OS/390 Firewall Technologies Overview

Paul de Graaff ITSO Poughkeepsie S/390 Security

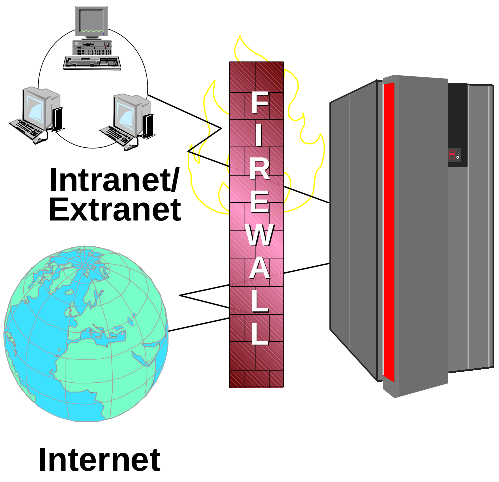
Agenda

- History
- OS/390 V2R7 Enhancements
- OS/390 V2R8 Enhancements
  - Dynamic Virtual Private Networks (internet Key Exchange)

History

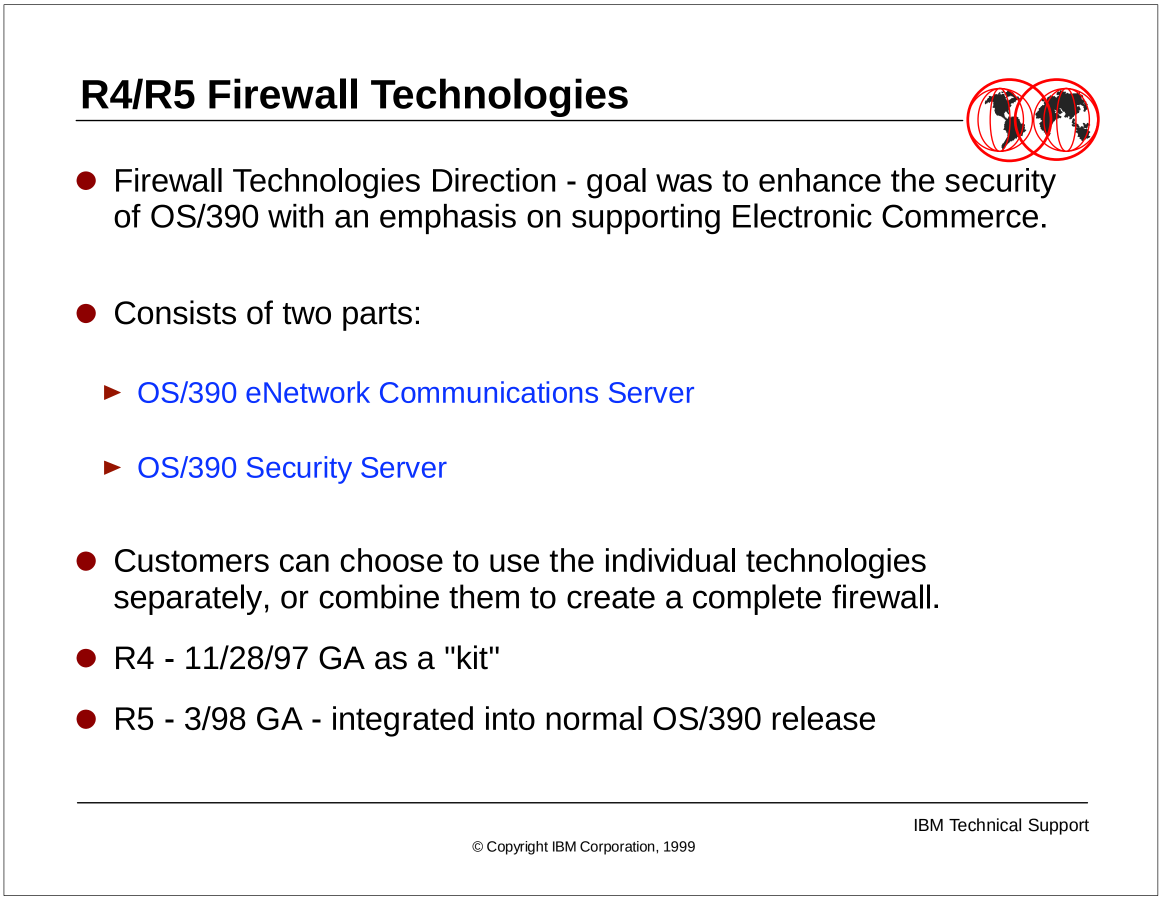
Firewall Technologies

Network Level



Firewall Technologies

- IP Filtering
- Network Address Translation (NAT)
- Virtual Private Network (VPN) with Crypto HW
- Proxy servers
- SOCKS server
- Domain Name services



R4/R5 Firewall Technologies

- Firewall Technologies Direction - goal was to enhance the security of OS/390 with an emphasis on supporting Electronic Commerce.
- Consists of two parts:
  - OS/390 eNetwork Communications Server
  - OS/390 Security Server
- Customers can choose to use the individual technologies separately, or combine them to create a complete firewall.
- R4 - 11/28/97 GA as a "kit"
- R5 - 3/98 GA - integrated into normal OS/390 release

OS/390 eNetwork Communications Server

- Packet Filtering
  - Works at the IP layer of TCP/IP and examines all IP packets.
  - Can filter on: IP address, adapter, direction, protocol, port, routing and date/time.
- IPSec (VPN/Tunnels)
  - Also known as Virtual Private Network (VPN) or tunneling.
  - Provides a secure channel for data across a network.
  - Supports IPSec RFCs 1825-1829 with KEYED_MD5 authentication and CDMF/DES encryption.
  - Key Management is done manually.
  - S/390 hardware cryptographic facility is used when available.
- Network Address Translation (NAT)
  - Provides translation from an internal IP address to an external address and vice versa.

OS/390 Security Server

- FTP Proxy/Socks Daemon
  - Intercept traffic at the TCP layer of the stack and checks authorization to come into or go out of the secure network.
  - FTP Proxy authorization is via an External Security Manager userid and password; Socks Server authorization is via socks rules.
  - The FTP Proxy only operates on traffic between an ftp client and an ftp server.
  - The Socks Server is not application specific but does require that the clients be "socksified". Currently OS/390 does not have any "socksified" client support.
- Enhanced Syslog Daemon
  - Replaces existing syslogd - more robust.
  - Uses message queues rather than UDP.

OS/390 Security Server *(continued)*

- Configuration and Administration commands
  - OE commands provided to configure and administer the OS/390 Firewall Technologies.
  - Compatible with AIX Firewall.

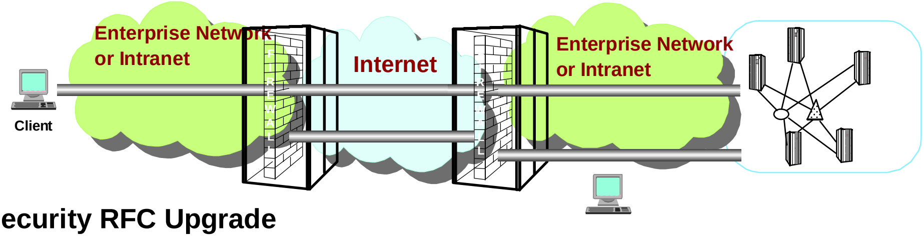
Release 6 Firewall Technologies

- OS/390 R6 - GA 9/98
- OS/390 eNetwork Communications Server
  - Performance/RAS
    - APAR released for R4/R5
- OS/390 Security Server
  - Performance/RAS
    - Added multi-threading support to FTP Proxy (APAR released for R4/R5).
  - VPN
    - Transport mode added to provide host to host tunneling.

OS/390 V2R7 Enhancements

IBM Technical Support

IPSec Enhancements (V2R7)


Enterprise Network or Intranet
Internet
Enterprise Network or Intranet
Client

IP Security RFC Upgrade

- Supports latest RFCs (2401-2406, 2410)
  - Maintains interoperability with previous IPSec RFC levels
- Increased security
  - Replay protection added
  - Improved authentication algorithms (HMAC-MD5, HMAC-SHA)

Strong Encryption

- Triple DES encryption
  - Exploits hardware S/390 cryptographic coprocessor

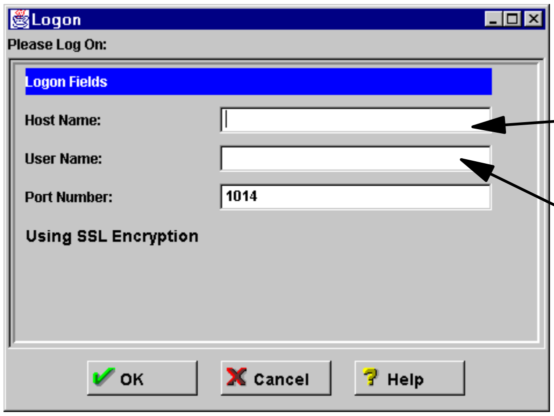
Configuration improvements

- Client-to-Server security associations
- JAVA-based GUI for IPSec configuration available with Security Server
  - Also configurable through UNIX command line

Release 7 Firewall Technologies

- OS/390 Security Server
  - IPSec Upgrade Configuration
    - fwtunnl command enhancements
  - Configuration GUI
    - Provides end user GUI interface to configure/manage the firewall.
    - Java based application with HTML help panels; runs on AIX and WIN NT.
    - Configuration Server
    - SSL used to protect data flowing from client to server.
    - RACF authenticated userids.
  - Multi-stack support
    - Allow Filter/Tunnel/NAT support to be executed on more than one stack.
  - Scalability
    - Remove thread limitations for FTP and Socks daemons
    - Configuration command added to configure daemons

FW GUI



Logon

Please Log On:

Logon Fields

Host Name:

User Name:

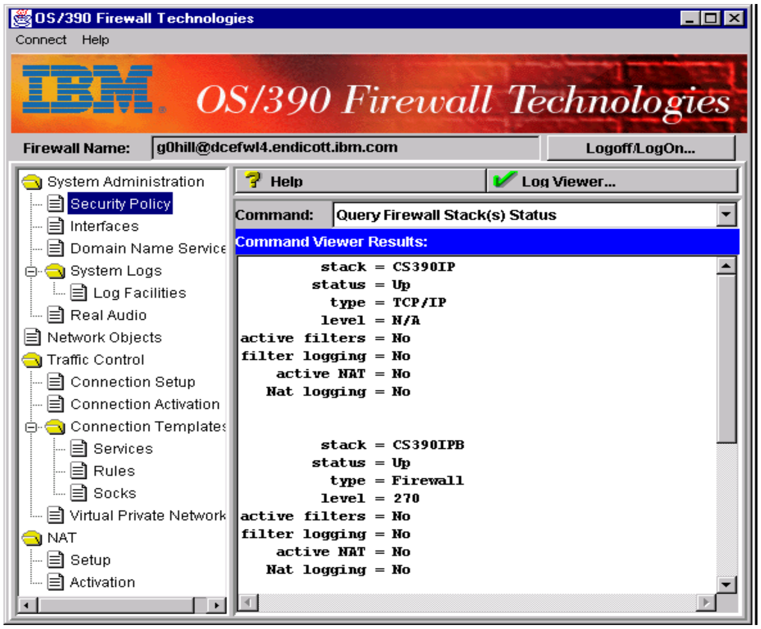
Port Number: 1014

Using SSL Encryption

Domain name or IP address

RACF Userid

JAVA GUI



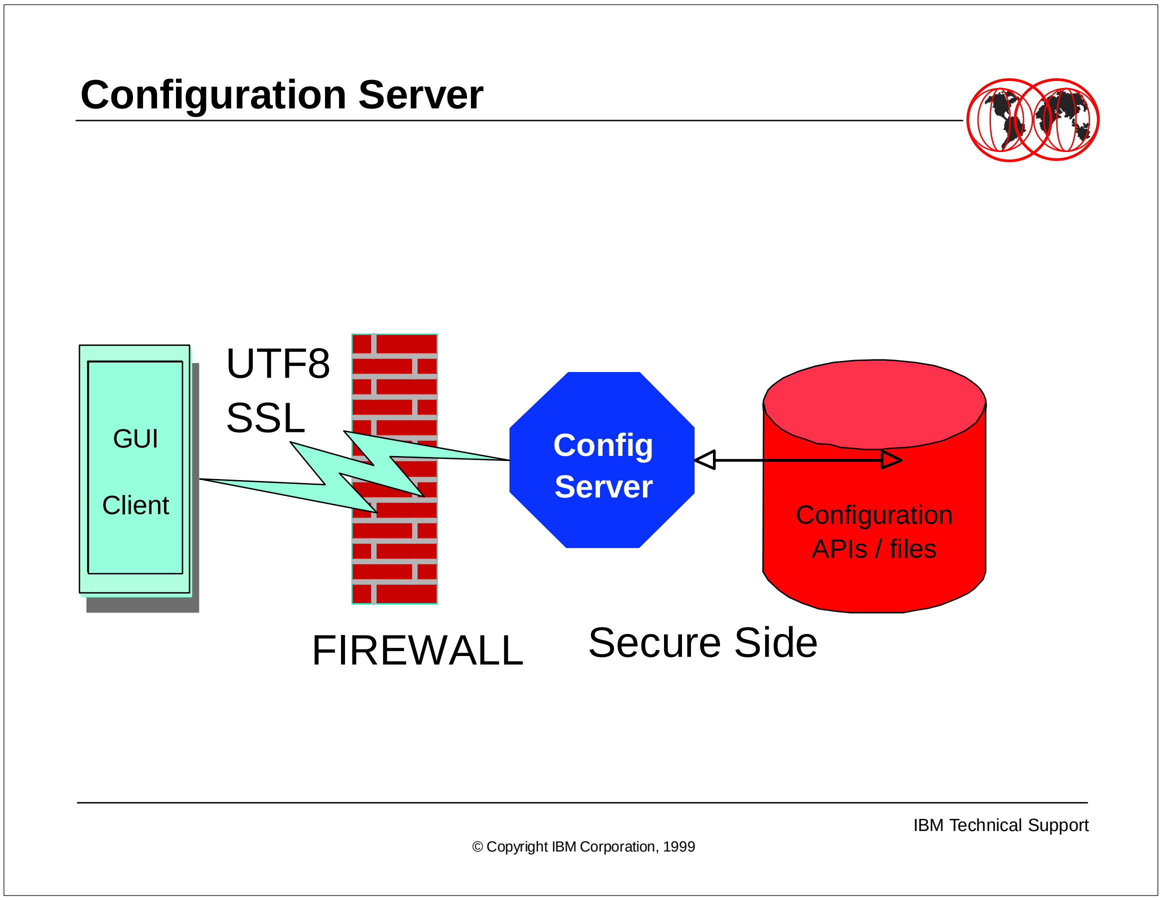
Firewall Name: g0hill@dcefwl4.endicott.ibm.com

Command: Query Firewall Stack(s) Status

```
         stack = CS390IP
        status = Up
          type = TCP/IP
         level = N/A
active filters = No
filter logging = No
    active NAT = No
   Nat logging = No


         stack = CS390IPB
        status = Up
          type = Firewall
         level = 270
active filters = No
filter logging = No
    active NAT = No
   Nat logging = No
```

- Main Panel
  - This is the main panel. I expanded all the tree structure on the left and the remaining screen captures pretty much following this order in the tree.
  - Notice the help (available on must panels).
  - The Log Viewer showing logging information (not shown in a screen capture).
  - The Command View lets you issue commands on the OS/390 Firewall and the results are displayed. ONLY SELECTED COMMANDS CAN BE PICKED.



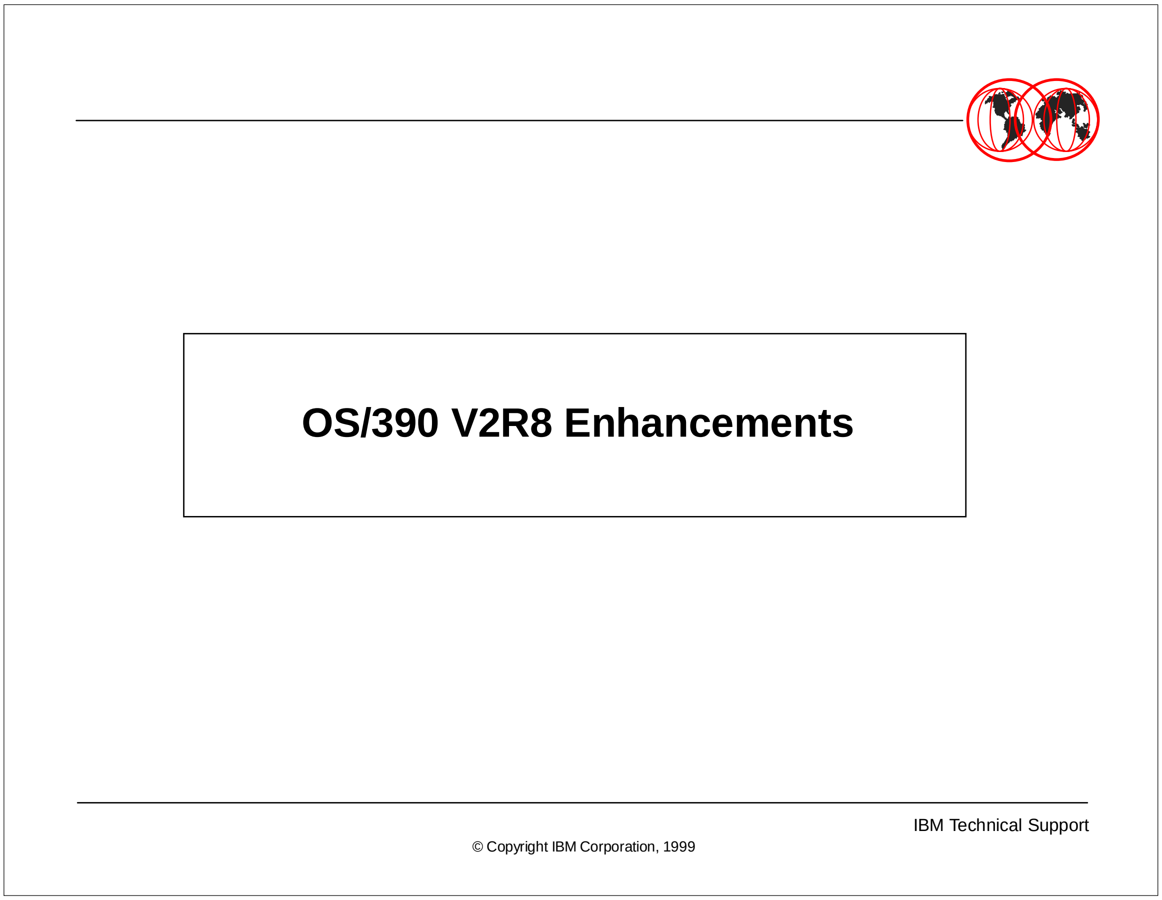
Configuration Server

IBM Technical Support

- Configuration Server
  - This new Firewall daemon communicates with the GUI client and the configuration database storage APIs.
  - Note: These same APIs are used by the command line so that the data is consistent between GUI and Command Line.

- Data from the GUI flows in UTF-8 format (unicode variation) and is encrypted using SSL (Security Sockets Layer).

- The configuration server authenicates and checks authorization. Uses External Security Manager (e.g. RACF).



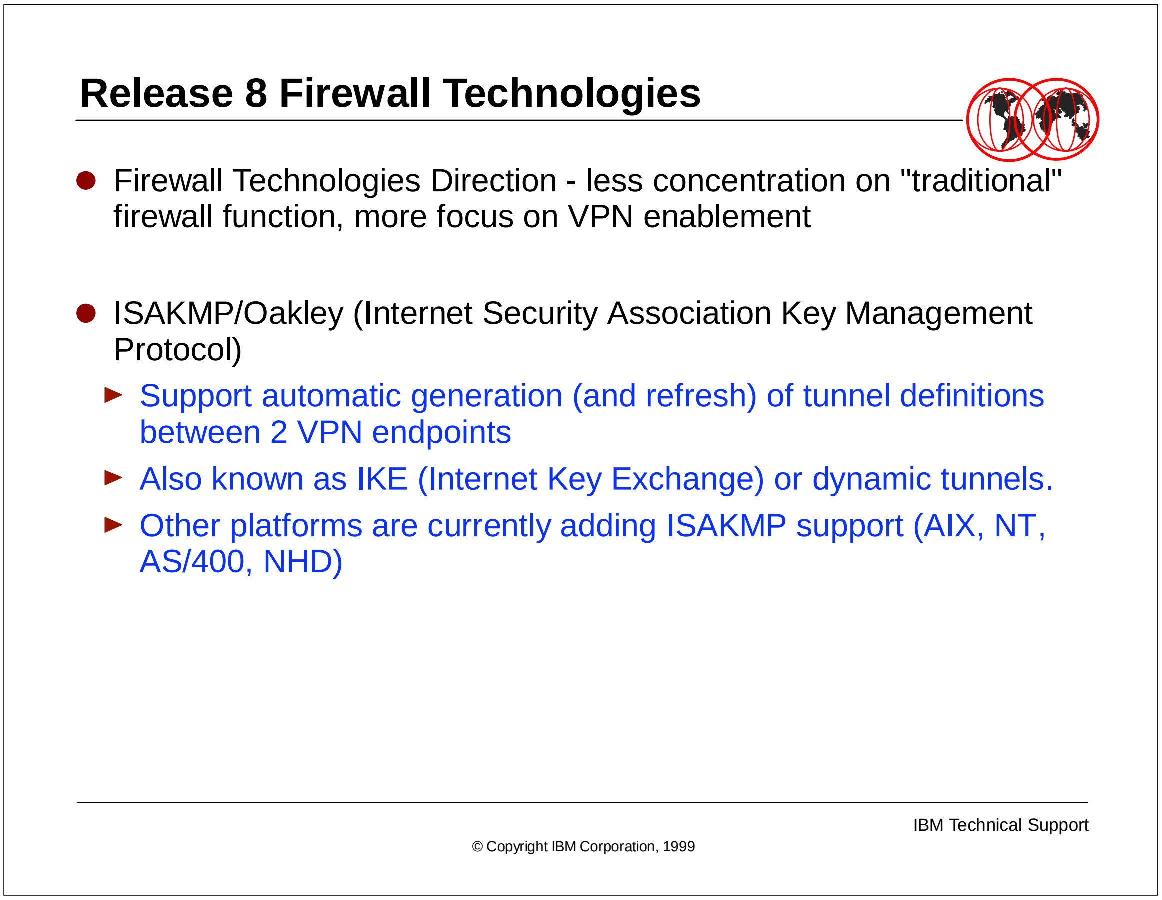
OS/390 V2R8 Enhancements

IBM Technical Support

OS/390 Firewall Technology Delivery in V2R8

- OS/390 eNetwork Communication Server
  - IP Packet Filtering
  - IP Security (VPN)
  - Network Address Translation (NAT)

- OS/390 Security Server
  - Internet Security Association Key Management Protocol (ISAKMP) Support
  - FTP Proxy support
  - Socks Server (daemon) support
  - Enhanced logging
  - Command Line Configuration/Administration
  - GUI Configuration/Administration



Release 8 Firewall Technologies

- Firewall Technologies Direction - less concentration on "traditional" firewall function, more focus on VPN enablement

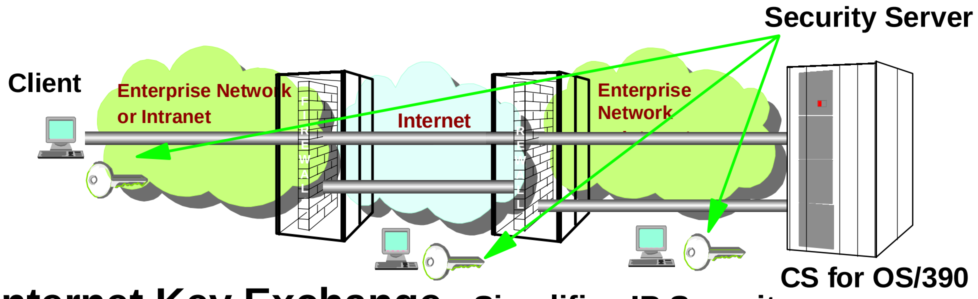
- ISAKMP/Oakley (Internet Security Association Key Management Protocol)
  - Support automatic generation (and refresh) of tunnel definitions between 2 VPN endpoints
  - Also known as IKE (Internet Key Exchange) or dynamic tunnels.
  - Other platforms are currently adding ISAKMP support (AIX, NT, AS/400, NHD)

Release 8 Firewall Technologies *(continued)*

- OS/390 eNetwork Communications Server
  - Enhancements being made in support of dynamic tunnels.

