The New z/OS CommServer Internet Key Exchange Version 2 - What Is It and How Does It Integrate With An Existing IKEv1 Deployment?

SHARE Session 8232
February 28, 2011

Lin Overby - overbylh@us.ibm.com
Chris Meyer – meyerchr@us.ibm.com

z/OS Communications Server Security

Trademarks, notices, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

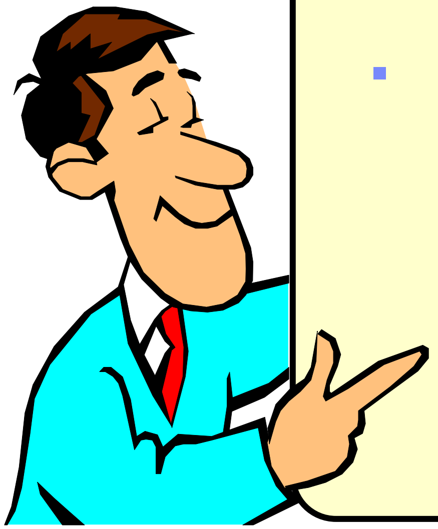
- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an “as is” basis, without warranty of any kind.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Refer to www.ibm.com/legal/us for further legal information.

Agenda



- **The new IKEv2 and related standards**
  - Why a new IKE?
  - IKEv1 and IKEv2 comparison
  - Additional function required to have a compliant IKEv2 implementation
- **IKEv2 implementation on z/OS**
  - New function overview and certifications
  - Administrative and operational externals
  - New certificate support
  - Migration considerations

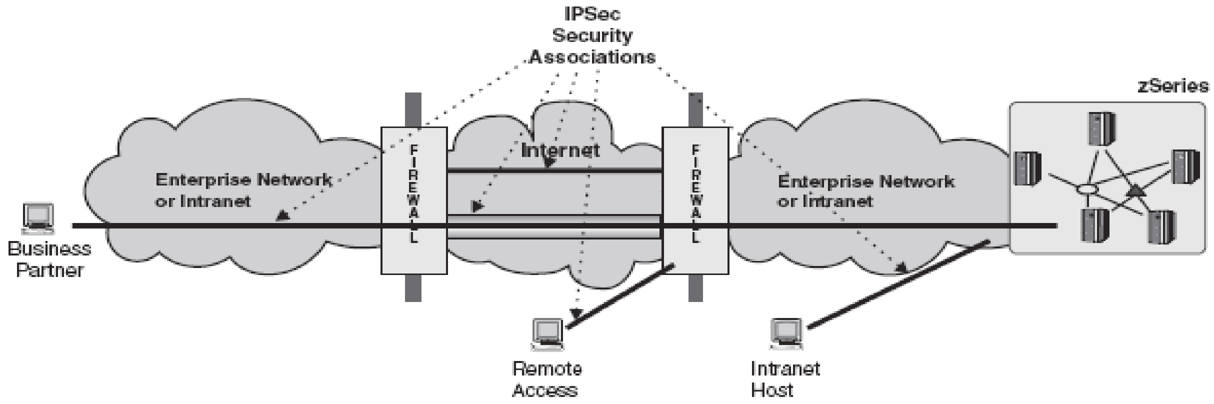
This presentation assume basic knowledge of IPSec and IKE concepts. See presentation for SHARE Session 8231, z/OS Communications Server IPSec and IP Packet Filtering for reference.

z/OS® V1R12 Communications Server

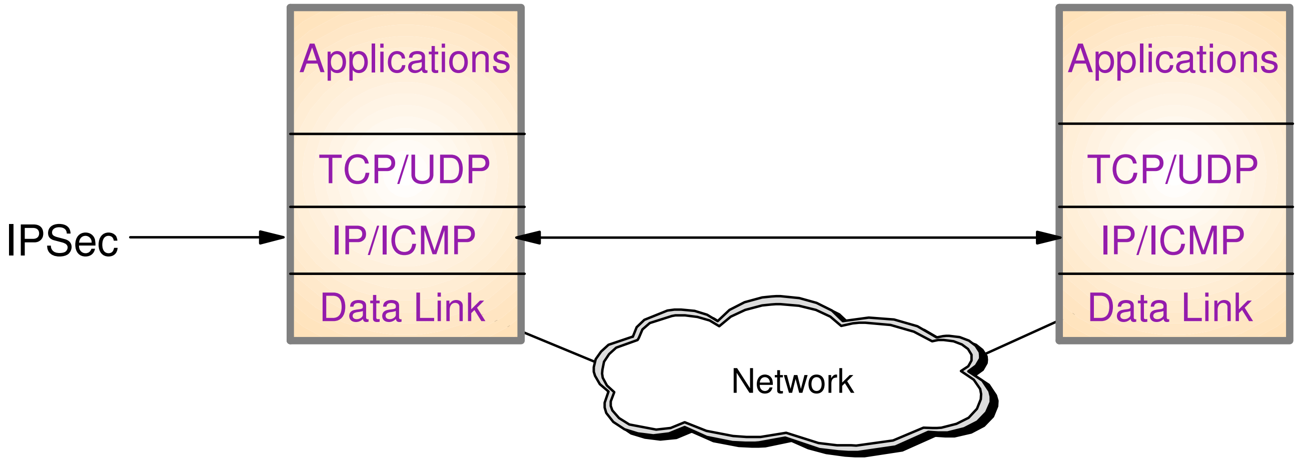
The New IKEv2 Protocol and Related Standards

Background information: IPSec and VPNs

- **IP Security (IPSec) is an industry standard architecture that provides authentication, integrity, and data privacy between any two IP entities**
- **Using IPSec, you can create virtual private networks (VPNs)**
  - A VPN enables an enterprise to extend its private network across a public network, such as the Internet, through a secure tunnel called a *security association (SA)*
- **Using IPSec, you can also create connection-level security associations**



Background Information: IPSec overview



- **IPSec is implemented at the IP layer**
  - Requires no application change
  - Secures traffic between any two IP resources under an “agreement” called a Security Association (SA)
- **IPSec defines 2 protocols to provides authentication, integrity, and data privacy**
  - Authentication Header (AH) - provides authentication / integrity
  - Encapsulating Security Protocol (ESP) - provides data privacy with optional authentication / integrity
- **Management of crypto keys and security associations can be:**
  - Manual
  - Automated via key management protocol (IKE)

Internet Key Exchange (IKE) versions

- **Two versions of the IKE protocol have been defined**
  - IKE version 1 (IKEv1)
    - Defined in the late 1990s
    - Widely implemented and deployed today
  - IKE version 2 (IKEv2)
    - Initially defined in 2005
    - Not as widely implemented or deployed, but growing steadily
- **IPv6 standard implementations are expected to support IKEv2**
  - Both DoD and NIST IPv6 standards require host systems to support IKEv2
  - US Government agencies, and vendors who do business with them, might be expected to use USGv6 compliant systems
    - And might be required to use IKEv2 to establish secure tunnels to US Government agency systems
- **It is anticipated that both IKE versions will be deployed for the foreseeable future because of widespread deployment of IKEv1**
- **Support for concurrent use of both versions is required as enterprises migrate from IKEv1 to IKEv2**

Why a new IKE?.... IKEv2 is better!

- **Goal was to simplify the protocol and correct issues identified with initial implementations and deployments**
  - Better performance characteristics
    - Fewer messages exchanged to establish security associations
    - Rekeying without reauthentication
  - Better operational characteristics
    - Minimal IKEv2 implementations must support newer cryptographic algorithms
    - Minimal IKEv2 implementations must support advanced certificate capabilities
    - Built-in dead-peer detection
    - Built-in NAT traversal
  - Better interoperability
    - Fewer supported combinations of parameters, less permutations
    - Some parameters negotiated in IKEv1 are separately and independently defined by each IKE peer in IKEv2

Primary IPSec standards

- **IPSec is defined and maintained by the Internet Engineering Task Force (IETF), through publications:**
  - RFC 4301: Security Architecture for the Internet Protocol
    - This RFC and its associated RFCs define the means of transporting data securely over an IP network
  - RFC 2409: The Internet Key Exchange (IKE)
    - This RFC and its associated RFCs define the initial version of IKE, now called IKEv1
  - RFC 4306: Internet Key Exchange (IKEv2) Protocol
    - This RFC and its associated RFCs define version 2 of IKE

Standards for IKEv2

- **IKEv2 originally defined by IETF RFC 4306**
  - Combines and Replaces:
    - RFC 2407 (ISAKMP)
    - RFC 2408 (Internet DOI)
    - RFC 2409 (IKE)
- **Clarified by RFC 4718 (IKEv2 Clarifications)**
- **IKEv2 recently reissued (no new requirements over 4306)**
  - RFC 5996 (IKEv2)
- **IKEv2 built on RFC 4301 (Security Architecture for the Internet Protocol)**
  - RFC 4301 supersedes RFC 2401
  - RFC 4301 is significantly different than RFC 2401
  - Much of support for RFC 4301 was added in z/OS V1R10
- **RFCs 4306/5996 includes support for solutions identified prior to IKEv2**
  - RFC 3947 Negotiation of NAT-Traversal in the IKE
  - RFC 3706 Dead peer detection
- **IKEv2 requires support for advanced Public Key Infrastructure (PKI) capabilities**
  - RFC 4945 – Profile for using PKI in the context of IKEv1 and IKEv2
  - RFC 4809 – IPSec certificate management profile requirements
- **RFCs 4306/5996 defers the definition of cryptographic algorithm requirements**
  - RFC 4307 - Cryptographic Algorithms for Use in the Internet Key Exchange Version 2

IKEv1 and IKEv2 protocol comparison overview

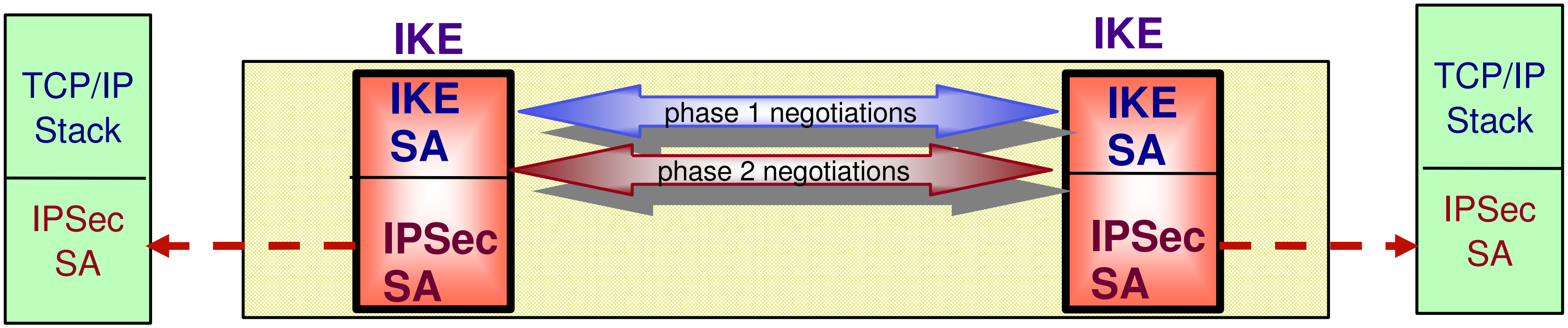
- **Both versions provide message authentication/integrity and data privacy**
  - IKEv2 requires support of some newer, stronger algorithms
- **Both versions support the same configurations**
  - Host-to-Host, Host-to-Gateway, Gateway-to-Gateway, Gateway-to-Host
- **Similar concepts, different terminology**
  - IKE peers establish a security association (SA) to securely carry IKE data between peers, then
  - They use IKE flows to establish one or more SAs to securely carry user data
    - IKEv1: “Phase 1 SA” or “ISAKMP SA”, “Phase 2 SA” or “IPsec SA”
    - IKEv2: “IKE SA”, “CHILD SA”
  - SAs can be refreshed, and have limited life span based on time or bytes
- **Format and sequences of network flows for security association activation and deactivation are different**
  - IKEv2 has preserved much of the IKEv1 header format
    - Coexistence is supported: both versions can run over the same UDP port HOWEVER
    - Interoperability is not supported: IKEv2 requests need IKEv2 responses
  - IKEv2 requires fewer network flows in most cases
    - IKEv1 has IKE negotiation modes (Main, Aggressive, Quick)
    - IKEv2 has “initial exchanges” and subsequent exchanges

IKEv2 topics

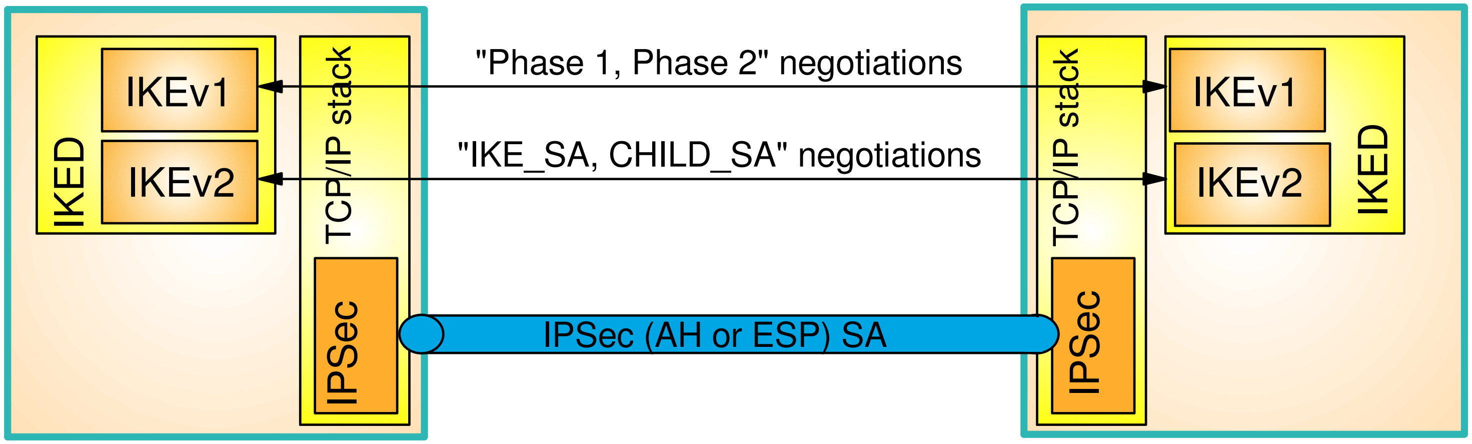
- **Terminology changes**
- **Message exchange changes**
- **Things no longer negotiated**
- **Things negotiated differently**
- **New Public Key Infrastructure (PKI) related requirements**

