IKEv2 RFC. IKEv2 fragmentation and the transportation of Security Group Tags (SGT) are described, along with the methods to delete a session should a peer be revoked or the peer's certificate expire. The lifetime of the IKEv2 session is examined and the effect this has is described in detail.

Chapter 7 IKEv2 deployments

This chapter described a number of scenarios to give the reader an understanding of the various types of IKEv2 deployments. Both IPv4 and IPv6 are covered, with authentication using pre-shared keys, RSA certificates, ECDSA certificates, and HTTP URL Cert. The IKE anti-DDoS mechanism is illustrated in detail.

Chapter 8 Introduction to FlexVPN

After an overview of FlexVPN, the tunnel interface types (static, virtual-template, and virtual-access) and IOS AAA infrastructure are described in detail. The building blocks of FlexVPN—Name Mangler, IKEv2 authorization policy, with user, group, and implicit authorization—are described. The configuration exchange is illustrated, along with advertising prefixes using IKEv2 routing.

Chapter 9 FlexVPN Server

The chapter provides an overview of FlexVPN Server. EAP authentication is described in detail, along with AAA-based pre-shared keys. Deriving virtual-access interfaces from virtual-templates is illustrated, along with automatic detection of the tunnel mode and encapsulation type using mode auto. RADIUS Packet of Disconnect and Change of Authorization (CoA) are described. The IKEv2 auto-reconnect, AnyConnect-EAP, and dual-factor authentication features are described. The FlexVPN Server supported clients are covered.

Chapter 10 FlexVPN Client

This chapter begins with an overview of FlexVPN Client. EAP authentication is described in detail. Client-specific attributes are described: split-DNS, WINS, and Domain Name. The FlexVPN client profile is described. The following specific features of FlexVPN client are illustrated: Backup gateways, dial backup, backup groups, tunnel interface types, tunnel initiation types, and FlexVPN with NAT. This chapter describes design considerations and troubleshooting specific to the FlexVPN client.

Chapter 11 FlexVPN Load Balancer

This chapter presents an overview of the FlexVPN Load Balancer. It details the core components and RFC 5685 “IKEv2 Redirect and Hot Standby Routing Protocol.” How the cluster operates, including cluster load, is detailed. FlexVPN client and server configurations are illustrated. Troubleshooting a specific FlexVPN load balancer configuration is described, and a number of example configurations are shown.

Chapter 12 FlexVPN Deployments

This chapter contains a number of example scenarios which illustrate the following FlexVPN deployments: AAA-based pre-shared key, user and group authorization, FlexVPN routing with dual-stack and tunnel mode auto, NAT with server-assigned IP addresses, WAN resilient using dynamic tunnel source, hub resiliency using backup peers, and FlexVPN backup tunnel using track-based activation.

Chapter 13 Monitoring IPsec VPNs

This chapter describes common methods for monitoring IPsec VPNs using AAA, SNMP, and syslog. A monitoring methodology is described that covers IP connectivity, VPN tunnel establishment, authentication, authorization, data encapsulation, data encryption, and overlay routing.

Chapter 14 Troubleshooting IPsec VPNs

This chapter describes the tools of troubleshooting: Event Trace Monitoring, IKEv2, IPsec, KMI and conditional debugging. Troubleshooting steps are described for IP connectivity, VPN tunnel establishment, authentication, authorization, data encapsulation, data encryption, and overlay routing.

Chapter 15 IPsec overhead and Fragmentation

This chapter describes computing IPsec overhead for ESP and AH and the effect that IPsec and fragmentation have for both IPv4 and IPv6. The following topics are illustrated: Path MTU Discovery (PMTUD), TCP MSS clamping, fragmentation, and PMTUD (specifically on tunnel interfaces). The impact of fragmentation is described.

Chapter 16 Migration Strategies

The chapter illustrates the considerations when migrating from IKEv1 to IKEv2. It covers hardware, VPN technologies, routing protocols, restrictions for IKEv1 and IKEv2, capacity planning, global commands, FlexVPN features, PKI authentication, high availability, and asymmetric routing. Migration strategies for hard and soft migrations are covered. It also discusses considerations for specific topologies: site-to-site, hub-and-spoke, and remote access. There is also an example migration scenario.

This page intentionally left blank

IKEv2 Deployments

This chapter introduces a number of designs where IKEv2 is used. Each design will use a simple deployment of two routers with the focus on the configuration of IKEv2. Although each scenario uses only two routers, the configuration can scale as required if needed.

The configuration is intended to be as simple as possible, and the emphasis is focused on the IKEv2 configuration.

Pre-shared-key Authentication with Smart Defaults

This configuration is the simplest to set up. By using smart defaults, a VPN is created between two peers using minimal configuration: only the IKEv2 profile and corresponding IKEv2 keyring are required.

Figure 7-1 illustrates the topology. The transport network is using IPv6, and the overlay network is using IPv4.

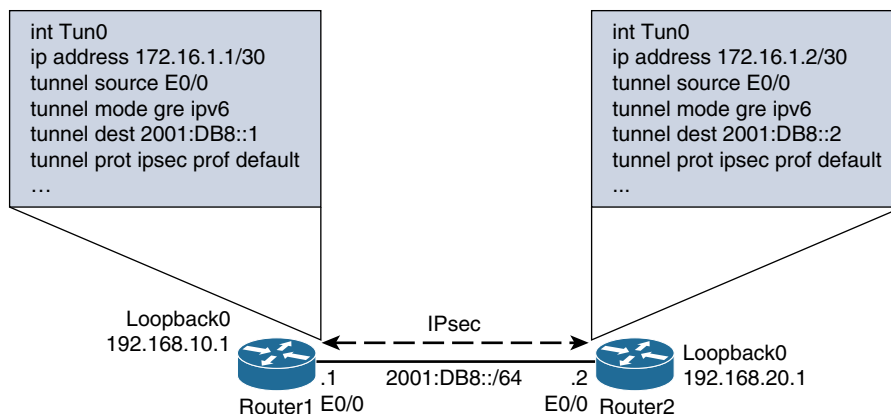


Figure 7-1 PSK Authentication with Smart Defaults Topology

The following example illustrates the relevant configuration used on Router1. This is a very minimal configuration which leaves little room for error.

Note that the shared secrets used in the example below are for illustrative purposes and, if used in a production environment, should contain sufficient entropy.

The example might seem complex as this scenario uses IPv4 and IPv6; however, the main focus of interest is to illustrate the IKEv2 configuration and the simplicity of using smart defaults.

An IKEv2 keyring is created with a peer entry which matches the peer's IPv6 address. Asymmetric pre-shared-keys are used with each device having a unique local and remote key.

```
crypto ikev2 keyring local_keyring
peer 2001:DB8::2
  address 2001:DB8::2/128
  pre-shared-key local bartlett
  pre-shared-key remote inamdar
```

The IKEv2 profile is the mandatory component and matches the remote IPv6 address configured on Router2. The local IKEv2 identity is set to the IPv6 address configured on E0/0. The authentication is set to pre-shared-key with the locally configured keyring defined previously.

```
crypto ikev2 profile default
match identity remote address 2001:DB8::2/128
identity local address 2001:DB8::1
authentication remote pre-share
authentication local pre-share
keyring local local_keyring
```

The local loopback interface is configured, which will allow testing over the IPsec Security Association.

```
interface Loopback0
ip address 192.168.10.1 255.255.255.0
```

The tunnel interface is created as tunnel mode GRE IPv6. This is required as the transport network is IPv6 and the overlay is IPv4. The default IPsec profile is used to protect this interface; this uses the default IKEv2 profile which was configured earlier.

```
interface Tunnel0
ip address 172.16.1.1 255.255.255.252
tunnel source Ethernet0/0
tunnel mode gre ipv6
tunnel destination 2001:DB8::2
tunnel protection ipsec profile default
```

The physical interface used as the tunnel source uses IPv6.

```
interface Ethernet0/0
  no ip address
  ipv6 address 2001:DB8::1/64
```

Enhanced interior gateway routing protocol (EIGRP) is used to establish a peer relationship over the tunnel interface and distribute the loopback prefix.

```
router eigrp 1
  network 172.16.1.0 0.0.0.3
  network 192.168.10.0
```

The following example illustrates the relevant configuration on Router2.

```
interface Loopback0
  ip address 192.168.20.1 255.255.255.0

interface Tunnel0
  ip address 172.16.1.2 255.255.255.252
  tunnel source Ethernet0/0
  tunnel mode gre ipv6
  tunnel destination 2001:DB8::1
  tunnel protection ipsec profile default

interface Ethernet0/0
  no ip address
  ipv6 address 2001:DB8::2/64

router eigrp 1
  network 172.16.1.0 0.0.0.3
  network 192.168.20.0
```

The following example illustrates the EIGRP neighbor relationship built over the tunnel interface. The prefix for IP address assigned to the loopback interface on Router2 is reachable via the protected tunnel.

```
Router1#show ip route 192.168.20.0
Routing entry for 192.168.20.0/24
  Known via "eigrp 1", distance 90, metric 27008000, type internal
  Redistributing via eigrp 1
  Last update from 172.16.1.2 on Tunnel0, 00:12:04 ago
  Routing Descriptor Blocks:
  * 172.16.1.2, from 172.16.1.2, 00:12:04 ago, via Tunnel0
    Route metric is 27008000, traffic share count is 1
    Total delay is 55000 microseconds, minimum bandwidth is 100 Kbit
    Reliability 255/255, minimum MTU 1418 bytes
    Loading 1/255, Hops 1
```

The following example illustrates the IKEv2 SA that is created. The IKEv2 SA is protected by the PRF and integrity algorithms using SHA512, encryption using AES-CBC-256, and Diffie-Hellman group 5, which are the most preferred algorithms within the IKEv2 default proposal. The authentication is performed using pre-shared-key.

```
Router1#show crypto ikev2 sa detailed
IPv4 Crypto IKEv2 SA

IPv6 Crypto IKEv2 SA

Tunnel-id      fvrf/ivrf              Status
1              none/none                READY
Local   2001:DB8::1/500
Remote   2001:DB8::2/500
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign:
PSK, Auth verify: PSK
Life/Active Time: 86400/10523 sec
CE id: 1002, Session-id: 2
Status Description: Negotiation done
Local spi: 261B9BD2F208A02A      Remote spi: 0B28D2A21FC6304D
Local id: 2001:DB8::1
Remote id: 2001:DB8::2
Local req msg id: 4                Remote req msg id: 4
Local next msg id: 4              Remote next msg id: 4
Local req queued: 4                Remote req queued: 4
Local window: 5                    Remote window: 5
...
```

The following example illustrates traffic being sent over the IPsec Security Association. The tunnel source and destination being the IPv6 addresses configured on the physical E0/0 interfaces.

Traffic is sent via the tunnel interface, from the locally configured loopback interface to the loopback on Router2.

```
Router1#ping 192.168.20.1 source 192.168.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.10.1

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms
```

The IPsec Security Association is verified where the default IPsec transform set is used, which is created using Encapsulation Security Payload with AES-CBC-256 for encryption and SHA1-HMAC for integrity. Transport mode is used.

```
Router1#show crypto ipsec sa
```

```
interface: Tunnel0
```

```
  Crypto-map tag: Tunnel0-head-0, local addr 2001:DB8::1
```

```
protected vrf: (none)
```

```
local ident (addr/mask/prot/port): (2001:DB8::1/128/47/0)
```

```
remote ident (addr/mask/prot/port): (2001:DB8::2/128/47/0)
```

```
current_peer 2001:DB8::2 port 500
```

```
  PERMIT, flags={origin_is_acl,}
```

```
  #pkts encaps: 523, #pkts encrypt: 523, #pkts digest: 523
```

```
  #pkts decaps: 523, #pkts decrypt: 523, #pkts verify: 523
```

```
  #pkts compressed: 0, #pkts decompressed: 0
```

```
  #pkts not compressed: 0, #pkts compr. failed: 0
```

```
  #pkts not decompressed: 0, #pkts decompress failed: 0
```

```
  #send errors 0, #recv errors 0
```

```
local crypto endpt.: 2001:DB8::1,
```

```
remote crypto endpt.: 2001:DB8::2
```

```
plaintext mtu 1462, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb Ethernet0/0
```

```
current outbound spi: 0x5FC3C94A(1606666570)
```

```
PFS (Y/N): N, DH group: none
```

```
inbound esp sas:
```

```
  spi: 0xB8435B94(3091422100)
```

```
  transform: esp-aes esp-sha-hmac ,
```

```
  in use settings = {Transport, }
```

```
  conn id: 10, flow_id: SW:10, sibling_flags 80000001, crypto-map:
```

```
Tunnel0-head-0
```

```
  sa timing: remaining key lifetime (k/sec): (4315844/2543)
```

```
  IV size: 16 bytes
```

```
  replay detection support: Y
```

```
  Status: ACTIVE(ACTIVE)
```

```
inbound ah sas:
```

```
inbound pcg sas:
```

```
outbound esp sas:
```

```
  spi: 0x5FC3C94A(1606666570)
```

```
  transform: esp-aes esp-sha-hmac ,
```

```
  in use settings = {Transport, }
```

```
  conn id: 9, flow_id: SW:9, sibling_flags 80000001, crypto-map:
```

```
Tunnel0-head-0
```

```
sa timing: remaining key lifetime (k/sec): (4315844/2543)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE (ACTIVE)
```

Elliptic Curve Digital Signature Algorithm Authentication

The scenario looks to use digital signatures to authenticate both peers. Rather than the more common RSA certificates, Elliptic Curve (EC) certificates are used that provide the ability to authenticate both parties, using the Elliptic Curve Digital Signature Algorithm (ECDSA).

The configuration in this example is intended to be simple, with the main focus on the IKEv2 configuration.

Figure 7-2 illustrates the physical IP addressing and the setup of the tunnel interface.

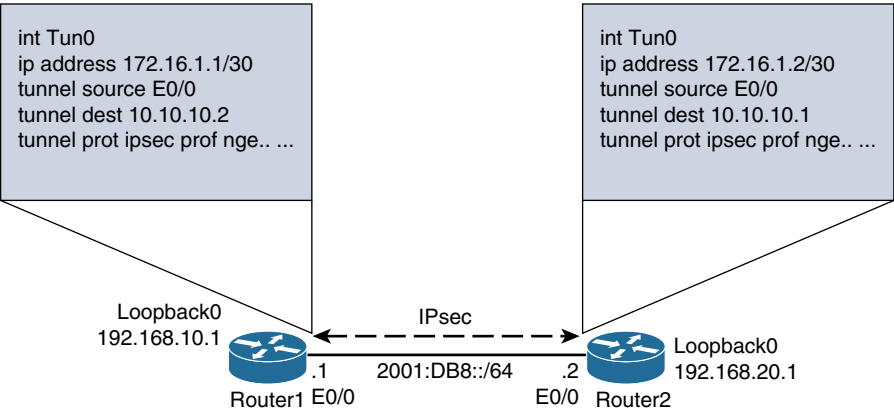


Figure 7-2 Topology with Configuration

In addition to ECDSA for authentication, Cisco Next Generation Encryption (NGE) algorithms secure the IKEv2 and IPsec session, as shown in Table 7-1.

Table 7-1 Security Algorithms Used

Method	Algorithm
IKEv2 encryption	AES-GCM-256
IKEv2 PRF	SHA512
Diffie-Hellman	Group 21
Authentication	Elliptic Curve Digital Signature Algorithm
IPsec encryption	AES-GCM-256
IPsec PFS	Group 21

Rather than using the default IKEv2 proposal, the default IKEv2 proposal is disabled, and a new IKEv2 proposal created containing the IKEv2 algorithms defined in Table 7-1.

Static routes are used to send traffic down the freshly created tunnel interface.

The following example illustrates the configuration that is used on Router1.

Note Although not shown, the trustpoint uses a locally configured elliptic curve keypair.

The trustpoint is configured using manual enrollment, with the local and CA certificate.

```
crypto pki trustpoint ecdsa_tp
  enrollment terminal

crypto pki certificate chain ecdsa_tp
  certificate 6156E3D50000000000009
    308203BF 30820365 A0030201 02020A61 56E3D500 00000000 09300A06 082A8648
  ...
    68656c6c 6f736861 627333E4FDDC 642DA416 F57D4962 C5DF6545 FEC931FA F84BAF40
    A9829E
  quit
  certificate ca 780887F0CDD97E9E49DB893FA5D74238
    30820206 308201AB A0030201 02021078 0887F0CD D97E9E49 DB893FA5 D7423830
    0A06082A 8648CE3D 04030230 4F311330 11060A09 92268993 F22C6401 19160363
  ...
    816AA443 9191FBAC 731C
  quit
```

A certificate map is created that will match certificates containing a subject name of *cisco.com*. This is used within the IKEv2 profile to anchor the certificates presented by the peers. As this is a site-to-site VPN with only two peers, the certificate map could have been more granular to include the peer DN.

```
crypto pki certificate map certmap 10
  subject-name co cisco.com
```

The default IKEv2 proposal is disabled, and a new IKEv2 proposal is created that contains the relevant cryptographic algorithms.