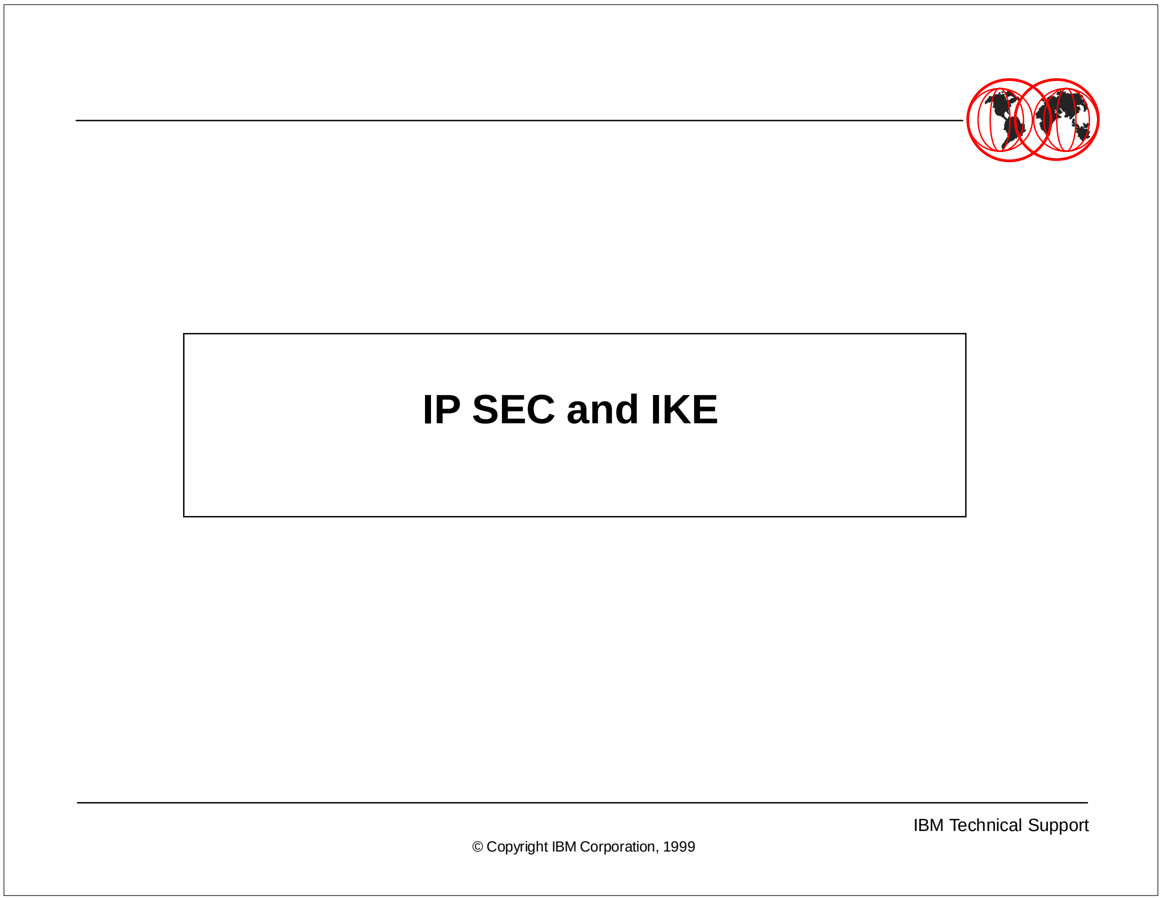
- Security Server
  - ISAKMP Daemon
    - Handles negotiation with remote system.
  - GUI/Commands
    - Support definition of dynamic tunnels.
  - Crypto/Certificates
    - Provide authentication/encryption necessary to securely negotiate tunnel definition.
  - VPN Manager
    - Manage interactions between ISAKMP/GUI/Stack.

Key Management (V2R8)

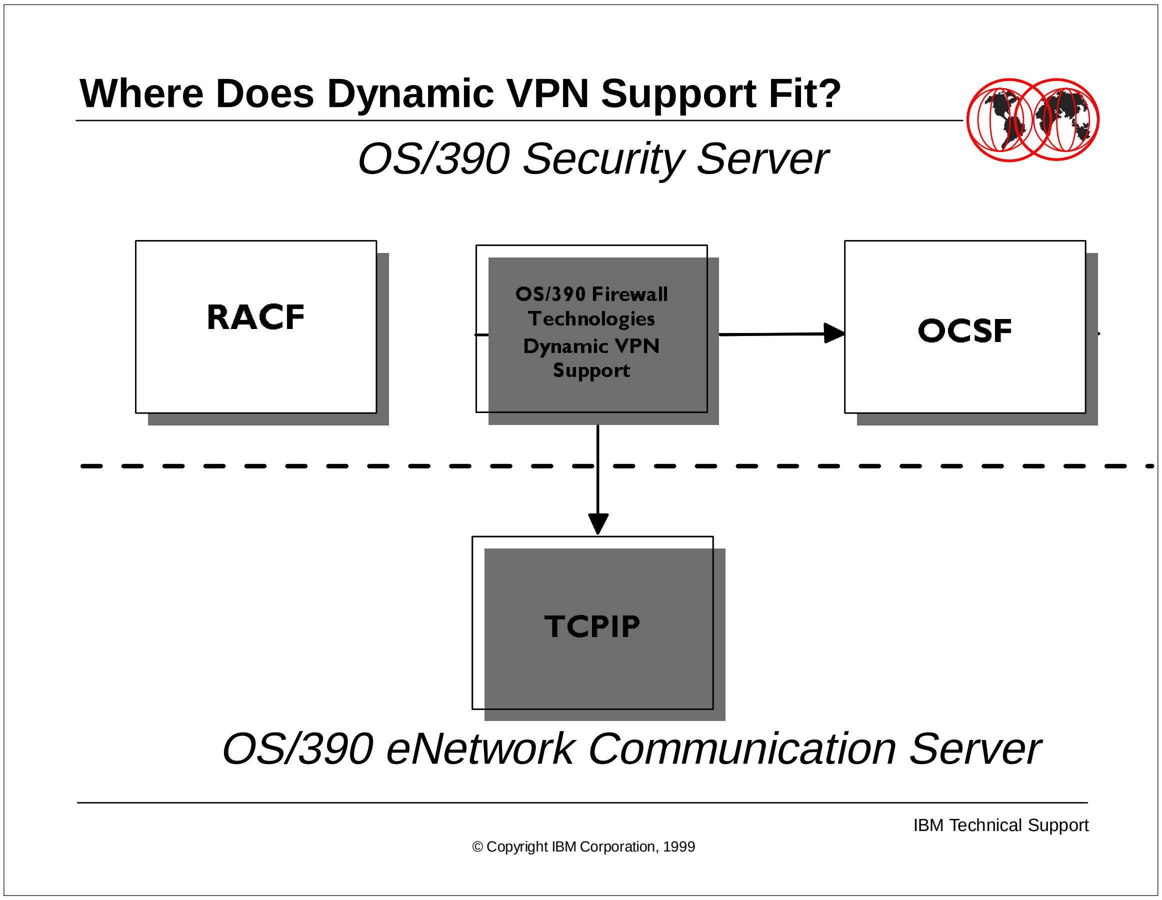


Internet Key Exchange - Simplifies IP Security

- Secure exchange of keys
- Reduces manual configuration
  - Dynamic tunnels
  - A critical element as VPNs grow
- Enables non-disruptive key refresh
- Enables network access with dynamic IP addresses
- Joint offering between the Communications and Security Servers for OS/390



IP SEC and IKE



Where Does Dynamic VPN Support Fit?

OS/390 Security Server

RACF

OS/390 Firewall Technologies Dynamic VPN Support

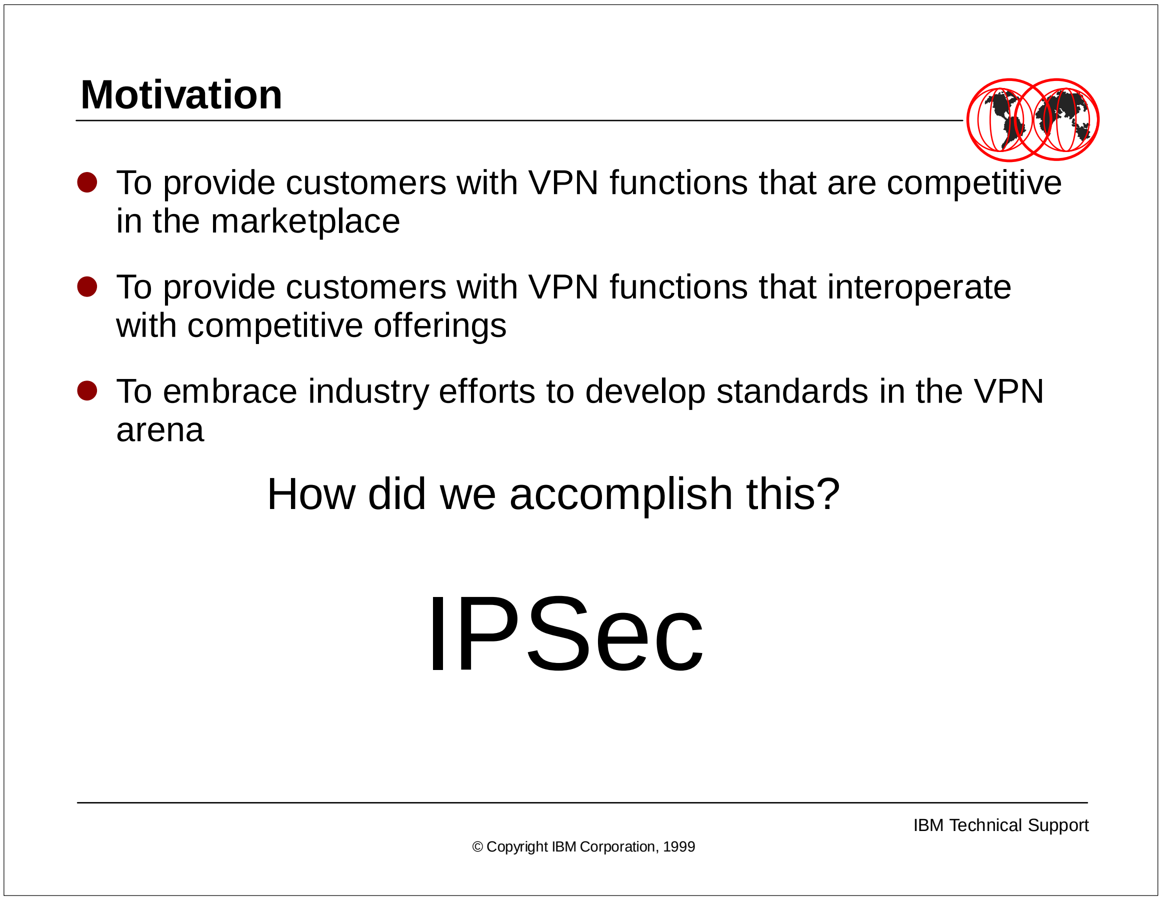
OCSF

TCPIP

OS/390 eNetwork Communication Server

IBM Technical Support

- This diagram indicates where dynamic VPN support fits into OS/390. The highlighted boxes indicate components that were modified in order to support dynamic VPNs. The unhighlighted boxes indicate OS/390 Security Server components that portions of the dynamic VPN support is dependent on. The arrows indicate dependencies.
- The OS/390 Firewall Technologies box includes a new daemon, ISAKMPD, to handle dynamic key management, dynamic attribute negotiation, and the creation of dynamic VPNs. It also includes the configuration client, configuration server, and configuration commands required to support dynamic VPNs. Everything in this box is shipped as part of the OS/390 Security Server.
- The TCPIP box includes changes to facilitate new structures and processing associated with dynamic VPNs. These changes are shipped as part of eNetwork Communication Server.
- The ISAKMP daemon utilizes RACF and OCSF functions.
- RACF is used to generate PKCS 10 certificate requests and to store the ISAKMP daemon's certificates. RACF supports storing private keys owned by the ISAKMP daemon in the Integrated Cryptographic Service Facility (ICSF) Public Key Data Store (PKDS).
- OCSF is used to verify these certificates as well as to perform the necessary public key encryption/decryption functions to support signing operations.



