

Securing and Encrypting Network Traffic with z/OS Communications Server and Policy Agent

Security Workshop

Protecting Traffic with IPsec



IBM Washington System Center
IBM Technical Sales Support

Trademarks

- **The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.**
 - IBM
 - z/OS
 - **The following are trademarks or registered trademarks of other companies.**
 - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.
 - All other products may be trademarks or registered trademarks of their respective companies.
 - Refer to www.ibm.com/legal for further legal information.
-
- OSA-Express Features
 - There are many different types of OSA-Express features. In this document where OSA is mentioned it refers to an OSA port on any of the OSA-Express features unless a specific OSA-Express feature is mentioned.

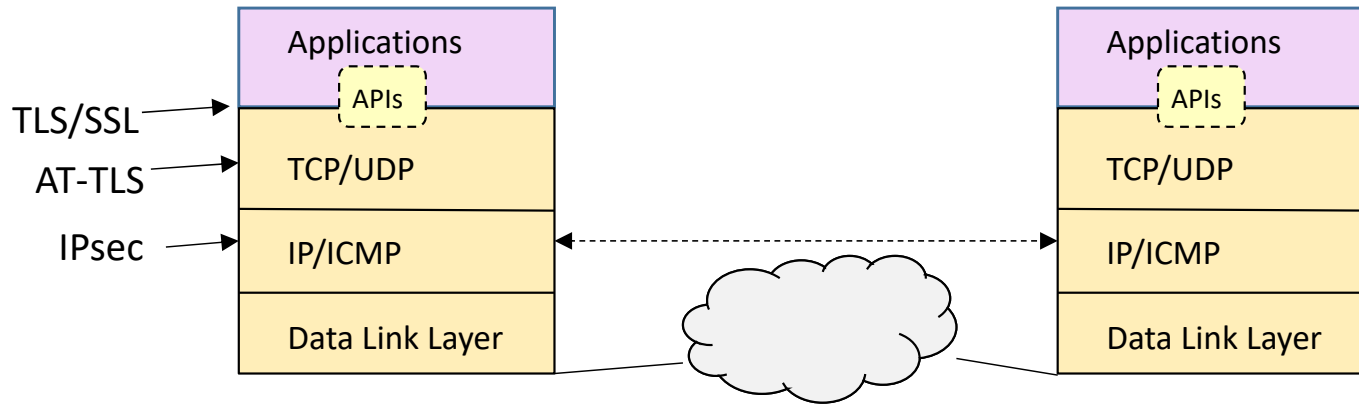
Agenda

- Overview of IPsec Protocol
- IPsec Tunnels
- Asymmetric and Symmetric Encryption
- Transport Mode Versus Tunnel Mode
- z/OS IPsec Implementation
- Network Security Services
- Sysplex Wide Security Associations

Overview of IPsec Protocol

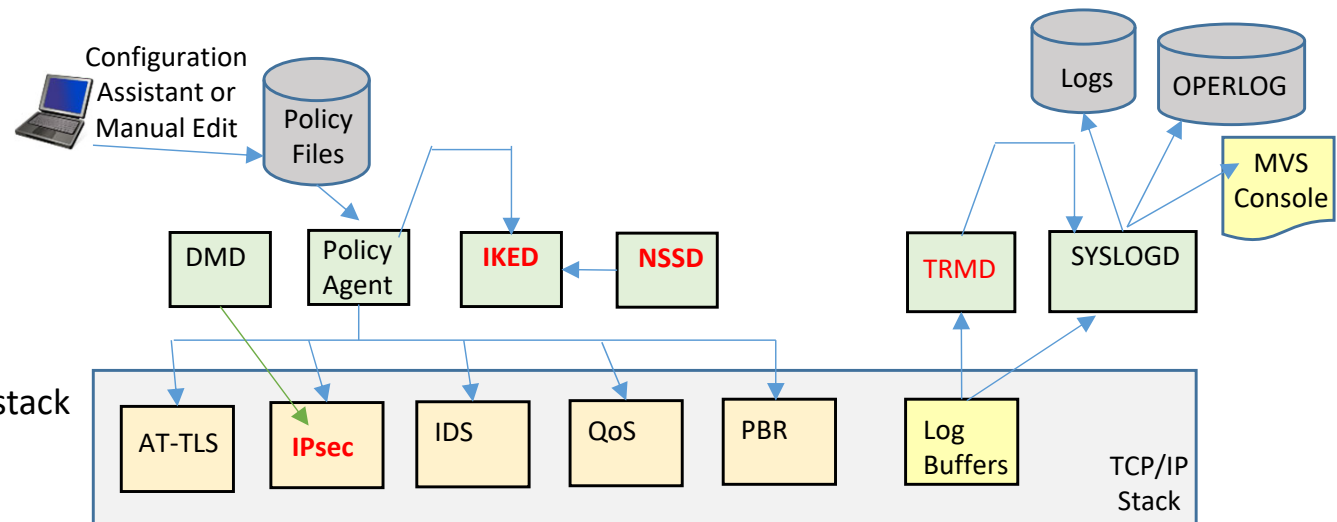


IPsec Protocol Overview



- Open standard network layer security protocol defined by IETF in RFCs
 - Provides authentication, integrity, and data privacy
 - RFC 5996
- IPSec security protocols
 - Authentication Header (AH) - provides authentication / integrity
 - Encapsulating Security Protocol (ESP) - provides data privacy with optional authentication/integrity
- Implemented at IP layer
 - Requires no application change
 - Secures traffic between any two IP resources
 - Security Associations (SA)
- Management of crypto keys and security associations can be
 - Manual
 - Automated via key management protocol (IKE)

Lots of Different Policy Types and Started Tasks

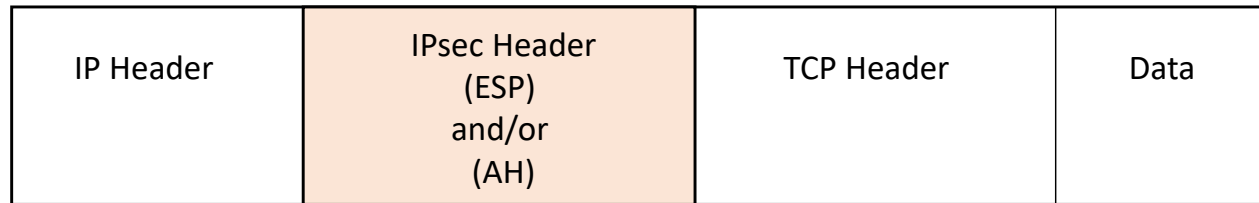


- Policy Agent
 - Installs Policies into the TCP/IP stack
- TCP/IP Stack
 - Enforces the Policies
- DMD (Defense Manager Daemon)
 - ipsec command can be used to install temporary IP Filter rules.
- **IKED (Internet Key Exchange Daemon)**
 - Required for IPsec Dynamic VPN (Virtual Private Network) Tunnels
- **NSSD (Network Security Server Daemon)**
 - Required for IKEv2
 - If digital signatures are used as the authentication method, then IKED must be configured to use the certificate services of the NSSD.
 - Provides central RACF certificate repository for remote IKED applications
 - Provides DataPower access to RACF
- **TRMD (Traffic Regulation Management Daemon)**
 - Required for log messages to syslogd for IP Filter, IPsec, and IDS
- SyslogD
 - Recommended for logging