Note Because this is a combined mode cipher, no integrity algorithm is required.

```
no crypto ikev2 proposal default
crypto ikev2 proposal nge
  encryption aes-gcm-256
  prf sha512
  group 21
```

An IKEv2 policy is created, which encompasses the IKEv2 proposal created above. Because the default IKEv2 proposal is disabled, this then ensures that only the IKEv2 proposal named *nge* will be used and minimizes the chance of mis-configuration.

```
crypto ikev2 policy default
  match fvrf any
  proposal nge
```

An IKEv2 profile is created, which uses the certificate map created earlier. The identity is set to DN, which will use the DN from the certificate. The authentication method is set to ECDSA and the PKI trustpoint used which was configured earlier. This profile will only match peer certificates, which contain the string *cisco.com* within the subject name. Dead-peer detection is enabled to ensure that the IKEv2 SA and corresponding IPsec Security Associations are torn down in a timely manner if IKE connectivity is lost.

```
crypto ikev2 profile nge
  match certificate certmap
  identity local dn
  authentication remote ecdsa-sig
  authentication local ecdsa-sig
  pki trustpoint ecdsa_tp
  dpd 10 2 on-demand
```

An IPsec transform set is created, which uses AES-GCM-256. Because this is a combined mode cipher, no integrity algorithm is required.

```
crypto ipsec transform-set nge-transform esp-gcm 256
  mode transport
```

The default IPsec profile is disabled, which ensures that it is not used due to mis-configuration. A new IPsec profile is created which uses the IKEv2 profile and IPsec transform-set created earlier. Additionally, perfect forward secrecy is enabled to ensure that a fresh Diffie-Hellman exchange is performed on rekey.

```
no crypto ipsec profile default

crypto ipsec profile nge-profile
  set transform-set nge-transform
  set pfs group21
  set ikev2-profile nge
```

A loopback interface is used that will allow traffic to be sourced from and destined to as it transverses the VPN.

```
interface Loopback0
  ip address 192.168.10.1 255.255.255.0
```

The tunnel interface is created with the relevant source interface configured and with the destination address of Router2. This is protected by the IPsec profile created above.

```
interface Tunnel0
 ip address 172.16.1.1 255.255.255.252
 tunnel source Ethernet0/0
 tunnel destination 10.10.10.2
 tunnel protection ipsec profile nge-profile
```

The E0/0 interface is used as the tunnel source.

```
interface Ethernet0/0
 ip address 10.10.10.1 255.255.255.0
```

A static route is configured to send all traffic for the 192.168.20.0/24 network, which is the subnet protected by the peer, via the peer tunnel IP address.

```
ip route 192.168.20.0 255.255.255.0 172.16.1.2
```

Router2 has a nearly similar configuration; the following example illustrates the unique configuration. The tunnel interface has a unique IP address, and the destination is configured as E0/0 on Router1.

Note the unique IP address and the tunnel destination of Router1.

```
interface Tunnel0
 ip address 172.16.1.2 255.255.255.252
 tunnel source Ethernet0/0
 tunnel destination 10.10.10.1
 tunnel protection ipsec profile nge-profile

interface Ethernet0/0
 ip address 10.10.10.2 255.255.255.0
```

The following example illustrates verification that the IKEv2 SA established. The algorithms used to secure the IKE session as described in Table 7-1 can be seen.

```
Router1#show crypto ikev2 sa detailed
```

```
IPv4 Crypto IKEv2 SA
```

Tunnel-id	Local	Remote	fvr/ivrf	Status
1	10.10.10.1/500	10.10.10.2/500	none/none	READY
Encr: AES-GCM, keysize: 256, PRF: SHA512, Hash: None, DH Grp:21, Auth sign: ECDSA, Auth verify: ECDSA				
Life/Active Time: 86400/6 sec				
CE id: 1030, Session-id: 13				


```

Status Description: Negotiation done
Local spi: 313404E23B3A5707      Remote spi: 13FE5BCC09FFAAAB
Local id: hostname=Router1.cisco.com
Remote id: hostname=Router2.cisco.com
Local req msg id: 2              Remote req msg id: 0
Local next msg id: 2            Remote next msg id: 0
Local req queued: 2             Remote req queued: 0
Local window: 5                 Remote window: 5
DPD configured for 10 seconds, retry 2
Fragmentation not configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

```

IPv6 Crypto IKEv2 SA

The creation of the IPsec Security Association can be seen in the following example. The tunnel interface is configured with the default GRE mode, the traffic selectors can be seen indicating this by the use of IP protocol 47.

Router1#**show crypto sockets**

Number of Crypto Socket connections 1

```

Tu0 Peers (local/remote): 10.10.10.1/10.10.10.2
  Local Ident (addr/mask/port/prot): (10.10.10.1/255.255.255.255/0/47)
  Remote Ident (addr/mask/port/prot): (10.10.10.2/255.255.255.255/0/47)
  IPSec Profile: "nge-profile"
  Socket State: Open
  Client: "TUNNEL SEC" (Client State: Active)
Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "nge-profile" Map-name: "Tunnel0-head-0"

```

The following example illustrates the route to 192.168.20.0/24, which be seen via the tunnel interface. All traffic intended for this network will be sent via the tunnel and encrypted by the corresponding IPsec Security Association.

```

Router1#show ip route 192.168.20.0 255.255.255.0
Routing entry for 192.168.20.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
    * 172.16.1.2
      Route metric is 0, traffic share count is 1
Router1#show ip route 172.16.1.2
Routing entry for 172.16.1.0/30

```

```
Known via "connected", distance 0, metric 0 (connected, via interface)
Routing Descriptor Blocks:
* directly connected, via Tunnel0
  Route metric is 0, traffic share count is 1
```

Traffic is sent from Router1 to Router2 via the tunnel interface. Note that this traffic has been protected by the IPsec Security Association, as indicated by the increasing *encaps* and *decaps* counters.

```
Router1#ping 192.168.20.1 source 192.168.10.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
```

```
Packet sent with a source address of 192.168.10.1
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/4/6 ms
```

```
Router1#show crypto ipsec sa
```

```
interface: Tunnel0
```

```
  Crypto-map tag: Tunnel0-head-0, local addr 10.10.10.1
```

```
protected vrf: (none)
```

```
local  ident (addr/mask/prot/port): (10.10.10.1/255.255.255.255/47/0)
```

```
remote ident (addr/mask/prot/port): (10.10.10.2/255.255.255.255/47/0)
```

```
current_peer 10.10.10.2 port 500
```

```
  PERMIT, flags={origin_is_acl,}
```

```
  #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
```

```
  #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
```

```
  #pkts compressed: 0, #pkts decompressed: 0
```

```
  #pkts not compressed: 0, #pkts compr. failed: 0
```

```
  #pkts not decompressed: 0, #pkts decompress failed: 0
```

```
  #send errors 0, #recv errors 0
```

```
local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.2
```

```
plaintext mtu 1466, path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
```

```
current outbound spi: 0x3FD4A2AF(1070899887)
```

```
PFS (Y/N): Y, DH group: group21
```

```
inbound esp sas:
```

```
  spi: 0x349334C6(882062534)
```

```
    transform: esp-gcm 256 ,
```

```
    in use settings = {Transport, }
```

```
    conn id: 6, flow_id: SW:6, sibling_flags 80000000, crypto-map:
```

```
Tunnel0-head-0
```

```
    sa timing: remaining key lifetime (k/sec): (4207250/3566)
```

```
    IV size: 8 bytes
```

```

        replay detection support: Y
        Status: ACTIVE (ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
    spi: 0x3FD4A2AF(1070899887)
    transform: esp-gcm 256 ,
    in use settings ={Transport, }
    conn id: 5, flow_id: SW:5, sibling_flags 80000000, crypto-map:
Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4207250/3566)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE (ACTIVE)

outbound ah sas:

outbound pcp sas:

```

RSA Authentication Using HTTP URL Lookup

In this scenario, we will use RSA certificates to authenticate both peers. However, for Router2, we will not send the certificate within the IKE AUTH exchange, but will send a HTTP URL from Router2 to Router1 to inform it where to obtain the certificate. Router1 will then retrieve the certificate from the HTTP URL and verify that the presented AUTH payload was signed by the private key relating to the public key contained within the certificate.

Router1 has been set up as a certificate authority; from this CA, a certificate is obtained for both Router1 and Router2. These certificates are used to authenticate the IKEv2 SA.

Figure 7-3 illustrates the operation of the HTTP URL lookup feature. Router2 will sign the AUTH payload with its private key. Router1 will retrieve the certificate from the HTTP server and validate the AUTH payload by using the public key obtained from the retrieved certificate.

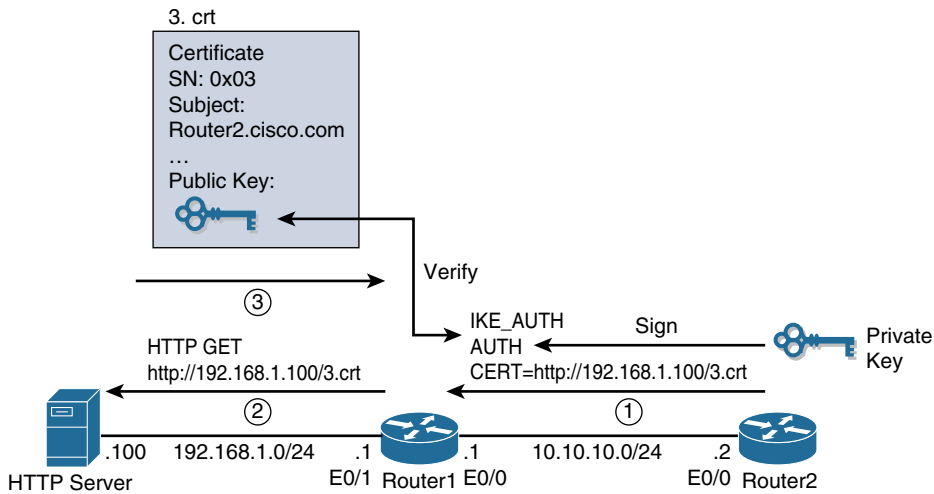


Figure 7-3 HTTP URL Lookup Feature

Note The certificate generated by the IOS CA is in Privacy Enhanced Mail (PEM) format. Although the IKEv2 RFC states that the HASH and URL feature returns a URL with the SHA1 hash of the requested certificate, Cisco IOS allows for any URL to be used. As per the IKEv2 RFC, Cisco IOS requires the obtained certificate to be in distinguished encoding rules (DER) encoding. The following example illustrates the OpenSSL commands to manually convert a certificate from PEM to DER encoding, with the PEM encoded certificate in file 3.crt.

```
openssl x509 -outform der -in 3.crt -out 3.der
```

Figure 7-4 illustrates the topology used in the tunnel interface configuration.

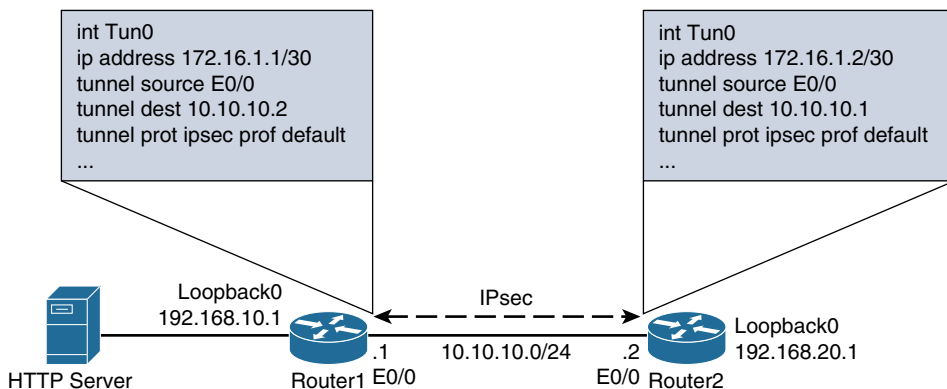


Figure 7-4 Topology with Configuration

The configuration is similar to the ECDSA example earlier, but RSA certificates are used, which results in a different authentication method. However, the base concepts are the same with regards to the PKI.

The subject information access (SIA) is an attribute within a certificate that defines some type of offered services. An example of where to access a server can be included in the SIA with a uniform resource identifier (URI). The SIA is amended to contain the URL that the peer will use for the HTTP URL lookup. This is achieved by the use of the certificate map that matches the locally used certificate and is attached to the trustpoint. This removes the inclusion of the certificate within the IKE exchange and uses the value defined in the SIA as the location for the peer to obtain the certificate.

The following example illustrates the configuration used on Router2.

The PKI trustpoint is defined; it has been authenticated, and the local device enrolled. The critical component to ensure that this client does not send its certificate but instead sends the HTTP URL is the *match certificate* command. This command will match the defined certificate map and override the SIA to contain the configured URL. This is then sent in replacement of the certificate in the IKE_AUTH exchange.

```
crypto pki trustpoint CA
  enrollment url http://10.10.10.1:80
  revocation-check crl
  match certificate local override sia 1 http://192.168.1.100/3.der
```

A certificate map is created that will match certificates containing a subject name of *router1.cisco.com*. This is used within the IKEv2 profile to anchor the peer's presented certificate.

```
crypto pki certificate map certmap 10
  subject-name eq router1.cisco.com
```

The following certificate map is used by the match statement within the trustpoint configuration to match the local certificate. This is achieved by matching the local subject name (which is not case sensitive) of *router2*.

```
crypto pki certificate map local 10
  subject-name co router2
```

The mandatory IKEv2 profile is configured which uses the certificate map created earlier. This will match any certificates which contain a subject name of *cisco.com*. The authentication method is set to RSA signatures, and the trustpoint configured earlier is used.

```
crypto ikev2 profile default
  match certificate certmap
  identity local dn
  authentication remote rsa-sig
  authentication local rsa-sig
  pki trustpoint CA
```

The tunnel interface is created with the relevant source interface configured and the destination address of Router1. This is protected by the default IPsec profile which uses the default IKEv2 profile which was created earlier.

```
interface Tunnel0
 ip address 172.16.1.2 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 10.10.10.1
 tunnel protection ipsec profile default
```

The following physical interface is used as the tunnel source.

```
interface Ethernet0/0
 ip address 10.10.10.2 255.255.255.0
```

Note Should a certificate hierarchy exist where there is a requirement to send a certificate chain with multiple URLs in multiple CERT payloads starting from “ID cert url,” “subca1,” “subca2,” until “root CA”; then each additional certificate can be included as a separate line within the trustpoint configuration as illustrated below.

```
crypto pki trustpoint CA
 match certificate local override sia 1 http://192.168.1.100/3.der
 match certificate local override sia 2 http://192.168.1.100/subca1.der
 match certificate local override sia 3 http://192.168.1.100/subca2.der
 match certificate local override sia 4 http://192.168.1.100/root.der
```

The following example illustrates the configuration used on Router1.

The certificate authority function is enabled. Note that the automatic granting of certificates is used here for ease of configuration and should not occur in a production environment where un-authenticated access to the CA can occur.

```
crypto pki server local
 database level complete
 no database archive
 grant auto
```

The relating PKI trustpoint for the IOS CA is:

```
crypto pki trustpoint local
 revocation-check crl
 rsakeypair local
```

A trustpoint is used to enroll into the local CA.

```
crypto pki trustpoint CA
 enrollment url http://10.10.10.1:80
 revocation-check crl
```

A certificate map is created that will match certificates containing a subject name of *router2.cisco.com*. This is used within the IKEv2 profile to anchor the peer's presented certificate.

```
crypto pki certificate map certmap 10
  subject-name eq router2.cisco.com
```

The mandatory IKEv2 profile is configured that uses the certificate map created earlier. This will match any certificates, which contain a subject name of *cisco.com*. The authentication method is set to RSA signatures, and the trustpoint configured earlier is used.

```
crypto ikev2 profile default
  match certificate certmap
  identity local dn
  authentication remote rsa-sig
  authentication local rsa-sig
  pki trustpoint CA
```

The tunnel interface is created with the relevant source interface configured, and the destination address of Router1. This is protected by the default IPsec profile that uses the default IKEv2 profile, which was created earlier.

```
interface Tunnel1
  ip address 172.16.1.1 255.255.255.0
  tunnel source Ethernet0/0
  tunnel destination 10.10.10.2
  tunnel protection ipsec profile default
```

The physical interface used as the tunnel source.

```
interface Ethernet0/0
  ip address 10.10.10.1 255.255.255.0
```

The physical interface used to reach the HTTP server containing the certificates.

```
interface Ethernet0/1
  ip address 192.168.1.1 255.255.255.0
```

Note When using the HTTP URL lookup feature, the router that retrieves the HTTP URL should be protected from malicious intent by restricting HTTP access to only the server storing the certificates. As the certificate obtained via the HTTP URL method is processed prior to authentication, an intruder could redirect the gateway to a large file containing garbage, or a URI that will slowly introduce a file, a little at a time, causing a DoS on the gateway. Mitigation can be achieved using controls, such as access-control-lists, control-plane policing, or control-plane protection.