IPSec/IKE – Two phases of IKE

- Phase 1
  - Negotiates an IKE SA
  - Generates cryptographic keys that will be used to protect
    - Phase 2 negotiations
    - Informational exchanges
  - Authenticates the identity of the parties involved
- Phase 2
  - Negotiates an IPSec SA (a.k.a. 'CHILD SA') with a remote security endpoint
  - Generates cryptographic keys that are used to protect data
  - Created under the protection of an IKE SA



Phases – Terminology changes



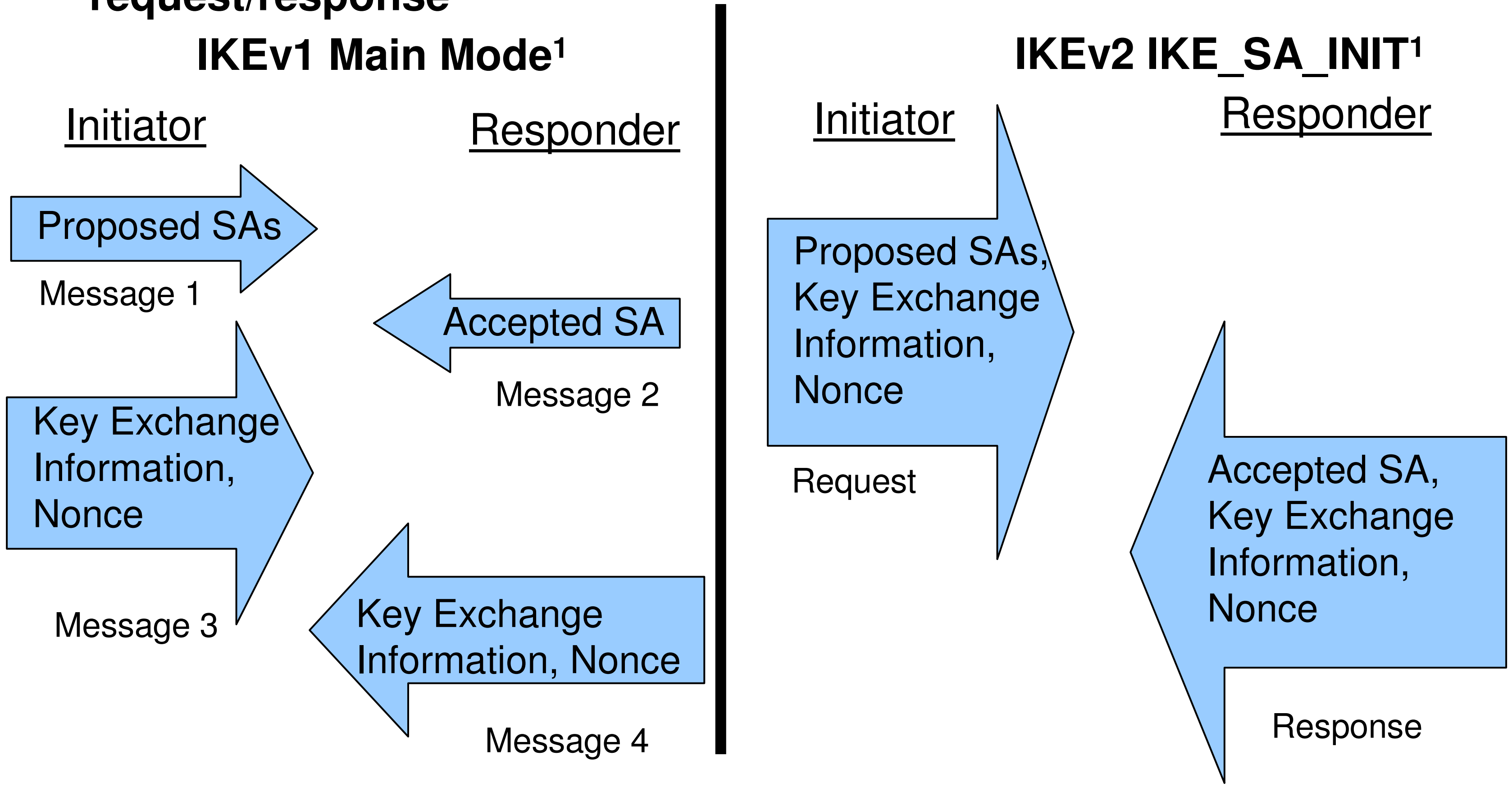
- **IKEv2 eliminates the terminology Phase 1 and Phase 2**
- **Conceptually still a two phase protocol, but optimized so the phases are less defined**
  - Phase 1 SAs become IKE_SAs
  - Phase 2 SAs become CHILD_SAs
- **First CHILD_SAs are created automatically when an IKE_SA is first created**
- **Additional CHILD_SAs are added as result of a new message called CREATE_CHILD_SA**

Negotiation Modes of Phase 1 SAs and IKE_SAs

- **IKEv1 (Phase 1)**
  - Main mode (better security, worse performance)
    - uses six messages
    - a.k.a. identity-protected mode (peer identities are encrypted)
  - Aggressive mode (less security, better performance)
    - uses three messages
    - peer identities are **not** encrypted
    - faster than main mode but potentially less secure
- **IKEv2 (IKE_SA)**
  - One exchange type
    - uses four messages
    - peer identities **are** encrypted
    - better performance than main mode
    - better protection than aggressive mode

Negotiating IKE SAs: IKE_SA_INIT message processing

- **Combines the first 4 messages of main mode into a single request/response**



IKEv1 Main Mode[1]

Initiator

Responder

Proposed SAs

Message 1

Accepted SA

Message 2

Key Exchange Information, Nonce

Message 3

Key Exchange Information, Nonce

Message 4

IKEv2 IKE_SA_INIT[1]

Initiator

Responder

Proposed SAs, Key Exchange Information, Nonce

Request

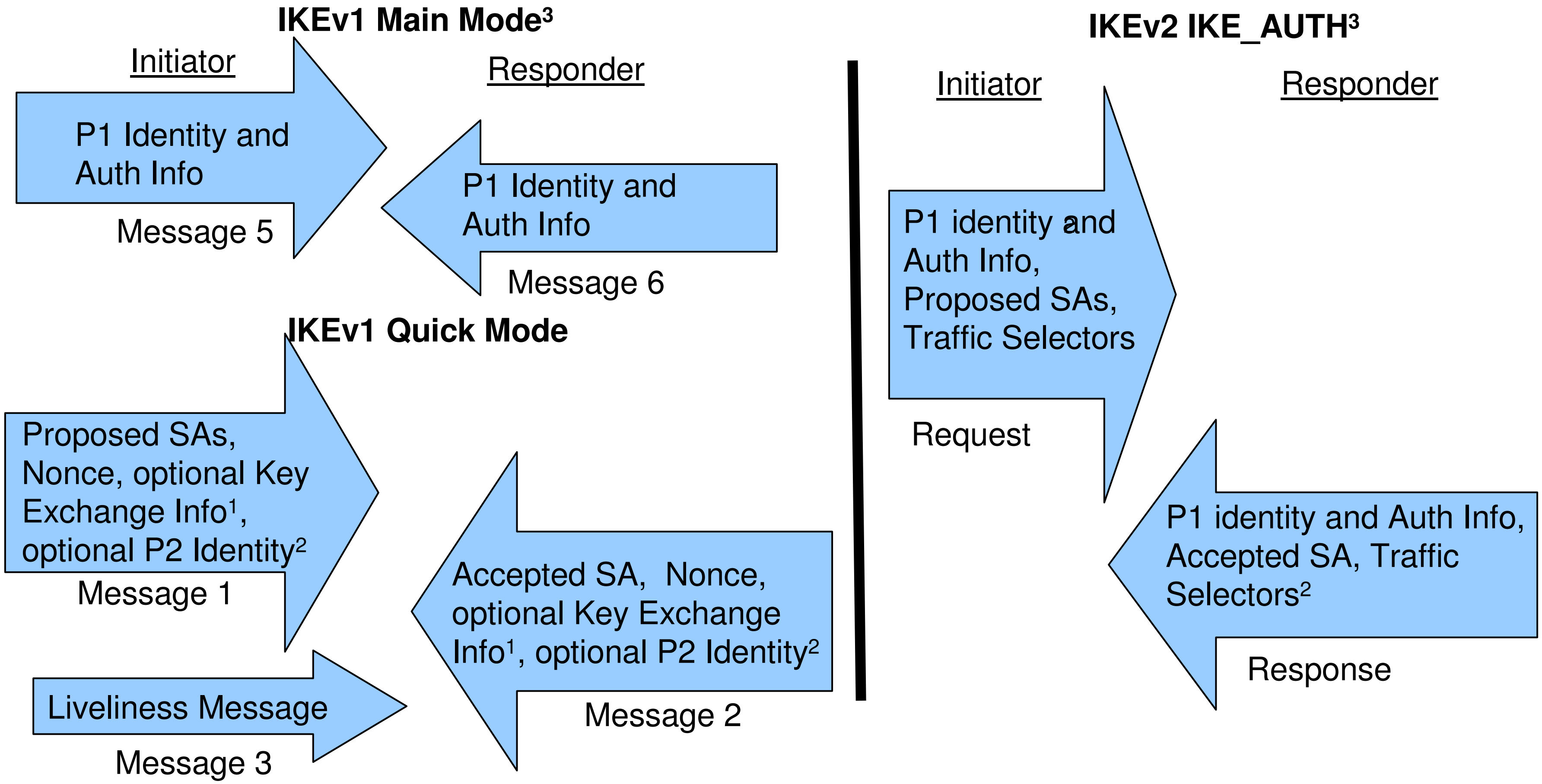
Accepted SA, Key Exchange Information, Nonce

Response

[1] Certificate related payloads and optional notifications not shown

Negotiating IKE and initial child SAs: IKE_AUTH processing

- **Combines messages 5 and 6 of main mode and the 1st quick mode exchange into a single request/response exchange**



IKEv1 Main Mode[3]

Initiator

Responder

P1 Identity and Auth Info

Message 5

P1 Identity and Auth Info

Message 6

IKEv1 Quick Mode

Proposed SAs, Nonce, optional Key Exchange Info[1], optional P2 Identity[2]

Message 1

Accepted SA, Nonce, optional Key Exchange Info[1], optional P2 Identity[2]

Message 2

Liveliness Message

Message 3

IKEv2 IKE_AUTH[3]

Initiator

Responder

P1 identity and Auth Info, Proposed SAs, Traffic Selectors

Request

P1 identity and Auth Info, Accepted SA, Traffic Selectors[2]

Response

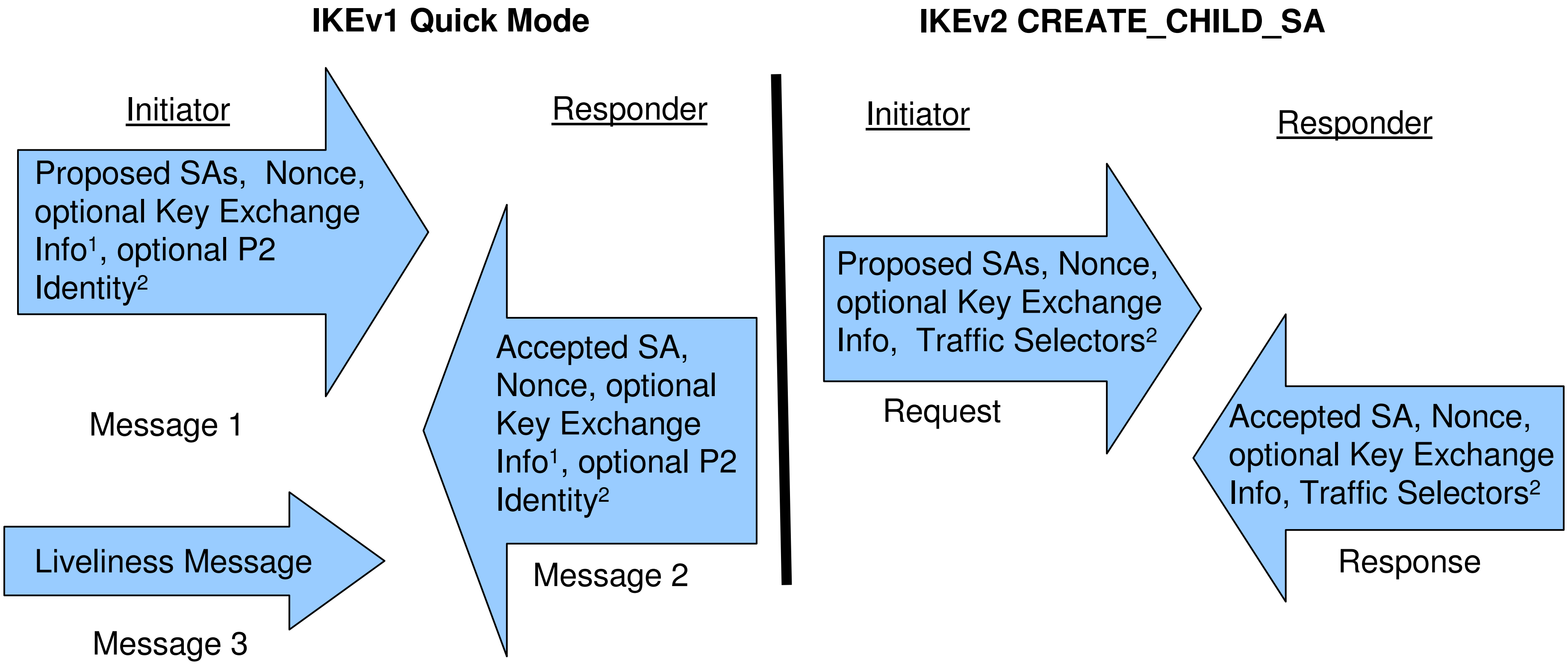
[1] The IKE_AUTH exchange does not support an optional Diffie-Hellman exchange when creating the first CHILD_SA.