Motivation

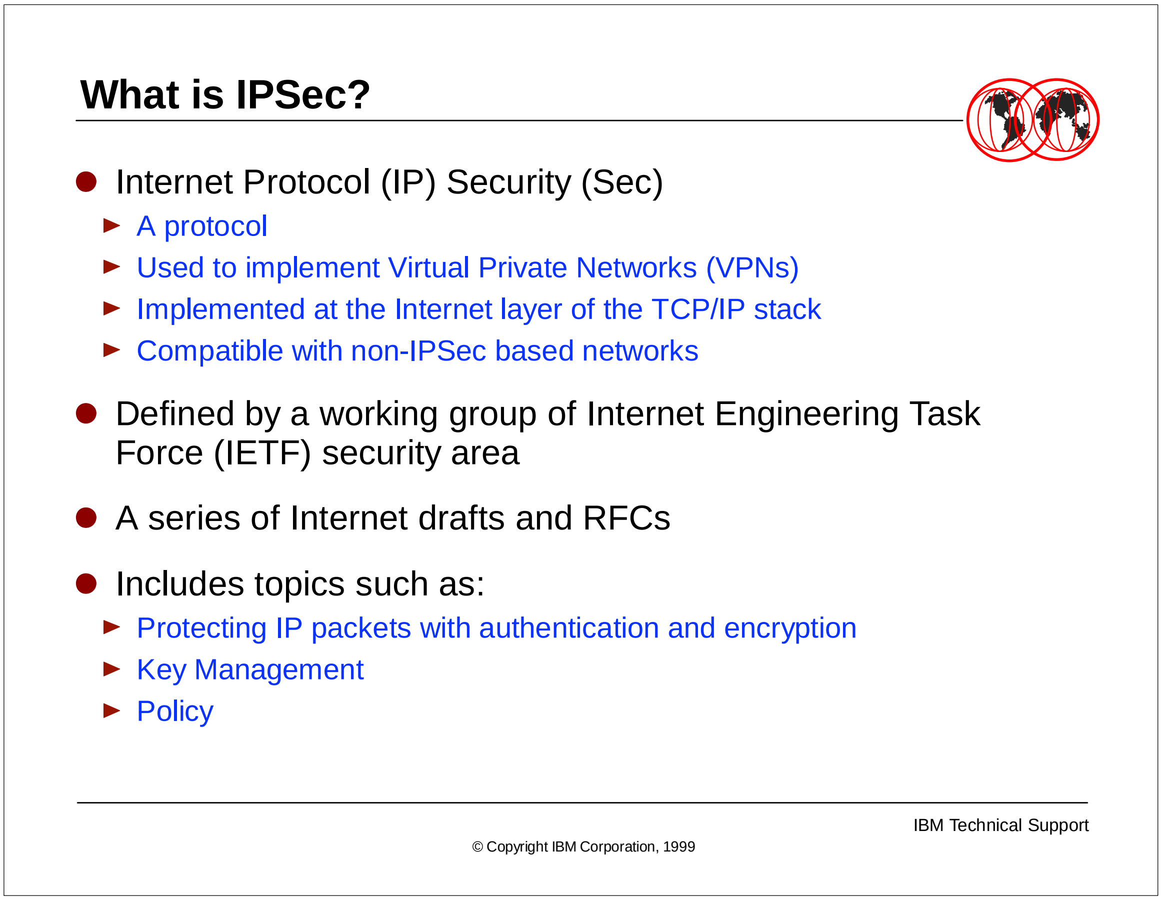
- To provide customers with VPN functions that are competitive in the marketplace
- To provide customers with VPN functions that interoperate with competitive offerings
- To embrace industry efforts to develop standards in the VPN arena

How did we accomplish this?

IPSec

IBM Technical Support

- The motivation behind this support is to provide OS/390 customers with a VPN implementation that is equivalent with VPN implementations being delivered by other vendors on other platforms. Today's marketplace is such that our customers have to deal with heterogeneous networks, whether it be their own private intranet or the Internet. Providing a VPN implementation that is functionally equivalent to the competition is not sufficient. Customers require a VPN implementation that works with other vendor's implementations. The best opportunity to achieve interoperabilty in the VPN arena is by supporting efforts to create standards.
- IPSec has emerged as the industry standard in the area of VPNs. IPSec addresses the needs of our customers and is widely implemented in the industry. As such our VPN enhancements are based on IPSec RFCs.
- In order to really understand what we did and why, it is beneficial to be familiar with IPSec. A presentation that provides a detailed overview of IPSec is provided in the appendix. The rest of this section will provide a quick summary of IPSec functions implemented on OS/390.

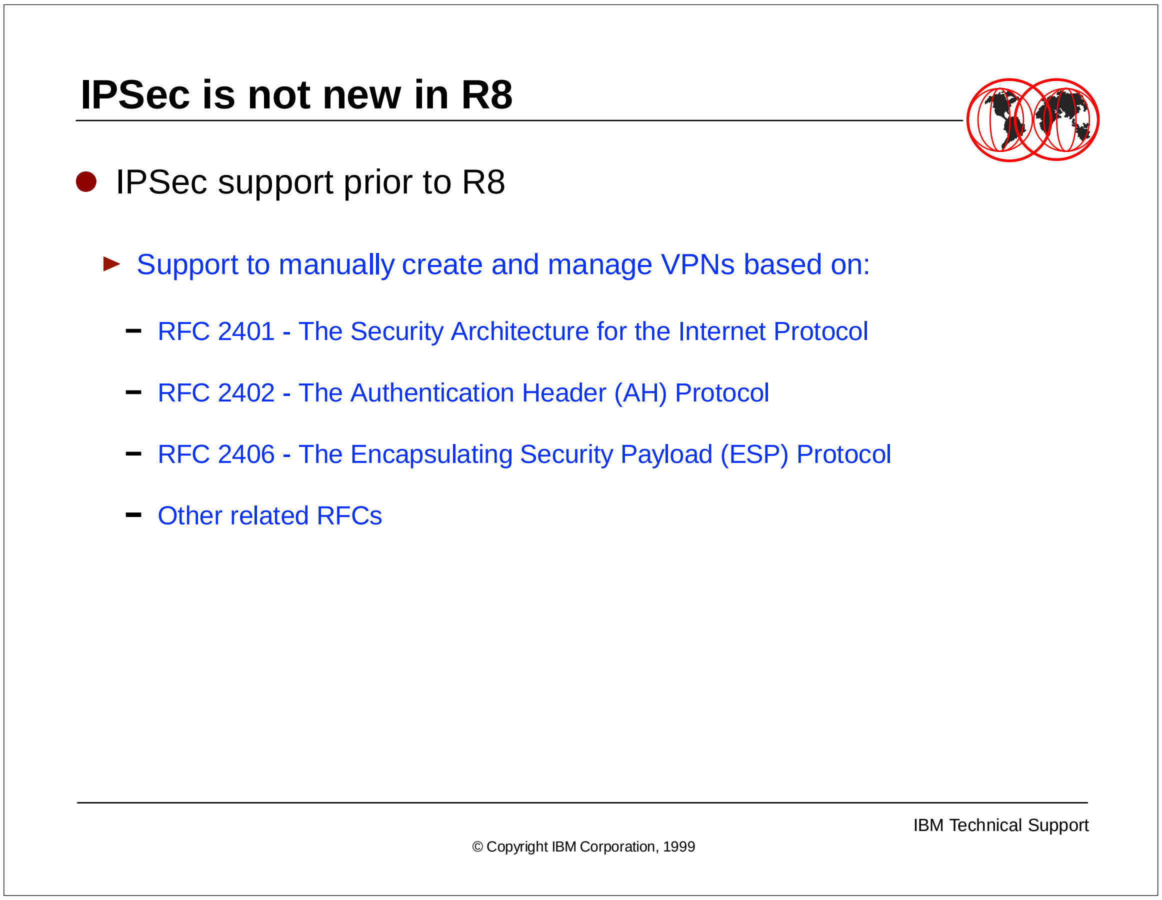


What is IPSec?

- Internet Protocol (IP) Security (Sec)
  - A protocol
  - Used to implement Virtual Private Networks (VPNs)
  - Implemented at the Internet layer of the TCP/IP stack
  - Compatible with non-IPSec based networks
- Defined by a working group of Internet Engineering Task Force (IETF) security area
- A series of Internet drafts and RFCs
- Includes topics such as:
  - Protecting IP packets with authentication and encryption
  - Key Management
  - Policy

IBM Technical Support

- The IPSec acronym stands for IP Security. The IP acronym stands for Internet Protocol.
- IPSec is the name of a working group in the security area of the IETF (Internet Engineering Task Force). The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. See www.ietf.org for more details.
- The TCP/IP protocol suite stack consists of 4 layers. Those layers are the network interface layer, the Internet layer, transport layer, and application layer. The IPSec protocol is implemented at the Internet (IP) level of the TCP/IP stack. The IPSec protocol is such that IPSec protected packets can flow through IP networks that do not support IPSec.
- IPSec is the protocol used to implement Virtual Private Networks (VPNs).
- The IPSec working group has published a series of working drafts and RFCs. At the time this presentation was created there were 26 working drafts and 17 RFCs. There are a variety of different topics covered in these publications. Examples include techniques to protect IP traffic (e.g. AH and ESP), key management (e.g. ISAKMP and IKE), and centralized policy management (e.g. LDAP schema).

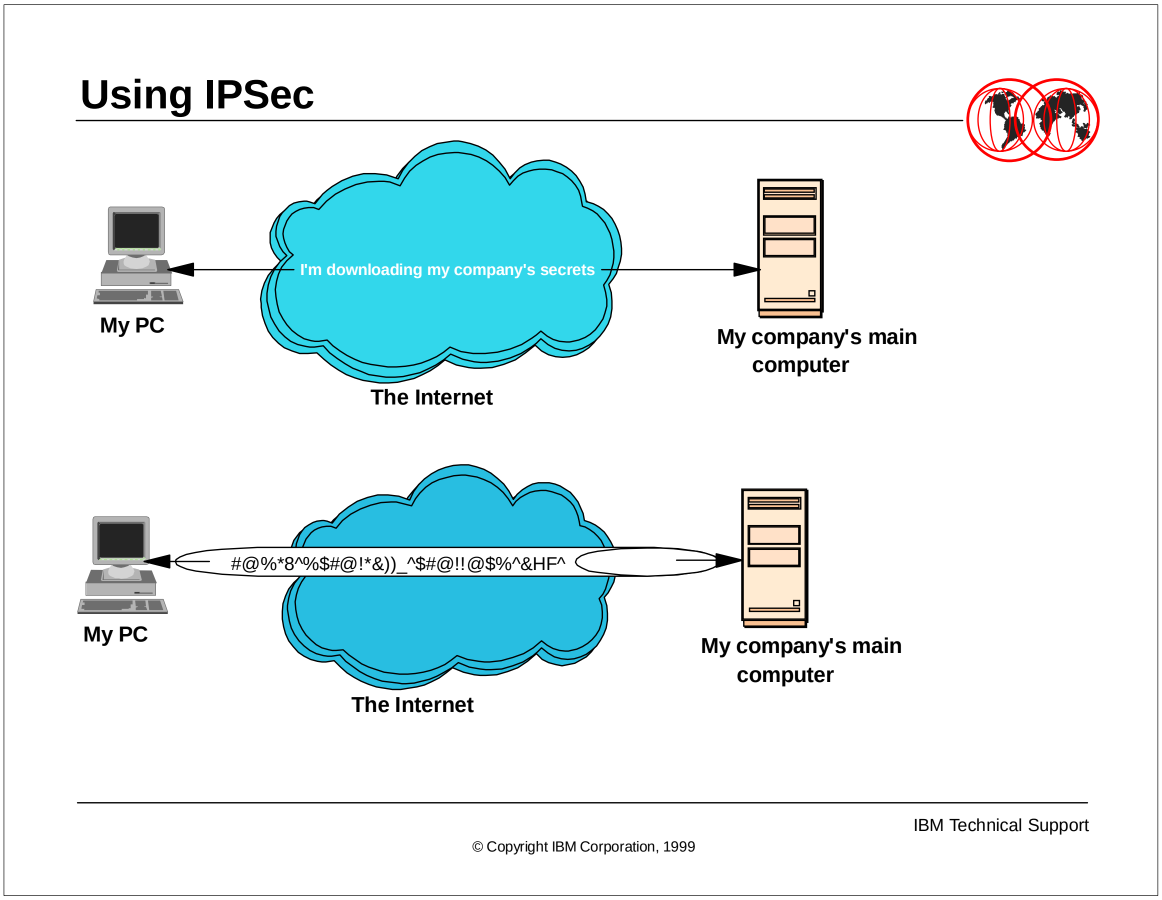


IPSec is not new in R8

- IPSec support prior to R8
  - ► Support to manually create and manage VPNs based on:
    - RFC 2401 - The Security Architecture for the Internet Protocol
    - RFC 2402 - The Authentication Header (AH) Protocol
    - RFC 2406 - The Encapsulating Security Payload (ESP) Protocol
    - Other related RFCs

IBM Technical Support

- Support of IPSec is not new for R8. Support for various IPSec drafts and RFCs have been included in OS/390 Firewall Technologies since R5 (as well as the R4 kit). The support offered in the R7 timeframe requires manual definition and creation of VPNs based on these RFCs. This means that the VPN administrators must manually decide the characteristics of a VPN and they must manually exchange the encryption keys used by a VPN. This foil lists a few of the major RFCs supported in R7.
- RFC 2401 defines the main IPSec architecture. It is the building block upon which other IPSec RFCs are based. It defines the basic concepts that allow security information to be represented and identified. One of the key concepts introduced by this RFC is that of a Security Association (SA).
- RFC 2402 defines a protocol that provides authentication and integrity of IP packets. Authentication provides a mechanism to verify the origin of the IP packet. Integrity provides a mechanism to verify the IP packet has not been modified.
- RFC 2404 defines a protocol that provides encryption of IP packets as well as authentication and integrity. Encryption provides a mechanism to hide the contents of an IP packet while it is routed through "untrusted" networks..
- The RFCs mentioned above are only three of the IPSec RFCs that are included in the R7 support. There are many related RFCs that have also been implemented. These related RFCs define specific algorithms that can be supported with the AH and ESP protocols.