The following example illustrates IKEv2 debugs taken from Router1. It can be seen that Router2 sends the IKE_AUTH exchange with the CERT payload containing the HASH and URL format. Also note the NOTIFY payload which indicates the HTTP URL method is supported.

```
IKEv2:(SESSION ID = 4,SA ID = 1):Received Packet [From 10.10.10.2:500/To
10.10.10.1:500/VRF i0:f0]
Initiator SPI : 52D538043A8E330C - Responder SPI : 5CE063D07E8745EA Message id: 1
IKEv2 IKE_AUTH Exchange REQUEST
IKEv2-PAK:(SESSION ID = 4,SA ID = 1):Next payload: ENCR, version: 2.0 Exchange
type: IKE_AUTH, flags: INITIATOR Message id: 1, length: 816
Payload contents:
VID Next payload: IDi, reserved: 0x0, length: 20
IDi Next payload: CERT, reserved: 0x0, length: 44
    Id type: DER ASN1 DN, Reserved: 0x0 0x0
CERT Next payload: CERTREQ, reserved: 0x0, length: 52
    Cert encoding Hash and URL of PKIX
CERTREQ Next payload: NOTIFY, reserved: 0x0, length: 25
    Cert encoding Hash and URL of PKIX
    NOTIFY(HTTP_CERT_LOOKUP_SUPPORTED) Next payload: AUTH, reserved: 0x0, length: 8
        Security protocol id: Unknown - 0, spi size: 0, type:
HTTP_CERT_LOOKUP_SUPPORTED
AUTH Next payload: CFG, reserved: 0x0, length: 136
    Auth method RSA, reserved: 0x0, reserved 0x0
CFG Next payload: SA, reserved: 0x0, length: 304
    cfg type: CFG_REQUEST, reserved: 0x0, reserved: 0x0
```

A short time later, Router1 opens a TCP socket with 192.168.1.100, when the certificate is obtained.

```
TCP: sending SYN, seq 2191097267, ack 0
TCP0: Connection to 192.168.1.100:80, advertising MSS 1460
TCP0: state was CLOSED -> SYNSENT [42603 -> 192.168.1.100(80)]
TCP0: state was SYNSENT -> ESTAB [42603 -> 192.168.1.100(80)]
TCP: tcb 32417230 connection to 192.168.1.100:80, peer MSS 1460, MSS is 1460
```

The following example illustrates verification on Router1 that the certificate was obtained by way of HTTP.

```
Router1#show crypto ikev2 stats ext-service
```

```
-----
AAA OPERATION                                PASSED      FAILED
-----
RECEIVING PSKEY                             0           0
AUTHENTICATION USING EAP                     0           0
```


START ACCOUNTING	0	0
STOP ACCOUNTING	0	0
AUTHORIZATION	0	0

IPSEC OPERATION	PASSED	FAILED

IPSEC POLICY VERIFICATION	3	0
SA CREATION	3	0
SA DELETION	3	0

CRYPTO ENGINE OPERATION	PASSED	FAILED

DH PUBKEY GENERATED	34	0
DH SHARED SECKEY GENERATED	29	0
SIGNATURE SIGN	28	0
SIGNATURE VERIFY	3	0

PKI OPERATION	PASSED	FAILED

VERIFY CERTIFICATE	3	0
FETCHING CERTIFICATE USING HTTP	1	0
FETCHING PEER CERTIFICATE USING HTTP	1	0
GET ISSUERS	31	0
GET CERTIFICATES FROM ISSUERS	28	0
GET DN FROM CERT	3	0

GKM OPERATION	PASSED	FAILED

GET_POLICY	0	0
SET_POLICY	0	0

The certificate that is obtained via HTTP is cached locally. By default, 200 certificates will be cached. As the certificate is cached, if the IKE session drops and is re-established, the certificate will not be required to be obtained via HTTP as it is already cached. This saves numerous HTTP requests to occur if the peer is required to re-authenticate. The following example illustrates viewing the contents of the certificate cache.

```
Router1#show crypto ikev2 certificate-cache
No of entries in ikev2 certificate-cache = 1
```

```
Certificate entry:
```

```
Certificate
```

```
  Status: Available
```

```
  Certificate Serial Number (hex): 03
```

```
  Certificate Usage: General Purpose
```

```

Issuer:
  cn=CA.cisco.com
Subject:
  Name: Router2.cisco.com
  hostname=Router2.cisco.com
Validity Date:
  start date: 10:44:26 UTC Feb 8 2016
  end date: 10:44:26 UTC Feb 7 2017
Associated Trustpoints:

```

The following example illustrates the IKEv2 SA being verified. The cryptographic algorithms used have been negotiated via the use of smart defaults. The authentication method of RSA can be seen. There is no differentiation that the certificate was received via the HTTP URL method; the authentication is performed in the same manner as RSA authentication when certificates are sent in the IKE_AUTH exchange.

```
Router1#show crypto ikev2 sa detailed
```

```
IPv4 Crypto IKEv2 SA
```

Tunnel-id	Local	Remote	fvrf/ivrf	Status
1	10.10.10.1/500	10.10.10.2/500	none/none	READY

```

  Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign:
RSA, Auth verify: RSA
  Life/Active Time: 86400/509 sec
  CE id: 1034, Session-id: 7
  Status Description: Negotiation done
  Local spi: 5CE063D07E8745EA      Remote spi: 52D538043A8E330C
  Local id: hostname=Router1.cisco.com
  Remote id: hostname=Router2.cisco.com
  Local req msg id: 0                Remote req msg id: 2
  Local next msg id: 0                Remote next msg id: 2
  Local req queued: 0                Remote req queued: 2
  Local window: 5                    Remote window: 5
  DPD configured for 0 seconds, retry 0
  Fragmentation not configured.
  Extended Authentication not configured.
  NAT-T is not detected
  Cisco Trust Security SGT is disabled
  Initiator of SA : No

```

IKEv2 Cookie Challenge and Call Admission Control

The following scenario highlights the use of the cookie challenge and the maximum in negotiation SA features, and the benefits that each brings.

IKEv2 call admission control (CAC) limits the maximum number of IKEv2 SAs that can be established. CAC limits the number of simultaneous negotiations with the default being 40 in-negotiation SAs, although this value is configurable using the **crypto ikev2 limit max-in-negotiation-sa** command.

To illustrate the CAC in action, the architecture in Figure 7-5 was developed. This setup consists of an IOS device acting as a VPN headend. Imagine a device created to send many IKE_SA_INIT requests to the headend from random spoofed source IP addresses. The IOS headend is configured with a default gateway, which is where all replies to any received IKE_SA_INIT messages will be sent and then discarded. The IKEv2 generator is pre-configured with an IKEv2 proposal that will be accepted by the IKEv2 headend and sends approximately 12 spoofed packets every second.

The IKEv2 generator sends an IKE_SA_INIT request with a spoofed source IP address of 192.168.1.1 to 10.10.10.1. The IKEv2 headend receives the IKE_SA_INIT, checks that the transforms are valid, allocates state and returns its IKE_SA_INIT response. This response will be received by the router and then forwarded to the 192.168.1.1 destination where it will be discarded.

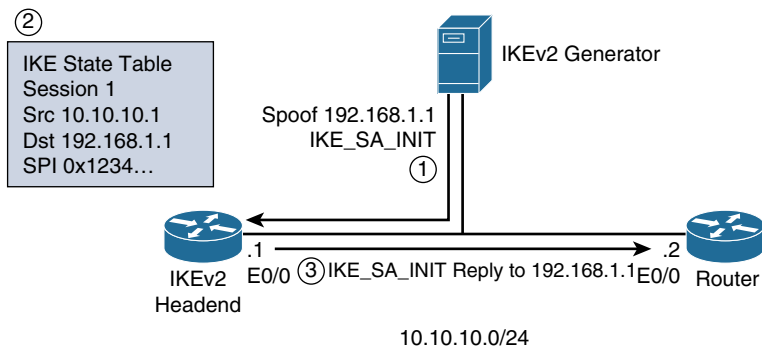
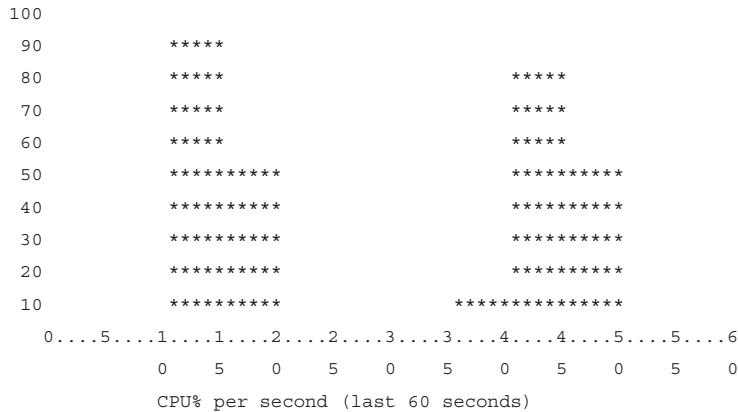


Figure 7-5 CAC Architecture

The hardware used for the IKEv2 headend was purposely chosen as a low-powered device. This was to illustrate the load when generating a large number Diffie-Hellman calculations and the software crypto engine was used. The following example illustrates the CPU history when a constant stream of spoofed IKEv2 SA_INIT requests is sent from the IKEv2 generator. The sudden initial spike in CPU (40 to 60 seconds) is due to the device processing the first forty spoofed IKE_SA_INIT requests, these are processed and replies sent. The CPU then drops to zero percent for approximately fifteen seconds and once again rises back to near full CPU at ninety percent. The drop in CPU processing was due to the CAC feature becoming active. Once forty IKE SAs are in negotiation, no more IKE_SA_INIT requests will be processed. Although the IKEv2 generator is sending a constant stream of these, the IKEv2 headend will only process forty at any given time (although this value is configurable). Some of the initial forty requests time out, and the state for these are removed before any new requests are processed and state allocated.

```
Router#show processes cpu history
```



When an IKEv2 device acting as a responder receives a number of half-open IKE_SA_INIT requests, the cookie challenge mechanism can be deployed. This will enable the responder to include the cookie notification payload in the response to the initiator. The responder does not allocate any state to the session. If the initiator was legitimate, the response containing the cookie will reach the initiator who will then re-attempt the IKE_SA_INIT exchange, including the cookie notification payload, which is then verified by the responder. The responder will then allocate state to the IKE session.

If a device is under a Denial-of-Service (DoS) attack where spoofed IKE_SA_INIT are sent with the purpose of overloading the CPU, the device can be configured to activate the cookie-challenge mechanism. In this situation, the responder will reply with the cookie notification payload. Because this reply is sent to an IP address that was spoofed by an attacker, this reply will be discarded, or dropped by the receiver.

To illustrate this behavior, the IKEv2 headend was amended to allow 1000 in negotiation SAs. The following example shows the command used to achieve this.

```
Router(config)#crypto ikev2 limit max-in-negotiation-sa 1000
```

The CPU of the IKEv2 headend was then constantly at 100 percent. This was due to the amount of constant spoofed IKE_SA_INIT requests from the IKEv2 generator that overwhelmed the IKEv2 state machine.

To rectify this issue, the cookie-challenge is enabled by default. This was enabled, using the value of 0, so all received IKE_SA_INIT requests will be returned with the cookie notification payload.

```
Router(config)#crypto ikev2 cookie-challenge 0
```

The value configured can be between 0 and 1000, which denotes the maximum number of in-negotiation IKE SAs before the cookie challenge is engaged.

No state is allocated to any IKE sessions as all IKE_SA_INIT replies are resent. The following example illustrates the impact that enabling the cookie challenge mechanism has. Once cookie challenge is enabled, the CPU drops from 100 to 0 percent. This is due to the fact that no state is allocated to any of the received IKE_SA_INIT requests.

```
Router#show processes cpu history
```

```

100 *****
 90 *****
 80 *****
 70 *****
 60 *****
 50 *****
 40 *****
 30 *****
 20 *****
 10 *****
    0...5...1...1...2...2...3...3...4...4...5...5...6
      0   5   0   5   0   5   0   5   0   5   0   5   0
      CPU% per second (last 60 seconds)

```

The cookie challenge is a useful feature when an IKEv2 headend is under a DoS attack whereby source IP addresses are spoofed. It can be enabled by default. However, this will incur an additional two-packet exchange to any IKE negotiation which might not be optimal in some situations. Using a value for the maximum in negotiation SAs that is a little higher than what is observed in a known good state will allow this mechanism to engage should a DoS condition occur.

Summary

The examples used in this chapter illustrate a variety of IKEv2 configurations. Numerous authentication methods were used to illustrate the broad range of options available and the benefits that they bring. Smart defaults were used to show the simplicity of the configuration when these are employed. PKI is mandatory when using RSA or EC digital signatures which isn't needed when using pre-shared-key authentication. However, this is not as scalable.

The use of the HTTP URL cert feature was described, where the certificate is not sent in the exchange but instead is retrieved by the IKEv2 peer. This allows for a substantially reduced packet size of the IKE_AUTH exchange.

The use of the maximum in-negotiation SAs and the cookie challenge mechanism was observed to illustrate how IKE can be susceptible to DoS attacks. The use of the cookie notification payload can reduce the impact of a DoS attack; however, in non-DoS conditions, it does add an additional round trip to any IKEv2 exchange.

Index

Symbols

+ (plus sign), 54

3DES, 111

3DS-CBC, 514

A

AAA (authentication, authorization, and accounting), 418

debug commands, 501

RADIUS-based AAA, authorization, 436

AAA accounting, FlexVPN server, 287

AAA authentication method, 278

AAA authorization, deriving virtual access configurations from (FlexVPN server), 293–294

aaa authorization, 223, 225, 229, 233

aaa authorization group, 229, 237–238

aaa authorization group override, 294

aaa authorization user, 285, 484

AAA infrastructure, 221–223
configuring for FlexVPN, 222–223

aaa new-model command, 222

AAA-based pre-shared keys

FlexVPN, 381–382

branch-1 router configuration, 382–383

branch-2 router configuration, 383

hub router configuration, 383–384

RADIUS server configuration, 384–386

FlexVPN server, 283–284

configuring, 284

examples, 285–287

RADIUS attributes, 285

acceptable algorithms, 111

access, remote access (migration strategies), 565–566

access control, 4

accounting, 418

FlexVPN server, 287–290

advanced features, FlexVPN client, 336

AES CBC mode, 112

AES GCM mode, 112

AES-CBC, 514

- AES-CBC-128, 111, 513
- AES-CBC-192, 111
- AES-CBC-256, 111
- AES-CCM, 513–514
- AES-CMAC-96, 515
- AES-CTR, 514
- AES-GCM, 513–514
- AES-GCM-128, 111
- AES-GCM-256, 111
- AES-GMAC1, 514
- AES-GMAC2, 513
- AES-XCBC-MAC-96, 515
- Aggregate authentication, 315
- Aggregate XML, 315
- Aggregation Services Routers (ASR), 87
- Aggressive Mode, 70
- AH (Authentication Header), 2–3
 - anti-replay services, 16–17
 - overhead, 509–510
- algorithms
 - combined algorithm overhead, 512–513
 - hash algorithms, for signatures, 163
 - IKEv2 encryption, Cisco IOS, 111
 - IKEv2 integrity, in Cisco IOS, 113
 - integrity algorithms, 511–512
 - PRF (pseudorandom function), 41
 - pseudorandom function, IKEv2, 115
- AMx, 70
- anti-denial of service, 72
- anti-replay services, 4, 16–17
 - ESP (Encapsulating Security Payload), 18
 - IPsec, 491–494
- AnyConnect
 - EKU (Extended Key Usage), 469
 - FlexVPN server
 - auto-reconnect configuration attributes*, 310–311
 - IKEv2*, 309–310
 - name verification, 468
 - XML configurations, 282–283
- AnyConnect-EAP, 146, 315, 315–316
 - configuring dual-factor authentication, 324–325
 - dual-factor authentication, 320–324
 - user authentication, 316–318
 - configuring*, 318–319
- AnyConnect-EAP XML messages, dual-factor authentication, 322–324
- APPLICATION_VERSION, 257
- ASR (Aggregation Services Routers), 87
- asymmetric cryptography, 8, 11–12
 - Diffie-Hellman exchange, 8–11
- asymmetric keys, 132
- asymmetric routing, considerations when moving to IKEv2, 547–548
- attacks, MITM (man-in-the-middle) attack, 13
- attributes
 - APPLICATION_VERSION, 257
 - backup-gateway attributes, configuring, 347
 - certificate attributes, 469
 - Cisco private use configuration attributes, 257–258
 - configuration attributes
 - FlexVPN*, 253–258
 - IKEv2 auto-reconnect*, 310–311
 - configuration payload, 59
 - default-domain attributes, configuring, 344

- FlexVPN, 253
- interface-config AAA attribute, 293–294
- interface-config attribute, 296
- INTERNAL_IP4_ADDRESS, 254
- INTERNAL_IP4_DNS, 256
- INTERNAL_IP4_NBNS, 257
- INTERNAL_IP4_NETMASK, 254
- INTERNAL_IP4_SUBNET, 255
- INTERNAL_IP6_ADDRESS, 255
- INTERNAL_IP6_DNS, 256
- INTERNAL_IP6_SUBNET, 256
- IPv4 DNS server attributes, configuring, 340
- IPv4 WINS attribute, 343
- IPv6 DNS server attributes, configuring, 341
- RADIUS attributes
 - AAA-based pre-shared keys*, 285
 - CoA (change-of-authorization)*, 303–304
 - FlexVPN server*, 325–329
- Split-DNS, configuring, 341
- audit-session-id**, 299, 303–304
- AUTH payload**, 13
- authentication**, 14, 73–74, 418
 - certificate-based authentication, 147–149
 - considerations when moving to IKEv2, 546–547
 - data origin authentication, 4
 - debugging, with PKI, 470
 - dual-factor authentication, AnyConnect-EAP, 320–324
 - EAP (Extensible Authentication Protocol), 434–436
 - EAP authentication. *See* EAP authentication, FlexVPN server
 - ECDSA (Elliptic-Curve Digital Signature Algorithm) authentication, 194–200
 - troubleshooting, 465–468
 - ESP (Encapsulating Security Payload), 17–18
 - IKE_AUTH, 42–43
 - IKEv2, 45
 - EAP (Extensible Authentication Protocol)*, 48–50
 - pre-shared-key-based authentication*, 47
 - signature-based authentication*, 46
 - IP Authentication Header, 15
 - PKI (Public Key Infrastructure), 431–434
 - pre-shared key authentication, 147
 - pre-shared keys, 478–479
 - pre-shared-key-based authentication, 47
 - PSK, 429
 - RSA authentication, troubleshooting, 465–468
 - RSA authentication using HTTP URL lookup, 200–207
 - signature-based authentication, 46
 - user authentication, AnyConnect-EAP, 315
- authentication command**, 147
- Authentication Header (AH)**, 2–3, 15
- authentication header overhead**, 509–510
- authentication methods**, IKEv2 profiles, 145–149
- authentication pre-shared keys**, 429–431
- authentication remote anyconnect-eap**, 323
- authorization**, 418, 485–487