[2] The IKE_AUTH exchange requires traffic selectors. Traffic selectors serve the same function as the optional identities exchanged in a quick mode exchange.

[3] Certificate related payloads and optional notifications not shown

Negotiating Child SAs: CREATE_CHILD_SA processing

- **Combines subsequent quick mode exchanges into a single request/response exchange**

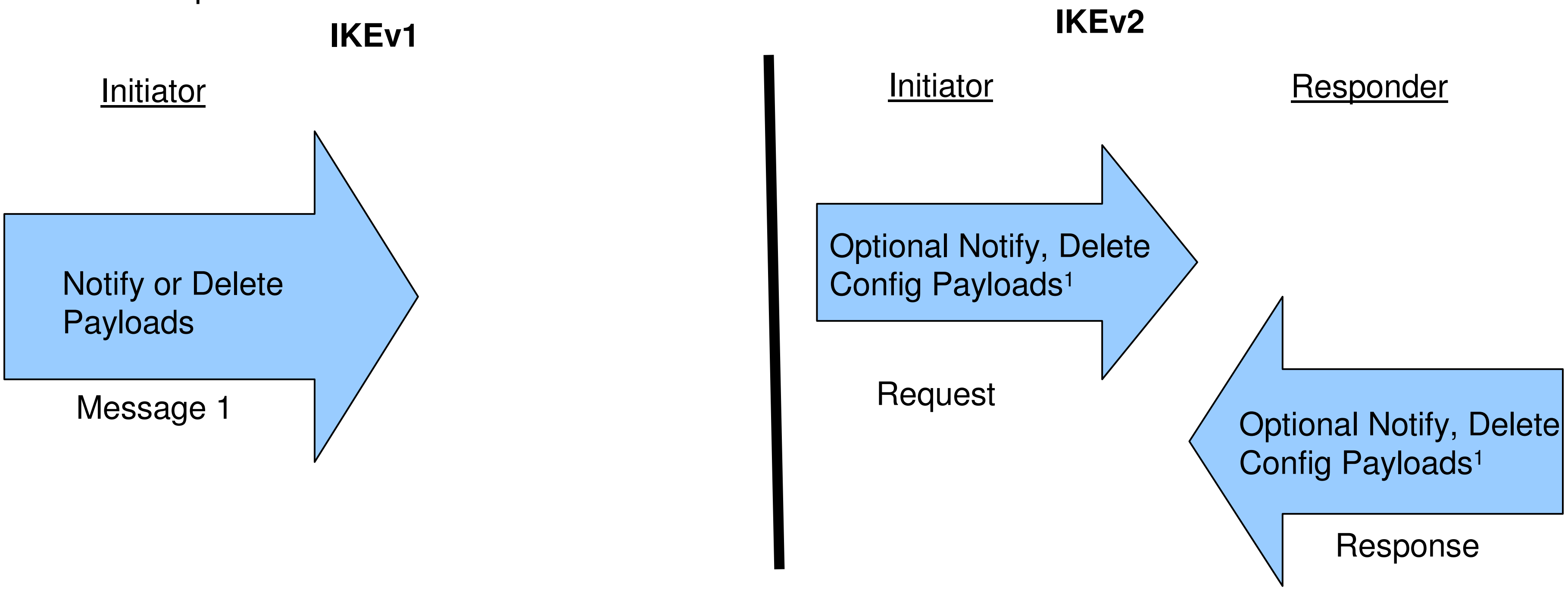

IKEv1 Quick Mode
IKEv2 CREATE_CHILD_SA
Initiator
Responder
Proposed SAs, Nonce, optional Key Exchange Info¹, optional P2 Identity²
Message 1
Accepted SA, Nonce, optional Key Exchange Info¹, optional P2 Identity²
Message 2
Liveliness Message
Message 3
Initiator
Responder
Proposed SAs, Nonce, optional Key Exchange Info, Traffic Selectors²
Request
Accepted SA, Nonce, optional Key Exchange Info, Traffic Selectors²
Response

[1] The CREATE_CHILD_SA exchange does support an optional Diffie-Hellman exchange when creating a CHILD_SA.

[2] The CREATE_CHILD_SA exchange requires traffic selectors (unless it is being done to rekey an SA). Traffic selectors serve the same function as the optional identities exchanged in a quick mode exchange.

Informational Exchange processing

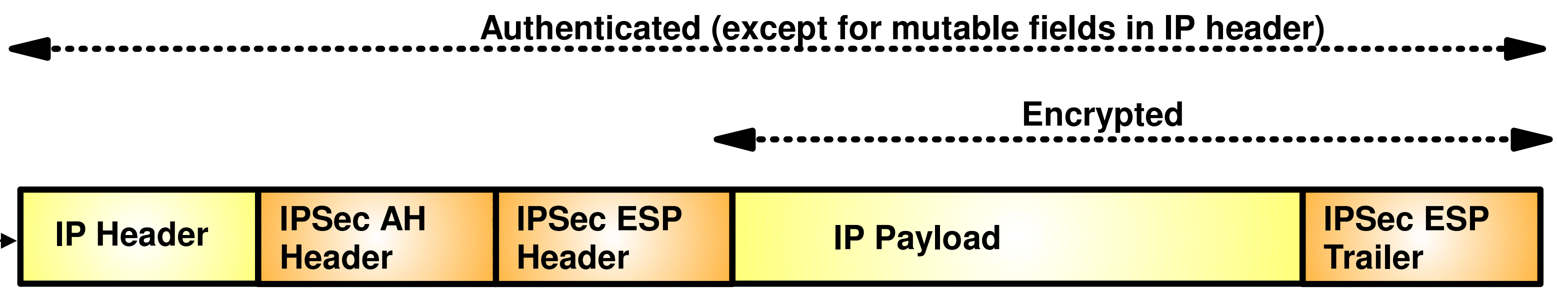
- **All informational exchanges are defined as a 2 message request/response flow**
  - Only the initiator of an exchange should retransmit a message based on the lack of a response


IKEv1
Initiator
Notify or Delete Payloads
Message 1
IKEv2
Initiator
Responder
Optional Notify, Delete Config Payloads[1]
Request
Optional Notify, Delete Config Payloads[1]
Response

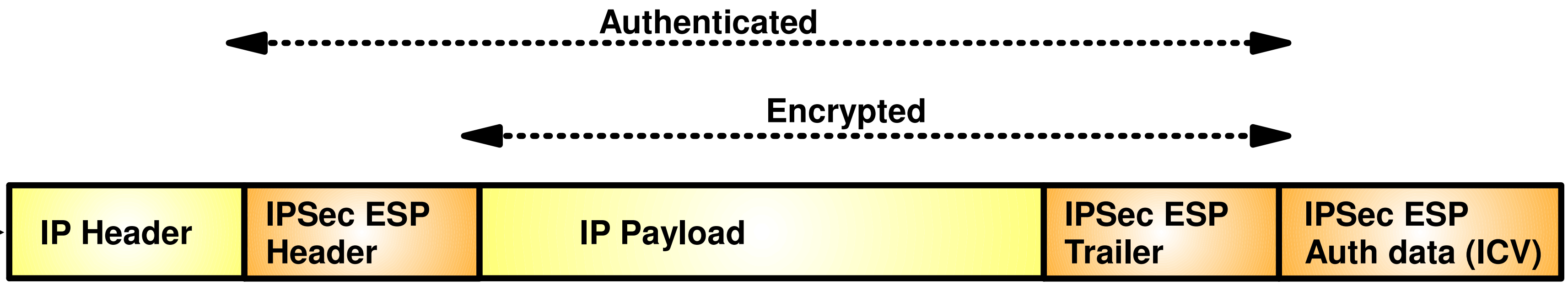
[1] Both the response and request could contain 0 payloads. A request with no payloads would typically be sent as inquiry or to determine whether a peer is still alive (dead peer detection)

Things no longer negotiated

- **SA bundles**
  - Can't negotiate combined AH and ESP with IKEv2

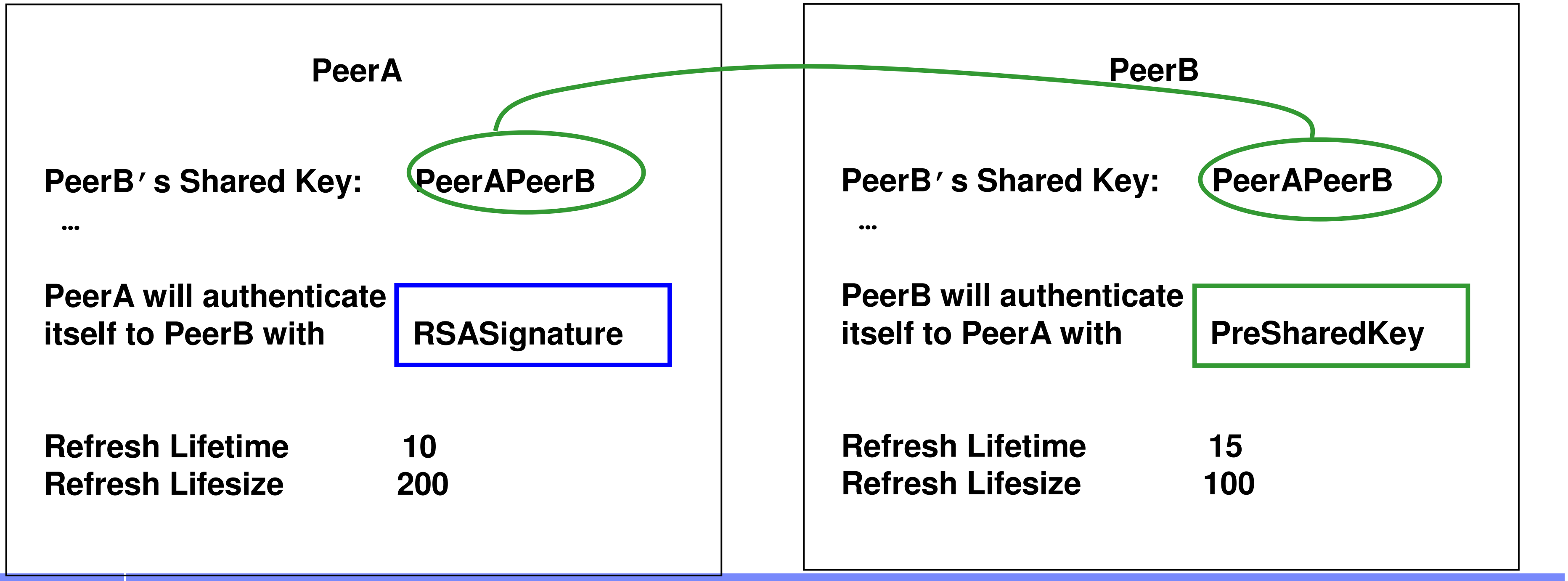


  - For combined encryption and authentication, ESP with authentication should be negotiated



Things no longer negotiated (continued)

- **How to authenticate**
  - Each security endpoint decides how it wants to authenticate to its peer
    - Digital signature (RSA or ECDSA)
    - Pre-shared key
      - ***If one IKEv2 peer uses PreSharedKey, both must have the same key defined!***
- **Lifetimes/Lifesizes**
  - Each security endpoint enforces it own local lifetime/lifesize policy
    - Refreshes/Expires an SA whenever it wants to



PeerA

PeerB′s Shared Key: PeerAPeerB

…

PeerA will authenticate itself to PeerB with RSASignature

Refresh Lifetime	**10**
Refresh Lifesize	**200**

PeerB

PeerB′s Shared Key: PeerAPeerB

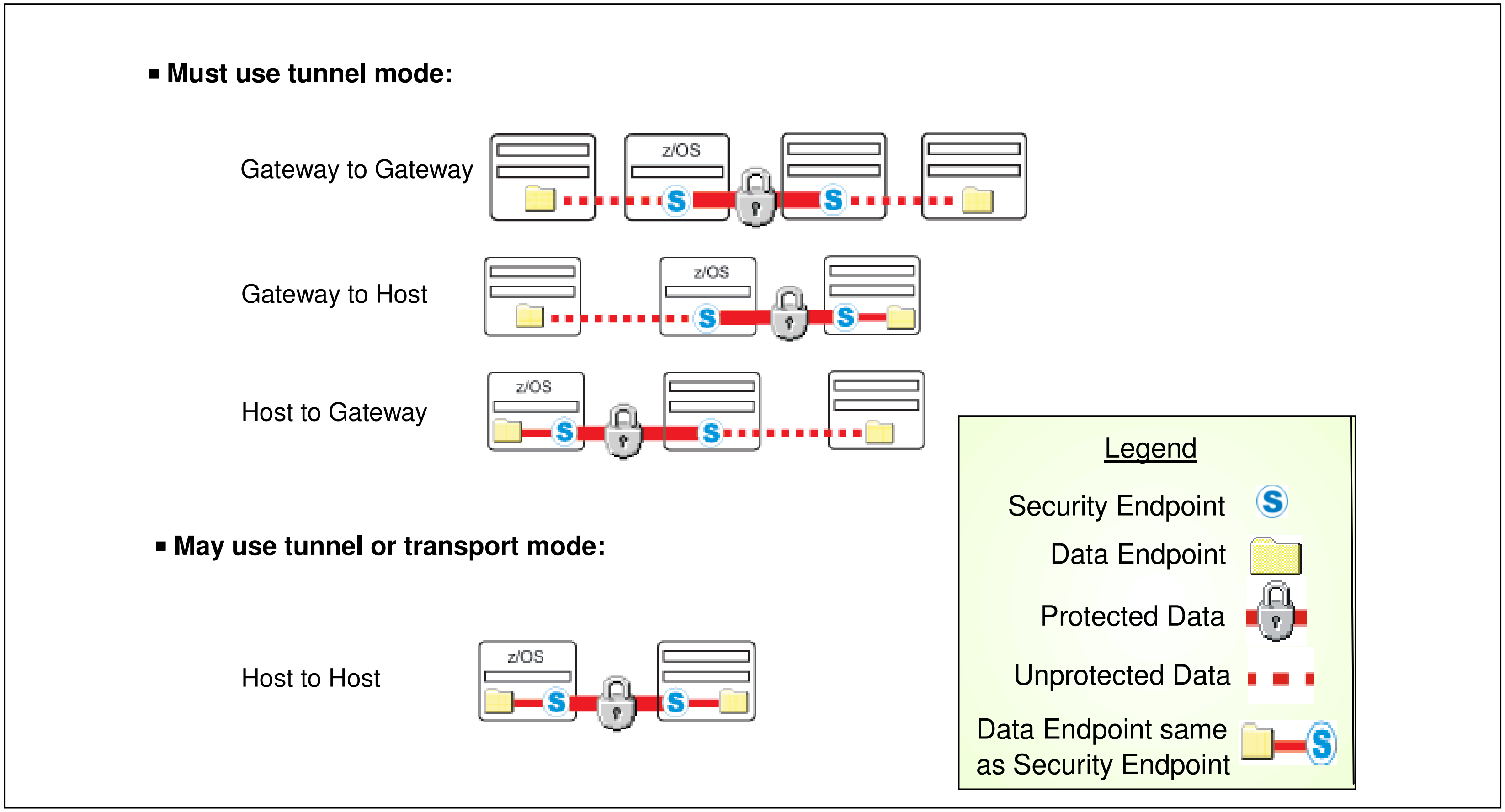
…

PeerB will authenticate itself to PeerA with PreSharedKey

Refresh Lifetime	**15**
Refresh Lifesize	**100**

Things negotiated differently

- **Encapsulation mode**
  - All CHILD_SAs are assumed to be TUNNEL mode, unless the initiator sends a USE_TRANSPORT_MODE notify
    - Responder can still pick tunnel mode