IPsec Tunnels

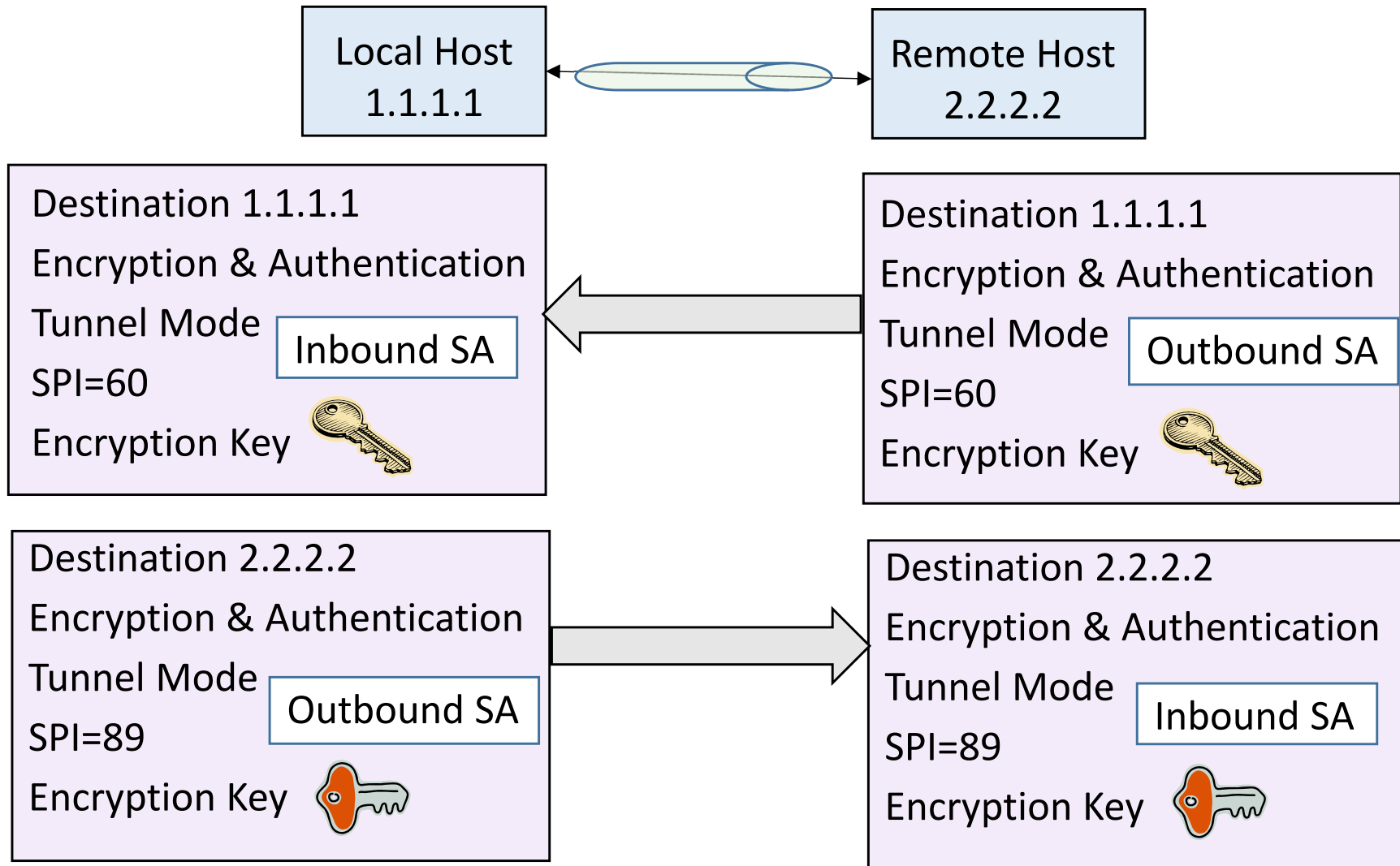


IPsec Header



- Data (original IP packets) passing through a tunnel can be:
 - authenticated (AH)
 - encrypted (ESP; authentication is optional)
 - Commonly used alone since it can perform both functions (encrypt + authenticate)
 - both: encrypted and then authenticated (ESP and AH)
- AH Protocol:
 - Protocol #51
 - Defined in RFC 2402 (supersedes RFC 1826)
 - Provides integrity and authentication
 - Includes selected fields of the IP header
 - Requires authentication algorithms:
 - HMAC-MD5-96 (RFC 2403)
 - HMAC-SHA-1-96 (RFC 2404)
 - Provides optional replay protection
 - May be used in combination with ESP
- Encapsulating Security Protocol
 - Protocol #50
 - Defined in RFC 2406 (supersedes RFC 1827)
 - Provides integrity, authentication, and encryption
 - Does not include fields of the IP header
 - Required authentication algorithms:
 - HMAC-MD5-96
 - HMAC-SHA-1-96
 - Null Authentication (i.e. none)
 - Required encryption algorithms:
 - DES_CBC (RFC 2405)
 - NULL (RFC 2410)
 - 3DES (RFC 2451)
 - Provides optional replay protection
 - May be used in combination with AH

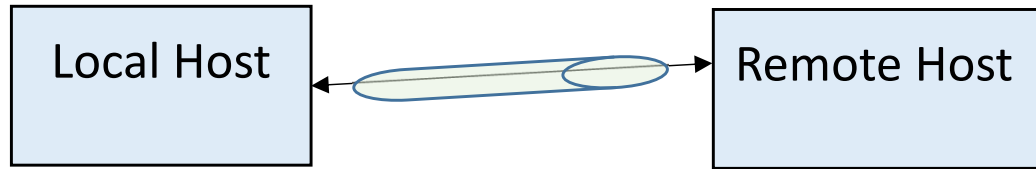
Security Association (SA)



One symmetric key for inbound and another symmetric key for outbound.

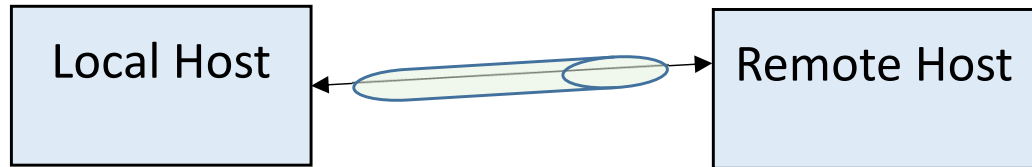
SPI = Security Parameter Index

Manual versus Dynamic Tunnels



- Two different types of IPsec Virtual Private Network (VPN) Tunnels
 - Manually
 - Dynamically using the Internet Key Exchange (IKE) protocol
- Manual Tunnels
 - All attributes are manually configured
 - Attributes of the Security Associations must match
- Dynamic Tunnels
 - IKE protocol is implemented by IKE Daemon (IKED)
 - IKE securely negotiates IPsec SAs
 - Uses a special "IKE SA" to protect the IKE protocol messages
 - Much more flexible and scalable than manual tunnels, but requires more infrastructure
 - Additional security definitions for authentication and encryption

Manual (Static) Tunnel



- Manual VPN Flows:

- IPsec Manual VPNs use Symmetric Cryptography.
 - There is no partner key exchange or verification at VPN setup time.
 - All partner information - including a shared secret key for authentication -- is manually configured ahead of time and shared without threat of compromise
- Security Associations:
 - Security Association for ESP, including Authentication and Encryption
 - Most commonly used
 - Authentication Security Associations for Inbound & Outbound
 - Encryption Security Associations for Inbound & Outbound
- All right for testing, but
 - Open to intrusion and security threats