- FlexVPN, 231–233
 - configuring*, 233
 - group authorization*, 237–241
 - implicit authorization*, 242–245
 - user authorization*, 235–237
- RADIUS-based AAA, 436
- authorization policy**
 - default IKEv2 authorization policy, 229–231
 - IKEv2, 228–229
- authorization types, FlexVPN**, 245–250
- auto detection of tunnel transport and encapsulation**, 297–298
- auto tunnel mode**, 99
- auto-detection of tunnel encapsulation and transport**, FlexVPN, 219–221
- automatic mode, tunnel initiation**, FlexVPN client, 350
- AutoReconnect**, 311
- auto-reconnect, FlexVPN server**, 309–310
 - configuration attributes, 310–311
 - configuring, 313–315
 - smart DPD, 311–313
- AutoReconnectBehavior**, 311
- avoid**, 111

B

- backup gateway lists, FlexVPN client**, 347
- backup gateways, FlexVPN client**, 346
- backup group command**, 353
- backup groups, FlexVPN client**, 353–354
- Backup Peers**, 544

- backup peers, FlexVPN hub resiliency**, 411
- backup-gateway attributes**, *configuring*, 347
- boolean expressions, tracking, lists of objects**, 350–352
- branch 1 configuration**
 - FlexVPN, spoke configuration, 392–394
 - FlexVPN client, group and user authorization, 386
 - FlexVPN spoke, verification, 397–399
- branch 2 configuration**
 - FlexVPN, spoke configuration, 394–395
 - FlexVPN client, user authorization, 387
 - FlexVPN spoke, verification, 399–400
- branch-1 router configuration**, AAA-based pre-shared keys (FlexVPN), 382–383
- branch-2 router configuration**, AAA-based pre-shared keys (FlexVPN), 383
- buffers, capture buffers**, 457
- building blocks of**
 - FlexVPN
 - Cisco IOS AAA infrastructure*, 221–223
 - IKEv2*, 213–214
 - point-to-point tunnel interfaces*, 214–221
 - FlexVPN client
 - FlexVPN client profiles*, 334
 - IKEv2 configuration exchange*, 334
 - NAT (Network Address Translation)*, 335
 - object tracking*, 334
 - static P2P tunnel interfaces*, 334

IPsec, 2

- access control*, 4
- anti-replay services*, 4
- confidentiality*, 4
- connectionless integrity*, 4
- data origin authentication*, 4
- key management protocol*, 3
- SAs (Security Associations)*, 3
- security protocols*, 2–3
- security services*, 3
- TFC (Traffic Flow Confidentiality)*, 4–5

C

CA (certificate authority), 12

- PKI (Public Key Infrastructure), 160–162

CAC (Call Admission Control),
IKEv2, 157

- Cookie Challenge and Call Admission Control, 207–210

cached keyword, 234

capacity, considerations when moving
to IKEv2, 542–543

capture buffers, 457

capture points, 458

CBC (cipher block chaining), 510

CDP (Cisco Discovery Protocol), 86

CDP (CRL Distribution Point), 475

CEF (Cisco Express Forwarding), 94

CERT, 148

certificate attributes, 469

certificate authority. *See* CA
(certificate authority)certificate authority (CA), PKI (Public
Key Infrastructure), 160–162certificate expiry, 470–472
session deletion, IKEv2, 184certificate maps, matching, peers,
472–473

certificate requests, 148

IKEv2, 38–39

HTTP_CERT_LOOKUP_
SUPPORTED, 39

certificate revocation, 473–476

Certificate Revocation Lists
(CRLs), 163

certificate revocation method, 163

session deletion, 182

certificate-based authentication,
147–149

certificates

certificate expiry, 470–472

digital certificates, 12

HTTP URL-based certificate
lookup, 156

matching peers, 140–141

certification revocation list
(CRL), 181

CERTREQ, 148

CertReq, 26

CERTREQ payload, 476

CFG_ACK, 252

CFG_REPLY, 251

CFG_REQUEST, 251

CFG_SET, 252

change-of-authorization (CoA),
303–304

Child SAs, 24

childless initiation, 66

cipher block chaining (CBC), 510

Cisco AV pair, 325

Cisco Discovery Protocol (CDP), 86

Cisco Express Forwarding (CEF), 94

Cisco IKEv2 AnyConnect clients, 330

Cisco IOS

- algorithms
 - IKEv2 encryption*, 111
 - IKEv2 integrity*, 113
- Diffie-Hellman group, 114
- IPsec configuration, 166–167
- IPsec configuration
 - examples*, 168
 - smart defaults*, 168–169
- IPsec profiles, 167
- PKI (Public Key Infrastructure), 159–160
 - CA (certificate authority)*, 160–162
- pseudorandom function algorithms, IKEv2, 115
- Cisco IOS AAA infrastructure, 221–223
 - configuring for FlexVPN, 222–223
- Cisco IPsec flow monitor MIB, 425
- Cisco meta data (CMD), 179
- Cisco private use configuration attributes, 257–258
- Cisco unity attributes, 253
- clamping, MSS (Maximum Segment Size), 526–527
- clear crypto ikev2 diagnose error, 461
- clear crypto ikev2 sa, 186
- clear crypto ikev2 sa remote, 360
- clear crypto session, 186
- clearing, IKEv2 FlexVPN client sessions, 360
- client awareness, considerations when moving to IKEv2, 545
- client connect tunnel
 - interface-number, 348
- client connect Tunnelo, 408, 412
- client debugging, IKEv2, 450
- client inside, 338
- clients, remote access clients. *See* remote access clients
- cloning, virtual access cloning, 295–297
- cluster debugging, IKEv2, 450
- cluster loads, FlexVPN load balancer, 369–372
- CMD (Cisco meta data), 179
- CoA (change-of-authorization), 303–304
- co-existence, FlexVPN, authorization types, 245
- collect command, 441
- combined algorithm overhead, 512–513
- combined mode ciphers, 77, 112
- combined-mode ciphers, IKEv2 proposals, 110
- commands
 - aaa authorization, 223, 225, 229, 233
 - aaa authorization group, 229, 237–238
 - aaa authorization group override, 294
 - aaa authorization user, 285, 484
 - aaa new-model command, 222
 - authenticate remote
 - anyconnect-eap, 323
 - authentication, 147
 - backup group, 353
 - CFG_ACK, 252
 - CFG_REPLY, 251
 - CFG_REQUEST, 251
 - CFG_SET, 252
 - clear crypto ikev2 diagnose error, 461
 - clear crypto ikev2 sa, 186
 - clear crypto ikev2 sa remote, 360
 - clear crypto session, 186

- client connect tunnel
 - interface-number, 348
- client connect Tunnelo, 408, 412
- client inside, 338
- collect, 441
- config-exchange request, 251
- config-exchange set accept, 251
- config-exchange set send, 251
- config-set, 497
- copy run start, 346
- crypto eap credential profile1, 337
- crypto ikev2 authorization policy, 222, 228
- crypto ikev2 cluster, 368
- crypto ikev2 cookie-challenge
 - number, 37
- crypto ikev2 diagnose error, 159
- crypto ikev2 disconnect-revoked-peers, 182
- crypto ikev2 keyring, 129
- crypto ikev2 name-mangler, 224
- crypto ikev2 profile, 138
- crypto ikev2 redirect client, 374
- crypto ikev2 redirect gateway
 - init, 377
- crypto ipsec df-bit {clear | set | copy}, 528
- crypto ipsec fragmentation, 532
- crypto ipsec fragmentation {before-encryption | after-encryption}, 528
- crypto ipsec profile, 348
- crypto ipsec security-association
 - replay window-size disable, 494
- crypto key generate, 162
- crypto mib ipsec flow history tunnel
 - size, 427
- crypto pki authenticate, 164
- crypto pki profile enrollment, 163
- debug aaa authorization, 436
- debug aaa proto {local | radius}, 486–487
- debug commands
 - AAA (authentication, authorization, and accounting)*, 501
 - EAP (Extensible Authentication Protocol)*, 501
 - IKEv2*, 501
 - IPsec*, 501
 - PKI (Public Key Infrastructure)*, 502
 - RADIUS*, 501
- debug crypto condition
 - unmatched, 456
- debug crypto ikev2, 187, 360, 436, 466, 473
- debug crypto ikev2 client
 - flexvpn, 353
- debug crypto ikev2 cluster
 - detail, 369
- debug crypto ikev2 packet, 126, 450
- debug crypto ikev2 packet
 - debugging, 491
- debug crypto ikev2 packet
 - hexdump, 451
- debug crypto ipsec, 453
- debug crypto ipsec metadata sgt, 181
- debug crypto kmi, 455
- debug crypto pki, 473
- debug ip dns name-list, 341
- debug ip tcp, 474
- debug vtemplate, 487
- debug vtemplate cloning, 487
- default crypto ikev2 authorization
 - policy, 230
- dn all, 227
- dn common-name, 226

- dn country, 227
- dn domain, 226
- dn locality, 227
- dn organization, 227
- dn organization-unit, 226
- dn state, 227
- eap all, 227
- eap dn common-name, 227
- eap dn country, 227
- eap dn domain, 227
- eap dn locality, 227
- eap dn organization, 227
- eap dn organization-unit, 227
- eap prefix delimiter., 227
- eap prefix delimiter @, 227
- eap prefix delimiter (backslash), 227
- eckeypair, 163
- email all, 226
- email domain, 226
- email username, 226
- enrollment url, 163
- fqdn all, 226
- fqdn domain, 226
- fqdn hostname, 225
- import all, 343
- ip | ipv6 unnumbered, 102
- ip address, 215
- ip address negotiated, 334, 348
- ip http server, 161
- ip mtu, 499, 531
- ip nat inside, 405
- ip nat outside, 355, 405
- ip tcp adjust-mss, 527
- ip unnumbered, 218
- ip unreachable, 520
- ip vrf, 100
- ip vrf forwarding, 100, 296
- IPsec commands, considerations
 - when moving to IKEv2, 543–544
- ipv6 address, 215
- ipv6 mtu, 531
- ipv6 tcp adjust-mss, 527
- ipv6 unnumbered, 218
- ipv6 unreachable, 522
- ivrf, 152
- keyring aaa, 223, 225
- match, 441
- match certificate, 472
- method-est, 163
- monitor event-trace, 449
- mtu, 518, 533
- no crypto ipsec nat-transparency
 - udp-encapsulation, 64
- no lifetime, 184
- no logging event link-status, 424
- no route accept, 267
- no shutdown, 161
- peer reactivate, 346
- pki trustpoint, 470
- reconnect, 310
- responder-only, 167
- revocation-check method
 - command, 164
- route accept, 266, 498
- route accept any, 230
- route set interface, 230, 496
- route set local, 267, 498
- service-policy, 296
- set ikev2-profile, 167
- set mixed-mode, 99, 167
- set peer hostname dynamic, 130
- set pfs, 167
- set reverse-route, 167

- set security-association, 167
- set security-association replay window-size disable, 494
- set transform-set, 167
- show aaa attribute protocol radius, 228
- show cef interface, 531
- show commands
 - IKEv2*, 500
 - IPsec*, 500
 - PKI (Public Key Infrastructure)*, 501
 - troubleshooting*, 447
- show crypto ikev2 authorization policy, 229
- show crypto ikev2 client flexvpn, 348, 358
- show crypto ikev2 client flexvpn flex1 detail, 342
- show crypto ikev2 client flexvpn name, 358
- show crypto ikev2 cluster, 369, 374
- show crypto ikev2 diagnose error, 460
- show crypto ikev2 proposal, 34, 109, 463
- show crypto ikev2 sa detail, 282
- show crypto ikev2 sa detailed, 144, 177, 313, 343, 359, 397, 399
- show crypto ikev2 session detailed, 144
- show crypto ipsec sa, 530, 560
- show crypto ipsec transform-set, 488
- show crypto pki certificate verbose, 480
- show crypto pki certificates, 467
- show crypto pki counters, 475
- show crypto pki trustpoints, 467
- show crypto session, 220, 559
- show crypto session brief, 560
- show crypto session detail, 186, 313
- show crypto sessions, 95
- show crypto sockets, 95
- show derived-config, 91
- show ip dhcp import, 343
- show ip dns name-list, 341, 343
- show ip interfaces, 518
- show ip nat statistics, 406
- show ip nat translations, 406
- show ip route, 266–267
- show ip route vrf, 460
- show ip traffic, 522
- show ipv6 interfaces, 518
- show ipv6 traffic, 523
- show ntp associations, 471
- show platform hardware qfp active feature ipfrag global, 522–523
- show run all, 184
- show running-config, 91
- show running-configuration, 478
- show standby, 375
- show track, 359
- SNMP IKE trap commands, 427
- SNMP IPsec trap commands, 438–439
- snmp-server enable traps ike tunnel start, 425
- snmp-server enable traps ike tunnel stop, 426
- snmp-server enable traps ipsec tunnel start, 438
- snmp-server enable traps ipsec tunnel stop, 439
- snmp-server enable traps snmp linkdown linkup, 439–440
- snmp-server enable traps snmp linkup linkdown, 424–425

- test aaa, 481
- tunnel destination, 215, 218
- tunnel destination dynamic, 349, 412
- tunnel destination peer-address, 90
- tunnel mode, 215, 218
- tunnel mode gre ip, 94
- tunnel path-mtu-discovery, 499, 534–535
- tunnel protection, 139, 167–168, 216, 218
- tunnel protection ipsec, 123
- tunnel source, 218
- tunnel source dynamic, 348, 408
- tunnel vrf name, 102
- virtual-template 1 mode auto, 396
- vrf definition, 100
- vrf forwarding, 100, 216, 218, 486
- vrf forwarding name, 102
- components of**
 - FlexVPN load balancer, 363
 - HSRP (hot standby routing protocol)*, 366–367
 - IKEv2 redirect*, 363–366
 - IPsec, 5
 - PAD (Peer Authorization Database)*, 6
 - SAD (Security Association Database)*, 6
 - SPD (Security Policy Database)*, 5–6
 - SPI (Security Parameter Index)*, 5
 - Split-DNS, FlexVPN client, 340–343
- conditional debugging**, 456–457
- confidentiality**, 4
- Config Payload**, 58
- config-exchange request**, 251
- config-exchange set accept**, 251
- config-exchange set send**, 251
- config-set command**, 497
- configuration attributes**
 - auto-reconnect, FlexVPN server, 310–311
 - FlexVPN, 253–258
- configuration constructs, IKEv2**, 106
- configuration examples**
 - EAP (Extensible Authentication Protocol), FlexVPN server, 278–283
 - keyring, IKEv2, 134–135
- configuration exchange, FlexVPN**, 250
 - enabling, 250–251
 - examples, 259–264
- configuration payload**, 75
 - attributes, 59
 - FlexVPN, 251–253, 258
- configuration payload exchange**, 58–59
- configurations, IKEv2**, 106
- configuring**
 - AAA infrastructure, FlexVPN, 222–223
 - AAA-based pre-shared keys, FlexVPN server, 284
 - authorization, FlexVPN, 233
 - auto-reconnect, FlexVPN server, 313–315
 - backup-gateway attributes, 347
 - default-domain attributes, 344
 - Diffie-Hellman group, IKEv2, 113–114
 - dual-factor authentication, AnyConnect-EAP, 324–325
 - EAP (Extensible Authentication Protocol), FlexVPN server, 277–278
 - configuration examples*, 278–283