Using IPSec

My PC

I'm downloading my company's secrets

The Internet

My company's main computer

My PC

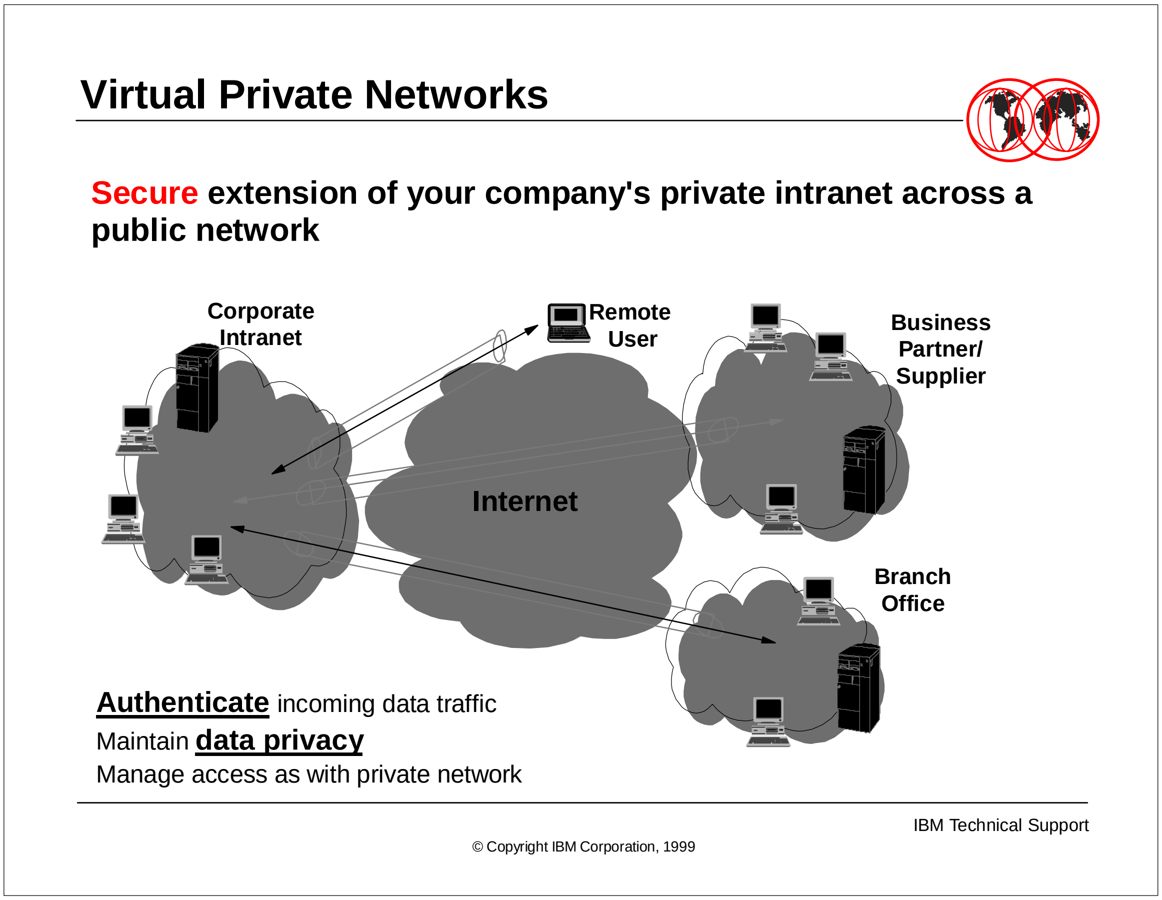
#@%*8^%$#@!*&))_^$#@!!@$%^&HF^

The Internet

My company's main computer

IBM Technical Support

- The top diagram represents using IP without IPSec. The bottom diagram represents using IPSec with the ESP protocol. The point is that in the top picture my company's secrets are exposed as they flow over the Internet. In the bottom picture they are protected. That's what IPSec and VPNs are all about.



Virtual Private Networks

Secure extension of your company's private intranet across a public network

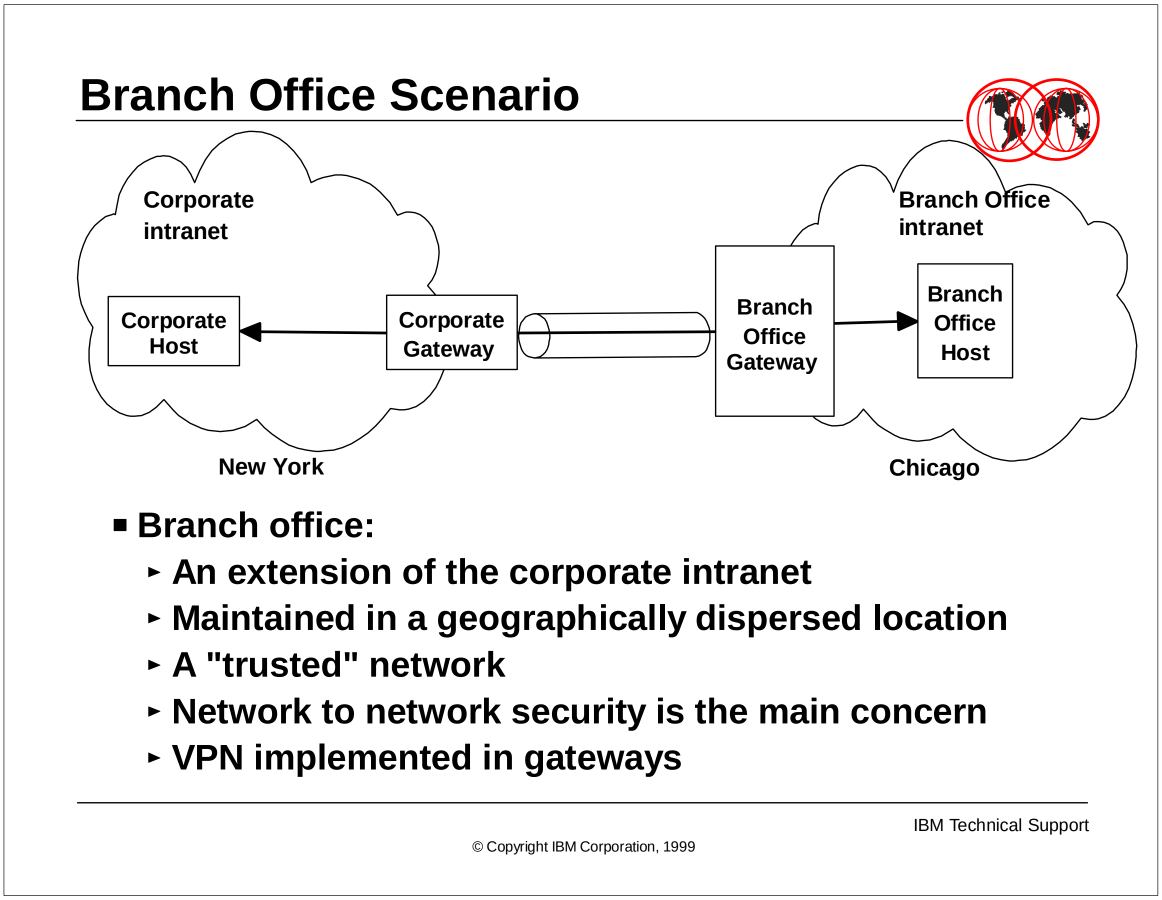
Authenticate incoming data traffic

Maintain **data privacy**

Manage access as with private network

IBM Technical Support

- There are three classic scenarios where VPNs can play a vital role in a customer environment. They are the Branch Office, Business Partner/Supplier, and Remote User.
- In each of these scenarios VPNs can be used to authenticate incoming traffic and provide a means to protect data as it flows across "untrusted" networks such as the Internet.
- VPNs provide customers with the ability to selectively open their private networks to the outside world, while maintaining control of who can access their network and in what manner.
- The remainder of this section describes these scenarios at a high level. These high level descriptions are not meant to apply to every situation.

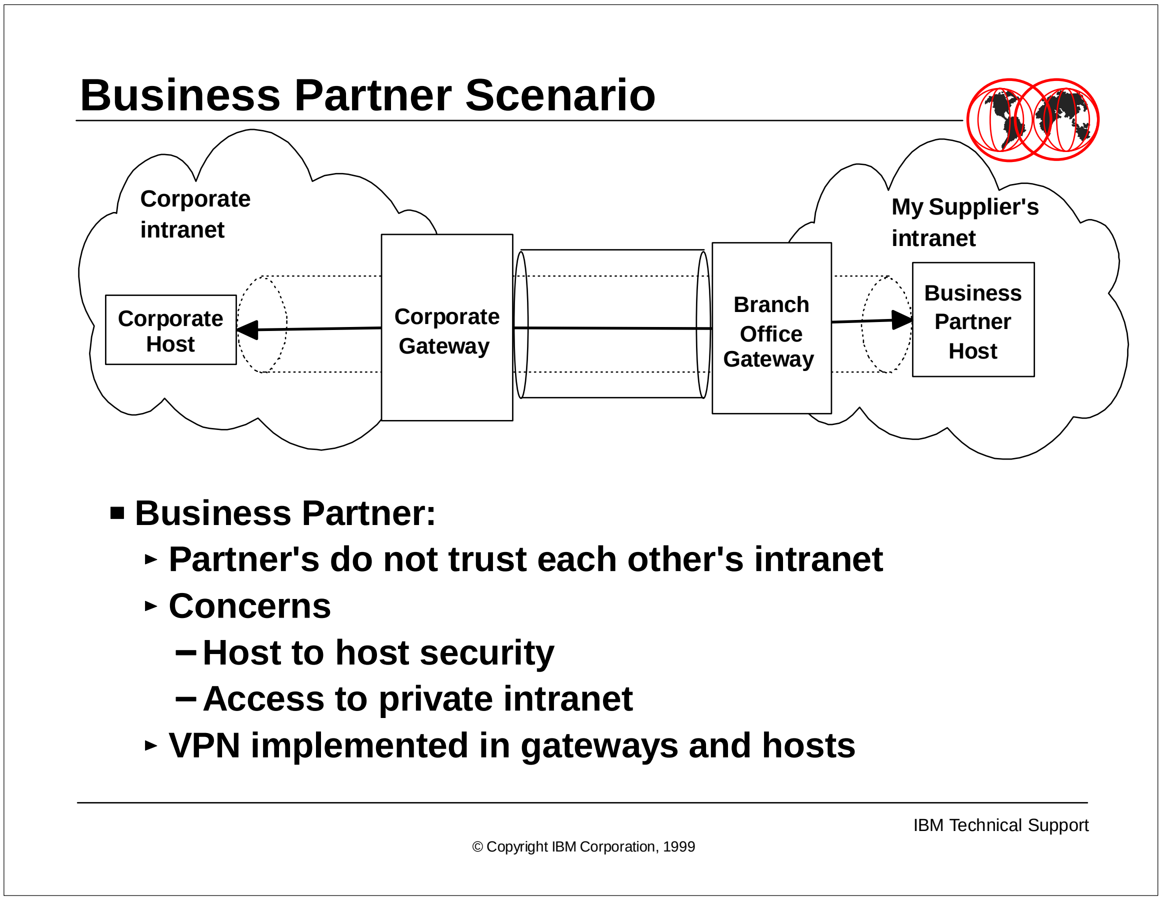


Branch Office Scenario

- Branch office:
  - An extension of the corporate intranet
  - Maintained in a geographically dispersed location
  - A "trusted" network
  - Network to network security is the main concern
  - VPN implemented in gateways

IBM Technical Support

- A branch office scenario usually involves a corporate intranet and at least one branch office's intranet. The intranets are part of the same company. Each intranet trusts that the other's intranet is secure.
- The branch office is often located in a different geographical location. The desire is to securely connect both intranets in an economical fashion. In doing this traffic between the intranets may pass through one or more untrusted networks.
- The main concern is network to network security. This includes authenticating the origin of IP packets upon entering the intranet and encrypting the data between the intranets. In order to accomplish this VPNs are employed between gateways that connect the intranets.
- The terms host and gateway as the apply to IPSEC need to be discussed. A device is acting as a host if it is encapsulating data that originated from that device or if it decapsulating data whose final destination is that device. A device is acting as a gateway if it is encapsulating data that originated from another device or if it is decapsulating data whose final destination is another device. Firewalls and routers are typically acting as gateways. The workstation on your desktop is typically acting as a host.