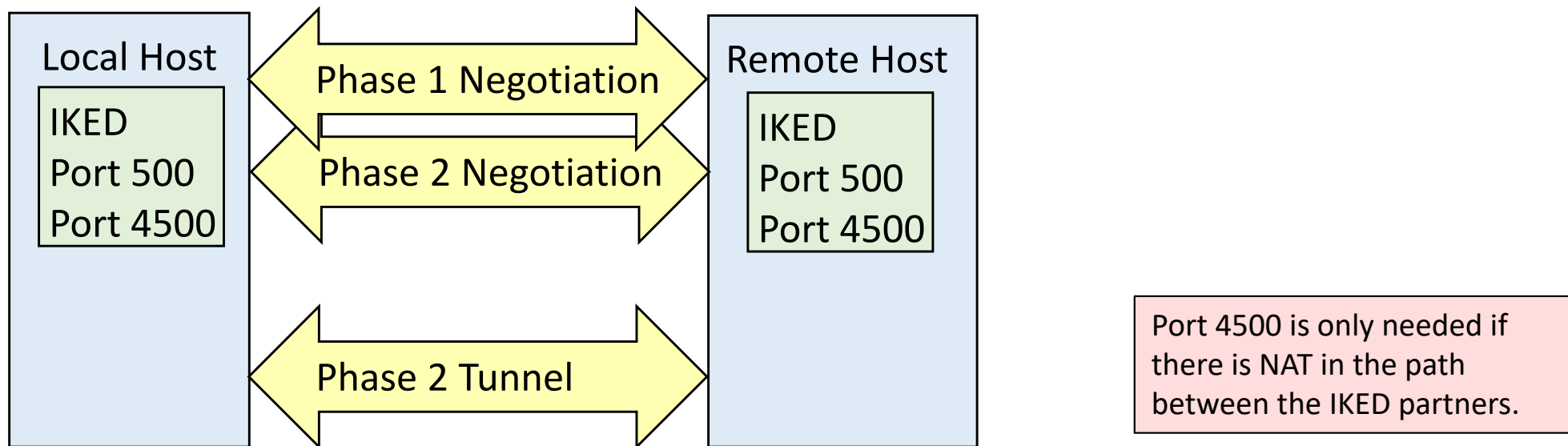
Sample Manual Tunnel Definition

```
IpManVpnAction          IPsecGoldStatic~0
{
  Active                Yes
  LocalSecurityEndpointAddr Any4
  RemoteSecurityEndpointAddr Any4
  HowToAuth              ESP Hmac_Sha
  HowToEncrypt            3DES
  HowToEncap              Transport
  AuthOutboundSa          300 0x0123456789ABCDEF0123456789ABCDEF01234567
  AuthInboundSa            301 0x9876543219876543219876543219876543219876
  EncryptOutboundSa        300 0x0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  EncryptInboundSa          301 0x9876543219876543219876543219876543219876
}

## Connectivity Rule    MyStaticVPNTweenOSAs combines the following items:
## Local data endpoint  All4
## Remote data endpoint All4
## Topology              Host to Host
## Requirement Map       LPAR1toOtherLPARs
## EE                     => IPsecGoldStatic

IpFilterRule            MyStaticVPNTweenOSAs~1
{
  IpSourceAddr           All4
  IpDestAddr              All4
  IpServiceRef            EE
  IpGenericFilterActionRef IpSec~LogYes
  IpManVpnActionRef        IPsecGoldStatic~0
}
```

Dynamic Tunnel

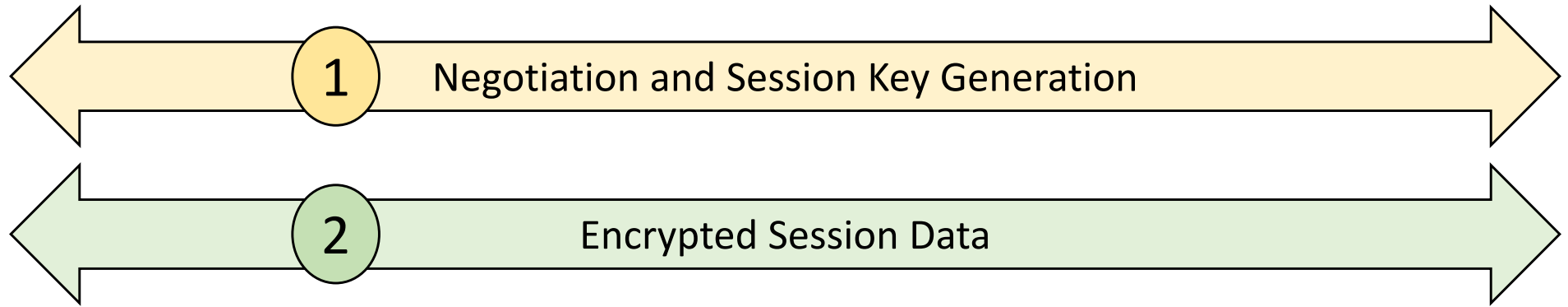


- **Dynamic VPN Negotiation Flow (Ports 500 or 4500):**
 - **Phase 1 (Authentication and Negotiation of SAs for IKE Messages)**
 - Pre-shared Key Mode: Negotiation authenticates with a value derived from a shared matching "key" or "password".
 - RSA Signature Mode: Definitions matching certificate result in authentication
 - **Phase 2 (Negotiation of SAs for Encrypted Data Transfer)**
 - IPsec generates Symmetric Session Keys (one for each direction)
 - **Phase 2 Tunnel (Data Payload with IPsec Headers)**
 - IPsec uses Symmetric Cryptography for data transfer.

Asymmetric and Symmetric Encryption



General Architecture of Encryption Flow



Encryption Flow	What Happens	SSL/TLS Terminology	IPsec Terminology	OpenSSH Terminology
Stage 1 Asymmetric Algorithms	Negotiation of Secure Connection: Authentication and Generation of and encrypted Transmit of Session Key	Handshake Layer	Phase I Phase II	No official terminology; just negotiation stage
Stage 2 Symmetric Algorithms	Encryption and Decryption of Data Payload (Session Data)	Record Layer	Phase II Tunnel	No official terminology, just data transfer stage

- Essentially all these security protocols use the same basic architecture:

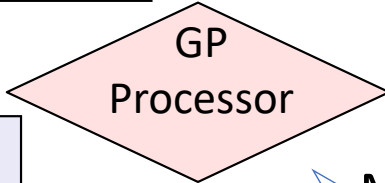
- 1 Authenticate the partner; generate a symmetric key
 - Encrypt symmetric key with asymmetric algorithm and send
- 2 Encrypt session data with symmetric ("Session") key and transmit session data

- Two Stages

Cryptographic Cards
(Accelerator or Coprocessor)

CPACF

Software



- Phase 1 Negotiation / Key Generation

- Dynamic Tunnels IPsec IKED Phases 1 and 2

- Authenticates Partners and generates SA Keys

- Uses ICSF or Crypto Card if available
Accelerator Card (Clear Key Mode)
Coprocessor Card (Secure Key Mode)

- Manual (Static) Tunnels

- Uses prior agreement instead of dynamic negotiation

- Phase 2 Data Tunnel

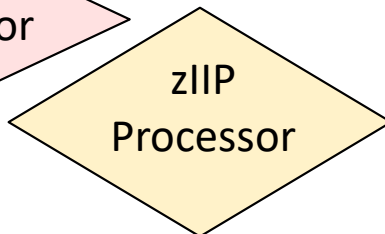
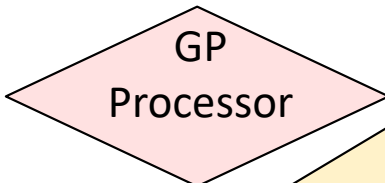
- Dynamic or Manual Tunnels

- Encrypts/Decrypts data

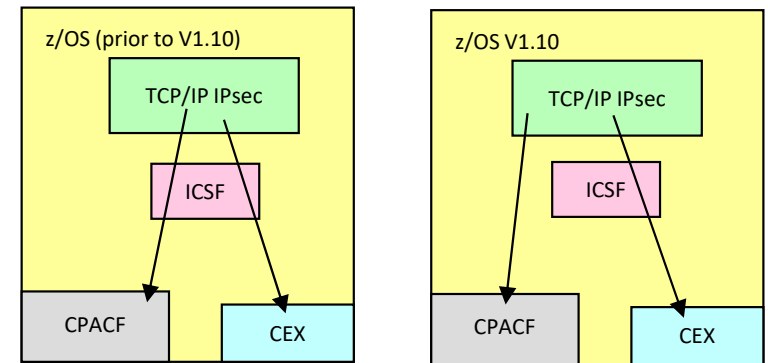
- Uses CPACF (clear key only) if available