- FlexVPN, RADIUS servers, 388–390
- FlexVPN client, dual-homed branch routers, 411–412
- FlexVPN server, 387–388
- IKEv2, encryption, 111–112
- IKEv2 proposals, 108–111
 - under IKEv2 policies, 119–120*
- integrity, IKEv2, 113
- IPsec configuration, 166–167
- IPv4 DNS server attributes, 340
- IPv4 WINS attribute, 343
- IPv6 DNS server attributes, 341
- keyring, IKEv2, 129–132
- keys, in peer blocks, 132
- match statements
 - under IKEv2 policies, 120–121*
 - IKEv2 profiles, 139–142*
- name mangler, IKEv2, 224–227
- peer blocks, in keyring, 130
- peers, in peer blocks, 130–131
- policies, IKEv2, 118–119
- profiles, IKEv2, 138–139
- pseudorandom function, IKEv2, 115
- RADIUS change-of-authorization (CoA), 304
- RADIUS Packet-of-Disconnect, FlexVPN server, 300
- Split-DNS attributes, 341
- static P2P tunnel interfaces, 214–216
- trustpoints (TP), 476
- user authentication,
 - AnyConnect-EAP, 318–319
 - virtual-template interfaces, 216–219
- connect auto mode, 408**
- connectionless integrity, 4**
- context-specific configuration, IKEv2, 106**
- continuous channel mode, 77**
- Cookie Challenge, IKEv2, 156**
 - IKEv2 Cookie Challenge and Call Admission Control, 207–210
- cookie notification, IKEv2, 36–38**
- copy run start, 346**
- Counter (CTR) mode, 510**
- CREATE_CHILD_SA, 24, 53, 53–54**
- CRL (certification revocation list), 181**
- CRL Distribution Point (CDP), 475**
- CRLs (Certificate Revocation Lists), 163**
- crypto eap credential profile1, 337**
- crypto ikev2 authorization policy, 222, 228**
- crypto ikev2 cluster, 368**
- crypto ikev2 cookie-challenge number, 37**
- crypto ikev2 diagnose error, 159**
- crypto ikev2 direct gateway init, 377**
- crypto ikev2 disconnect-revoked-peers, 182**
- crypto ikev2 error, 448**
- crypto ikev2 keyring, 129**
- crypto ikev2 name-mangler, 224**
- crypto ikev2 profile, 138**
- crypto ikev2 redirect client, 374**
- crypto ipsec df-bit {clear | set | copy}, 528**
- crypto ipsec error, 448**
- crypto ipsec fragmentation, 532**
- crypto ipsec fragmentation {before-encryption | after-encryption}, 528**
- crypto ipsec profile, 348**
- crypto ipsec security-association replay window-size disable, 494**
- crypto key generate, 162**
- crypto maps, 79**
 - demise of, 86–87
 - versus tunnel protection, 80–81

crypto mib ipsec flow history tunnel size, 427

crypto pki authenticate, 164

crypto pki profile enrollment, 163

crypto pki server, 160

crypto sockets, 94–95

cryptographic exchange bloat, 77

cryptographic strength, 111–112

pre-shared keys, 135

cryptography

asymmetric cryptography, 11–12

IPsec VPNs, 7

asymmetric cryptography, 8

Diffie-Hellman exchange, 8–11

symmetric cryptography, 7–8

CTR (Counter) mode, 510

D

data encapsulation, GRE (generic routing encapsulation), 495

data encryption, 488

SNMP with IPsec, 437–439

data origin authentication, 4

data usage, monitoring, 440–443

datagram format, ESP (Encapsulating Security Payload), 18–19

dead peer detection, 59–61

Dead Peer Detection (DPD), IKEv2, 149–151, 158–159

debug aaa authorization, 436

debug aaa proto {local | radius}, 486–487

debug commands

AAA (authentication, authorization, and accounting), 501

EAP (Extensible Authentication Protocol), 501

IKEv2, 501

IPsec, 501

PKI (Public Key Infrastructure), 502

RADIUS, 501

debug crypto condition unmatched, 456

debug crypto ikev2, 123, 187, 360, 436, 466, 473

debug crypto ikev2 client flexvpn, 353, 450

debug crypto ikev2 cluster detail, 369, 375

debug crypto ikev2 packet, 126, 450

debug crypto ikev2 packet debugging, 491

debug crypto ikev2 packet hexdump, 451

debug crypto ipsec, 453

debug crypto ipsec metadata sgt, 181

debug crypto kmi, 123, 455

debug crypto pki, 473

debug ip dns name-list, 341

debug ip packet, 406

debug ip tcp, 474

debug vtemplate, 487

debug vtemplate cloning, 487

debugging, 449

authentication, with PKI, 470

conditional debugging, 456–457

FlexVPN client, 360

IKEv2, 449–453

IPsec, 488–491

IPsec debugging, 453

KMI (Key Management Interface), 453–455

- PKI (Public Key Infrastructure), 456
- verbose debugging, 181
- default crypto ikev2 authorization policy, 230
- default IKEv2 authorization policy, 229–231
- default IKEv2 policies, 121–122
- default proposals, IKEv2, 115–117
- default-domain attributes, configuring, 344
- defaults, smart defaults, 168–169
- deleting, SAs (Security Associations), 57–58
- deployments
 - IKEv1, 551–552
 - IKEv2
 - Cookie Challenge and Call Admission Control*, 207–210
 - pre-shared-key authentication with smart defaults*, 189–194
 - RSA authentication using HTTP URL lookup*, 200–207
- depo, IKEv2, ECDSA (Elliptic-Curve Digital Signature Algorithm) authentication, 194–200
- DER (Distinguished Encoding Rules), 44, 201
- DES, 111
- detail keyword, 559
- detection, NAT-D (Network Address Translation-Detection), 64
- DF (Don't Fragment), 172
- diagnose error, IKEv2, 460–461
- diagnostics, IKEv2, 159
- dial backups, FlexVPN client, 352–353
- Differentiated Services Code Point (DSCP), 74
- Diffie, Whitfield, 9
- Diffie-Hellman exchange, 8–11, 24, 26–29
 - initiators, 28
 - MITM (man-in-the-middle) attack, 45
 - SA_INIT, 29
- Diffie-Hellman group, 30
 - IKEv2, configuring, 113–114
- Diffie-Hellman tests, IKEv2, 65
- digital certificates, 12
- digital signatures, 12
 - IKEv2, 12–13
- disabling
 - anti-replay, 494
 - default IKEv2 policies, 122
 - default IKEv2 proposals, 116
 - profiles, IKEv2, 153
- disconnecting revoked peers, 182
- DisconnectOnSuspend, 311
- displaying, profiles, IKEv2, 153–154
- Distinguished Encoding Rules (DER), 44, 201
- distinguished name (DN), 224
- DN (distinguished name), 224
 - extracting names from, 226–227
- dn all, 227
- dn common-name, 226
- dn country, 227
- dn domain, 226
- dn locality, 227
- dn organization, 227
- dn organization-unit, 226
- dn state, 227
- domain names, FlexVPN client, 344–345
- Don't Fragment (DF), 172

DPD (Dead Peer Detection), IKEv2,
149–151, 158–159

DSCP (Differentiated Services Code
Point), 74

dual stack

FlexVPN, 391–392

FlexVPN client, 335

GRE (generic routing
encapsulation), 96

dual-factor authentication

AnyConnect-EAP, 320–324

AnyConnect-EAP XML messages,
322–324

configuring with AnyConnect-EAP,
324–325

dual-homed branch routers, FlexVPN
client configuration, 408–409,
411–412

dummy packets, ESP (Encapsulating
Security Payload) version 3, 20

dVTI (dynamic VTI), 92, 153

dynamic keyword, 346

dynamic routing, FlexVPN client, 335

dynamic routing protocols, 498–499

dynamic tunnel interfaces, 91–92

dynamic tunnel source, FlexVPN
WAN resiliency, 407–408

dynamic VTI (dVTI), 92

E

EAP (Extensible Authentication
Protocol), 48–50, 146, 480–485

authentication, 434–436

debug commands, 501

FlexVPN server, configuring,
277–278

examples, 278–283

eap all, 227

EAP authentication

FlexVPN client, 335, 337–338

FlexVPN server, 271–272

EAP methods, 272

steps for, 275–277

eap dn common-name, 227

eap dn country, 227

eap dn domain, 227

eap dn locality, 227

eap dn organization, 227

eap dn organization-unit, 227

eap dn state, 227

EAP identity, 224

extracting names, 227

FlexVPN server, 273–275

EAP message flow, FlexVPN
server, 273

EAP methods

FlexVPN server, 272

TLS (transport layer security), 272

eap prefix delimiter., 227

eap prefix delimiter @, 227

eap prefix delimiter \, 227

EAP timeout, FlexVPN server, 275

EAP tunneled methods, 272

ECDH (Elliptic Curve Diffie
Hellman), 10

ECDSA (Elliptic-Curve Digital
Signature Algorithm), 73

ECDSA (Elliptic-Curve Digital
Signature Algorithm)
authentication

IKEv2, 194–200

troubleshooting, 465–468

ECDSA (Elliptic-Curve Digital
Signature Algorithm)
signatures, 12

eckeypair, 163

- EEM (embedded event manager), 356–358
- EIGRP (enhanced interior gateway routing protocol), 191
- EKU (Extended Key Usage), 469, 480
 - AnyConnect, 469
- Elliptic Curve Diffie Hellman (ECDH), 10
- email
 - extracting names from, 226
 - IKE identity types, 224
- email all, 226
- email domain, 226
- email username, 226
- embedded event manager (EEM), 356–358
- enabling, configuration exchange, FlexVPN, 250–251
- encapsulating security payload overhead, 507–509
- Encapsulating Security Payload (ESP). *See* ESP (Encapsulating Security Payload), 2–3
- encapsulation, modes of
 - encapsulation, 82
 - GRE encapsulation, 82–83
 - GRE over IPsec, 83
- encipherment, 7
- ENCR (Encryption algorithm), 30
- encrypted payload structures, 43–44
- encryption, 7, 14
 - ESP (Encapsulating Security Payload), 17–18
 - IKE_AUTH, 42–43
 - IKEv2
 - algorithms in Cisco IOS*, 111
 - configuring*, 111–112
 - IP Authentication Header, 15–16
 - Encryption Algorithm (ENCR), 30
 - encryption overhead, 510–511
 - enhanced interior gateway routing protocol (EIGRP), 191
 - Enrollment over Secure Transport (EST), 163
 - enrollment url, 163
 - error debugging, IKEv2, 450
 - ESN (Extended Sequence Numbers), 30
 - ESP (Encapsulating Security Payload), 2–3, 17, 504
 - anti-replay services, 18
 - authentication, 18
 - confidentiality, 4
 - datagram format, 18–19
 - encryption, 18
 - ESP (Encapsulating Security Payload) version 3, 19
 - dummy packets, 20
 - extended sequence numbers, 19
 - TFC (Traffic Flow Confidentiality), 20
 - ESP-NUL, 16
 - EST (Enrollment over Secure Transport), 163
 - event-trace monitoring, 447–449
 - examples
 - AAA-based pre-shared keys, FlexVPN server, 285–287
 - configuration exchange, FlexVPN, 259–264
 - EAP (Extensible Authentication Protocol), FlexVPN server, 278–283
 - FlexVPN, implicit authorization, 243–245
 - IKEv2 load balancing, FlexVPN load balancer, 376–378

- IPsec configuration, 168
- overhead, 516–517
- RADIUS change-of-authorization (CoA), FlexVPN server, 305–309
- RADIUS Packet-of-Disconnect, FlexVPN server, 301–303
- soft migration, transitioning from IKEv1 to IKEv2, 551–552
- virtual access cloning, 295–297
- exchange modes
 - IKEv1, 70–71
 - IKEv2, 71–72
- Exchange Type, 55
- explicit padding, 510
- Extended Authentication within IKE (XAUTH), 69
- Extended Authentication within ISAKMP/Oakley, 69
- Extended Key Usage (EKU), 469
- extended sequence numbers, ESP (Encapsulating Security Payload) version 3, 19
- Extensible Authentication Protocol. *See* EAP (Extensible Authentication Protocol)
- extensions, for EAP-only authentication, 65
- external AAA servers, FlexVPN, group authorization, 239–241
- extra overhead, 516–517
- extracting names
 - from DN identity, 226–227
 - from EAP identity, 227
 - from email identity, 226
 - from FQDN identity, 225–226
- EzVPN, 333, 540
 - FlexVPN client, 336

F

- familiarization, considerations when moving to IKEv2, 545
- FIB (Forwarding Information Base), 94
- flapping tunnel interface, 499
- Flexible NetFlow, 418
- FlexVPN, 540
 - AAA-based pre-shared keys, 381–382
 - branch-1 router configuration*, 382–383
 - branch-2 router configuration*, 383
 - hub router configuration*, 383–384
 - RADIUS server configuration*, 384–386
 - attributes, 253
 - authorization, 231–233
 - configuring*, 233
 - group authorization*, 237–241
 - implicit authorization*, 242–245
 - user authorization*, 235–237
 - authorization types, 245–250
 - precedence*, 247–250
 - auto-detection of tunnel encapsulation and transport, 219–221
 - building blocks of
 - Cisco IOS AAA infrastructure*, 221–223
 - IKEv2*, 213–214
 - point-to-point tunnel interfaces*, 214–221
 - Cisco private use configuration
 - attributes, 257–258
 - configuration attributes, 258
 - and authorization*, 253–258

- configuration exchange, 250
 - enabling*, 250–251
 - examples*, 259–264
- configuration payload, 251–253, 258
- configuring
 - AAA infrastructure*, 222–223
 - RADIUS servers*, 388–390
- default IKEv2 authorization policy, 229–231
- group and user authorization, 386
- IKEv2, authorization policy, 228–229
- migration strategies, 544
- overview, 211–213
- remote subnets, 264
 - learning from peer*, 266–267
 - learning locally*, 265–266
- routing, 264–265
- routing, dual stack, and tunnel mode
 - auto, 391–392
- spoke configuration
 - branch 1 configuration*, 392–394
 - branch 2 configuration*, 394–395
 - hub configuration*, 395–397
 - verification at branch 1*, 397–399
 - verification at branch 2*, 399–400
 - verification on hub*, 401–404
- value proposition, 213
- virtual access cloning examples, 295–297
- FlexVPN backup tunnels, track-based tunnel activation, 414–415**
- FlexVPN client**
 - advanced features, 336
 - backup gateway lists, 347
 - backup gateways, 346
 - backup groups, 353–354
 - building blocks of
 - FlexVPN client profiles*, 334
 - IKEv2 configuration exchange*, 334
 - NAT (Network Address Translation)*, 335
 - object tracking*, 334
 - static P2P tunnel interfaces*, 334
 - clearing IKEv2 FlexVPN client sessions, 360
 - configuring on dual-homed branch routers, 411–412
 - debugging, 360
 - dial backups, 352–353
 - domain names, 344–345
 - dual stack, 335
 - dual-homed branch routers, configuring, 408–409
 - dynamic routing, 335
 - EAP authentication, 335, 337–338
 - EzVPN, 336
 - FlexVPN load balancer, 374
 - group and user authorization
 - branch 1 configuration*, 386
 - branch 2 configuration*, 387
 - logs, 390–391
 - NAT (Network Address Translation), 354–355, 404–405
 - verification*, 405–407
 - network extension modes, 336
 - overview, 332–333
 - PKI (Public Key Infrastructure), 356
 - pre-shared keys, 356
 - profiles, 345–346
 - reactivating peers, 346
 - resolution of FQDN (fully qualified domain names), 346

- setting up FlexVPN servers, 336–337
- Split-DNS, 338–340
 - components of*, 340–343
- tracking, 356
 - EEM (embedded event manager)*, 356–358
- tracking lists of objects, with boolean expressions, 350–352
- troubleshooting
 - debugging*, 360
 - show commands*, 358–359
- tunnel destination, 349
- tunnel initiation, 350
 - automatic mode*, 350
 - manual mode*, 350
 - track mode*, 350
- tunnel interface, 347–348
- tunnel source, 348–349
- verification, 409–410, 412–413
- WINS (Windows Internet Naming Service), 343–344
- FlexVPN client profiles, 334**
- FlexVPN feature, 90**
- FlexVPN hub resiliency, backup peers, 411**
- FlexVPN IKEv2 Load Balancer, 367–369**
- FlexVPN load balancer**
 - cluster loads, 369–372
 - components of, 363
 - HSRP (hot standby routing protocol)*, 366–367
 - IKEv2 redirect*, 363–366
- FlexVPN client, 374
- FlexVPN IKEv2 Load Balancer, 367–369
- IKEv2 load balancing, examples, 376–378
- IKEv2 redirect, 372–373
- redirect loops, 373–374
- troubleshooting, IKEv2 load balancing, 374–375
- FlexVPN server**
 - AAA-based pre-shared keys, 283–284
 - configuring*, 284
 - examples*, 285–287
 - RADIUS attributes*, 285
 - accounting, 287–290
 - AnyConnect-EAP, 315–316
 - configuring dual-factor authentication*, 324–325
 - configuring user authentication*, 318–319
 - dual-factor authentication*, 320–324
 - user authentication*, 316–318
 - auto detection of tunnel transport and encapsulation, 297–298
 - configuring, 387–388
 - EAP (Extensible Authentication Protocol)
 - configuration examples*, 278–283
 - configuring*, 277–278
 - EAP authentication, 271–272
 - EAP methods*, 272
 - steps for*, 275–277
 - EAP identity, 273–275
 - EAP message flow, 273
 - EAP timeout, 275
 - IKEv2 auto-reconnect, 309–310
 - configuration attributes*, 310–311
 - configuring*, 313–315
 - smart DPD*, 311–313
 - per-session interface, 290–291
 - query-identity, 277

- RADIUS attributes, 325–329
- RADIUS change-of-authorization (CoA), 303–304
 - configuring*, 304
 - examples*, 305–309
- RADIUS Packet-of-Disconnect, 299–300
 - configuring*, 300
 - examples*, 301–303
- remote access clients, 329
 - Cisco IKEv2 AnyConnect clients*, 330
 - Microsoft Windows 7 IKEv2 clients*, 329–330
- sequence of events, 270–271
- setting up, 336–337
- timeout option, 278
- virtual access configurations
 - AAA authorization*, 293–294
 - incoming sessions*, 294
 - virtual templates*, 291–293
- FlexVPN WAN resiliency**
 - dual-homed branch routers, FlexVPN client configuration, 408–409
 - dynamic tunnel source, 407–408
 - FlexVPN client, verification, 409–410
- flow monitors, 441**
- flow records, 441**
- Forwarding Information Base (FIB), 94**
- FQDN (fully qualified domain name), 224**
 - extracting names from, 225–226
 - resolution of FQDN (fully qualified domain names), FlexVPN client, 346
- fqdn all, 226**
- fqdn domain, 226**

- fqdn hostname, 225**
- fragmentation**
 - IKEv2, 171–172, 173–178
 - session authentication*, 181–182
 - session deletion on certificate expiry*, 184
 - session deletion on certificate revocation*, 182–184
 - session lifetime*, 185–187
- IP fragmentation, 172–173
- IPsec
 - impact of*, 535–536
 - IPv4*, 519–522
 - IPv6*, 522–523
 - MTU (maximum transmission unit)*, 518–519
 - PMTUD (path MTU discovery)*, 523–525, 527–531
 - TCP MSS clamping*, 525
- tunnels, 531
 - GRE (generic routing encapsulation)*, 532–533
 - GRE over IPsec*, 534
 - IPsec only (VTI)*, 531–532
 - PMTUD (path MTU discovery)*, 534–535
- Front-door VRF (VFRF), 118**