▪ Must use tunnel mode:

Gateway to Gateway

Gateway to Host

Host to Gateway

▪ May use tunnel or transport mode:

Host to Host

Legend

Security Endpoint

Data Endpoint

Protected Data

Unprotected Data

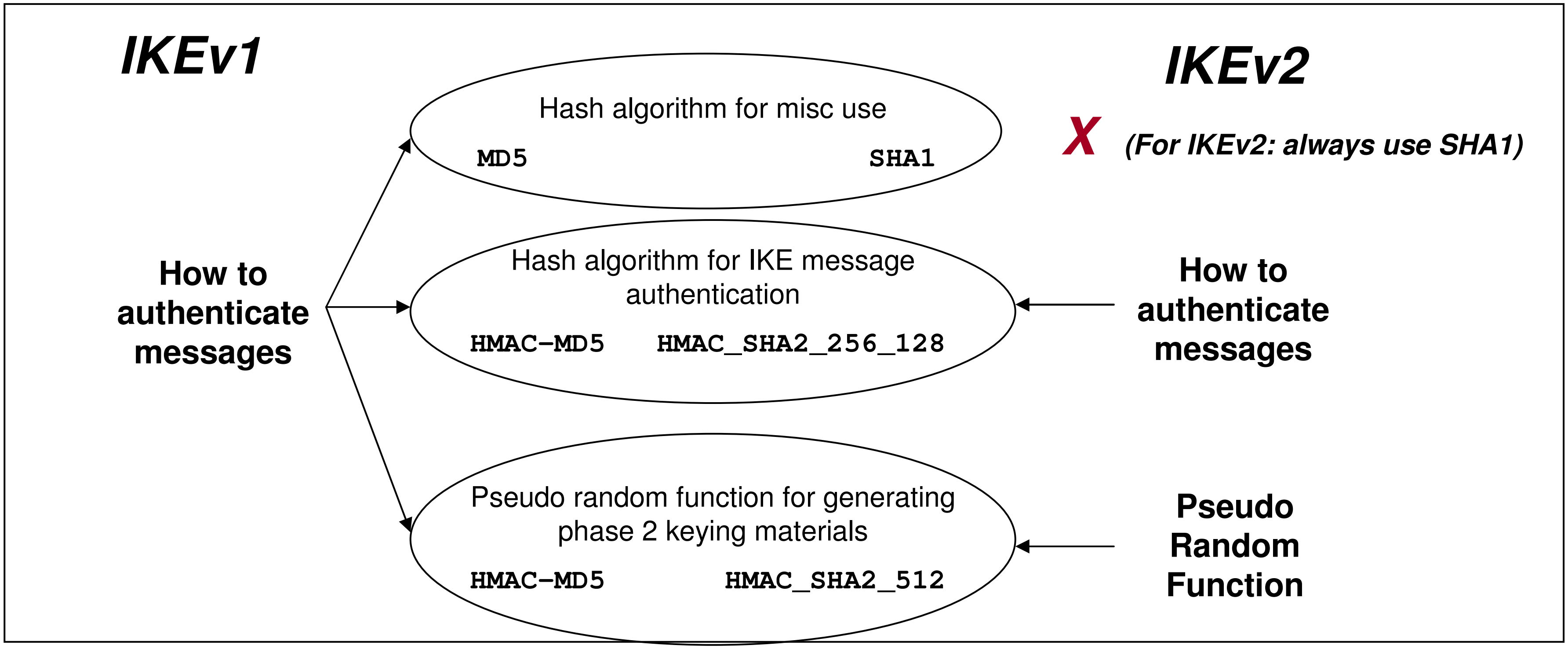
Data Endpoint same as Security Endpoint

Things negotiated differently (continued)

- **Peer-to-peer authentication during security association refresh**
  - In IKEv1, peer-to-peer authentication is always required for Phase 1 SA refresh
  - In IKEv2, peer-to-peer authentication is *optional* for IKE SA refresh
    - Reduces cost of expensive asymmetric cryptography required for re-authentication

Things negotiated differently (continued)

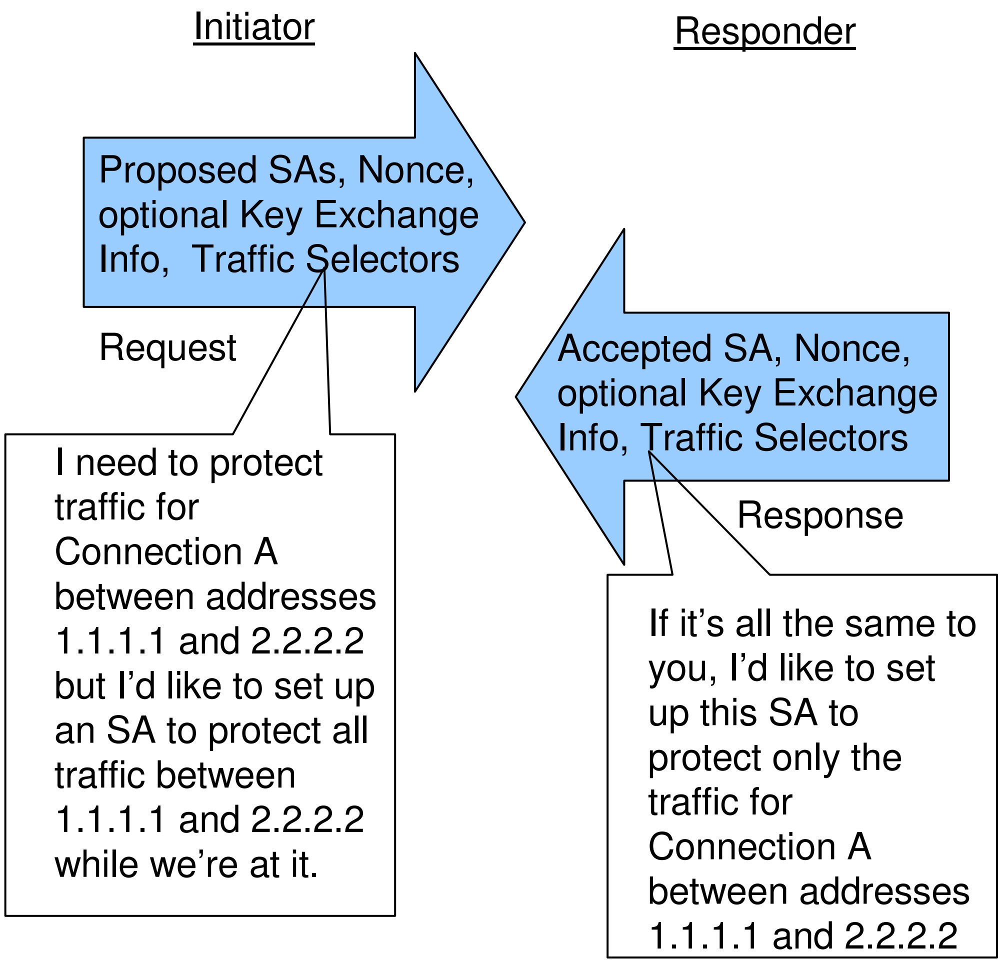
- **Hashing algorithms now separately specified**
  - In IKEv1, a single negotiated hash algorithm is used for three purposes
    - Message authentication, Pseudo random function (PRF), miscellaneous purposes
  - In IKEv2, hash algorithms are selected differently
    - Message authentication and PRF separately specified
    - SHA1 always used for miscellaneous purposes


IKEv1
IKEv2
Hash algorithm for misc use
MD5
SHA1
X
(For IKEv2: always use SHA1)
How to authenticate messages
Hash algorithm for IKE message authentication
HMAC-MD5
HMAC_SHA2_256_128
How to authenticate messages
Pseudo random function for generating phase 2 keying materials
HMAC-MD5
HMAC_SHA2_512
Pseudo Random Function

Things Negotiated Differently (continued)

- **Negotiating traffic selectors**
  - New Traffic Selector (TS) payloads allow endpoints to communicate some information (IP address, ports, and protocols) from their SPD to their peers.
    - TS payloads specify the selection criteria for packets that will be protected using the newly set up SA.
    - TS payloads replace Phase 2 identity used in IKEv1
  - IKEv2 allows the responder to choose a subset of the traffic proposed by the initiator
    - The responder can return one or more TS payloads that are more "narrow"
      - Allows successful SA negotiation when the configurations of the two endpoints are not perfectly matched.

IKEv2 CREATE_CHILD_SA



Initiator

Responder

Proposed SAs, Nonce, optional Key Exchange Info, Traffic Selectors

Request

I need to protect traffic for Connection A between addresses 1.1.1.1 and 2.2.2.2 but I'd like to set up an SA to protect all traffic between 1.1.1.1 and 2.2.2.2 while we're at it.

Accepted SA, Nonce, optional Key Exchange Info, Traffic Selectors

Response

If it's all the same to you, I'd like to set up this SA to protect only the traffic for Connection A between addresses 1.1.1.1 and 2.2.2.2

PKI Related Changes – Required for IKEv2, Applicable to IKEv1

- **Certificate Trust Chain support**
  - Sending/receiving up to 4 certificates in a certificate hierarchy
    - End entity (host) certificate and up to 3 CA certs
- **Certificate revocation checking (RFCs 4945 and 4809)**
  - Possible options include Certificate Revocation List (CRL) checking via http server
- **Remote identity checking**
  - New requirements in IP-based ID types
  - Ability to enable/disable checks
- **Additional certificate content requirements (RFCs 4945 and 4809)**
  - CRLDistributionPoints and AuthorityInfoAccess extensions should be in IPSec certificates
  - Allow certificates with *nonRepudiation* bit in the *KeyUsage* extension to be used for creating signatures
  - If a certificate has an *ExtendedKeyUsage* extension it should contain either the *keyPurposeID id-kp-ipsecIKE* or *anyExtendedKeyUsage*

PKI Related Changes
Required for IKEv2, Not applicable to IKEv1

- **Certificate Request Payload Changes**
  - The Certification Authority value is a concatenated list of SHA-1 hashes
  - Each is a SHA-1 hash of the Subject Public Key Info element from each trust anchor certificate.
  - The hashes are concatenated and included with no other formatting.
  - A CERTREQ should be seen as a suggestion for a certificate to select, not a mandate
- **Certificate Payload Changes**
  - New certificate types
    - Must support
      - Hash and URL of X.509 certificate
      - Hash and URL of X.509 bundle

z/OS® V1R12 Communications Server

IKEv2 on z/OS

z/OS Communications Server IKEv2 support overview

- **In V1R12, IKEv2 support is added to the existing z/OS IPSec function to support activation, rekeying, and deactivation of IPSec security associations using the IKEv2 protocol, in addition to the current IKEv1 support**
  - One IPSec policy file can contain both IKEv1 and IKEv2 policies
  - IKE daemon supports both IKEv1 and IKEv2 tunnels simultaneously
  - Each TCP/IP stack can support tunnels activated by IKEv1 and IKEv2, and supports a wider variety of cryptographic algorithms.
  - The ipsec command can display, activate, refresh, and deactivate both IKEv1 and IKEv2 tunnels
- **The NSS daemon provides ALL certificate services for IKEv2 including the advanced certificate services that are required by IKEv2**
- **In V1R12, advanced certificate functions required for IKEv2 that were previously optional for IKEv1 are now available for IKEv1 through NSS**

z/OS CommServer V1R12 successfully completed USGv6 interoperability testing including the IPSec, IKE and ESP test suites