CPACF

Software

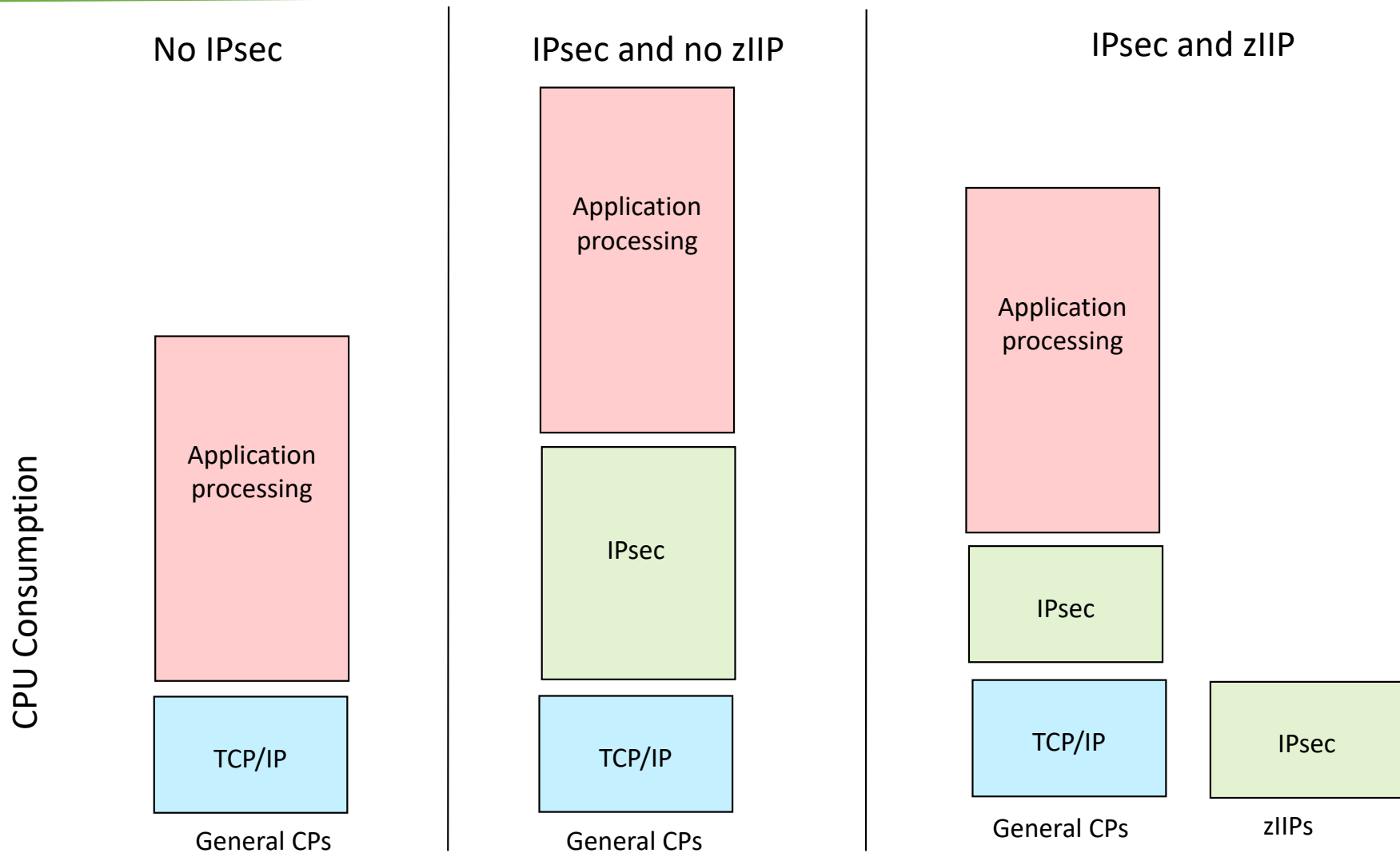


Cryptography Hardware Support



- IPsec work utilizes System z cryptography hardware if the hardware is enabled, and if the required cryptographic algorithm is supported by the hardware.
 - See the "Using Hardware Cryptographic Features with System SSL" table in the z/OS Cryptographic Services System Secure Sockets Layer Programming, manual for details about which cryptographic algorithms are supported by the System z cryptographic hardware.
- Prior to z/OS V1.10, IPsec uses (Integrated Cryptographic Service Facility) ICSF for all hardware crypto functions.
- Starting in z/OS V1.10, IPsec will save cycles by using the CPACF (CP Assist for Cryptographic Function) functions directly, for the crypto functions that are supported by the CPACF.
 - Throughput improvements are greatest for especially short datagrams (< 1K).
 - Anything else that currently is directed to ICSF will continue to be directed to ICSF.
 - Workload for Crypto Express will continue to flow through ICSF.
 - Note: Plans for the z/OS Communications Server are subject to change prior to general availability.

CPU Consumption for IPsec with zIIP Processor



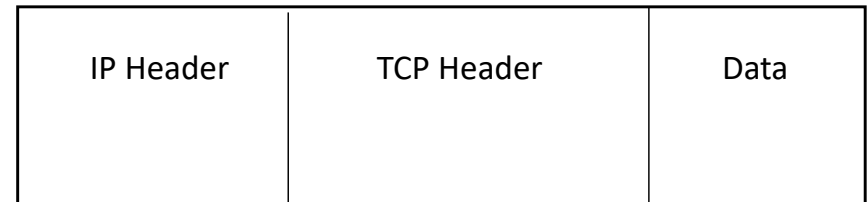
- When enabled on a z/OS image that includes zIIP(s), the zIIP IPSECURITY (zIIP assist for IPsec) function can reduce IPsec processing load on General Purpose CPs well beyond what is achievable using just CPACF or the Cryptographic hardware.
 - zIIP has access to the same hardware cryptographic facilities that are used for symmetric key encrypt/decrypt operations as a general purpose CP would have, so the benefits of using CPACF still apply when using zIIP.
 - The zIIP offload is focused on the encrypt/decrypt of the "user" data in the stack under control of the Security Association (SA). The offload does NOT have anything to do with the public key operations that occur during IKE key exchange flows.
 - The biggest zIIP benefit is realized for cases where there is a large amount of data transferred over long-lived ESP tunnels.

Transport Mode Versus Tunnel Mode



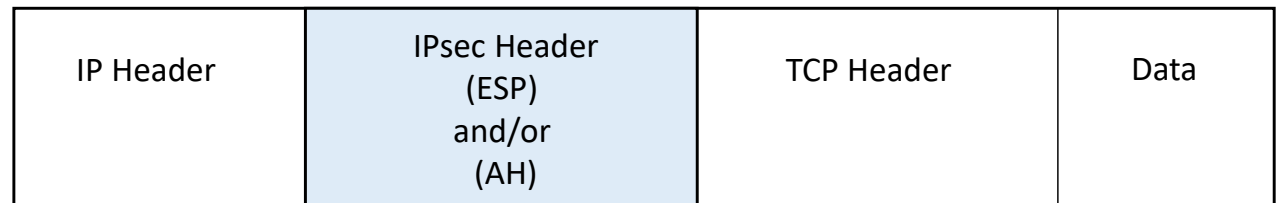
Transport Mode or Tunnel Mode

- Original Packet



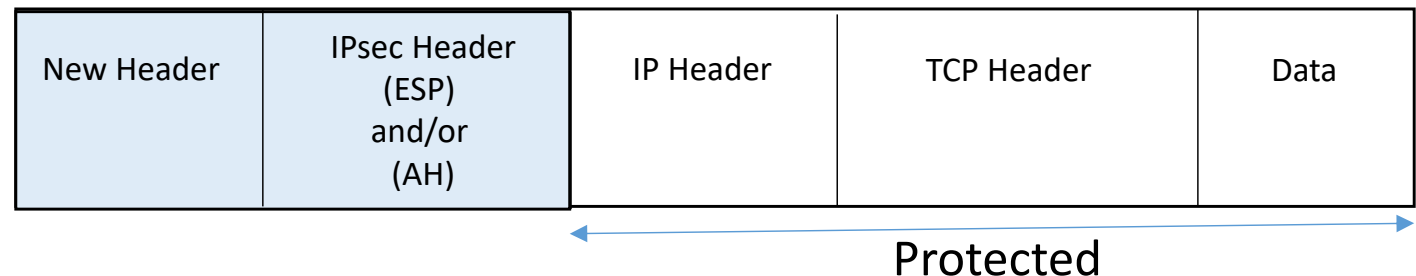
- Transport Mode

- Original data is protected but certain header fields are not.
- Typically used when Security Endpoint and Data Endpoint are the same IP Address.

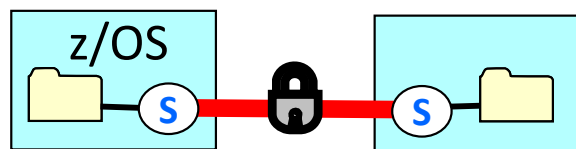


- Tunnel Mode

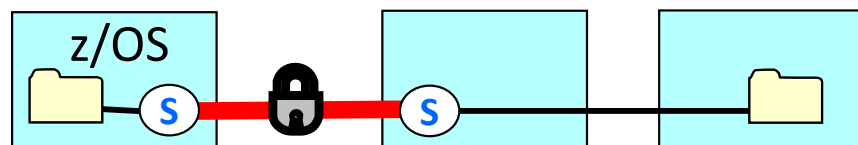
- Protects the entire IP packet
- A new IP header and IPsec header are placed in front of the original IP packet
- Always used when Security Endpoint and Data Endpoint are different IP Addresses on either or both ends.
- Optionally can be used to protect all contents of Original Packet where the Security Endpoint and Data Endpoints are the same IP Address.