G

- generic routing encapsulation.**
 - See GRE (generic routing encapsulation)*
- GET VPN, 212–213**
- global configuration, IKEv2, 106, 155**
 - HTTP URL-based certificate lookup, 156
- global IKE, considerations when moving to IKEv2, 543–544**

GRE (generic routing encapsulation), 80, 495

fragmentation, tunnels, 532–533

implementation modes

auto tunnel mode, 99*dual stack*, 96*mixed mode*, 96–99

mGRE (multipoint GRE), 92–94

overhead, 505–507

traffic, non-IP protocols, 86

traffic selectors, 51–52

tunnel configuration, 88–89

VRF aware, 101–102

GRE encapsulation, 82–83**GRE over IPsec, 83**

fragmentation, tunnels, 534

IPsec transport mode, 83–84

IPsec tunnel mode, 84–85

VRF aware, 102–103

GRE tunnel keys, mismatching, 495**GRE/IP encapsulation, 82****GRE/IPsec, 87–88****group authorization**

FlexVPN, 237–241

precedence, 249–250

FlexVPN client, 386

branch 1 configuration, 386*branch 2 configuration*, 387**guiding principles, IKEv2, 106**

H**half connections, 36****hard migration, transitioning from IKEv1 to IKEv2, 548–549****hardware limitations, considerations when moving to IKEv2, 540****Hardware Security Modules (HSM), 545**

hash algorithms, signatures, 163

Hashed Message Authentication Code (HMAC), 4

HDR, 26, 76

Hellman, Martin, 9

high availability, 74

considerations when moving to IKEv2, 547

history of, IKEv1, 67–69

HMAC (Hashed Message Authentication Code), 4, 18, 40

integrity overhead, 511–512

HMAC-MD5–96, 515

HMAC-SHA-1–96, 513, 515

HMAC-SHA-256–128, 515

HMAC-SHA-384–192, 515

HMAC-SHA-512–256, 515

hot standby routing protocol (HSRP), FlexVPN load balancer, 366–367

HSM (Hardware Security Modules), 545

HSRP (hot standby routing protocol), FlexVPN load balancer, 366–367

HTTP URL lookup, RSA authentication using HTTP URL lookup, 200–207

HTTP URL-based certificate lookup, 156

HTTP_CERT_LOOKUP_SUPPORTED, 39

hub configuration, FlexVPN, spoke configuration, 395–397

hub router configuration, AAA-based pre-shared keys, FlexVPN, 383–384

hub-and-spoke topology

hard migration, 549

migration strategies, 562–564



ICV (Integrity Check Value), 4, 15

ID_DER_ASN1_DN, 45

ID_DER_ASN1_GN, 45

ID_IPV4_ADDR, 44

ID_IPV6_ADDR, 45

ID_KEY_ID, 45

ID_RFC 822_ADDR, 45

identities, 44–45

 EAP identity, FlexVPN server,
 273–275

 IKEv2, 74

 local IKE identities, defining local
 IKE identities, 143–145

 matching peers, 141–142

identity local, 143–144

identity services engine (ISE), 178

identity times, IKE identity
 types, 224

IETF (Internet Engineering Task
 Force), 23

IKE (Internet Key Exchange), 2

IKE identity types, 224

IKE trap commands, SNMP
 (Simple Network Management
 Protocol), 427

IKE_AUTH, 13, 24, 42, 174, 175–176

 authentication, 42–43

 encryption, 42–43

 identity, 45

 parameters, 43

 traffic selectors, 50–52

 troubleshooting, 464

*ECDSA (Elliptic-Curve Digital
 Signature Algorithm)*
 authentication, 465–468

RSA authentication, 465–468

IKE_SA_INIT, 25–26, 124

IKEv1

anti-denial of service, 72

authentication, 73–74

combined mode ciphers, 77

configuration payload, 75

continuous channel mode, 77

cryptographic ciphers, 77

deployments, 551–552

exchange modes, 70–71

high availability, 74

history of, 67–69

lifetime, 72–73

matching on identity, 75–76

NAT (Network Address Translation),
 74–75

traffic selectors, 74

IKEv2 (Internet Key Exchange protocol version 2), 1

anti-denial of service, 72

authentication, 14, 45, 73–74

*EAP (Extensible Authentication
 Protocol)*, 48–50

*pre-shared-key-based
 authentication*, 47

*signature-based
 authentication*, 46

authorization policy, 228–229

auto-reconnect

FlexVPN server, 309–310

*FlexVPN server, configuration
 attributes*, 310–311

FlexVPN server, configuring,
 313–315

FlexVPN server, smart DPD,
 311–313

CAC (Call Admission Control), 157

certificate requests, 38–39

*HTTP_CERT_LOOKUP_
 SUPPORTED*, 39

- clearing IKEv2 FlexVPN client sessions, 360
- client debugging, 450
- cluster debugging, 450
- combined mode ciphers, 77
- configuration constructs, 106
- configuration exchange, FlexVPN client, 334
- configuration payload, 75
- considerations when moving to IKEv2
 - asymmetric routing*, 547–548
 - authentication*, 546–547
 - client awareness*, 545
 - current capacity*, 542–543
 - current VPN technology*, 540–541
 - familiarization*, 545
 - FlexVPN*, 544
 - global IKE*, 543–544
 - hardware limitations*, 540
 - high availability*, 547
 - IP addresses*, 543
 - IPsec commands*, 543–544
 - IPv6*, 546
 - PKI (Public Key Infrastructure)*, 545–546
 - restrictions when running IKEv1 and IKEv2 simultaneously*, 541–542
 - routing protocols*, 541
 - software*, 543
 - VPN gateways*, 543
- context-specific configuration, 106
- continuous channel mode, 77
- Cookie Challenge, 156
- cryptographic exchange bloat, 77
- debug commands, 501
- debugging, 449–453
- deployments
 - Cookie Challenge and Call Admission Control*, 207–210
 - ECDSA (Elliptic-Curve Digital Signature Algorithm) authentication*, 194–200
 - pre-shared-key authentication with smart defaults*, 189–194
 - RSA authentication using HTTP URL lookup*, 200–207
- diagnose error, 460–461
- diagnostics, 159
- Diffie-Hellman group, configuring, 113–114
- digital signatures, 12–13
- DPD (Dead Peer Detection), 149–151, 158–159
- dual-factor authentication, 320–321
- encrypted payload structures, 43–44
- encryption, 14
 - algorithms in Cisco IOS*, 111
 - configuring*, 111–112
- error debugging, 450
- exchange modes, 71–72
- FlexVPN, 213–214
- fragmentation, 171–172, 173–178
- global configuration, 106, 155
 - HTTP URL-based certificate lookup*, 156
- guiding principles, 106
- high availability, 74
- identities, 44–45, 74
- initial contact, 52, 151
- integrity, configuring, 113
- internal debugging, 450

- key material generation, 39–42
- keyring, 106, 128–129
 - configuration examples*, 134–135
 - configuring*, 129–132
 - key lookup on initiators*, 132–133
 - key lookup on responders*, 133–134
 - overview*, 136
 - pre-shared-key authentication with smart defaults*, 190
- lifetime, 72–73
- load balancing, 374–375
 - examples*, 376–378
- matching on identity, 75–76
- MOBIKE, 75
- name mangler, 223–224
 - configuring*, 224–227
- NAT (Network Address Translation), 61–64, 74–75
- NAT keepalives, 152, 159
- overview, 23–24
- packet debugging, 450
- packet structure, 55–56
- PKI (Public Key Infrastructure)
 - examples*, 164–166
 - trustpoints (TP)*, 163–164
- policies, 106, 117–118
 - configuring*, 118–119
 - configuring match statements*, 120–121
 - configuring proposals under*, 119–120
 - default IKEv2 policies*, 121–122
- policy configuration examples
 - multiple proposals*, 126–127
 - per-peer IKEv2 policies*, 125–126
- policy selection
 - initiators*, 122–124
 - responders*, 124–125
- pre-shared key lookup parameters, 134
- profiles, 106, 136–137
 - configuring*, 138–139
 - configuring match statements*, 139–142
 - defining local and remote authentication methods*, 145–149
 - defining local IKE identities*, 143–145
 - defining scope*, 143
 - disabling*, 153
 - displaying*, 153–154
 - initial contact*, 151
 - initiators and responders*, 154
 - IVRF (inside VRF)*, 152
 - lifetime*, 151
 - matching peer identity*, 142–143
 - matching peers by identity*, 141–142
 - NAT keepalives*, 152
 - overview*, 154–155
 - peer authorization database*, 137–138
 - pre-shared-key authentication with smart defaults*, 190
 - virtual template interface*, 153
- proposals, 106, 107–108
 - configuring*, 108–111
 - configuring under IKEv2 policies*, 119–120
 - default proposals*, 115–117

- protected tunnel interface, 558
- pseudorandom function,
 - configuring, 115
- public-private key pair, 162
- redirect, FlexVPN load balancer,
 - 363–366, 372–373
- reliability, 77
- request-response, 61
- routing, 496–498
- SAs (Security Associations)
 - lifetime*, 151
 - rekey*, 54–55
- session authentication, 181–182
- session deletion on certificate
 - expiry, 184
- session deletion on certificate
 - revocation, 182–184
- session lifetime, 185–187
- SGT (security group tags), 178–181
- shared secrets, 13
- show commands, 500
- smart defaults, 168–169
- SNMP (Simple Network
 - Management Protocol), 425–427
- standard attributes, 253
- syslog messages, 428–429
- traffic selectors, 74
- window size, 158
- IKEv2 exchange, 24–25**
 - cookie notification, 36–38
 - Diffie-Hellman exchange, 26–29
 - IKE_SA_INIT, 25–26
 - nonce, 35–36
 - SPI (Security Parameter Index),
 - 34–35
- IKEv2 SA, 555–556**
- IKEv2-password-local, 285**
- IKEv2-password-remote, 285**
- impact of, fragmentation, 535–536**
- implementation modes, GRE (generic

 - routing encapsulation)
 - auto tunnel mode, 99
 - dual stack, 96
 - mixed mode, 96–99**
- implicit authorization, 269, 285**
 - FlexVPN, 242–245
- import all, 343**
- inac1, 304**
- incoming sessions, deriving virtual

 - access configurations, FlexVPN
 - server, 294**
- INFORMATIONAL exchange, 56**
 - deleting SAs (Security Associations),
 - 57–58
- initial contact, IKEv2, 52, 151**
- initial exchange, 24**
- initial handshake, 24**
- INITIAL_CONTACT, 52**
- initialization vector (IV), 18**
- initiators**
 - Diffie-Hellman exchange, 28
 - key lookup, 132–133
 - policy selection, IKEv2, 122–124
 - profiles, IKEv2, 154
- inside VRF (IVRF), 118**
- INTEG (Integrity algorithm), 30**
- integrity, 15**
 - connectionless integrity, 4
 - IKEv2, configuring, 113
- Integrity Algorithm, 30**
- integrity algorithms, 511–512**
- Integrity Check Value (ICV), 4, 15**
- integrity overhead, 511–512**
- interface Tunnel number, 91**
- interface-config AAA attribute,

 - 293–294**

interface-config attribute, 296, 303

interfaces

dynamic tunnel interfaces, 91–92

flapping tunnel interface, 499

P2P (point-to-point) tunnel interfaces, 214–221

per-peer P2P tunnel interfaces, 221

static P2P tunnel interfaces, 214–216

static tunnel interfaces, 90

traffic selection by routing, 88–90

virtual access interfaces, 217

virtual interfaces, 87–88

virtual template interface, 291–293

configuring, 216–219

virtual-access interface, 290–291

internal debugging, IKEv2, 450

INTERNAL_IP4_ADDRESS, 254

INTERNAL_IP4_DNS, 256

INTERNAL_IP4_NBNS, 257

INTERNAL_IP4_NETMASK, 254

INTERNAL_IP4_SUBNET, 255

INTERNAL_IP6_ADDRESS, 255

INTERNAL_IP6_DNS, 256

INTERNAL_IP6_SUBNET, 256

Internet Engineering Task Force. *See*
IETF (Internet Engineering Task
Force)

Internet Key Exchange (IKE), 2

Internet Key Exchange protocol
version 2. *See* IKEv2

ip | ipv6 unnumbered, 102

ip address, 215

ip address negotiated, 334, 348

IP addresses, considerations when
moving to IKEv2, 543

IP Authentication Header, 15–16

IP connectivity, 423–424

troubleshooting, 457–460

IP fragmentation, overview, 172–173

ip http server, 161

ip mtu command, 499, 531

ip nat inside, 405

ip nat outside command, 355, 405

IP protocol numbers, 50

ip tcp adjust-mss, 527

ip unnumbered, 218

ip unreachable, 520

ip vrf, 100

ip vrf forwarding, 100, 296

IP_FQDN, 44

IPsec

anti-replay services, 491–494

building blocks of, 2

access control, 4

anti-replay services, 4

confidentiality, 4

connectionless integrity, 4

data origin authentication, 4

key management protocol, 3

SAs (Security Associations), 3

security protocols, 2–3

security services, 3

*TFC (Traffic Flow
Confidentiality)*, 4–5

components of, 5

*PAD (Peer Authorization
Database)*, 6

*SAD (Security Association
Database)*, 6

*SPD (Security Policy
Database)*, 5–6

*SPI (Security Parameter
Index)*, 5

debug commands, 501

debugging, 488–491

fragmentation

- impact of*, 535–536
- IPv4, 519–522
- IPv6, 522–523
- MTU (*maximum transmission unit*), 518–519
- PMTUD (*path MTU discovery*), 523–525, 527–531
- TCP MSS clamping, 525
- GRE over IPsec, 83
- modes of, 20
 - transport mode*, 20–21
 - tunnel mode*, 21
- overlay routing, 495
- show commands, 500
- VRF (Virtual Routing and Forwarding), 99–101
- VRF aware, 101–102
- IPsec commands, considerations when moving to IKEv2, 543–544
- IPsec configuration, 166–167
 - examples, 168
 - smart defaults, 168–169
- IPsec debugging, 453
- IPsec dVTI, 153
- IPsec mode overhead (without GRE), 505
- IPsec overhead, 504–505
 - encapsulating security payload overhead, 507–509
 - examples, 516–517
 - GRE overhead, 505–507
 - IPsec mode overhead (without GRE), 505
 - plaintext MTU, 513–514
- IPsec profiles, 167
- IPsec Remote Access Client (IRAC), 58
- IPsec Remote Access Server (IRAS), 58
- IPsec SA
 - creating, 53–54
 - rekey, 54
 - traffic selectors, 51
- IPsec security services, 3
- IPsec SNMP trap, 437
- IPsec transport mode, GRE over IPsec, 83–84
- IPsec tunnel encapsulation, 92
- IPsec tunnel mode
 - GRE over IPsec, 84–85
 - GRE/IPsec, 87–88
 - VTI (Virtual Tunnel Interface), 87–88
- IPsec VPN methodology, troubleshooting, 446
- IPsec VPNs, 2
 - cryptography, 7
 - asymmetric cryptography*, 8
 - Diffie-Hellman exchange*, 8–11
 - symmetric cryptography*, 7–8
- IPsec-v3 standards, 504
- IPv4
 - IPsec, fragmentation, 519–522
 - tunnels, mixed mode, 96
- IPv4 DNS server attributes, configuring, 340
- IPv4 WINS attribute, 343
- ipv4-pool, 250
- IPv6
 - considerations when moving to IKEv2, 546
 - IPsec, fragmentation, 522–523
 - pre-shared-key authentication with smart defaults, 191
- ipv6 address, 215
- IPv6 DNS server attributes, configuring, 341
- ipv6 mtu, 531

ipv6 tcp adjust-mss, 527
 ipv6 unnumbered, 218
 ipv6 unreachable, 522
 IRAC (IPsec Remote Access Client), 58
 IRAS (IPsec Remote Access Server), 58
 irvf command, 152
 ISAKMP, 68
 ISE (identity services engine), 178
 IV (initialization vector), 18
 IVRF (inside VRF), 118
 profiles, IKEv2, 152