Business Partner Scenario

Corporate intranet

Corporate Host

Corporate Gateway

Branch Office Gateway

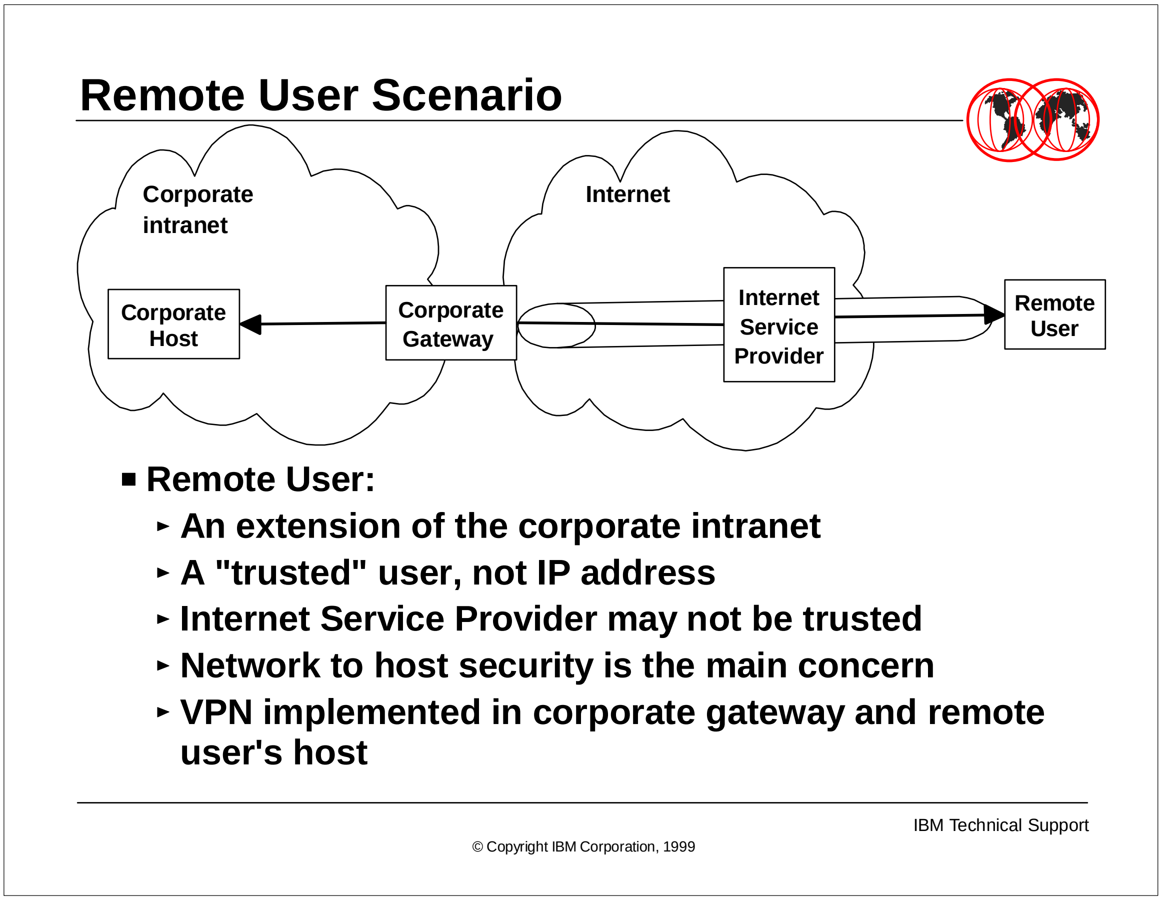
My Supplier's intranet

Business Partner Host

- Business Partner:
  - Partner's do not trust each other's intranet
  - Concerns
    - Host to host security
    - Access to private intranet
  - VPN implemented in gateways and hosts

IBM Technical Support

- A business partner scenario usually involves a corporate intranet and at least one business partner's intranet. The intranets are not part of the same company. Neither intranet trusts that the other's intranet is secure.
- There is a business need to allow limited access between each company's intranet.
- Since neither company trusts the other's intranet host to host security is a concern. In order to accomplish this VPNs are employed between communicating hosts.
- In addition, neither company desires to blindly allow traffic from the outside world into its intranet. In order to accomplish this VPNs are employed between gateways that connect the intranets. These VPNs will authenticate that an IP packet is authorized to enter the intranet.

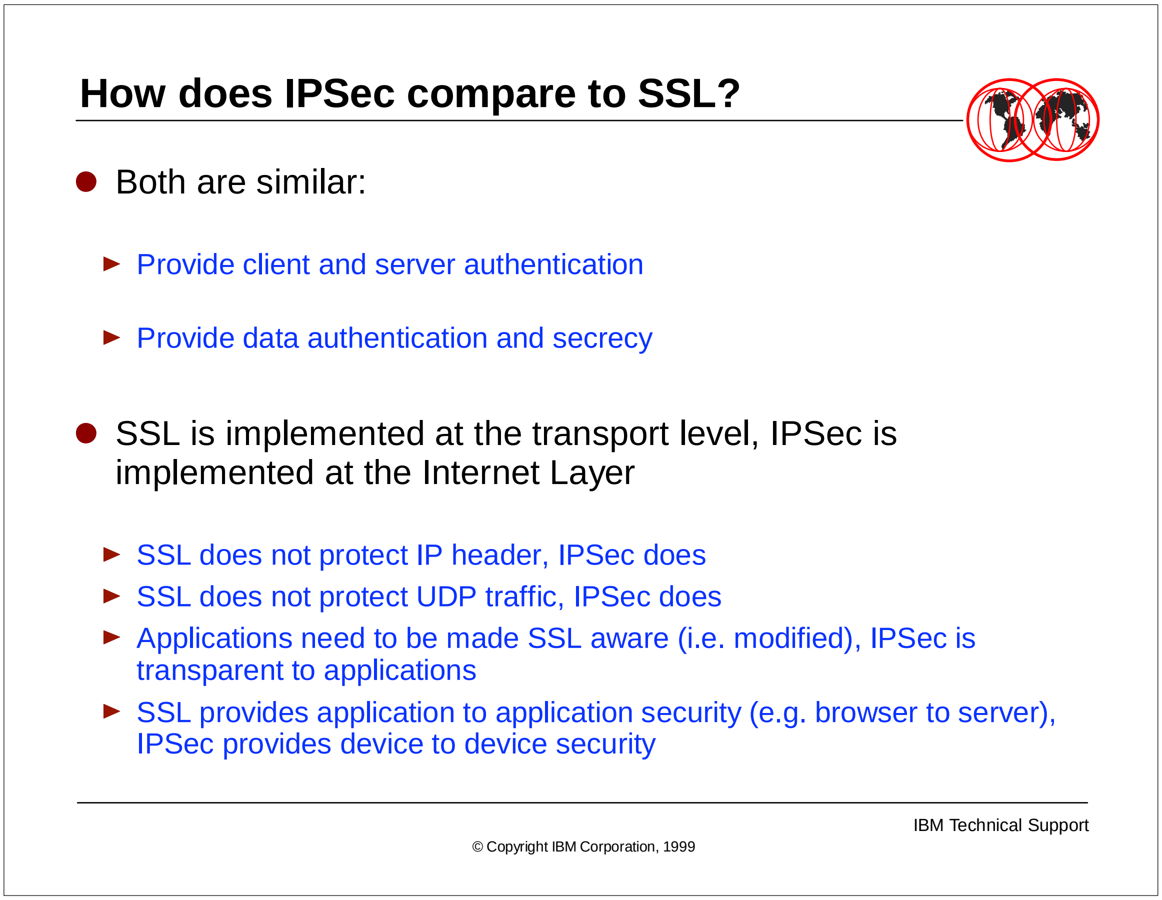


Remote User Scenario

- Remote User:
  - An extension of the corporate intranet
  - A "trusted" user, not IP address
  - Internet Service Provider may not be trusted
  - Network to host security is the main concern
  - VPN implemented in corporate gateway and remote user's host

IBM Technical Support

- The remote user scenario is a variation of the business partner scenario. In this case both the remote user and the corporate intranet are trusted. Networks between the remote user and corporate intranet may or may not be trusted. A VPN is implemented between the remote host and the corporate gateway.
- The remote user is likely to have a dynamic IP address. This makes the manual establishment and maintenance of a VPN difficult.

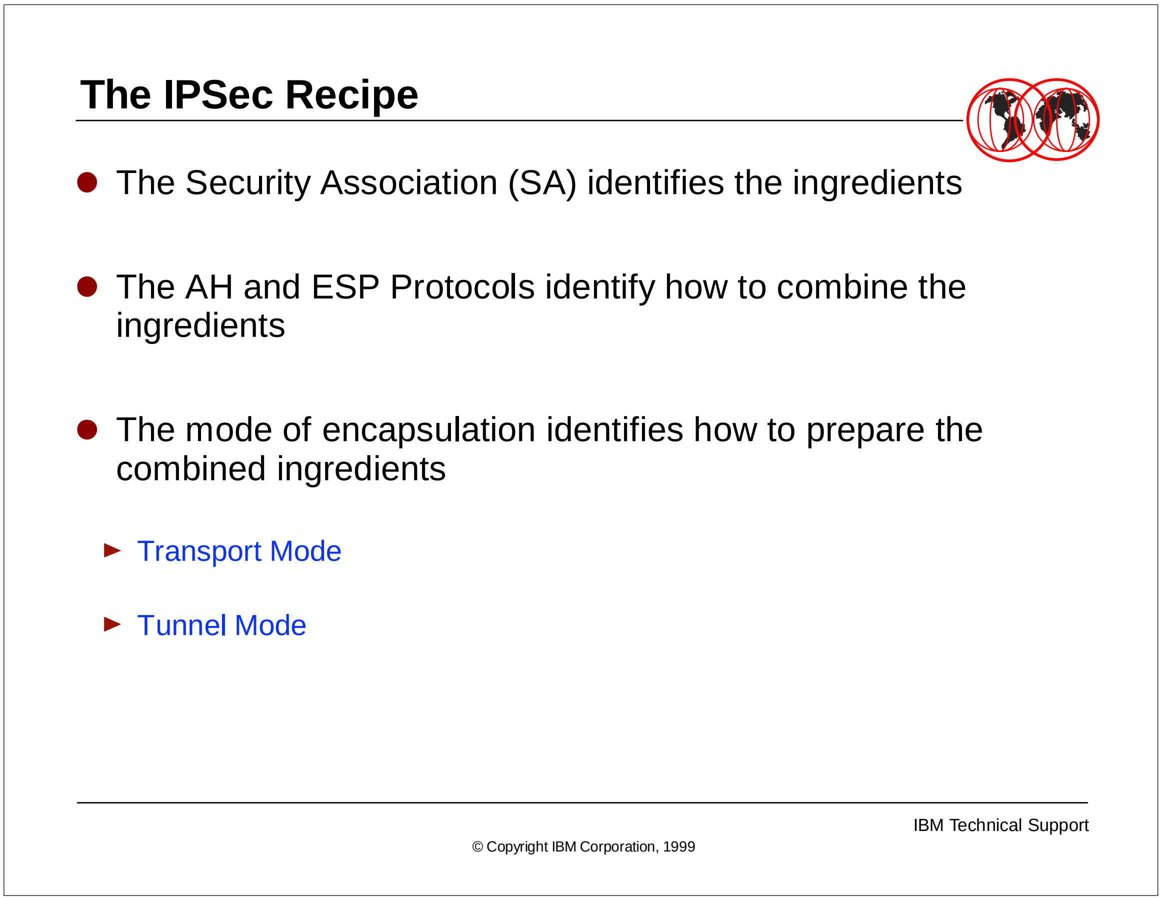


How does IPSec compare to SSL?