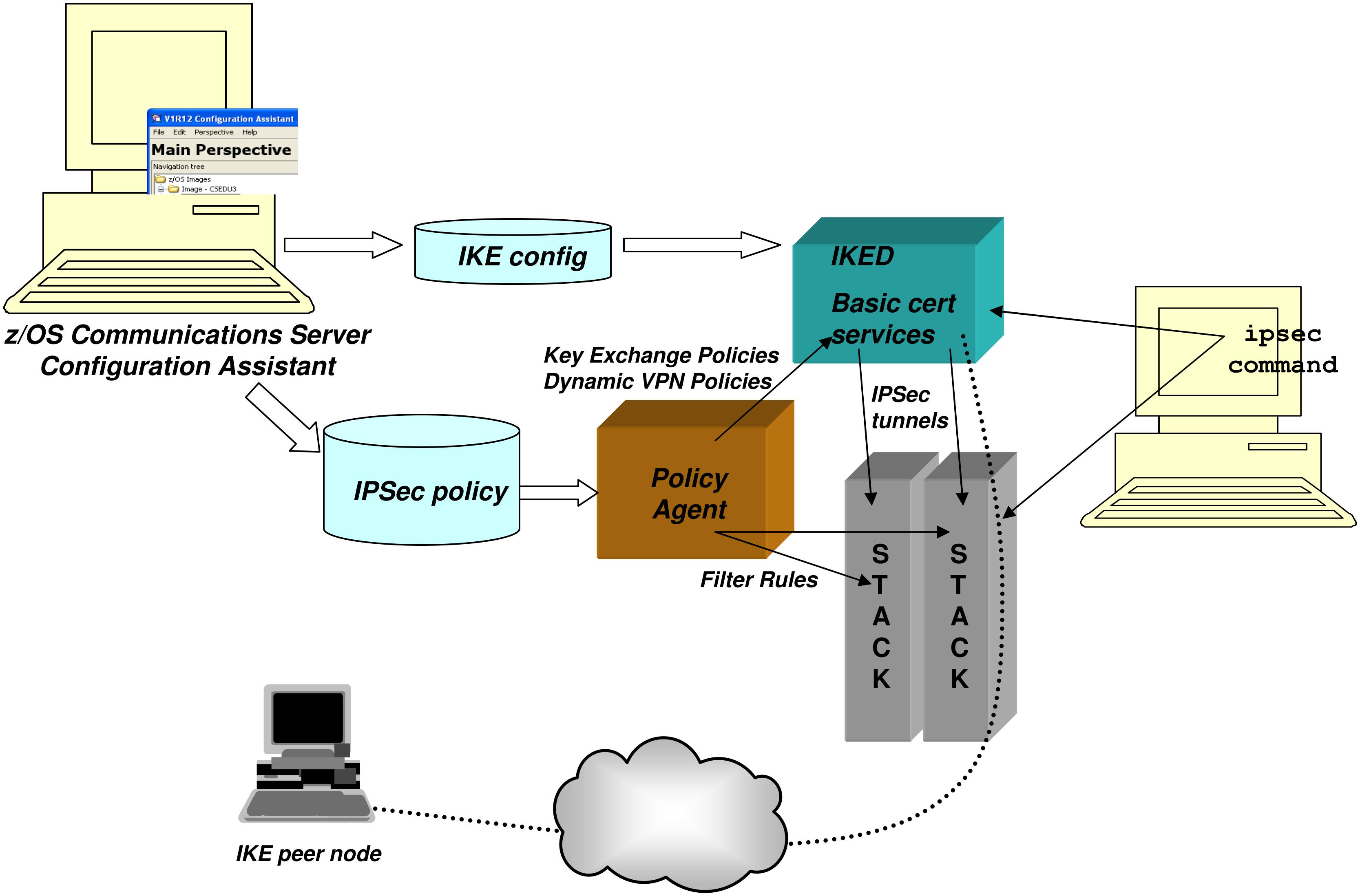
http://www.iol.unh.edu/services/testing/ipv6/usgv6tested.php

z/OS IKEv2 restrictions

- **In z/OS V1R12**
  - z/OS IKEv2 support does not include Network Address Translation (NAT)
    - Continue to use IKEv1 if there is a NAT between the security endpoints
  - z/OS IKEv2 support does not include support for Sysplex-Wide Security Associations (SWSA)
    - Continue to use IKEv1 for SWSA in your sysplex

- **Coming in z/OS V1R13**
  - Support for both NAT and SWSA are planned

z/OS processes in IPSec for IKEv1 using basic certificate services



z/OS Communications Server Configuration Assistant

IKE config

IPSec policy

IKED

Basic cert services

Key Exchange Policies
Dynamic VPN Policies

Policy Agent

IPSec tunnels

Filter Rules

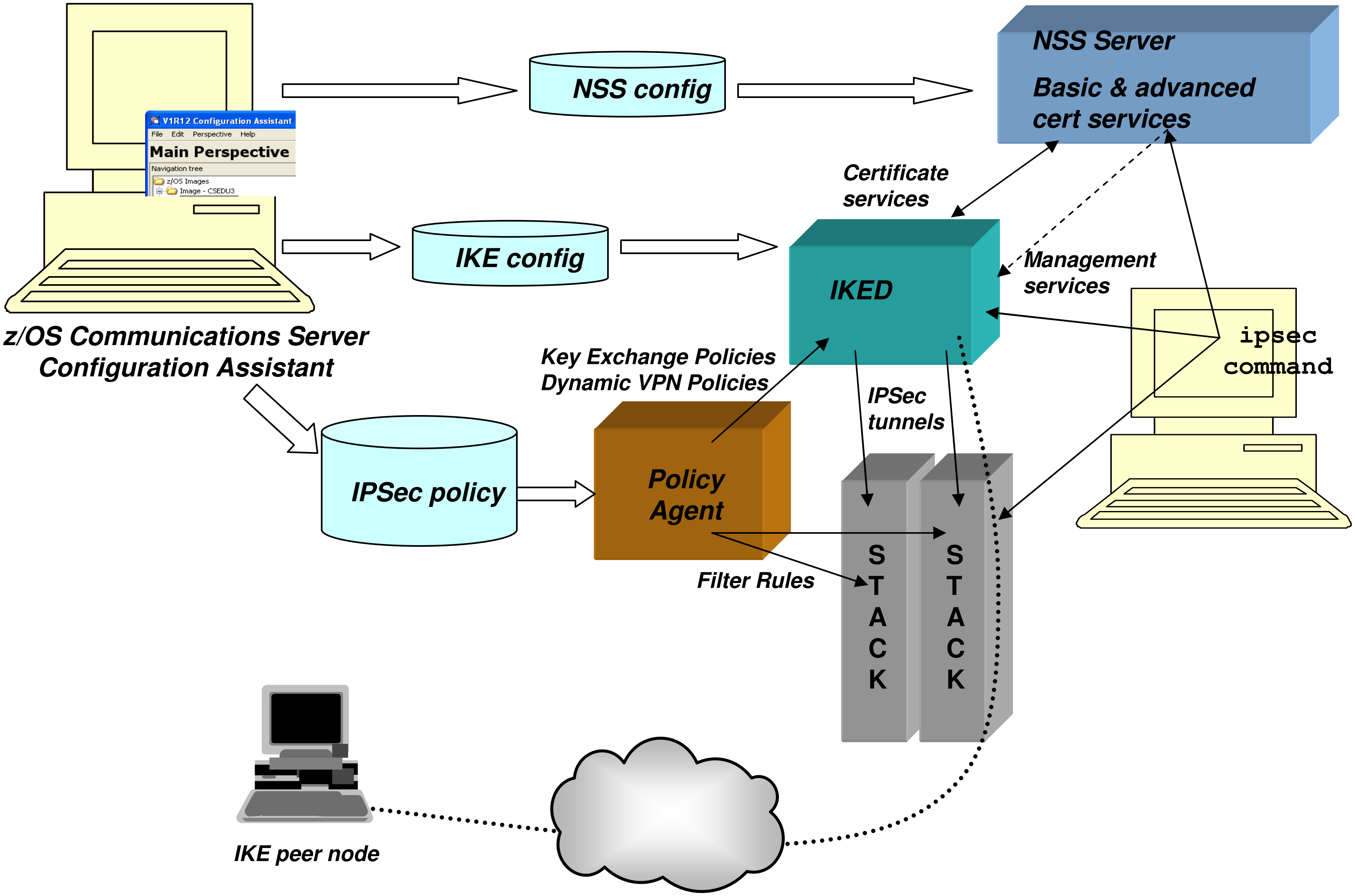
STACK

STACK

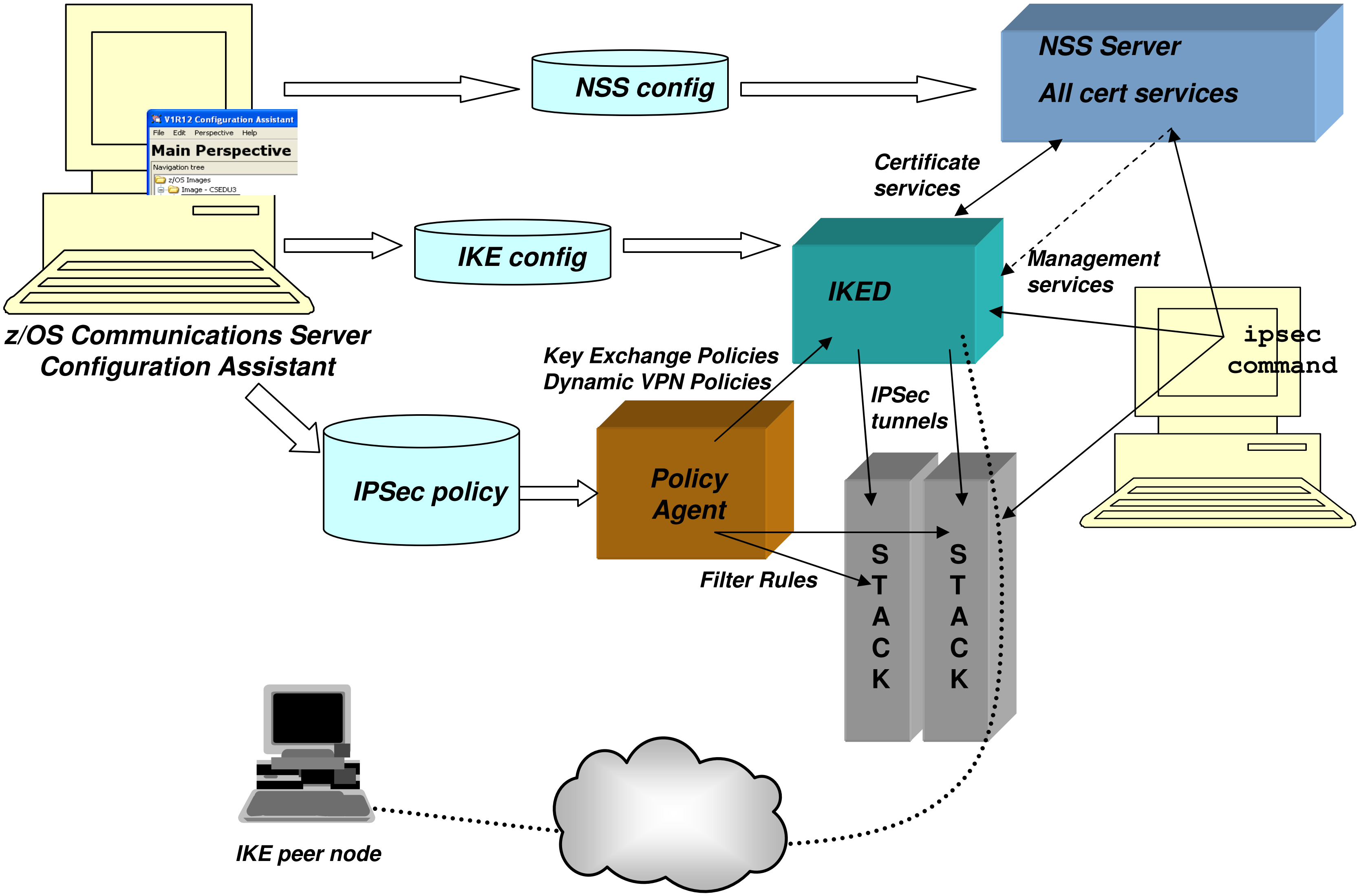
ipsec command

IKE peer node

z/OS processes in IPSec for IKEv1 using advanced certificate services (V1R12)



z/OS processes in IPSec for IKEv2

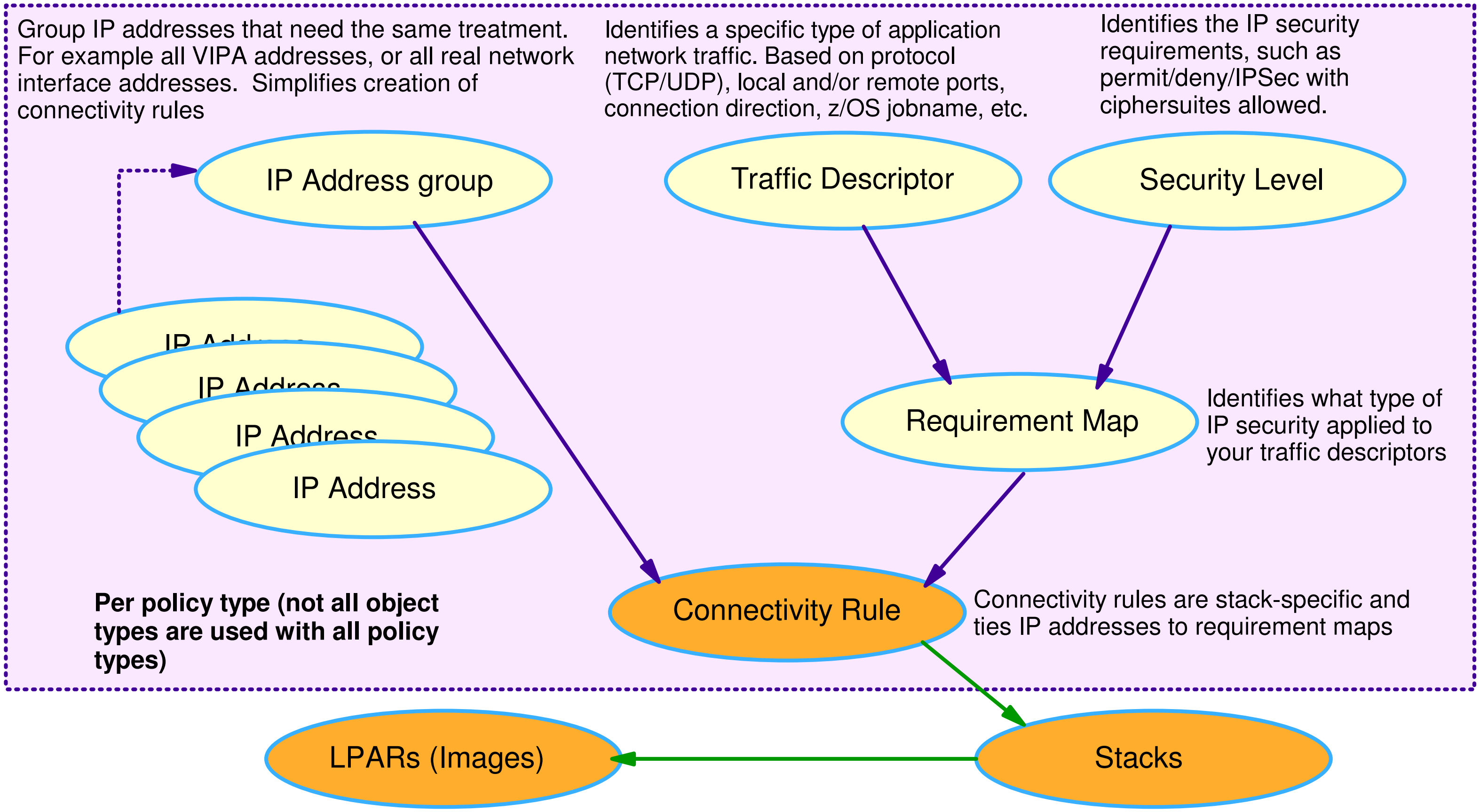


Externals overview – seamless support

- **Documentation and externals are made common for IKEv1 and IKEv2 where possible**
- **IKEv1 and IKEv2 terminology differences are minimized**
  - e.g. “Phase 1 / phase 2” terms are used for both IKEv1 and IKEv2
- **Configuration Assistant visibility to IKEv2-unique protocol definitions is minimized and configuration is based on good defaults**
  - Changes to these defaults are made through “advanced” panels in Configuration Assistant
- **Configuration to govern IKE negotiations default to using the same parameters for IKEv1 and IKEv2 when possible**
- **Ipsec commands display both IKEv1 and IKEv2 information in same output where appropriate**

Before we talk about IKEv2 policy configuration…

- **The Configuration Assistant for z/OS Communication Server is recommended for configuring IPSec**
  - Focus on high level concept vs. low level file syntax



Group IP addresses that need the same treatment. For example all VIPA addresses, or all real network interface addresses. Simplifies creation of connectivity rules

Identifies a specific type of application network traffic. Based on protocol (TCP/UDP), local and/or remote ports, connection direction, z/OS jobname, etc.