K

KE (Key Exchange) payload, 27–28
 keepalives, 24, 59–61
 NAT keepalives, 152, 159
 KEi, 26
 KEr, 26
 Key Exchange payload. *See* KE (Key Exchange) payload
 key lookup
 initiators, 132–133
 responders, 133–134
 Key Management Interface. *See* KMI (Key Management Interface)
 key management protocol, 3
 key material generation, IKEv2, 39–42
 key pairs, public-private key pair, 162
 Key Usage, 469
 KEY_ENG_DELETE_SAS, 454
 KEY_ENG_IPSEC_READY, 454
 KEY_ENG_NOTIFY_INCR_COUNT, 454
 KEY_ENG_REQUEST_SAS, 454
 KEY_MGR_CREATE_IPSEC_SAS, 454
 KEY_MGR_DELETE_SAS, 455
 KEY_MGR_IKMP_READY, 454
 KEY_MGR_SESSION_CLOSED, 455
 KEY_MGR_VALIDATE_IPSEC_PROPOSALS, 454
 keyring, IKEv2, 106, 128–129
 configuration examples, 134–135
 configuring, 129–132
 configuring peer blocks, 130
 key lookup on initiators, 132–133
 key lookup on responders, 133–134
 overview, 136
 pre-shared-key authentication with smart defaults, 190
 keyring aaa, 223, 225
 keys
 AAA-based pre-shared keys
 FlexVPN, 381–382
 FlexVPN server, 284
 RADIUS attributes, 285
 asymmetric keys, 132
 authentication pre-shared keys, 429–431
 configuring, in peer blocks, 132
 GRE tunnel keys, mismatching, 495
 pre-shared keys, 478–479
 cryptographic strength, 135
 FlexVPN client, 356
 symmetric keys, 132
 KeyUsage extension, 469
 keywords
 cached, 234
 detail, 559
 dynamic keyword, 346

max-redirects, 374

sign, 476

timeout, 484

KMI (Key Management Interface),
debugging, 453–455

L

legacy algorithms, 111

lifetime, 72–73

IKEv2, SAs (Security
Associations), 151

SAs (Security Associations), 7

session lifetime, 185–187

lifetime certificate, 184

liveness checking, 24

load balancers, FlexVPN load
balancer. *See* FlexVPN load
balancer

load balancing

high availability, 547

troubleshooting, IKEv2 load
balancing, 374–375

local AAA database, FlexVPN, group
authorization, 238–239

local authentication methods, IKEv2
profiles, 145–149

local IKE identities, defining,
143–145

logs, FlexVPN client, 390–391

M

Main mode, 70

Management Information
Base (MIB), 419

man-in-the-middle (MITM) attack, 13

manual mode, tunnel initiation
(FlexVPN client), 350

maps, crypto maps, 86–87

master session key (MSK), 272

match address local, 142–143

match certificate command, 472

match certificates, 139, 140–141

match command, 441

match fvrfr, 142–143

match fvrfr any statement, 122

match identity, 139, 141–142

match identity remote any, 142

match statements, configuring
under IKEv2 policies, 120–121
in IKEv2 profiles, 139–142

matching

peer identity, 142–143

peers

with certificate maps,
472–473

by certificates, 140–141

by identity, 141–142

matching on identity, 75–76

maximum authentication header
overhead, 516

maximum ESP overhead, 515

maximum ICV padding, 514

maximum input padding, 514

maximum output overhead, 514

maximum overhead, 514–515

Maximum Segment Size (MSS),
525–527

maximum transmission
unit (MTU), 172

IPsec, fragmentation, 518–519

max-redirects, 374

MD5, 115

Md5, 113

- messages, Proposal Incomplete message, 108
- method-est, 163
- methodologies, monitoring methodology, 422–423
- mGRE, 83
- mGRE (multipoint GRE), 92–94
- MIB (Management Information Base), 419
- Microsoft Windows 7 IKEv2 clients, FlexVPN server, 329–330
- migration strategies, 539, 548
 - considerations when moving to IKEv2
 - asymmetric routing*, 547–548
 - authentication*, 546–547
 - client awareness*, 545
 - current capacity*, 542–543
 - current VPN technology*, 540–541
 - familiarization*, 545
 - FlexVPN*, 544
 - global IKE*, 543–544
 - hardware limitations*, 540
 - high availability*, 547
 - IP addresses*, 543
 - IPsec commands*, 543–544
 - IPv6*, 546
 - PKI (Public Key Infrastructure)*, 545–546
 - restrictions when running IKEv1 and IKEv2 simultaneously*, 541–542
 - routing protocols*, 541
 - software*, 543
 - VPN gateways*, 543
- remote access, 565–566
- topologies, 561
 - hub-and-spoke topology*, 562–564
 - site-to-site*, 561–562
- transitioning from IKEv1 to IKEv2, 548
 - hard migration*, 548–549
 - soft migration*, 549–559
- verification, 559–561
- migration verification, 559–561
- mismatching, GRE tunnel keys, 495
- MITM (man-in-the-middle) attack, 13
 - Diffie-Hellman exchange, 45
- mixed mode, GRE (generic routing encapsulation), 96–99
- MMx, 70
- MOBIKE, 75
- Mobility and Multihoming protocol, 75
- modes, 79
 - continuous channel mode, 77
- modes of encapsulation, 82
 - GRE encapsulation, 82–83
 - GRE over IPsec, 83
- modes of IPsec, 20
 - transport mode, 20–21
 - tunnel mode, 21
- modifying
 - default IKEv2 policies, 122
 - default IKEv2 proposals, 116–117
- monitor even-trace crypto ipsec, 448
- monitor event-trace, 449
- monitor event-trace crypto ikev2, 448
- monitoring
 - AAA (authentication, authorization, and accounting), 418
 - authentication EAP, 434–436

- authentication PKI, 431–434
- authentication pre-shared keys, 429–431
- authorization, RADIUS-based AAA, 436
- data encryption, SNMP with IPsec, 437–439
- data usage, 440–443
- IP connectivity, 423–424
- NetFlow, 418–419
- overlay routing, 439–440
- SNMP (Simple Network Management Protocol), 419–420
- syslog, 421
- VPN tunnel establishment, 425
 - Cisco IPsec flow monitor MIB*, 425
 - SNMP with IKEv2*, 425–427
- monitoring methodology, 422–423
- MPLS (Multiprotocol Label Switching), 2
- MSK (master session key), 272
- MSS (Maximum Segment Size), 525–526
 - adjustments, 526–527
- MSS clamping, 525
- MTU (maximum transmission unit), 172
 - IPsec, fragmentation, 518–519
 - plaintext MTU, 513–514
- mtu command, 518, 533
- multicast traffic, 85–86
- multiple proposals, IKEv2 policy
 - configuration examples, 126–127
- multipoint GRE (mGRE), 92–94
- Multiprotocol Label Switching. *See* MPLS (Multiprotocol Label Switching)
- multi-SA dVTI, 92

N

- name mangler, IKEv2, 223–224
 - configuring, 224–227
- name verification, AnyConnect, 468
- names, extracting
 - from DN identity, 226–227
 - from EAP identity, 227
 - from email identity, 226
 - from FQDN identity, 225–226
- NAPT (network address port translation), 221
- NAT (Network Address Translation), 74–75
 - FlexVPN client, 335, 354–355, 404–405
 - verification*, 405–407
 - IKEv2, 61–64
 - keepalives, 59–61
- NAT keepalives, 59–61, 152, 159
- NAT-D (Network Address Translation-Detection), 64
- negotiations, SGT (security group tags), IKEv2, 178–181
- NetFlow, 418–419, 440–441
- network address port translation (NAPT), 221
- network extension modes, FlexVPN client, 336
- Network Time Protocol (NTP), 471
- Next Hop Resolution Protocol (NHRP), 86, 93
- next-generation encryption (NGE), 112
- NGE (next-generation encryption), IKEv2, 125
- NHRP (Next Hop Resolution Protocol), 86, 93
- Ni, 26

- no crypto ipsec nat-transparency
 udp-encapsulation, 64
- no lifetime, 184
- no logging event link-status, 424
- no route accept, 267
- no shutdown command, 161
- non-broadcast, 93
- nonce, IKEv2 exchange, 35–36
- non-IP protocols, 86
- Notification payload, 56
- notifications, REDIRECT notification,
 363–366
- Nr, 26
- NTP (Network Time Protocol), 471
- null encryption, 16

O

- OAKLEY, 68
- object tracking, FlexVPN client, 334
 - EEM (embedded event manager),
 356–358
- OCSP (online certificate status
 protocol), 181
- on-demand mode, DPD (Dead Peer
 Detection), 150
- online certificate status protocol
 (OCSP), 181
- outacl, 304
- outbound IPsec SA parameters, 215
- overhead
 - authentication header overhead,
 509–510
 - combined algorithm overhead,
 512–513
 - encryption overhead, 510–511
 - extra overhead, 516–517
 - GRE (generic routing encapsulation),
 505–507

- integrity overhead, 511–512
- IPsec. *See* IPsec overhead
- maximum authentication header
 overhead, 516
- maximum ESP overhead, 515
- maximum overhead, 514–515
- plaintext MTU, 513–514
- overlay routing, 439–440, 495
- overload limit, 372

P

- P2P (point-to-point) tunnel interfaces,
 214–221
- packet debugging, IKEv2, 450
- packet structure, IKEv2, 55–56
- Packet-of-Disconnect (PoD), 299
- PAD (Peer Authorization Database), 6
 - IKEv2, profiles, 137–138
- parameters
 - global configuration, IKEv2, 155
 - IKE_AUTH, 43
 - outbound IPsec SA parameters, 215
 - pre-shared key lookup parameters,
 IKEv2, 134
 - profiles, IKEv2, 136–137
- path MTU discovery (PMTUD),
 172–173
 - IPsec, fragmentation, 523–525
- Peer Authorization Database. *See*
 PAD (Peer Authorization Database)
- peer blocks, configuring
 - in keyring, 130
 - keys, 132
 - peers, 130–131
- peer identity, matching, IKEv2,
 142–143
- peer reactivate, 346

peers, 2

- backup peers, FlexVPN hub resiliency, 411
- configuring, in peer blocks, 130–131
- matching
 - with certificate maps*, 472–473
 - by certificates*, 140–141
 - by identity*, 141–142

- reactivating, FlexVPN client, 346
- remote subnets, FlexVPN, 266–267
- revoked peers, disconnecting, 182

PEM (Privacy Enhanced Mail), 201**periodic mode, DPD (Dead Peer Detection), 150****per-peer IKEv2 policies, 125–126****per-peer P2P tunnel interfaces, 221****per-session interface, FlexVPN server, 290–291****PKI (Public Key Infrastructure), 11, 159–160, 456**

- authentication, 431–434
- CA (certificate authority), 12, 160–162
- certificate-based authentication, 147–148
- considerations when moving to IKEv2, 545–546
- debug commands, 502
- debugging, 456
 - authentication*, 470
- digital certificates, 12
- examples, 164–166
- FlexVPN client, 356
- public-key cryptography, 11–12
- public-private key pair, 162
- show commands, 501
- trustpoints (TP), 163–164

PKI trustpoints, 148–149, 470**plaintext MTU, 513–514****PMTUD (path MTU discovery), 172–173**

- fragmentation, tunnels, 534–535
- IPsec, fragmentation, 523–525, 527–531

PoD (Packet-of-Disconnect), FlexVPN server, 299**point-to-point tunnel interfaces. *See* P2P (point-to-point) tunnel interfaces****policies, IKEv2, 106, 117–118**

- authorization policy, 228–229
- configuring, 118–119
- configuring match statements, 120–121
- configuring proposals under, 119–120
- default IKEv2 policies, 121–122

policy configuration examples, IKEv2

- multiple proposals, 126–127
- per-peer IKEv2 policies, 125–126

policy selection, IKEv2

- initiators, 122–124
- responders, 124–125

precedence, FlexVPN

- group authorization, 249–250
- user authorization, 247–249

pre-shared key authentication, 147**pre-shared key lookup parameters, IKEv2, 134****pre-shared keys, 13, 478–479**

- cryptographic strength, 135
- FlexVPN client, 356

pre-shared-key authentication with smart defaults, 189–194**pre-shared-key-based authentication, 47**

PRF (pseudorandom function), 7, 30, 31, 40–41

algorithms, 41

IKEv2, configuring, 115

Privacy Enhanced Mail (PEM), 201

profiles

FlexVPN client, 345–346

IKEv2, 106, 136–137

configuring, 138–139

configuring match statements, 139–142

defining local and remote authentication methods, 145–149

defining local IKE identities, 143–145

defining scope, 143

disabling, 153

displaying, 153–154

initial contact, 151

initiators and responders, 154

IVRF (inside VRF), 152

lifetime, 151

matching peer identity, 142–143

matching peers by identity, 141–142

NAT keepalives, 152

overview, 154–155

peer authorization database, 137–138

pre-shared-key authentication with smart defaults, 190

virtual template interface, 153

IPsec, 167

Proposal Incomplete message, 108

proposals

IKEv2, 106, 107–108

configuring, 108–111

configuring under IKEv2 policies, 119–120

default proposals, 115–117

multiple proposals, IKEv2 policy configuration examples, 126–127

Security Association Proposals, 29–34

protected tunnel interface, IKEv2, 558

proto id, 491

protocols

AAA (authentication, authorization, and accounting), 418

AH (Authentication Header), 2–3

Authentication Header (AH), 15

CDP (Cisco Discovery Protocol), 86

dynamic routing protocols, 498–499

EAP (Extensible Authentication Protocol). *See* EAP (Extensible Authentication Protocol), 48–50

ESP (Encapsulating Security Payload), 2–3, 17

HSRP (hot standby routing protocol), FlexVPN load balancer, 366–367

ISAKMP, 68

NetFlow, 418–419

NHRP (Next Hop Resolution Protocol), 86, 93

non-IP protocols, 86

OAKLEY, 68

OCSP (online certificate status protocol), 181

RFC 4301, 3

routing protocols, considerations when moving to IKEv2, 541

SKEME, 68

SNMP (Simple Network Management Protocol), 419–420

- SXP (Security group tag exchange), 179
- syslog, 421
- UDP (User Datagram Protocol), 25
- pseudorandom function. *See* PRF (pseudorandom function)
- PSK, authentication, 429
- Public Key Infrastructure. *See* PKI (Public Key Infrastructure)
- public-key cryptography, 8
 - PKI (Public Key Infrastructure), 11–12
- public-private key pair, 162

Q

- QCR (quantum computer resistant), 112
- query, 337
- query-identity, 274, 482
 - FlexVPN server, 277
- Quick Mode, 70

R

RADIUS

- debug commands, 501
- PSK configuration, 478

RADIUS accounting, 287

- authentication pre-shared keys, 429

RADIUS attributes

- AAA-based pre-shared keys, 285
- CoA (change-of-authorization), 303–304
- FlexVPN server, 325–329

RADIUS change-of-authorization (CoA)

- configuring, 304

- FlexVPN server, 303–304

- examples*, 305–309

- updating

- session ACL*, 307–309

- session QoS policies*, 305–307

RADIUS Packet-of-Disconnect, FlexVPN server, 299–300

- configuring, 300
- examples, 301–303

RADIUS servers

- configuring

- AAA-based pre-shared keys*, 384–386

- FlexVPN*, 388–390

- EAP (Extensible Authentication Protocol), configuration
- examples, 278, 280–281

- reactivating peers, FlexVPN
- client, 346

- reconnect, 310

- Reconnect capable active session count, 315

- Reconnect capable inactive session count, 315

- ReconnectAfterResume, 311

- reconnect-cleanup-interval, 311

- reconnect-dpd-interval, 311

- reconnect-session-id, 310

- reconnect-timeout, 313

- reconnect-token-id, 310

- redirect, IKEv2, FlexVPN load balancer, 363–366, 372–373

- redirect loops, FlexVPN load balancer, 373–374

- redirect mechanisms, 65

- REDIRECT notification, 363–366

- REDIRECT payload, 65

- re-enabling

default IKEv2 policies, 122	RFC 791, 520
default IKEv2 proposals, 116	RFC 2401, 68
rekey, 7	RFC 2402, 68
IPsec SA, 54	RFC 2403, 68
SAs (Security Associations), IKEv2, 54–55	RFC 2404, 68
RFC 2405, 68	RFC 2405, 68
reliability, 77	RFC 2406, 68
remote access, migration strategies, 565–566	RFC 2407, 68
remote access clients, FlexVPN server, 329	RFC 2408, 68
Cisco IKEv2 AnyConnect clients, 330	RFC 2409, 68, 73
Microsoft Windows 7 IKEv2 clients, 329–330	RFC 2410, 68
remote authentication methods, IKEv2 profiles, 145–149	RFC 2411, 68
remote subnets, FlexVPN, 264	RFC 2412, 68
learning from peer, 266–267	RFC 2459, 469
learning locally, 265–266	RFC 3164, 421
request-response, IKEv2, 61	RFC 3526, 11
resolution of FQDN (fully qualified domain names), FlexVPN client, 346	RFC 3706, 69
responder-only, 167	RFC 3715, 69
responders	RFC 3748, 48, 73
key lookup, 133–134	RFC 3947, 69
policy selection, IKEv2, 124–125	RFC 3948, 69
profiles, IKEv2, 154	RFC 4301, 3, 137
restoring modified default IKEv2 proposals, 117	RFC 4302, 15
restrictions when running IKEv1 and IKEv2 simultaneously, 541–542	RFC 4304, 69
revocation, certificate revocation, 473–476	RFC 4478, 182
revocation-check method command, 164	RFC 4555, 75
revoked peers, disconnecting, 182	RFC 4739, 321
RFC (Request for Comments), 23	RFC 4754, 69, 73
	RFC 4821, 525
	RFC 4945, 469
	RFC 5114, 11
	RFC 5685, 65
	RFC 5716, 299
	RFC 5998, 65
	RFC 6023, 66
	RFC 6989, 65

RFC 7296, 44, 65
 RFC 7383, 174
 RIB (Routing Information Base),
 440, 495
 Rivest-Shamir-Adleman (RSA) key
 pair, 160
 Rivest-Shamir-Adleman Signature, 12
 route accept, 266, 498
 route accept any, 230
 route set interface, 230, 496
 route set interface statement, 497
 route set local, 267, 498
 routing
 asymmetric routing, considerations
 when moving to IKEv2, 547–548
 FlexVPN, 264–265, 391–392
 IKEv2, 496–498
 overlay routing, 439–440, 495
 static routing, 496
 traffic selection, 88–90
 routing adjacency, 498
 Routing Information Base (RIB),
 440, 495
 routing protocols, considerations
 when moving to IKEv2, 541
 RSA (Rivest-Shamir-Adleman) key
 pair, 160
 RSA authentication, troubleshooting,
 465–468
 RSA authentication using HTTP URL
 lookup, 200–207