- Both are similar:
  - Provide client and server authentication
  - Provide data authentication and secrecy
- SSL is implemented at the transport level, IPSec is implemented at the Internet Layer
  - SSL does not protect IP header, IPSec does
  - SSL does not protect UDP traffic, IPSec does
  - Applications need to be made SSL aware (i.e. modified), IPSec is transparent to applications
  - SSL provides application to application security (e.g. browser to server), IPSec provides device to device security

IBM Technical Support

- SSL (Secure Socket Layer) was developed by Netscape and is widely used today. TLS (Transport Layer Security) is an IETF RFC which is based on SSL.
- IPSec and SSL (Secure Socket Layer) are similar. Both attempt to solve similar problems. Both provide a way to encrypt data contained in IP packets. Both prevent modification of data contained in IP packets. Both provide a way to authenticate the communicating parties.
- The biggest difference is where each protocol is implemented. SSL sits between the application layer and the transport layer of the TCP/IP protocol stack. As such it does not provide protection for the IP header, it does not protect UDP traffic, and it requires applications to be modified to make use of it.
- IPSec and SSL both have pluses and minuses. There is not a clear answer as to which is better. There are situations where one is more applicable than the other.

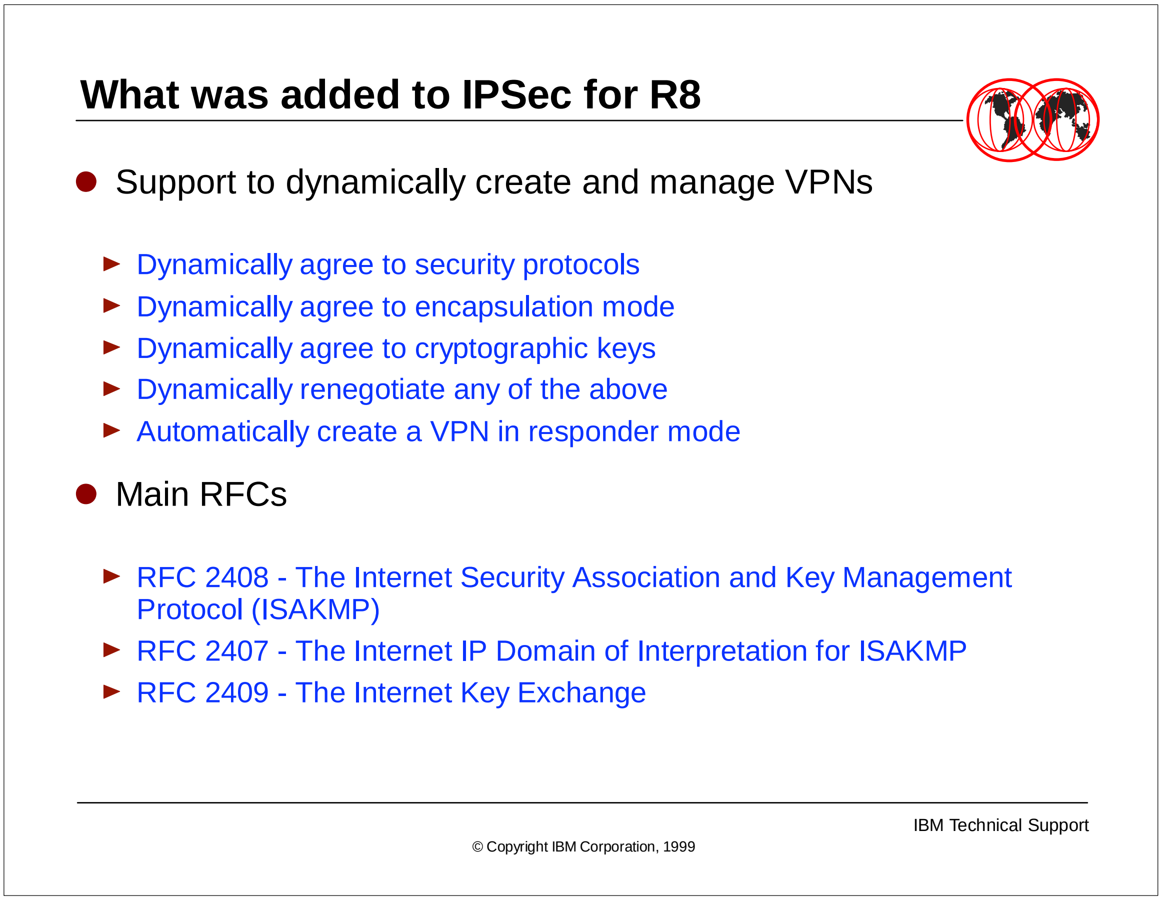


The IPSec Recipe

- The Security Association (SA) identifies the ingredients
- The AH and ESP Protocols identify how to combine the ingredients
- The mode of encapsulation identifies how to prepare the combined ingredients
  - Transport Mode
  - Tunnel Mode

IBM Technical Support

- For our purposes think of a IPSec as providing a standard way to create a recipe for a VPN. The Security Association identifies the ingredients needed to create IPSec packets for a particular VPN. If you are baking an apple pie the ingredients would include information like how many apples and what spices to use. If you are creating a IPSec packet for a specific VPN the ingredients would include information like cryptographic keys and attributes specific to the security service being used. This information is contained in an SA.
- If you were making an apple pie the directions would tell you how to combine the ingredients listed to that make the pie. In the case of IPSec the security protocol (e.g. AH or ESP) being provided identifies how to combine the information contained in the SA to create an IPSec packet.
- After the ingredients of the apple pie are combined it is necessary to bake the pie. The recipe provides the details involved in baking the pie. Likewise with IPSec the recipe identifies how to package the IPSec packet into an IP packet. There are two options transport mode and tunnel mode.
- This analogy is an over simplification, but will suffice for the purpose of this T3.
- The process of constructing an IPSec protected IP packet is called encapsulation. Encapsulation includes combining the SA using an IPSec protocol and packaging it into an IP packet.
- The process of deconstructing an IPSec protected IP packet is called decapsulation. When a packet is decapsulated the IPSec recipe is applied in reverse.

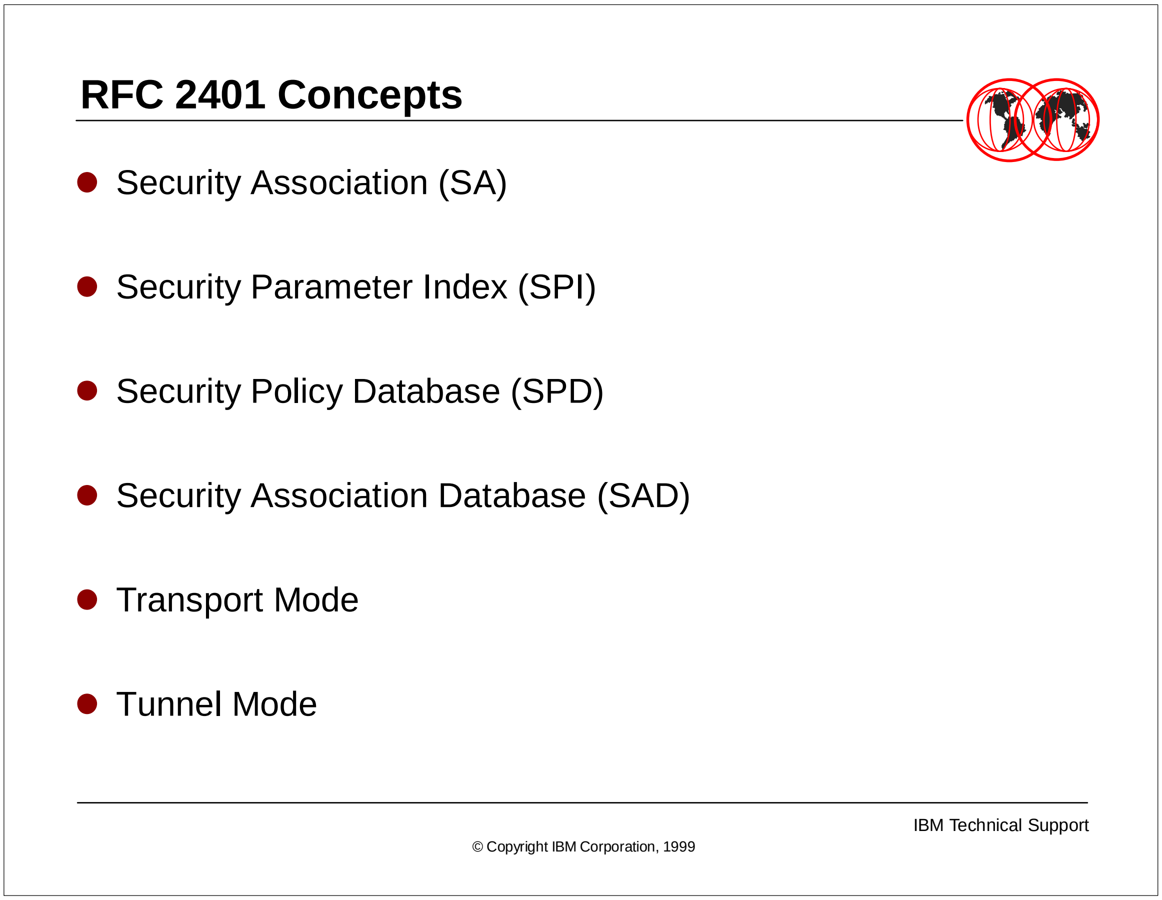


What was added to IPSec for R8

- Support to dynamically create and manage VPNs
  - Dynamically agree to security protocols
  - Dynamically agree to encapsulation mode
  - Dynamically agree to cryptographic keys
  - Dynamically renegotiate any of the above
  - Automatically create a VPN in responder mode
- Main RFCs
  - RFC 2408 - The Internet Security Association and Key Management Protocol (ISAKMP)
  - RFC 2407 - The Internet IP Domain of Interpretation for ISAKMP
  - RFC 2409 - The Internet Key Exchange

IBM Technical Support

- In R8 support to dynamically create and manage VPNs was added to OS/390 Firewall Technologies. This includes the ability to dynamically agree to any of the information needed to create the IPSec recipe. Perhaps the most important aspect of this support is the ability to dynamically agree to and refresh the cryptographic keys. In R8 the ability to automatically create a VPN is restricted to responder mode processing, TCP/IP stack start up, and ISAKMP daemon startup. This means that in R8 there are still cases where a dynamic VPN must be manually started.
- As mentioned earlier this support will be shipped as part of the OS/390 Security Server and it has dependencies on changes made to the OS/390 eNetwork Communication Sever.
- This new support implements additional IPSec RFCs. These RFCs include 2408, 2407, and 2409.
- RFC 2408 enables SA and key management procedures. It defines message formats and message flows that will allow two devices to dynamically agree to the contents of an SA that will be shared between them (including keys). It also provides the ability to authenticate the agreeing parties.
- RFC 2408 does not define how to exchange keys or authenticate peers. It is a framework that facilitates the definition of SA attributes, negotiating SAs, modifying SAs, deleting SAs, authenticating peers, and exchanging keys. This means that RFC 2408 provides the building blocks needed to do this, but does not define the specifics. That is left up to other RFCs such as RFC 2407 (The Internet IP Domain of Interpretation for ISAKMP) and RFC 2409 (The Internet Key Exchange).