Identifies the IP security requirements, such as permit/deny/IPSec with ciphersuites allowed.

Identifies what type of IP security applied to your traffic descriptors

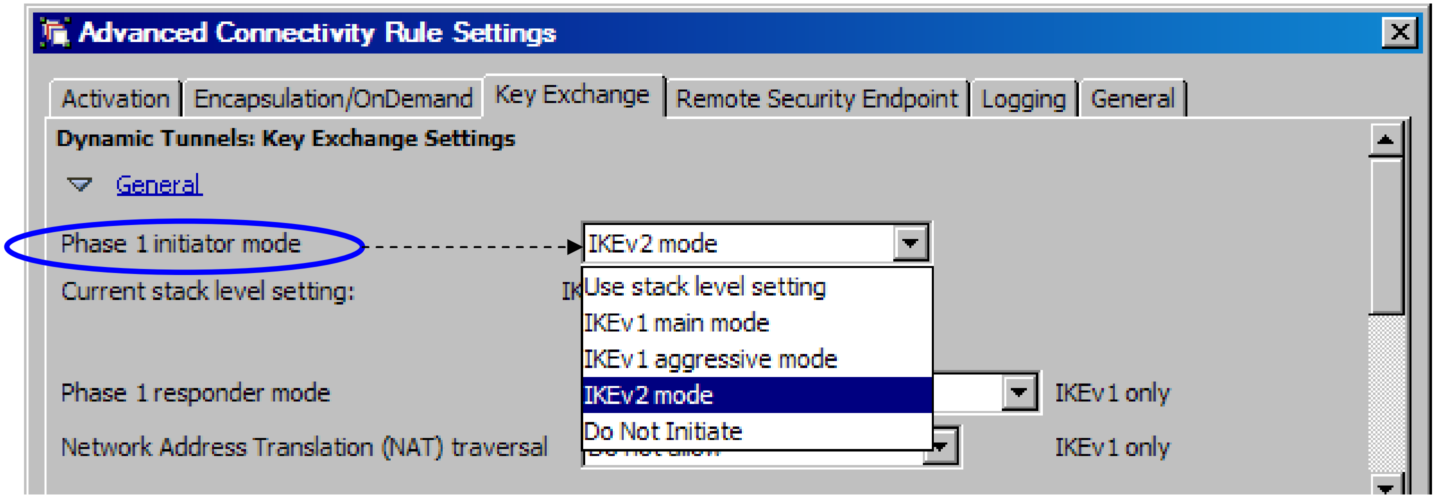
Connectivity rules are stack-specific and ties IP addresses to requirement maps

Per policy type (not all object types are used with all policy types)

- **In order to reduce configuration requirements, the Configuration Assistant generates good defaults for IKE based on the security levels associated with the connectivity rule and configuration selected**
  - IKE parameter defaults can be changed through advanced connectivity rule settings panel

How do you configure policy to start using IKEv2?

- **At z/OS V1R12, IKED will automatically respond to either IKEv1 or IKEv2 activation requests when acting as the *responder***
  - In most cases, no additional configuration required to support IKEv2 as a responder!
    - See migration considerations for exceptions
- **The IKE policy provides configuration to specify whether IKED will use IKEv1 or IKEv2 when acting as the *initiator* of a security association**
  - Initiator mode can be set at the “stack” level
    - IKEv1 Main Mode is default
  - Stack default can be overridden at the “connectivity rule” level



Dynamic Tunnels: Key Exchange Settings

General

Phase 1 initiator mode: IKEv2 mode

Current stack level setting: IK

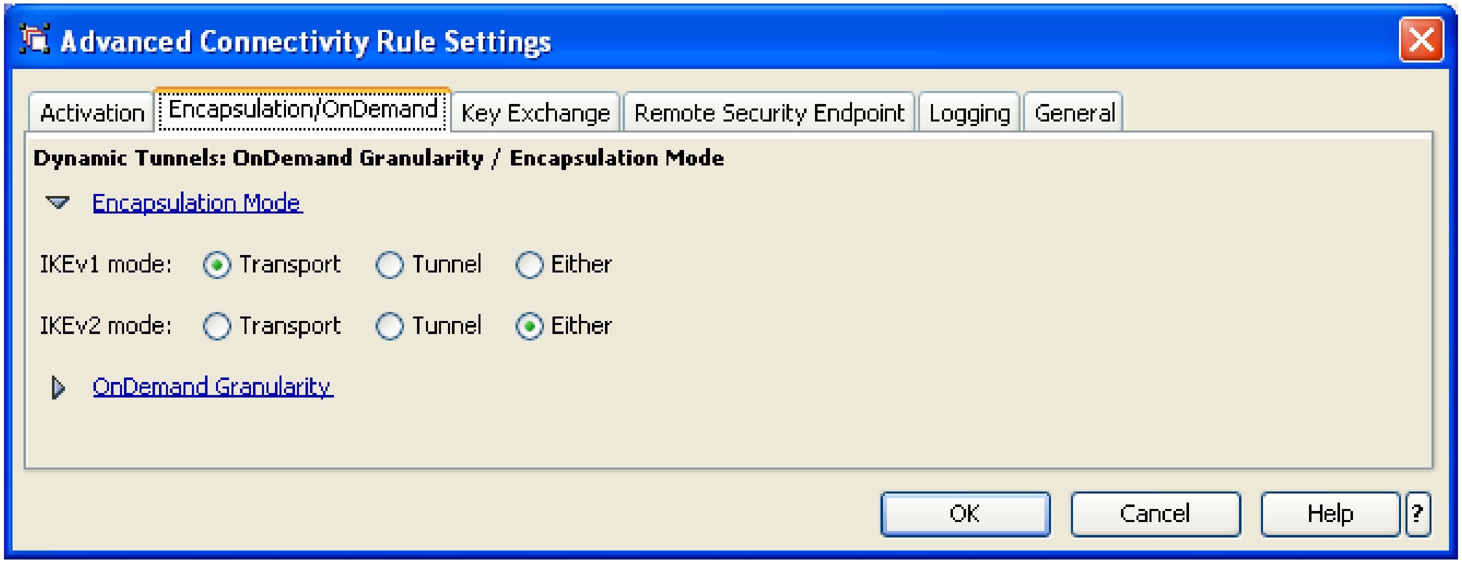
Phase 1 responder mode — IKEv1 only

Network Address Translation (NAT) traversal: Do not allow — IKEv1 only

Tunnel or Transport mode?

- **The Configuration Assistant will pick an encapsulation mode based on the topology selection for connectivity rule.**
  - This selection can be overidden on the advanced connectivity rule settings panel.
- **For IKEv1, encapsulation mode (tunnel or transport) is a negotiated attribute of the SA**
  - Tunnel, Transport, or Either
    - Either means two data offers are sent to IKE peer
  - Local value must match peer's value and be correct for the actual topology
- **For IKEv2, encapsulation mode is negotiated based on topology and user preference**
  - Tunnel, Transport or Either
    - **Either** means prefer transport mode for host-to-host SAs, else use tunnel mode

Default settings for Host-to-Host topology



Advanced Connectivity Rule Settings

Dynamic Tunnels: OnDemand Granularity / Encapsulation Mode

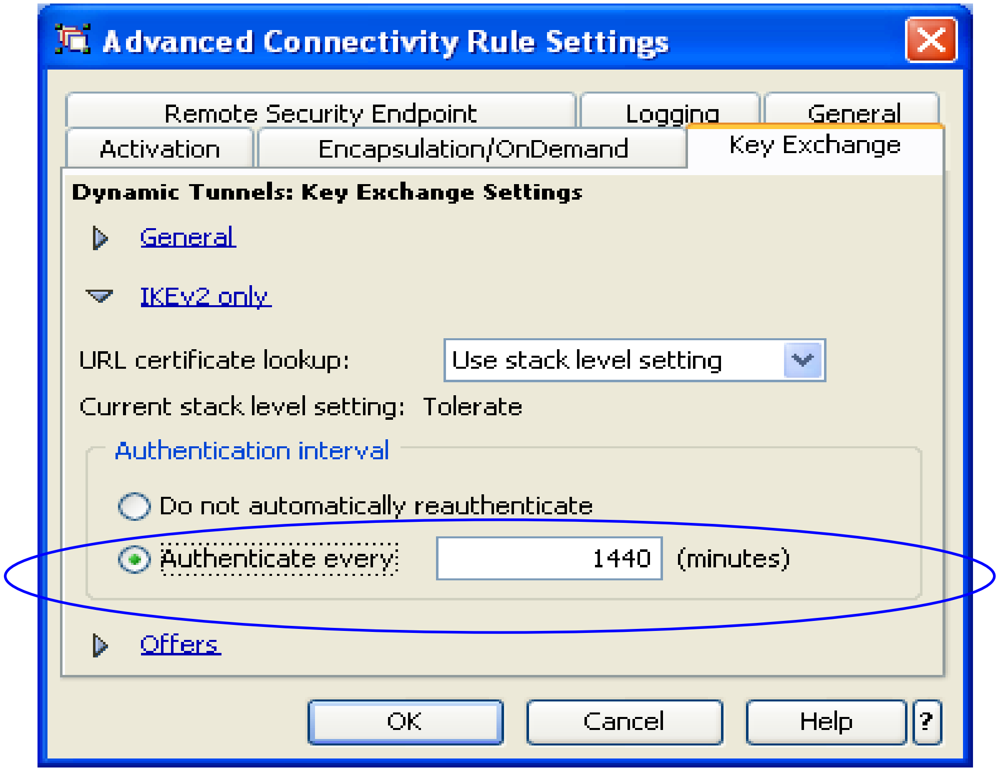
Encapsulation Mode

IKEv1 mode: ● Transport ○ Tunnel ○ Either

IKEv2 mode: ○ Transport ○ Tunnel ● Either

IKEv2 rekey without reauthentication

- **For IKEv1, refreshing a Phase 1 SA includes rekeying the SA and reauthenticating the peer.**
  - Reauthenticating is expensive!
- **IKEv2 supports rekeying a Phase 1 SA *without* reauthenticating the peer**
  - For IKEv2, lifetime or lifesize expiration will cause a rekey only
  - An authentication interval can be specified in Configuration Assistant
    - The authentication interval setting will periodically cause a rekey and authentication for the Phase 1 SA and termination and renegotiation of all the underlying Phase 2 SAs
  - “ipsec –k refresh” will force a rekey and reauthentication



Dynamic Tunnels: Key Exchange Settings

URL certificate lookup: Use stack level setting

Current stack level setting: Tolerate

Authentication interval

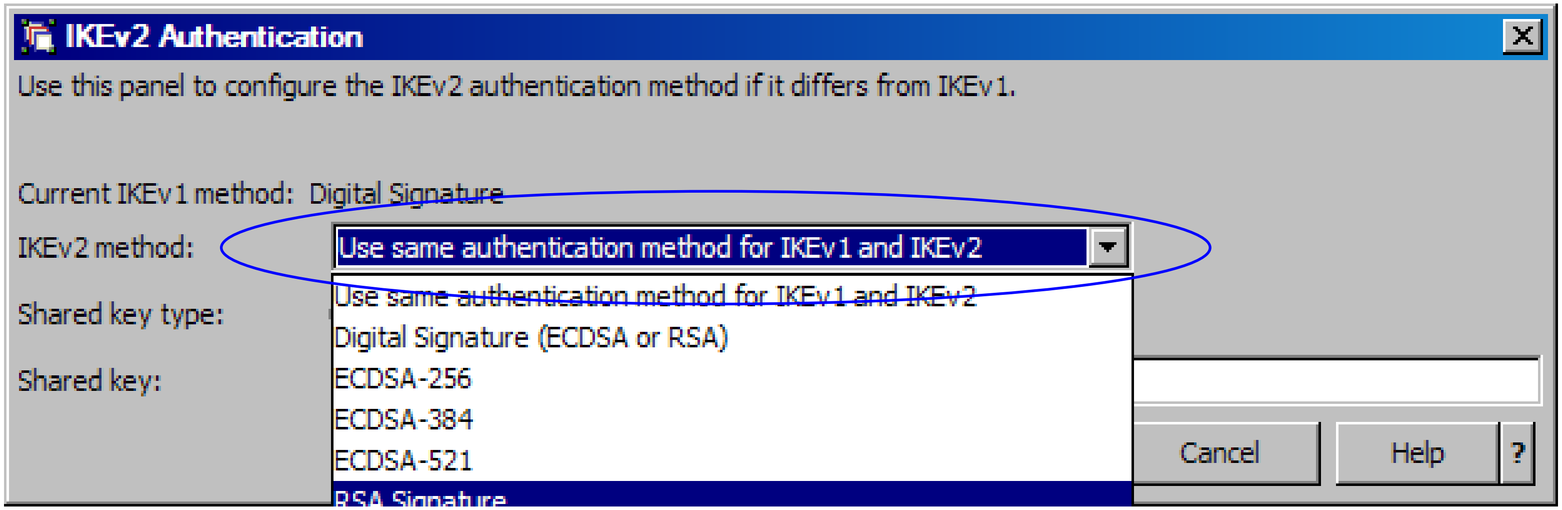
○ Do not automatically reauthenticate

● Authenticate every: 1440 (minutes)

Independent selection of authentication method, lifetime/lifesize

- **The Configuration Assistant defaults to using the same peer-to-peer authentication method as selected for IKEv1**
  - This selection can be overridden at the advanced connectivity rule settings level
  - *Note:* Elliptical Curve Digital Signature Algorithm (ECDSA) is currently supported in IKEv2 only. To use ECDSA, an override of the IKEv1 authentication method is required.