S

SAD (Security Association
 Database), 6
 SAil, 26
 SArl, 26
 SAs (Security Associations), 2–3
 creating, 53–54
 deleting, 57–58
 IKEv2
 lifetime, 151
 rekey, 54–55
 lifetime, 7
 rekey, 54
 scope, IKEv2, profiles, 143
 Security Association Database.
 See SAD (Security Association
 Database)
 Security Association Proposals,
 29–34
 Security Association (SA), creating,
 53–54
 Security Associations. *See* SAs
 (Security Associations)
 Security Associations (SA), deleting,
 57–58
 Security group tag exchange
 (SXP), 179
 security group tags. *See* SGT
 (security group tags)
 Security Information Event
 Management. *See* SIEM (Security
 Information Event Management)
 security levels
 IKEv2 syslog messages, 428–429
 syslog, 421
 Security Parameter Index (SPI), 3, 5
 security payload overhead,
 encapsulating, 507–509
 Security Policy Database. *See* SPD
 (Security Policy Database)

SA_INIT, 24, 174–175
 Diffie-Hellman exchange, 29
 parameters, 26
 SA_INIT exchange, troubleshooting,
 461–464

- security protocols, 2–3
- security services, IPsec, 3
- selecting, trustpoints (TP), 476–477
- sequence of events, FlexVPN server, 270–271
- service-policy, 296
- service-policy input, 304
- session ACL, updating, with RADIUS CoA, 307–309
- session authentication, IKEv2, 181–182
- session deletion on certificate expiry, 184
- session deletion on certificate revocation, IKEv2, 182–184
- session lifetime, IKEv2, 185–187
- session QoS policies, updating, with RADIUS CoA, 305–307
- set ikev2-profile, 167
- set mixed-mode, 99, 167
- set peer hostname dynamic, 130
- set pfs, 167
- set reverse-route, 167
- set security-association, 167
- set security-association replay window-size disable, 494
- set transform-set, 167
- SET_WINDOW_SIZE notification payload, 158
- SGT (security group tags), 171
 - IKEv2, 178–181
- SHA1, 115
- Sha1, 113
- SHA256, 115
- SHA384, 115
- Sha384, 113
- SHA521, 115
- Sha521, 113
- shared secrets, 13
- shared-key-based authentication, 47
- show aaa attribute protocol radius, 228
- show cef interface, 531
- show commands
 - IKEv2, 500
 - IPsec, 500
 - PKI (Public Key Infrastructure), 501
 - troubleshooting, 447
 - FlexVPN client*, 358–359
- show crypto ikev2 authorization policy, 229
- show crypto ikev2 client flexvpn, 358
- show crypto ikev2 client flexvpn flex1 detail, 342
- show crypto ikev2 client flexvpn name, 358
- show crypto ikev2 cluster, 369, 374
- show crypto ikev2 diagnose error, 460
- show crypto ikev2 flexvpn, 348
- show crypto ikev2 proposal, 34, 109, 463
- show crypto ikev2 sa detail, 282
- show crypto ikev2 sa detailed, 144, 177, 313, 343, 359, 397, 399
- show crypto ikev2 session detailed, 144
- show crypto ikev2 stats reconnect, 315
- show crypto ipsec sa, 530, 560
- show crypto ipsec transform-set, 488
- show crypto pki certificate verbose, 480
- show crypto pki certificates, 467
- show crypto pki counters, 475
- show crypto pki trustpoints, 467

- show crypto session, 220, 559
- show crypto session brief, 560
- show crypto session detail, 186, 313
- show crypto sessions, 95
- show crypto sockets, 95
- show derived-config, 91
- show ip dhcp import, 343
- show ip dns name-list, 341, 343
- show ip interfaces, 518
- show ip nat statistics, 406
- show ip nat translations, 406
- show ip route, 266, 267
- show ip route vrf, 460
- show ip traffic, 522
- show ipv6 interfaces, 518
- show ipv6 traffic, 523
- show ntp associations, 471
- show platform hardware qfp active
 - feature ipfrag global, 522–523
- show run all, 184
- show running-config, 91
- show running-configuration
 - command, 478
- show standby, 375
- show track, 359
- SIA (subject information access), 202
- SIEM (Security Information Event Management), 417
- sign keyword, 476
- signature-based authentication, 46
- signatures
 - digital signatures, IKEv2, 12–13
 - hash algorithms, 163
- Simple Network Management Protocol (SNMP), 419–420
- site-to-site, migration strategies, 561–562
- SKEME, 68
- SKEY, 76
- SKEYID, 75
- SKEYSEED, 40, 54, 75
- slave priority, 370
- smart defaults, 106, 168–169
 - pre-shared-key authentication with
 - smart defaults, 189–194
- smart DPD, auto-reconnect, FlexVPN
 - server, 311–313
- SNMP (Simple Network Management Protocol), 419–420
 - IKE trap commands, 427
 - with IKEv2, VPN tunnel
 - establishment, 425–427
 - with IPsec, data encryption, 437–439
 - trap commands, 438–439
 - versions, 419–420
 - VRF-aware SNMP, 420
- SNMP agent, 419
- SNMP manager, 419
- snmp-server enable traps, 425
- snmp-server enable traps ike tunnel
 - start, 425
- snmp-server enable traps ike tunnel
 - stop, 426
- snmp-server enable traps ipsec tunnel
 - start, 437, 438
- snmp-server enable traps ipsec tunnel
 - stop, 439
- snmp-server enable traps snmp
 - linkdown linkup, 439–440
- snmp-server enable traps snmp linkup
 - linkdown, 424
- soft migration, transitioning from
 - IKEv1 to IKEv2, 549–559
- software, considerations when
 - moving to IKEv2, 543
- SPD (Security Policy Database), 5–6

SPI (Security Parameter Index), 3, 5

IKEv2 exchange, 34–35

Split-DNS

attributes, configuring, 341

FlexVPN client, 338–340

components of, 340–343

spoke configuration, FlexVPN

branch 1 configuration, 392–394

branch 2 configuration, 394–395

hub configuration, 395–397

verification at branch 1, 397–399

verification at branch 2, 399–400

verification on hub, 401–404

static P2P tunnel interfaces, 214–216

FlexVPN client, 334

static routing, 496

static tunnel interfaces, 90

static VTI (sVTI), 92

subject information access (SIA), 202

sub-modes

IKEv2 policies, 119

IKEv2 proposals, 108

subnets, remote subnets, FlexVPN,

264, 265–266

sub-policy-in, 304

sub-policy-out, 304

sub-qos-policy-in, 304

sub-qos-policy-out, 304

sVTI (static VTI), 92

SXP (Security group tag

exchange), 179

symmetric cryptography, IPsec

VPNs, 7–8

symmetric keys, 132

syslog, 421

IKEv2, 428–429

syslog messages,

troubleshooting, 447

T

TCAM (ternary content-addressable memory), 87

test aaa command, 481

TFC (Traffic Flow Confidentiality),
4–5, 504

ESP (Encapsulating Security Payload)
version 3, 20

timeout keyword, 484

timeout option, FlexVPN server,
275, 278

TLS (transport layer security), EAP
methods, 272

tools for troubleshooting, 446–447

event-trace monitoring, 447–449

show commands, 447

syslog messages, 447

topologies, migration strategies, 561

hub-and-spoke topology, 562–564

site-to-site, 561–562

TP (trustpoints), 148, 163–164, 195

configuring, 476

selecting, 476–477

track mode, tunnel, FlexVPN

client, 350

track-based tunnel activation,

FlexVPN backup tunnels, 414–415

tracking

FlexVPN client, 356

*EEM (embedded event man-
ager)*, 356–358

lists of objects, with boolean expres-
sions, 350–352

traffic

multicast traffic, 85–86

non-IP protocols, 86

routing, 88–90

Traffic Flow Confidentiality. *See* TFC (Traffic Flow Confidentiality), 504

traffic selectors, 4, 50–52, 74

GRE, 51–52

IPsec SA, 51

Transforms, 29–30

transitioning from IKEv1 to

IKEv2, 548

hard migration, 548–549

soft migration, 549–559

transport mode

FlexVPN, 219–221

GRE over IPsec, 83–84

IPsec, 20–21

traps

IPsec SNMP trap, 437

SNMP IPsec trap commands,
438–439

troubleshooting

debugging, 449

conditional debugging,
456–457

IKEv2, 449–453

IPsec, 488–491

IPsec debugging, 453

*KMI (Key Management
Interface),* 453–455

PKI (Public Key Infrastructure),
456

FlexVPN client

debugging, 360

show commands, 358–359

IKE_AUTH, 464

*ECDSA (Elliptic-Curve Digital
Signature Algorithm)*

authentication, 465–468

RSA authentication, 465–468

IKEv2, diagnose error, 460–461

IKEv2 load balancing, 374–375

IP connectivity, 457–460

IPsec VPN methodology, 446

SA_INIT exchange, 461–464

tools, 446–447

event-trace monitoring,
447–449

show commands, 447

syslog messages, 447

VPN tunnel establishment, 460

trustpoints (TP), 148, 163–164, 195

configuring, 476

selecting, 476–477

tunnel destination, 88, 215, 218

FlexVPN client, 349

tunnel destination dynamic, 349, 412

tunnel destination peer-address, 90

tunnel encapsulation modes, 215

auto detection, FlexVPN server,
297–298

FlexVPN, 219–221

tunnel endpoints, 88

tunnel initiation, FlexVPN client, 350

automatic mode, 350

manual mode, 350

track mode, 350

tunnel interface, 79, 229

FlexVPN client, 347–348

tunnel mode, 87–88, 215, 218

AH (Authentication Header), 3

ESP (Encapsulating Security
Payload), 3

GRE over IPsec, 84–85

IPsec, 21

tunnel mode auto, FlexVPN, 391–392

tunnel mode gre, 214–215, 217

tunnel mode gre ip, 94, 214, 216

- tunnel mode IPSEC, 298
- tunnel mode ipsec, 214–215, 217
- tunnel mode ipsec ipv4, 96
- tunnel mode ipsec ipv4 v6-overlay, 97
- tunnel mode ipsec ipv6, 96
- tunnel mode ipsec ipv6 v4-overlay, 97–98
- tunnel path-mtu-discovery, 499, 534–535
- Tunnel Pivot, 544
- tunnel protection, 80, 94–95, 139, 216, 218
 - versus crypto maps, 80–81
 - IPsec parameters, 167
- tunnel protection command, 167–168
- tunnel protection ipsec, 123
- tunnel source, 88, 218
 - FlexVPN client, 348–349
- tunnel source dynamic, 348, 408
- tunnel vrf name, 102
- Tunnel-Password, 285
- tunnels
 - FlexVPN backup tunnels, track-based tunnel activation, 414–415
 - fragmentation, 531
 - GRE (generic routing encapsulation)*, 532–533
 - GRE over IPsec*, 534
 - IPsec only (VTI)*, 531–532
 - PMTUD (path MTU discovery)*, 534–535
- type tunnel, 91

U

- UDP (User Datagram Protocol), 25
- uniform resource identifier (URI), 202

- updating
 - session ACL with RADIUS CoA, 307–309
 - session QoS policies, RADIUS change-of-authorization (CoA), 305–307
- URI (uniform resource identifier), 202
- user authentication, AnyConnect-EAP, 315, 316–318
- user authorization
 - FlexVPN, 235–237
 - precedence*, 247–249
 - FlexVPN client, 386
 - branch 1 configuration*, 386
 - branch 2 configuration*, 387
- User Datagram Protocol (UDP), 25

V

- value proposition, FlexVPN, 213
- verbose debugging, 181
- verification
 - FlexVPN client, 409–410, 412–413
 - NAT (Network Address Translation)*, 405–407
 - FlexVPN spoke
 - branch 1*, 397–399
 - branch 2 configuration*, 399–400
 - hub configuration*, 401–404
 - migration, 559–561
- VersionIDofSecret, 36–37
- VFRF (Front-door VRF), 118
- virtual access cloning, examples, 295–297
- virtual access configurations, FlexVPN server
 - deriving from AAA authorization, 293–294

- deriving from incoming sessions, 294
 - deriving from virtual templates, 291–293
- virtual access interfaces, 217
- virtual IPsec interfaces, 85–86
- Virtual Routing and Forwarding.
 - See* VRF (Virtual Routing and Forwarding)
- virtual template interface
 - configuring, 216–219
 - FlexVPN server, virtual access configurations, 291–293
 - IKEv2 profiles, 153
- Virtual Tunnel Interface (VTI), 87–88
- virtual-access interface, 290–291
- virtual-template 1 mode auto, 396
- virtual-template interfaces, FlexVPN feature, 91
- VPN gateways, considerations when moving to IKEv2, 543
- VPN peers, 2
- VPN technology, considerations when moving to IKEv2, 540–541
- VPN tunnel establishment, 425, 460
 - Cisco IPsec flow monitor MIB, 425
 - SNMP with IKEv2, 425–427
- VRF (Virtual Routing and Forwarding), 81, 118
 - IPsec, 99–101
- VRF aware, 101

- GRE (generic routing encapsulation), 101–102
 - GRE over IPsec, 102–103
 - IPsec, 101–102
- vrf definition, 100
- vrf forwarding, 100, 216, 218, 486
- vrf forwarding name, 102
- VRF-aware SNMP, 420
- VTI (Virtual Tunnel Interface), 87–88
 - fragmentation, 531–532

W

- wildcard keys, 130
- window size, IKEv2, 158
- WINS (Windows Internet Naming Service), FlexVPN client, 343–344
- worst case maximum overhead, 514–515

X-Y-Z

- XAUTH (Extended Authentication within IKE), 69
- XML
 - Aggregate XML, 315
 - AnyConnect-EAP XML messages, 322–324
- XML configurations, AnyConnect, 282–283



IKEv2 IPsec Virtual Private Networks

- Understand IKEv2 improvements: anti-DDoS cookies, configuration payloads, acknowledged responses, and more
- Implement modern secure VPNs with Cisco IOS and IOS-XE
- Plan and deploy IKEv2 in diverse real-world environments
- Configure IKEv2 proposals, policies, profiles, keyrings, and authorization
- Use advanced IKEv2 features, including SGT transportation and IKEv2 fragmentation
- Understand FlexVPN, its tunnel interface types, and IOS AAA infrastructure
- Implement FlexVPN Server with EAP authentication, pre-shared keys, and digital signatures
- Deploy, configure, and customize FlexVPN clients
- Configure, manage, and troubleshoot the FlexVPN Load Balancer
- Improve FlexVPN resiliency with dynamic tunnel source, backup peers, and backup tunnels
- Monitor IPsec VPNs with AAA, SNMP, and Syslog
- Troubleshoot connectivity, tunnel creation, authentication, authorization, data encapsulation, data encryption, and overlay routing
- Calculate IPsec overhead and fragmentation
- Plan your IKEv2 migration: hardware, VPN technologies, routing, restrictions, capacity, PKI, authentication, availability, and more

*This book is part of the **Networking Technology Series** from Cisco Press®, which offers networking professionals valuable information for constructing efficient networks, understanding new technologies, and building successful careers.*

Category: Networking Technology
Covers: Security

ciscopress.com

Create and manage highly-secure IPsec VPNs with IKEv2 and Cisco FlexVPN

The IKEv2 protocol significantly improves VPN security, and Cisco's FlexVPN offers a unified paradigm and command line interface for taking full advantage of it. Simple and modular, FlexVPN relies extensively on tunnel interfaces while maximizing compatibility with legacy VPNs. Now, two Cisco network security experts offer a complete, easy-to-understand, and practical introduction to IKEv2, modern IPsec VPNs, and FlexVPN.

The authors explain each key concept, and then guide you through all facets of FlexVPN planning, deployment, migration, configuration, administration, troubleshooting, and optimization. You'll discover how IKEv2 improves on IKEv1, master key IKEv2 features, and learn how to apply them with Cisco FlexVPN.

IKEv2 IPsec Virtual Private Networks offers practical design examples for many common scenarios, addressing IPv4 and IPv6, servers, clients, NAT, pre-shared keys, resiliency, overhead, and more. If you're a network engineer, architect, security specialist, or VPN administrator, you'll find all the knowledge you need to protect your organization with IKEv2 and FlexVPN.

Graham Bartlett, CCIE No. 26709, has designed large-scale VPNs throughout the U.K., and has collaborated with customers to deploy IKEv2 and Next Generation Encryption worldwide. Graham has discovered many IKEv2 zero-day vulnerabilities, contributed to IETF RFCs, and published intellectual property as prior art. A CiscoLive speaker, he has developed exam content for both CCIE and CCNP certifications. He holds CCP (Senior) IA Architect, CCP (Practitioner) Security & Information Risk Advisor, CCNP, CISSP, and Cisco Security Ninja credentials, and a BSc (Hons) in Computer Systems and Networks.

Amjad Inamdar, CISSP No. 460898, Senior Technical Leader with Cisco IOS Security Engineering in India, specializes in designing, developing, and deploying Cisco IOS secure connectivity solutions, including FlexVPN, DMVPN, GETVPN, and EzVPN. He is now working on Cisco's next generation SD-WAN solution. A contributor to IETF draft standards and regular presenter on security topics, Amjad holds a Cisco patent and has prior art publications. His credentials include CISSP, CCSK, CCNP Security, CCDP, CCNP R/S, CCNA (SP, Data Center, Wireless, and Voice) and Cisco Security Ninja; and a B.E. in Electronics and Communication Engineering.

ISBN-13: 978-1-58714-460-8
ISBN-10: 1-58714-460-3



\$64.99 USA / \$80.99 CAN