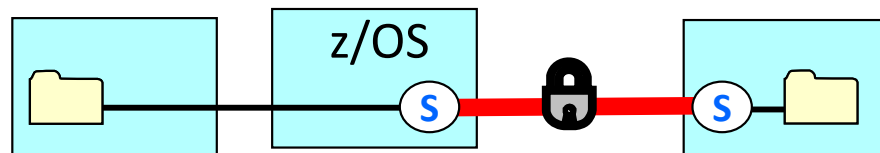
IPsec Network Topology



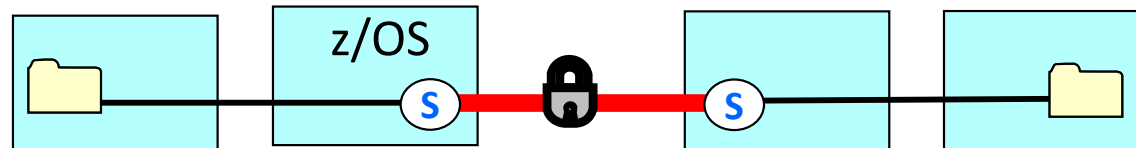
- Host-to-Host: z/OS node is data host and security endpoint both (host), connecting to a data host and security endpoint remote node (host).



- Host-to-Gateway: z/OS node is a data host and security endpoint both (host), connecting to a security endpoint (gateway) in front of a remote data host.



- Gateway-to-Host: Data host in front of z/OS node security endpoint (gateway), which connects to a data host and security endpoint remote node (host).



- Gateway-to-Gateway: Data host in front of z/OS node security endpoint (gateway), which connects to a security endpoint (gateway) in front of a remote data host.

- Security Endpoints are the endpoints of the IKE (phase 1) SA.
 - The negotiating IKEDs are located at the security endpoints.
 - The data is protected between the security endpoints.
- Data Endpoints are the endpoints of the Dynamic (phase 2) SA.
 - The data is not protected between the data endpoints and the security endpoints.
- When IPsec is configured, z/OS is considered a “Host” if it is the security endpoint (IKE/phase 1) and data endpoint (dynamic/phase 2), otherwise it is considered a “Gateway”.
- z/OS may be a data endpoint behind a security endpoint.
 - In this case z/OS does not require any IKE or IPsec customization.

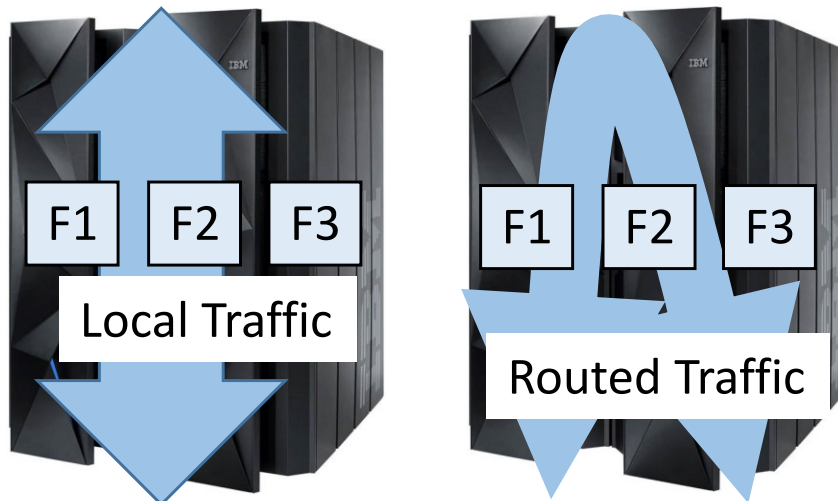
z/OS IPsec Implementation



Policy Definition

Criteria	Description
Packet	
Source address	Source IP Address in IP header of packet
Destination address	Destination IP Address in IP header of packet
Protocol	Protocol in the IP header of packet
Source port	Source Port in TCP or UDP transport header of packet
Destination port	Destination Port in TCP or UDP transport header of packet
ICMP type and code	ICMP type and code in ICMP header of packet
OSPF type	OSPF type in OSPF header of packet
Network Attributes	
Direction	Direction of packet
Routing	Traffic is Local if source or destination IP address exists on local host, otherwise traffic is Routed
Link security class	Class that allows you to group interfaces with similar security requirements. Non-VIPA addresses can be assigned a security class. Packets inherit the security class of the interface over which packet is sent/received.
Time Condition	
Time, Day, Week, Month	When filter rule is active.
Action	
Permit, Deny, or IPsec	Permit, Deny, or use IPsec

RFC4301



Fragment 1 = IP Header + TCP Header + Data
Fragment 2 = IP Header + Data
Fragment 3 = IP Header + Data

- Routed Traffic
 - First fragmented packet contains protocol header, while later packets do not.
- Local Traffic
 - Fragmented packets are put back together before being inspected.
- RFC4301 does not allow policy rules to be created for routed traffic that define those items that exist in the protocol header:
 - Port Numbers
 - ICMP(v6) Code Types
 - OSPF Types
- z/OS Network Configuration Assistant will prevent the creation of a rule that does not adhere to RFC4301.
- z/OS Policy Agent will not load a rule that does not adhere to RFC4301.

IPsec Dynamic Tunnel IKED Identity Choices

- X.509 Certificate

Label: IKED1

...

Issuer's Name:OU=MVSNM1 Certificate Authority.O=IBM.C=US

1 Subject's Name:CN=Server 1.T=IKED1.OU=MVS.O=IBM

Subject's AltNames:

2 IP:172.17.0.71

3 EMail: mvsnm1 at washington.ibm.com

4 Domain: washington.ibm.com

URI: mvsnm1.washington.ibm.com

...

A choice of four different Certificate fields. The Policy may define any one of the four fields. You can see how the optional AltNames fields are usually easier to define in the policy.

One AltName field may be used for to identify one side and a different AltName field may be used to identify the other side.

- Policy Configuration

Policy 2 "IP Address" = 172.17.0.71

Policy 4 "FQDN" = washington.ibm.com

Policy 3 "Userid@FQDN" = mvsnm1@washington.ibm.com

Policy 1 "X.509 Distinguished Name" = CN=Server 1,T=IKED1,OU=MVS,O=IBM

Policy 5 "Key ID" (ASCII, EBCDIC, or Hexadecimal) = gaithersburg

← Preshared Key Mode rather than RSA Signature Mode Authentication

- Sequence of Elements must be accurate!

Data Endpoint and IKED Identity Match

- IP Header of Packet

Local IP Address = 10.1.1.1 (1)

Remote IP Address = 10.2.2.2 (2)

- Local Policy Configuration

Local IKED Identity

IP Address = 172.17.0.71 (3)

Local data endpoint

10.1.1.1 (1)

Remote IKED Identity

IP Address = 172.17.0.72 (4)

Remote data endpoint

10.2.2.2 (2)

Key Exchange Rule

Preshared Key

*
Preshared Key
Mode
Authentication
does not use any
certificate so
there is no
certificate check.

- IP Header of Packet

(2) Local IP Address = 10.2.2.2

(1) Remote IP Address = 10.1.1.1

- Remote Policy Configuration

(4) Local IKED Identity

IP Address = 172.17.0.72

(2) Local data endpoint

10.2.2.2

(3) Remote IKED Identity

IP Address = 172.17.0.71

(1) Remote data endpoint

10.1.1.1

Key Exchange Rule

Preshared Key

IKED Identity = Distinguished Name

- IP Header of Packet

Local IP Address = 10.1.1.1 (1)

Remote IP Address = 10.2.2.2 (2)

- Local X.509 Certificate

Subject's Name:CN=Server 1.T=... (3)

- Local Policy Configuration

Local IKED Identity

X.509 Dist Name = CN=Server 1,T=... (3)

Local data endpoint

10.1.1.1 (1)

Remote IKED Identity

X.509 Dist Name = CN=Server 2,T=... (4)

Remote data endpoint

10.2.2.2 (2)

Key Exchange Rule

RSA Signature

- IP Header of Packet

(2) Local IP Address = 10.2.2.2

(1) Remote IP Address = 10.1.1.1

- Remote X.509 Certificate

(4) Subject's Name:CN=Server 2.T=IKED2...