- For IKEv1, the two peers must negotiate and agree on Authentication method, and on values for Lifetime and Lifesize
- For IKEv2, each peer chooses its own Authentication method, Lifetime, and Lifesize



IKEv2 Authentication

Use this panel to configure the IKEv2 authentication method if it differs from IKEv1.

Current IKEv1 method: Digital Signature

IKEv2 method: Use same authentication method for IKEv1 and IKEv2

Shared key type:

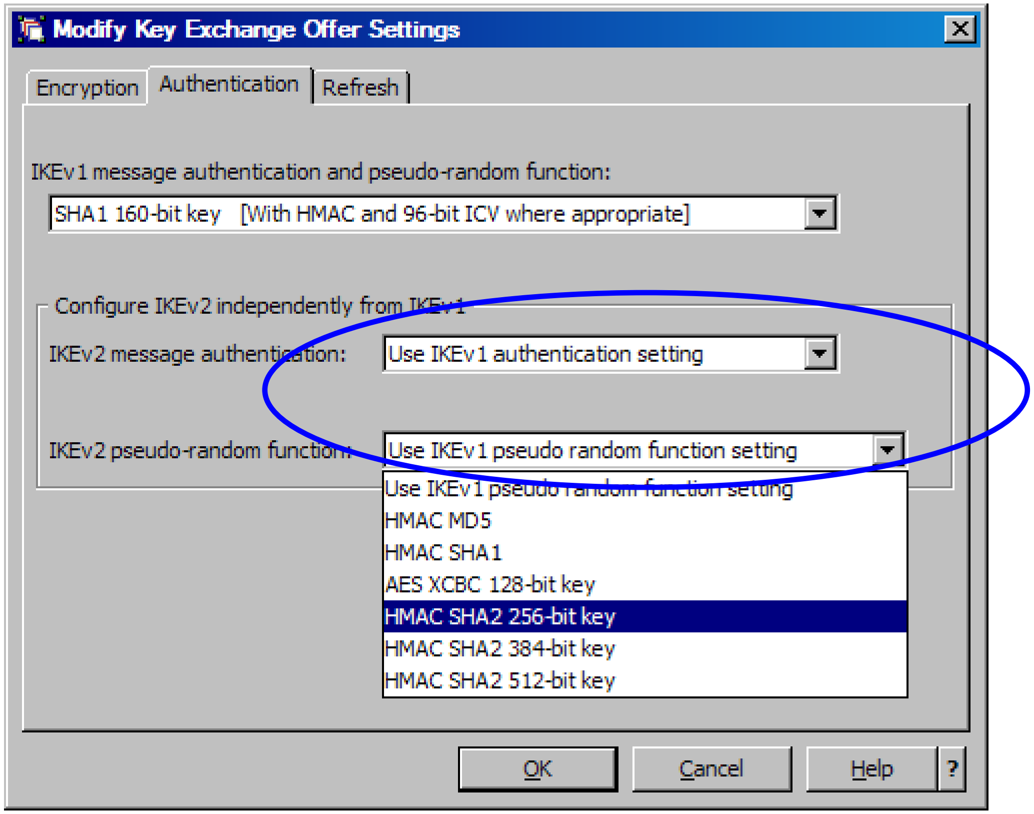
Shared key:

Use same authentication method for IKEv1 and IKEv2
Digital Signature (ECDSA or RSA)
ECDSA-256
ECDSA-384
ECDSA-521
RSA Signature
Cancel
Help
?

- **Lifetime and lifesize selections are the same for IKEv1 and IKEv2 !**

Configuring hash algorithm selections

- **In IKEv1, a single negotiated hash algorithm is used for both message authentication and pseudo-random key function (PRF)**
- **In IKEv2, these two algorithms can be separately specified in configuration**
  - The Configuration Assistant defaults to using the same single IKEv1 parameter for both algorithms OR they can be overridden at the advanced connectivity rule settings level



IKEv1 message authentication and pseudo-random function: SHA1 160-bit key [With HMAC and 96-bit ICV where appropriate]

Configure IKEv2 independently from IKEv1

IKEv2 message authentication: Use IKEv1 authentication setting

IKEv2 pseudo-random function: Use IKEv1 pseudo random function setting

Use IKEv1 pseudo random function setting
HMAC MD5
HMAC SHA1
AES XCBC 128-bit key
HMAC SHA2 256-bit key
HMAC SHA2 384-bit key
HMAC SHA2 512-bit key

ipsec command example

- **ipsec –k display commands show both IKEv1 and IKEv2 phase 1 SAs**

```
USER1@MVS124:/proj/UT # ipsec -k dis -p tcpcs2

CS V1R12 ipsec  Stack Name: TCPCS2  Thu Jan 14 15:29:05 2010
Primary:  IKE tunnel      Function: Display            Format:   Detail
Source:   IKED            Scope:    Current            TotAvail: n/a

TunnelID:                     K15
Generation:                   1
IKEVersion:                   2.0
KeyExchangeRuleName:          IKEv2-PSK-MD5-DES-v4
:
:
************************************************************************
TunnelID:                     K18
Generation:                   1
IKEVersion:                   1.0
KeyExchangeRuleName:          H-H-SingleIP-TCP-10062-5
:
:
************************************************************************

2 entries selected
USER1@MVS124:/proj/UT #
```

ipsec command example

- **ipsec –y display commands show both IKEv1 and IKEv2 phase 2 SAs**

```
USER1@MVS124:/proj/UT # ipsec -y dis -p tcpcs2

CS V1R12 ipsec  Stack Name: TCPCS2  Thu Jan 14 15:29:09 2010
Primary:  Dynamic tunnel  Function: Display            Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 2

TunnelID:                     Y16
Generation:                   1
IKEVersion:                   2.0
ParentIKETunnelID:            K15
:
:
***********************************************************************
TunnelID:                     Y19
Generation:                   1
IKEVersion:                   1.0
ParentIKETunnelID:            K18
:
:
***********************************************************************

2 entries selected
USER1@MVS124:/proj/UT #
```

z/OS V1R12 certificate support summary

The table below summarizes the support for digital certificates in IKED and NSSD and identifies the supported configurations

Function	IKEv1 Local	IKEv1 with NSSD	IKEv2 with NSSD
Rivest-Shamir-Adleman (RSA) Digital Signature algorithm [1,2]	✓	✓	✓
Elliptic Curve Digital Signature Algorithm (ECDSA) [2]			✓
Enhanced ID validation [2]	✓	✓	✓
Certificate Trust Chain Support [2]		✓	✓
Certificate Revocation Lists [2]		✓	✓
Hash and URL encoding for certificates [2,3]			✓
Hash and URL encoding for certificate bundles [2,3]			✓

[1] Available prior to z/OS V1R12

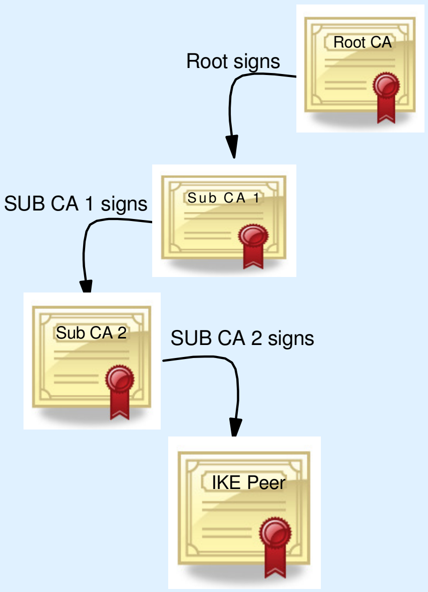
[2] Required by IKEv2

[3] Applicable to IKEv2 only

IKE certificate trust chain support

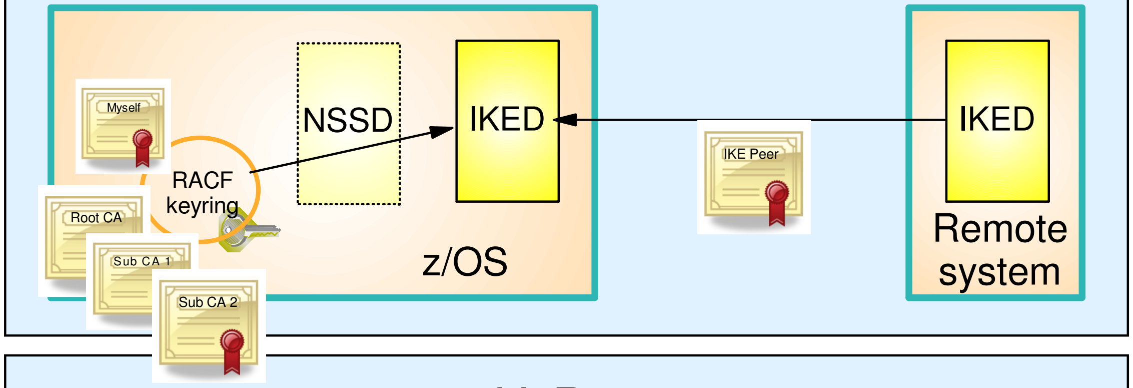
Eases administrative requirements by reducing the number of subordinate CA certificates needed on IKE keyrings

Given the following certificate hierarchy:



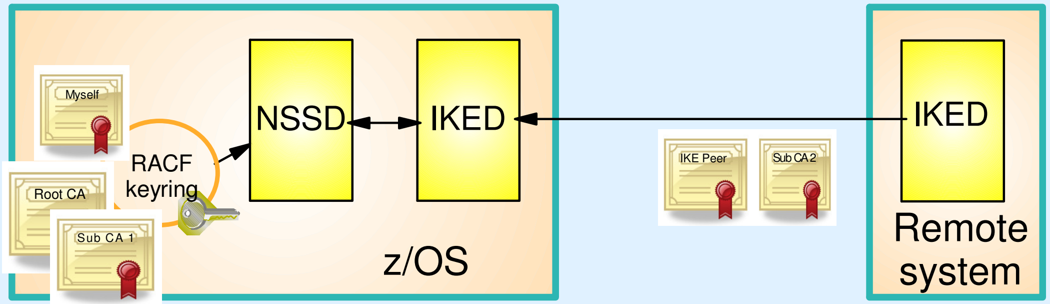
Pre-V1R12

Local keyring must contain cert of CA that signed peer cert



V1R12

IKED and NSSD cooperate to build and validate complete trust chain using keyring and intermediate certs sent by IKE peer

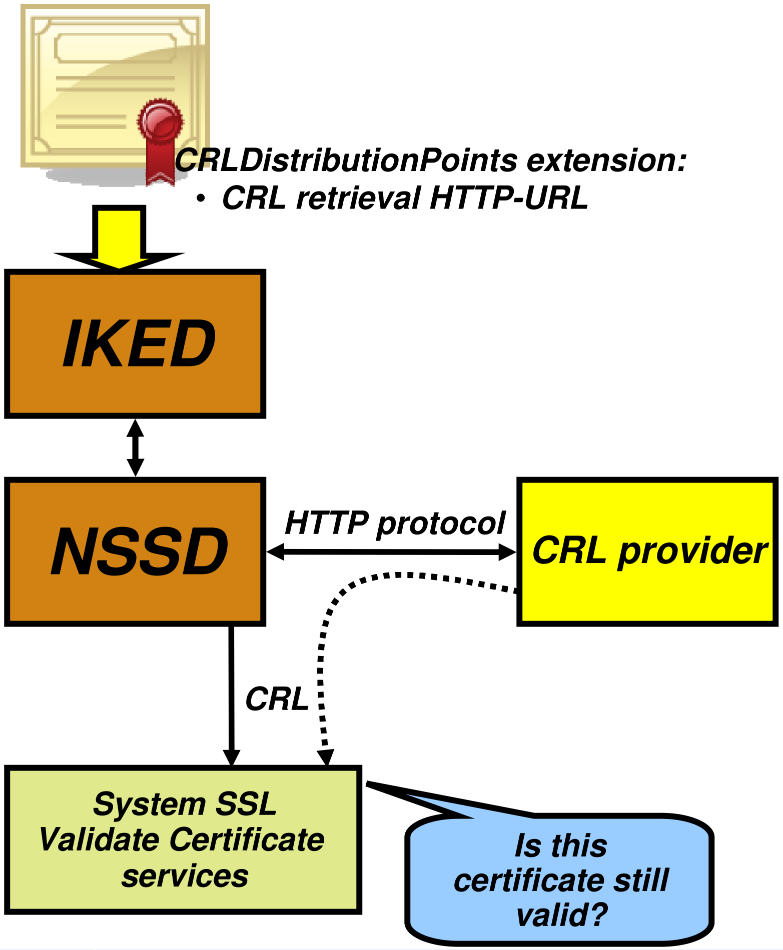


- RFCs 4306 and 4945 require support of trust chains
- Supported for both IKEv1 and IKEv2
- Requires NSSD
- NSSD supports certs on keyring as well as IKEv2 cert retrieval through HTTP

IKE Certificate Revocation List Support

A Certificate Revocation List is a list of certificates that have been revoked or are no longer valid.

CRLs are digitally signed by issuing certificate authority


CRLDistributionPoints extension:
• CRL retrieval HTTP-URL
IKED
NSSD
HTTP protocol
CRL provider
CRL
System SSL Validate Certificate services
Is this certificate still valid?

- RFC 4306 requires that when IPSec authenticates a digital signature, it needs to ensure that the certificate presented for authentication is still valid
- IKED controls level of CRL checking done based on configuration in IPSec policy
- IKED requests that NSSD retrieve CRLs using information in the CRLDistributionPoints extension in a certificate
  - HTTP-URLs only
  - Retrieval of CRLs from LDAP servers not supported
- NSSD will pass CRLs to z/OS System SSL services
- System SSL will validate the certificate against the CRL to ensure the certificate has not been revoked

IKEv2 hash and URL encoding of certificate and certificate bundles

An alternative to sending full certificates in IKE messages

Retrieval of certificates from HTTP servers

- **IKEv2 support includes new certificate payloads encoded using hash and URL encoding**
  - Hash and URL encoding is an alternative to transmitting the actual certificates
  - Amount of data carried on the IKE certificate payload flow is reduced
    - Offsets the potential for increased data introduced by Certificate Trust Chain support
    - Reduces exposure of exceeding UDP datagram maximum size of 64K bytes
- **The URL points to the location of either a certificate or “certificate bundle” in a binary file on HTTP server**
  - Facilitates creation of a shared public key infrastructure by using HTTP servers as digital certificate repositories
    - Certificates / certificate bundles can be retrieved using HTTP protocols by anyone who knows the URL and has network access to the HTTP server

Enhanced IKE ID processing

More stringent IKE ID validation now defined

- **RFC 4945 provides additional requirements relative to how the ID in a certificate must be used**
  - When the local ID is an X.509 distinguished name the ID payload in the IKE message flow MUST be populated with the content of the end-entity certificate's Subject field
  - MUST be capable of verifying an IP address used as an ID:
    - is the same as the peer's source address
    - is in the SubjectAltName extension of the peer's certificate
- **In z/OS V1R12, IKED is updated to populate the local identity using the binary subject distinguished name field from the signing certificate when:**
  - using digital signature authentication and
  - the local identity type is ID_DER_ASN1_DN (X.500 distinguished name)
- **In z/OS V1R12, IKED is updated to verify that the source IP address in the IKE message IP header matches the remote security endpoint's ID when the ID payload in the IKE message is an IPv4 or IPv6 address.**
  - The validation can be optionally disabled
  - Applies to:
    - IKEv1 and IKEv2 protocols
    - Preshared key and Digital Signature authentication

Migration considerations

- **If you're using IKEv1 with RSA signature authentication, and you want to migrate those security associations to IKEv2,**
  - The stack supporting the security associations *must* use NSS certificate services, and
  - the NSS Server providing them *must* be at z/OS V1R12 level
    - Configuration Assistant Health Checks point that out

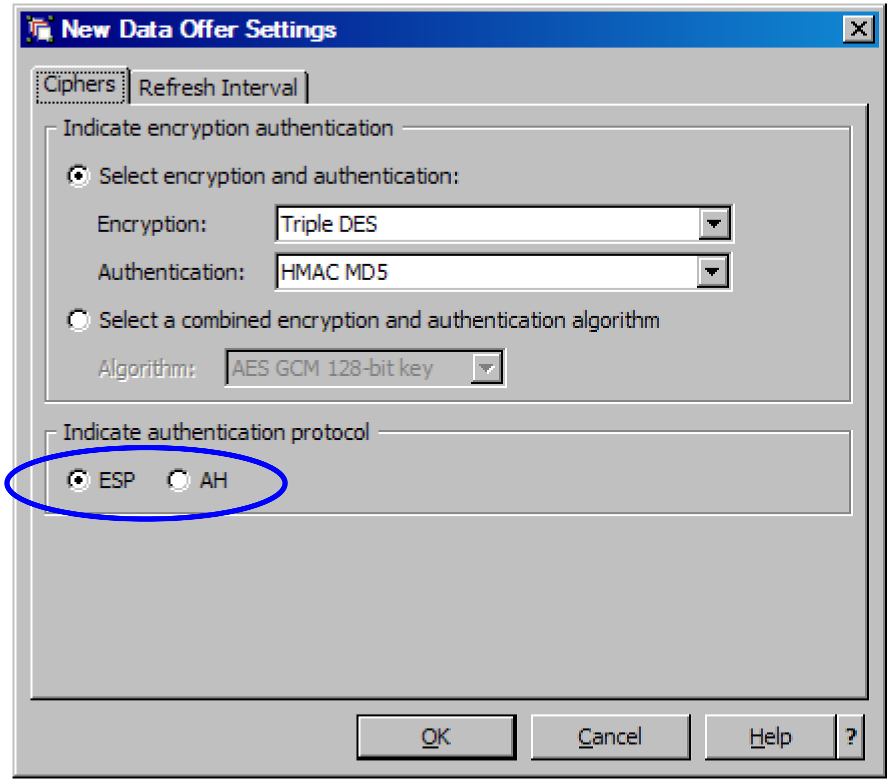
- **If you do NOT currently use NSS certificate services**
  - configure and start the NSS Server daemon
    - Give it access to the key ring that IKED currently uses
  - configure the stacks to request NSS certificate services

- **If you DO currently use NSS certificate services**
  - Migrate the NSS system to V1R12 first
    - Before migrating the systems running IKED that will activate IKEv2 tunnels using digital signature authentication

Migration considerations (continued)

- **Configuring z/OS to *initiate* with IKEv2 might cause security association activation failures to nodes that do not yet support IKEv2**
  - For example, z/OS V1R11 or earlier releases
  - Other peer nodes might need to be upgraded or reconfigured
  - If z/OS is not the initiator, this is less of a concern
    - z/OS will respond to both IKEv1 and IKEv2 security association activation requests

- **IKEv1 supports AH+ESP (SA bundles), but IKEv2 does not**
  - RFC 4718 clarifies that implementations should NOT support AH+ESP
  - If you directly edit the IPSec policy file, look for any IpDataOffer with HowToAuth **AH** *and* HowToEncrypt other than **DoNot**
    - Suggestion: change to HowToAuth **ESP**
  - If you use the Configuration Assistant, the IPSec protocol selection for message authentication is specified as a "Security Level" advanced setting and the default is **ESP**.
    - Ensure that this setting has not been changed to **AH** for any security level that also specifies encryption.



Indicate encryption authentication

● Select encryption and authentication:

Encryption: Triple DES

Authentication: HMAC MD5

○ Select a combined encryption and authentication algorithm

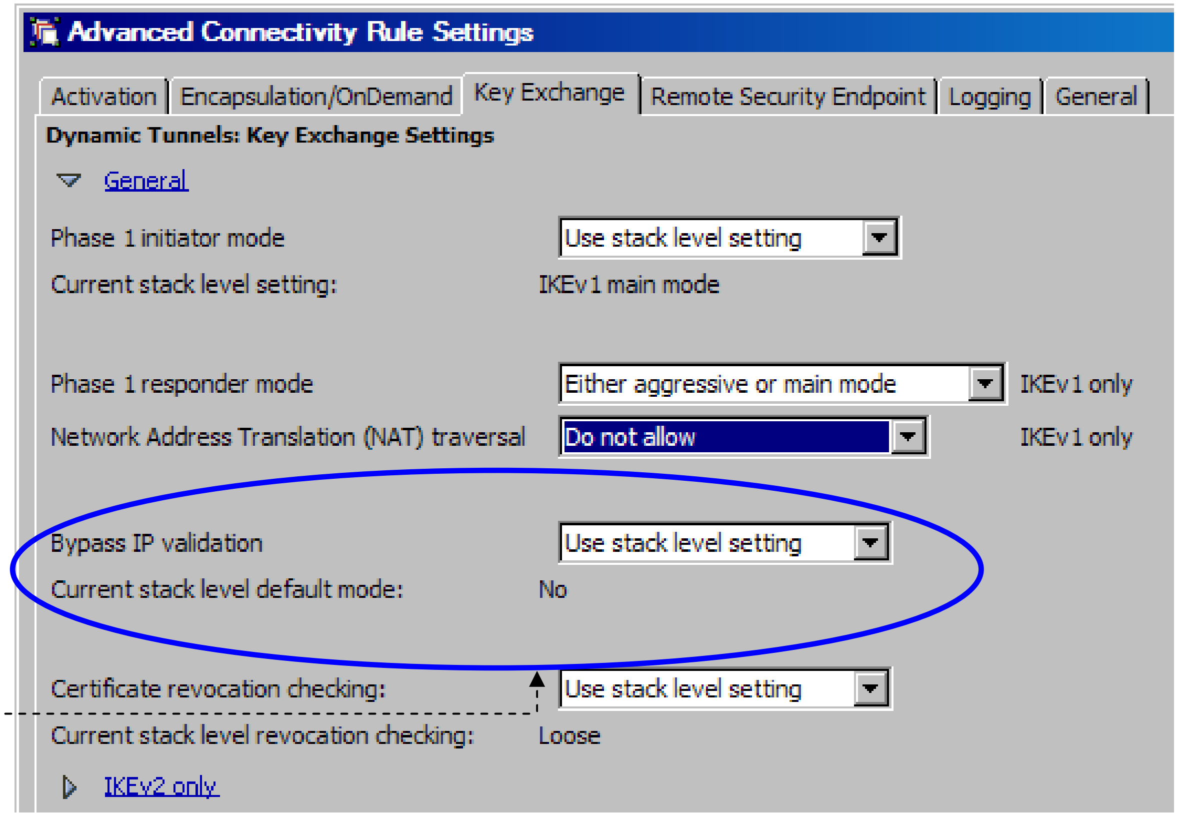
Algorithm: AES GCM 128-bit key

Indicate authentication protocol

● ESP ○ AH

Migration considerations (continued)

- **RFC 4945 requires IP address validation to be enabled by default.**
  - This check was not performed prior to V1R12 Comm Server
  - If you directly edit the IPSec policy you will need to code BypassIpValidation Yes on the KeyExchangePolicy to disable the check
    - If you use the Configuration Assistant, you just need to generate new IPSec configuration files
- **If the remote security endpoint is expected to be behind a NAT, its IP address will NOT match!**
  - Bypass IP validation needs to be set for this case
    - Default of “No” can be overridden in the advanced connectivity rules settings level



Dynamic Tunnels: Key Exchange Settings

General

Phase 1 initiator mode: Use stack level setting

Current stack level setting: IKEv1 main mode

Phase 1 responder mode: Either aggressive or main mode — IKEv1 only

Network Address Translation (NAT) traversal: Do not allow — IKEv1 only

Bypass IP validation: Use stack level setting

Current stack level default mode: No

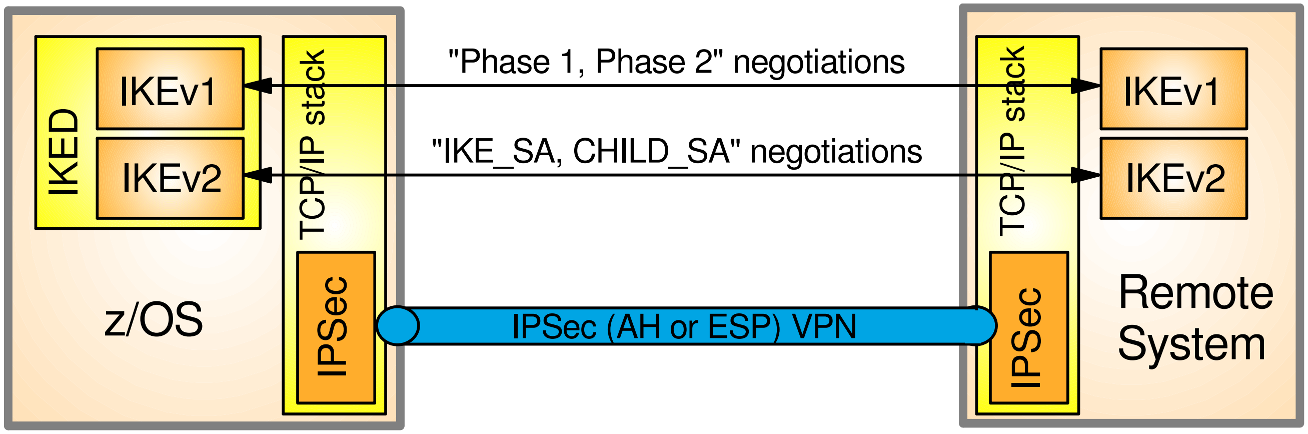
Certificate revocation checking: Use stack level setting

Current stack level revocation checking: Loose

IKEv2 only

IKEv2 Wrap-up

- IKE version 1 (IKEv1) specified by RFCs 2407-2409
- **IKE version 2 (IKEv2) specified by RFCs 4306/5996**



- **IKEv2 protocol**
  - Supports all of the same configurations as IKEv1
  - Different protocol than IKEv1
    - similar function
    - different messages and flows
    - different terminology
  - More efficient than IKEv1:
    - fewer messages per negotiation
    - new formats allow for smaller messages
  - More robust than IKEv1:
    - Request/response model for all flows
    - Built-in dead peer detection

- **z/OS IKEv2 implementation**
  - Coexists and concurrently supported with IKEv1 in IKED
  - Fully supported by Configuration Assistant for z/OS
  - Requires network security services (NSS) for certificate-based authentication
  - Does not support:
    - NAT traversal for IPv4, support planned for V1R13
    - System-Wide Security Associations (SWSA), support planned for V1R13

For more information

URL	Content
http://www.twitter.com/IBM_Commserver twitter	IBM Communications Server Twitter Feed
http://www.facebook.com/IBMCommserver facebook	IBM Communications Server Facebook Fan Page
http://www.ibm.com/systems/z/	IBM System z in general
http://www.ibm.com/systems/z/hardware/networking/	IBM Mainframe System z networking
http://www.ibm.com/software/network/commserver/	IBM Software Communications Server products
http://www.ibm.com/software/network/commserver/zos/	IBM z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	IBM Communications Server for Linux on System z
http://www.ibm.com/software/network/ccl/	IBM Communication Controller for Linux on System z
http://www.ibm.com/software/network/commserver/library/	IBM Communications Server library
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/zos/support/	IBM z/OS Communications Server technical Support – including TechNotes from service
http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs	Technical support documentation from Washington Systems Center (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)
http://www.ibm.com/systems/z/os/zos/bkserv/	IBM z/OS Internet library – PDF files of all z/OS manuals including CommServer