- Remote Policy Configuration

Local IKED Identity

(4) **X.509 Dist Name** = CN=Server 2,T=IKED2...

Local data endpoint

(2) 10.2.2.2

Remote IKED Identity

(3) **X.509 Dist Name** = CN=Server 1,T=IKED1...

Remote data endpoint

(1) 10.1.1.1

Key Exchange Rule

RSA Signature

IKED Identity AltName IP Address

- IP Header of Packet

Local IP Address = 10.1.1.1 (1)

Remote IP Address = 10.2.2.2 (2)

- Local X.509 Certificate

Subject's AltNames:

IP: 172.17.0.71 (3)

- Local Policy Configuration

Local IKED Identity

IP Address = 172.17.0.71 (3)

Local data endpoint

10.1.1.1 (1)

Remote IKED Identity

IP Address = 172.17.0.72 (4)

Remote data endpoint

10.2.2.2 (2)

Key Exchange Rule

RSA Signature

- IP Header of Packet

(2) Local IP Address = 10.2.2.2

(1) Remote IP Address = 10.1.1.1

- Remote X.509 Certificate

Subject's AltNames:

(4) IP: 172.17.0.72

- Remote Policy Configuration

Local IKED Identity

(4) **IP Address** = 172.17.0.72

Local data endpoint

(2) 10.2.2.2

Remote IKED Identity

(3) **IP Address** = 172.17.0.71

Remote data endpoint

(1) 10.1.1.1

Key Exchange Rule

RSA Signature

If Host Topology and IKED Identity is defined as IP address, IP Address in certificate is checked against IP Address, unless "BypassIpValidation" is defined. See KeyExchangeAction.

IKED Identity AltName Email

- IP Header of Packet
Local IP Address = 10.1.1.1 (1)
Remote IP Address = 10.2.2.2 (2)

- Local X.509 Certificate
Subject's AltNames: (3)
Email: mvsnm1 at washington.ibm.com

- Local Policy Configuration
Local IKED Identity
Userid@FQDN =
mvsnm1@washington.ibm.com (3)
Local data endpoint
10.1.1.1 (1)
Remote IKED Identity
Userid@FQDN =
mvsnm2@washington.ibm.com (4)
Remote data endpoint
10.2.2.2 (2)
Key Exchange Rule
RSA Signature

- IP Header of Packet
(2) Local IP Address = 10.2.2.2
(1) Remote IP Address = 10.1.1.1

- Remote X.509 Certificate
Subject's AltNames:
(4) Email: mvsnm2 at washington.ibm.com

- Remote Policy Configuration
Local IKED Identity
Userid@FQDN =
(4) mvsnm2@washington.ibm.com
Local data endpoint
(2) 10.2.2.2
Remote IKED Identity
Userid@FQDN =
(3) mvsnm1@washington.ibm.com
Remote data endpoint
(1) 10.1.1.1
Key Exchange Rule
RSA Signature

IKED Identity AltName Domain

- IP Header of Packet

Local IP Address = 10.1.1.1 (1)

Remote IP Address = 10.2.2.2 (2)

- Local X.509 Certificate

Subject's AltNames:

Domain: washington.ibm.com (3)

- Local Policy Configuration

Local IKED Identity

FQDN = washington.ibm.com (3)

Local data endpoint

10.1.1.1 (1)

Remote IKED Identity

FQDN = gbg.lab.ibm.com (4)

Remote data endpoint

10.2.2.2 (2)

Key Exchange Rule

RSA Signature

- IP Header of Packet

(2) Local IP Address = 10.2.2.2

(1) Remote IP Address = 10.1.1.1

- Remote X.509 Certificate

Subject's AltNames:

(4) Domain: gbg.lab.ibm.com

- Remote Policy Configuration

Local IKED Identity

(4) **FQDN** = gbg.lab.ibm.com

Local data endpoint

(2) 10.2.2.2

Remote IKED Identity

(3) **FQDN** = washington.ibm.com

Remote data endpoint

(1) 10.1.1.1

Key Exchange Rule

RSA Signature

IPSec Manual Tunnels

- Local IP Address = 10.1.1.1 (1)

Remote IP Address = 10.2.2.2

- ## Local data endpoint

10.1.1.1 (1)

Local Encryption SPI

256 (3)

Local Encryption key (4)

[illegible]

Local Authentication SPI

256 (3)

Local Authentication key (5)

48BBAAFF22116698EEAA322222222222

Remote data endpoint

10.2.2.2 (2)

Remote Encryption SPI

342 (6)

Remote Encryption key(7)

48BBAAFF22116698000000000000000000000000000000000000

Remote Authentication SPI

342 (6)

Remote Authentication key (8)

48BBAAFF22116698FFFFFFFFFFFFFFFF

- Local IP Address = 10.2.2.2

Remote IP Address = 10.1.1.1

- ## Local data endpoint

2) 10.2.2.2

Local Encryption SPI

6 342

Local Encryption key

(7) 48BBAAFF2211669800000000000000000000000000000000

Local Authentication SPI

6 342

Local Authentication key

8 48BBAAFF22116698FFFFFFFFFFFFFFFF

Remote data endpoint

① 10.1.1.1

Remote Encryption SPI

3 256

Remote Encryption key

(4) 48BBAAFF2211669833333333333333333333333333422111111

Remote Authentication SPI

3 256

Remote Authentication key

5) 48BBAAFF22116698EEAA322222222222

If ESP protocol is being used then SPI (Security Parameter Index) for Encryption and Authentication must match.

IPsec Steps

- Provide x.509 certificates if deploying IPsec with RSA Signature Mode.
- Build IPsec policies with IBM Configuration Assistant
- Implement PAGENT, SYSLOGD, and TRMD on z/OS
- Implement IKED for dynamic VPNs
- Enable IPsec in the TCP/IP Stack:
 - Set IPCONFIG IPSECURITY in PROFILE.TCPIP.
 - Optionally establish IPSECRULEs in TCP/IP Profile to override the Default Implicit "denyall" rule that is in effect until a set of PAGENT IPsec policy rules is activated.
- IKE daemon configuration file search order
 - The MVS data set or z/OS UNIX file specified by IKED_FILE environment variable
 - /etc/security/iked.conf
- Sample IKE daemon configuration file
 - /usr/lpp/tcpip/samples/iked.conf
- Reserve ports for IKED in the TCP/IP profile
 - PORT
500 UDP IKED
4500 UDP IKED
- Open Firewalls if necessary:
 - Ports
 - Protocols (UDP; ESP=50; AH=51; etc)
- IKED sample proc
 - TCPIP.SEZAINST(IKED)

Activate IPsec VPN

The screenshot shows the IBM z/OS Management Facility Configuration Assistant interface. The left sidebar contains navigation links: Welcome, Notifications, Workflows, Configuration (with sub-link Configuration Assistant), Links, z/OSMF Administration, and z/OSMF Settings. The main content area is titled 'Advanced Connectivity Rule Settings' and includes tabs for Activation, Encapsulation/OnDemand, Key Exchange, Remote Security Endpoint, Logging, and General. The 'Activation' tab is selected, showing 'Dynamic Tunnels: How to Activate' instructions. Below this is a table with columns: Traffic Descriptor, Protocol, Local Port, Remote Port, Connect Direction, IPsec Security Level, Allow Remote Activation, Allow On Demand Activation, Auto Activate, and ipsec Command Activation. The table lists six entries for FTP-Clients and FTP-Servers with various port ranges and activation settings.

Traffic Descriptor	Protocol	Local Port	Remote Port	Connect Direction	IPsec Security Level	Allow Remote Activation	Allow On Demand Activation	Auto Activate	ipsec Command Activation
FTP-Client	TCP	All Ephemeral	21	Outbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP-Client	TCP	All Ephemeral	20	Inbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP-Client	TCP	All Ephemeral	50000-50200	Outbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP-Server	TCP	21	All Ephemeral	Inbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP-Server	TCP	20	All Ephemeral	Outbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP-Server	TCP	50000-50200	All Ephemeral	Inbound	IPSec__Gold	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Specify when IPsec VPN will be activated:
 - Remote end causes activation
 - Activate this end when a filter applies (on-demand)(default)
 - Activate automatically at IPsec Policy initialization
 - Activate manually with 'ipsec' command

FIPS 140

- Configure the IKED, the NSSD, and the TCP/IP stack components to operate in FIPS 140 mode.
- Configure FIPS 140 mode in System SSL
 - Places restrictions on the cryptographic algorithms and key lengths that can be used for IP security.
 - See z/OS Cryptographic Services System Secure Sockets Layer Programming, SC14-7495, for the latest support.
- Recommendation: Initialize ICSF in FIPS 140 mode.

Enterprise Extender (EE) IPsec Performance

- z/OS V1R11 Improved performance for EE over IPsec
 - The “bursty” nature of HPR traffic can cause significant performance degradation when it is carried over IPsec tunnels.
 - Smaller bursts frequently get sent before larger bursts. This results in out-of-order segments that are dropped, forcing retransmits.
 - Primarily observed for streaming traffic over EE.
 - Very large volumes of SNA interactive traffic (such as tens of thousands of sessions that may appear similar to bulk data traffic) are likely to see issues too.
 - V1R11 breaks large bursts into batches of smaller bursts.
- z/OS V1R11 support for EE over IPsec offloaded to a zIIP
 - Support for offloading outbound EE over IPsec traffic to a zIIP processor.
 - Previously only inbound EE over IPsec traffic was processed on the zIIP.
- Protecting both interactive and streaming EE workload with IPsec is now fully recommended.

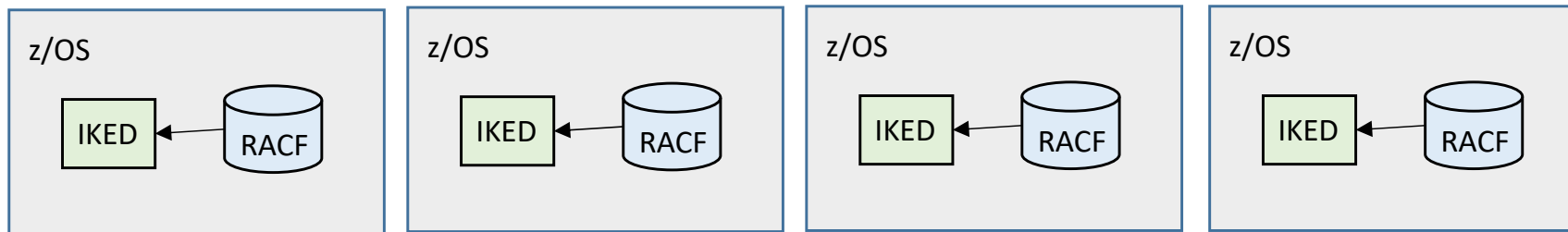
ipsec Command

- ipsec command SERVAUTH profile
 - EZB.IPSECCMD.sysname.stackname.command_type
 - SETROPTS GENERIC(SERVAUTH)
 - RDEFINE SERVAUTH EZB.IPSECCMD.sysname.tcpprocname.* UACC(NONE)
 - PERMIT EZB.IPSECCMD.sysname.tcpprocname.* CLASS(SERVAUTH)
 - ID(userid) ACCESS(READ)
 - SETROPTS GENERIC(SERVAUTH) REFRESH
- ipsec command options
 - -f for IP Filter
 - -F for Defensive Filter
 - -m for Manual Tunnel
 - -k for IKE Tunnel
 - -y for Dynamic Tunnel
 - -i for Interface
 - -t for IP Traffic Test
 - -o for NATT Port Translation
 - -w for IKED Network Security
 - -x for Network Security Server
 - -?

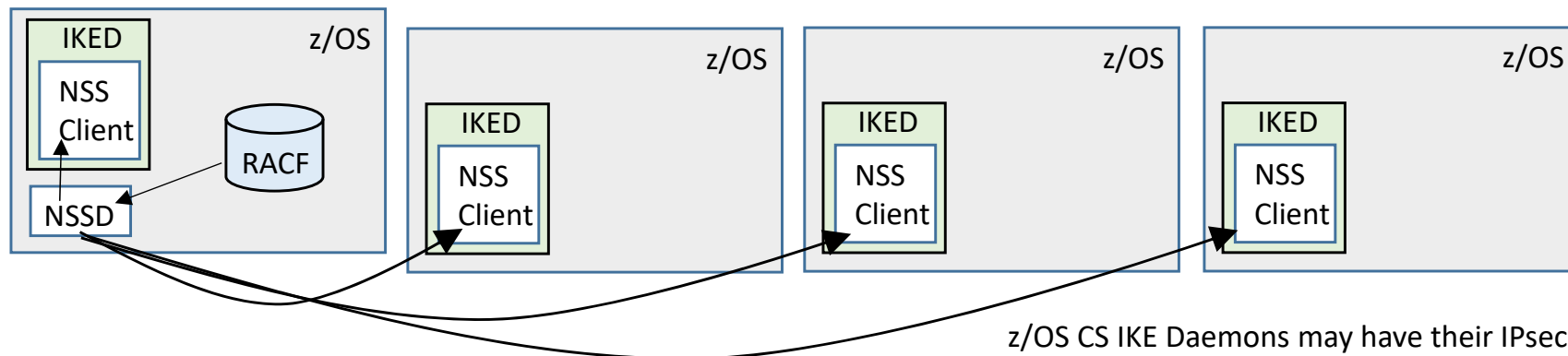
Network Security Services



Network Security Services



Without NSSD all z/OS CS IKE Daemons have their own key ring repository.



z/OS CS IKE Daemons may have their IPsec certificates stored on a single system.

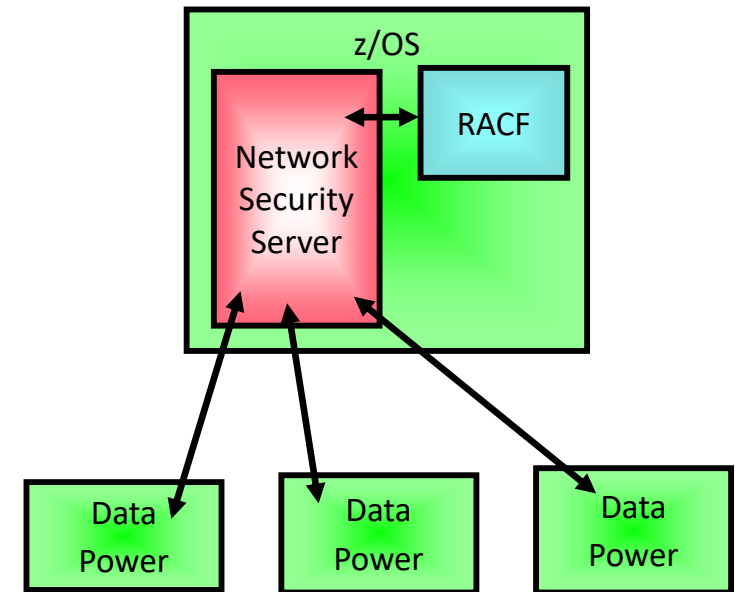
- Network Security Services Daemon (NSSD)
 - Centralized certificate services
 - Monitoring and management for IPsec security for z/OS systems within and across sysplexes
- IKE Daemon may be configured as a Network Security Client.
 - Configuration is on a per-stack basis
 - Each NSS-enabled stack will appear to the Network Security Server as an independent client.
 - For TCP/IP stacks that are not configured to use Network Security Services, the IKE daemon will continue to manage certificates out of a local keyring.
- NSSD required for IKEv2 support.

IKEv2 concurrently supported with IKEv1 in IKED.
Any individual connection uses one or the other.

IKEv2 Enhancements

- Reduced bandwidth used by control messages
 - This leaves increased bandwidth available for data transmission.
- Extensible Authentication Protocol (EAP) authentication
 - In addition to IKEv1 pre-shared key and certificate authentication
 - EAP is a protocol for wireless networks that expands on authentication methods used by the Point-to-Point Protocol (PPP), a protocol often used when connecting a computer to the Internet. EAP can support multiple authentication mechanisms, such as token cards, smart cards, certificates, one-time passwords, and public key encryption authentication.
- MOBIKE support
 - Required for mobile platforms like phones and by users with multi-homed setups. MOBIKE allows a host that has multiple simultaneous points of attachment to a network to change which interface is forwarding traffic while maintaining a VPN session.
- Network Address Translation (NAT) traversal support
 - NAT in routers between the two endpoints is supported.
- Liveness check
 - Detects whether the tunnel is still alive or not. If the tunnel is down, IKEv2 is able to re-establish the connection automatically.

DataPower NSSD Support



- Network Security Services Daemon z/OS XMLAppliance Discipline
 - Data Power are non-System z XML appliances that may receive Web Service requests, parse and transform the XML messages, and forward them on to WebSphere Application Servers (WAS).
 - <https://www.ibm.com/products/datapower-gateway>
 - NSSD API provides an interface for the remote Data Power boxes to SAF access services.
 - When WAS is on z/OS, zLinux, or even non-System z.
 - nssctl command for monitoring

RACF for NSSD

- NSSD userid requires UID 0.
- Define IRR.DIGTCERT FACILITY class resources in RACF if they do not already exist and PERMIT NSSD administrator userid to IRR.DIGTCERT FACILITY class resources.
- Define NSS client userid and password.
- Optionally define an NSSD profile in the APPL class with UACC(NONE) and issue PERMIT to authorize each NSS client.
- Define SERVAUTH profiles to authorize NSS clients.
 - Create SERVAUTH resource profile for NSS Service, and PERMIT READ access for the NSS clients.
 - IPsec certificate service EZB.NSS.sysname.clientname.IPSEC.CERT
 - IPsec certificate service EZB.NSS.sysname.clientname.IPSEC.NETMGMT
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.CERT
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.PRIVKEY
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.SAFACCESS
 - Create SERVAUTH resource profile for each NSS client certificate and CA cert added to NSSD key ring, and PERMIT NSS client to profile for their own certificates.
 - EZB.NSSCERT.sysname.mappedlabelname.HOST
 - EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH
 - Create SERVAUTH resource profile for each **XMLAppliance** client certificate (HOST or CERTAUTH as documented above) and private key, and PERMIT XMLAppliance clients to profile for their own certificates.
 - EZB.NSSCERT.sysname.mappedlabelname.PRIVKEY
 - Create SERVAUTH resource profile for remote monitor (IPSEC.DISPLAY) and manage (IPSEC.CONTROL) NSS clients.
 - EZB.NETMGMT.sysname.clientname.IPSEC.DISPLAY
 - EZB.NETMGMT.sysname.clientname.IPSEC.CONTROL

RACF for NSSD (cont.)

- If using **XMLAppliance** with ICSF-protected private keys, authorize NSS server to ICSF (CSFSERV).
 - ICSF is required for RSA operations in private key service.
 - When using cryptographic coprocessor, the callable ICSF service names are:
 - CSNDDSG
 - CSNDPKD
- If **XMLAppliance** clients using the SAF access service are using certificates for access checks, enable RACF certificate name filtering.
 - RACF certificate name filtering maps an X.500 distinguished name to a userid when performing SAF access checks.
- NSSD uses ICSF callable services for ECDSA digital signature support.
 - Services it uses are the PKCS11 private key sign service and the PKCS11 public key verify service.
 - Use the CSFSERV general resource class, and the CSF1PKS and CSF1PKV profiles.
 - PERMIT NSSD userid READ access to the profiles.

Configure and Modify NSSD

- Create NSSD configuration file.
 - Use z/OS Configuration Assistant
 - Or use sample /usr/lpp/tcpip/samples/nssd.conf
 - Search Order
 - Environment variable NSSD_FILE
 - /etc/security/nssd.conf
- Create NSSD key ring.
- Define NSSD port to PROFILE.TCPIP.
 - Default is TCP port 4159.
- Create AT-TLS policies to protect NSSD traffic to NSS clients.
- Create NSSD JCL procedure or start from Unix.
 - Sample proc in SEZAINST(NSSD)
 - Only one NSSD per z/OS.
- Modify
 - Flush all cached URLs and reread the NSS server configuration file.
 - MODIFY procname,REFRESH
 - MODIFY NSSD,REFRESH,FILE='/etc/security/nssd.conf2'
 - Display the configuration file parameters.
 - MODIFY procname,DISPLAY
 - Display the contents of the URL cache.
 - MODIFY procname,DISPLAY,URLCACHE

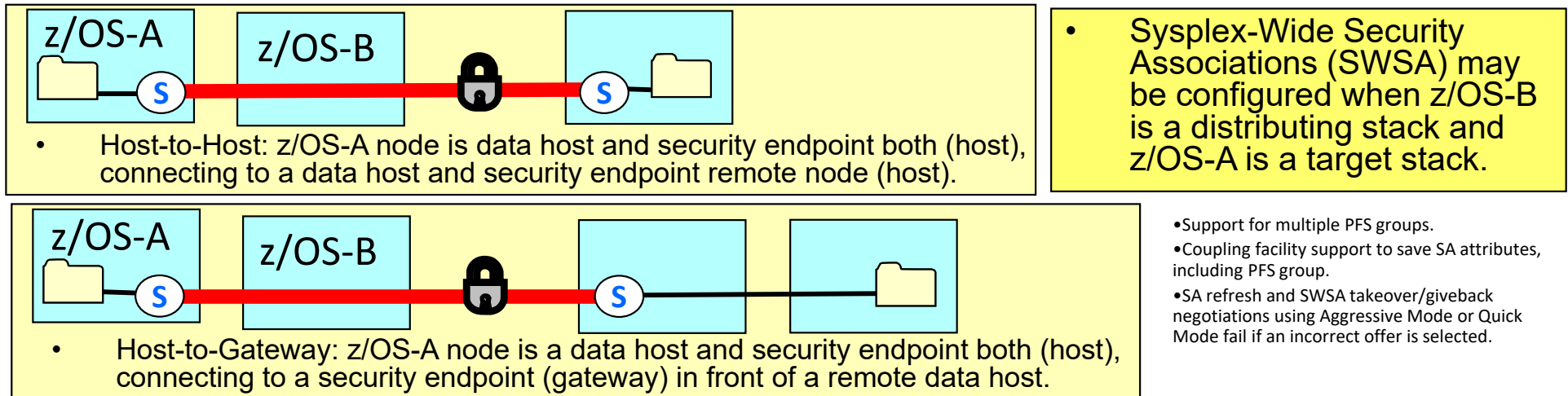
NSSD Display Commands

- Display NSS IPsec client connection information.
 - `ipsec -x display`
- Display connected NSS IPsec clients.
 - `nssctl -d`
- Filter ipsec command by client.
 - `ipsec -y display -z client4`

Sysplex Wide Security Associations



IPsec for Distributed DVIPA (Sysplex Distributor)



- Existing connections to target systems will be preserved if distributing stack fails and backup stack takes over distribution.
 - When a DVIPA is moved during DVIPA takeover (planned or unplanned), SWSA automatically reestablishes new IPsec SAs with the same security service characteristics as the SAs that existed on the host that previously owned the DVIPA. The SA reestablishment is transparent to the client that owns the other end of the SA. That is, the SA reestablishment looks like a normal SA refresh.
- The distributing stack(s) require: IPCONFIG IPSECURITY, IKED, IPCONFIG IPSECURITY, the filter/VPN rules for the IPsec traffic to/from the distributed DVIPA.
- Because the outbound traffic does not necessarily pass through the distributing stack SWSA has additional requirements on the target stacks: identical filter/VPN rules for the IPsec traffic on the target stacks as what is configured on the distributing stack(s). Stacks that are only distributed DVIPA targets do not need IKE, key exchange rules, or DVIPSEC.
- SWSA also requires the use of a coupling facility structure with a name in the form EZBDVIPAvvtt, where vv is the 2-digit VTAM group ID suffix specified on the XCFGRPID start option, and tt is the TCP group ID suffix specified on the GLOBALCONFIG statement in the TCP/IP profile.

End of Topic



End of Topic