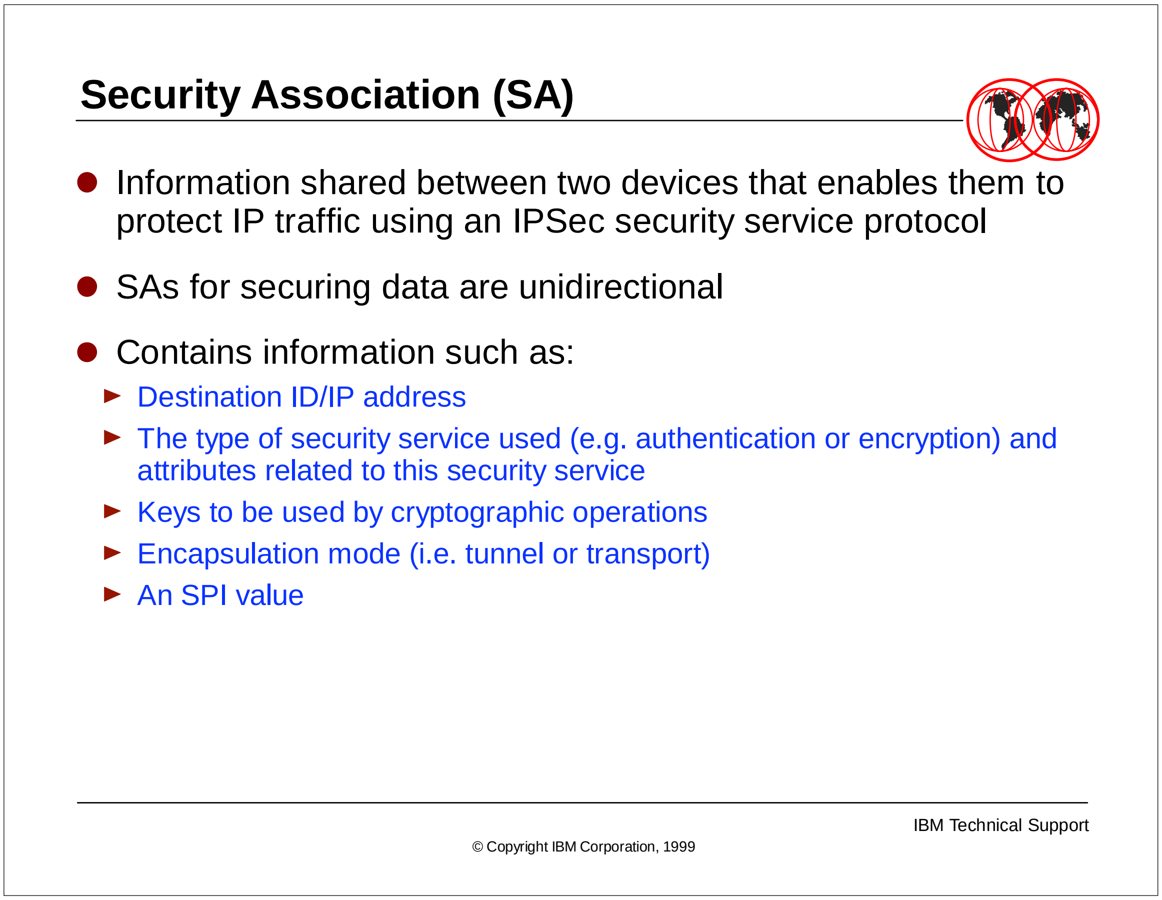


RFC 2401 Concepts

- Security Association (SA)
- Security Parameter Index (SPI)
- Security Policy Database (SPD)
- Security Association Database (SAD)
- Transport Mode
- Tunnel Mode

IBM Technical Support

- RFC 2401 introduces some important concepts. These concepts include Security Association (SA), Security Parameter Index (SPI), Security Policy Database (SPD), Security Association Database (SAD), transport mode, and tunnel mode. These concepts will be discussed in the next few pages.

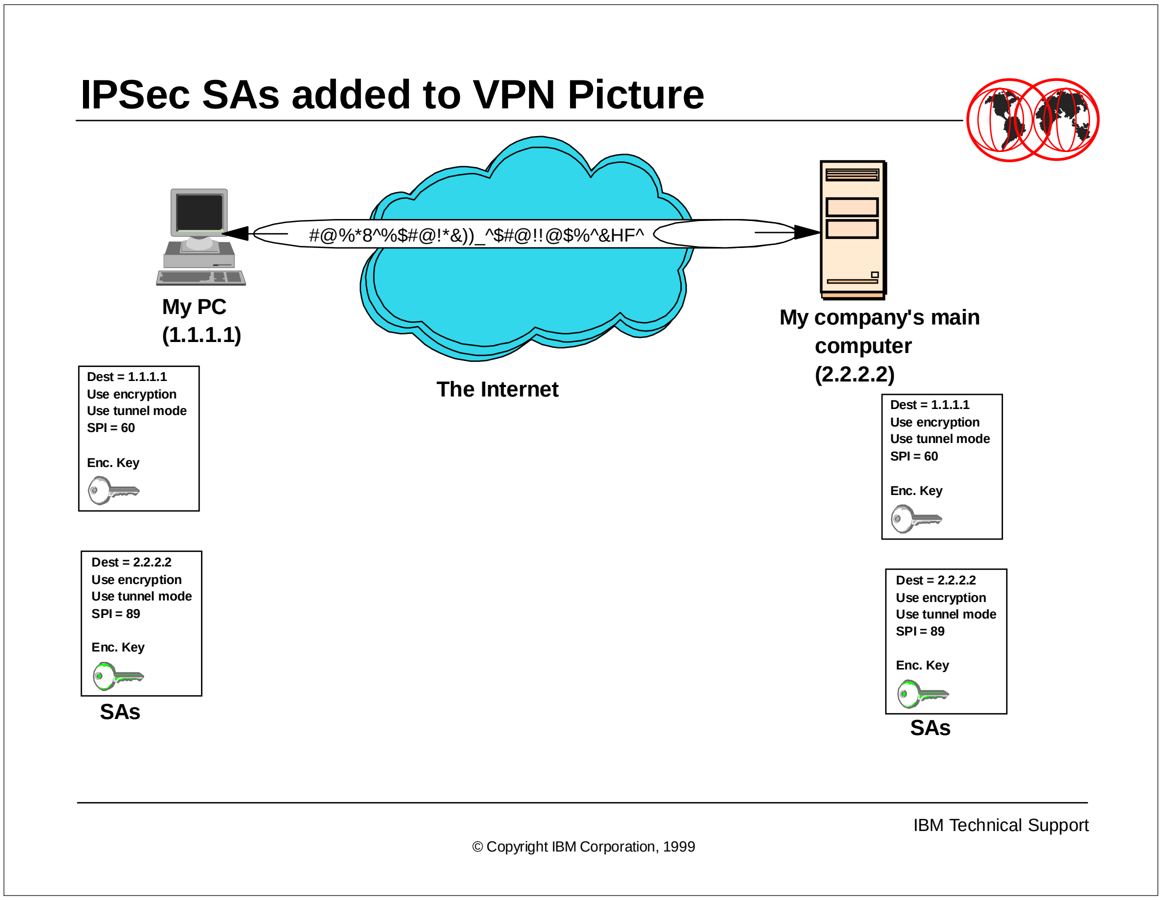


Security Association (SA)

- Information shared between two devices that enables them to protect IP traffic using an IPSec security service protocol
- SAs for securing data are unidirectional
- Contains information such as:
  - Destination ID/IP address
  - The type of security service used (e.g. authentication or encryption) and attributes related to this security service
  - Keys to be used by cryptographic operations
  - Encapsulation mode (i.e. tunnel or transport)
  - An SPI value

IBM Technical Support

- A security association is the set of parameters and their values used by IPSec compliant devices when applying IPSec protocols to IP traffic. Before two devices can protect IP traffic with IPSec they must agree on the values each will use for these parameters. In this regard an SA is an agreement or contract between two devices.
- For now we will talk about SAs that are used to secure data. There currently are two IPSec protocols defined for the purpose of securing IP traffic. These protocols are the AH and ESP protocols. These protocols will be discussed later on. They are introduced now so I can define what I mean by the term IPSec SA. Throughout this document when I say IPSec SA, I am referring to an SA for use with the ESP or AH protocol. Latter when we discuss key management we will find there is another type of SA called an ISAKMP SA.
- In essence an IPSec SA identifies the information needed by one device to construct an IPSec protected packet and the information needed by the other device to deconstruct the IPSec protected packet. IPSec SAs are unidirectional meaning that if two devices are going to send and receive IPSec protected packets then 2 SAs are needed. An inbound SA and an outbound SA.
- IPSec SAs are specific to the security service they provide. That is if more that one IPSec security service is being applied to an IP packet (e.g. AH and ESP), then each security service will have its own SA.
- The process of constructing an IPSec protected packet is referred to as encapsulation and the process of deconstructing an IPSec protected packet is referred to as decapsulation.
- Manual techniques can be used to define IPSec SAs or IPSec SAs can be defined using automated techniques such as those defined in RFC 2408 (The Internet Security Association and Key Management Protocol - ISAKMP)



IPSec SAs added to VPN Picture

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAs

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

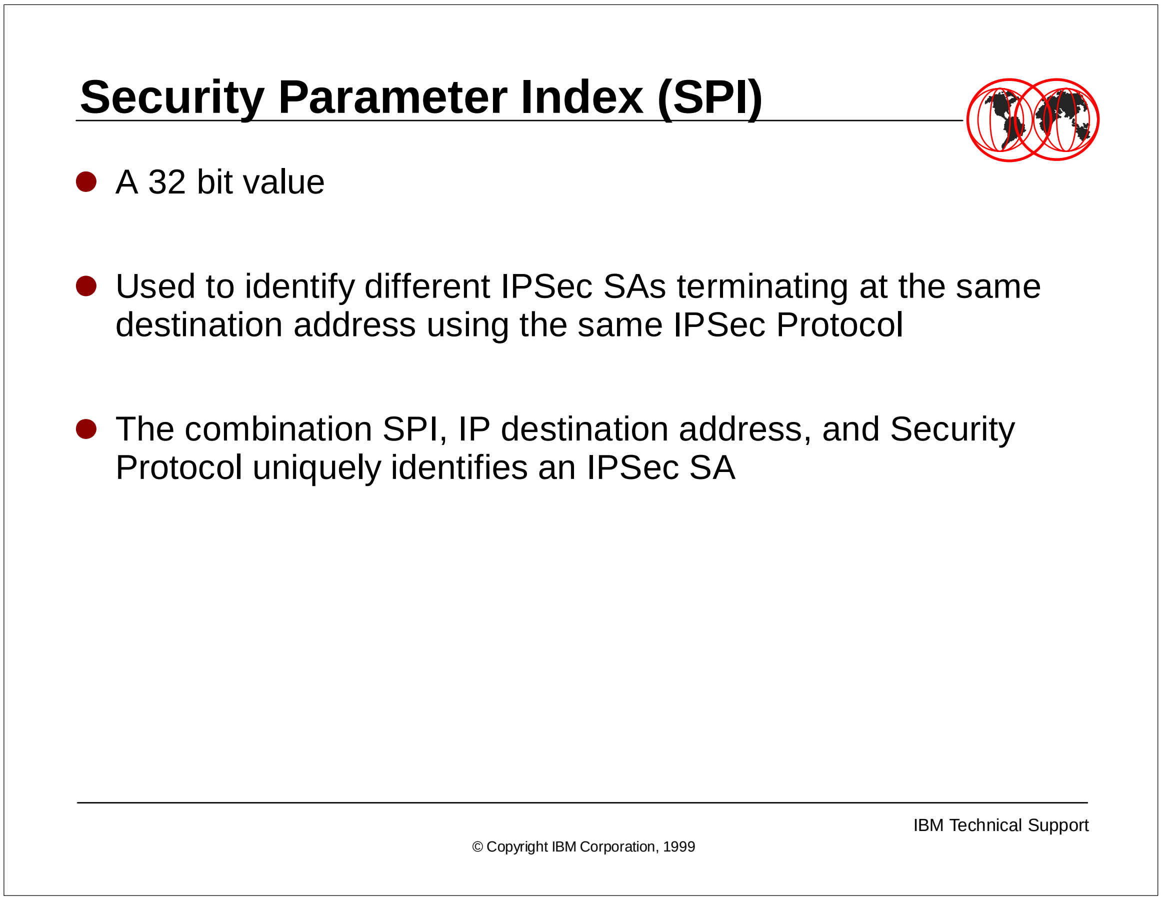
Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAs

IBM Technical Support

- This is a diagram to show the details of the previous page. Note that a real IPSec SA is more complicated than this.

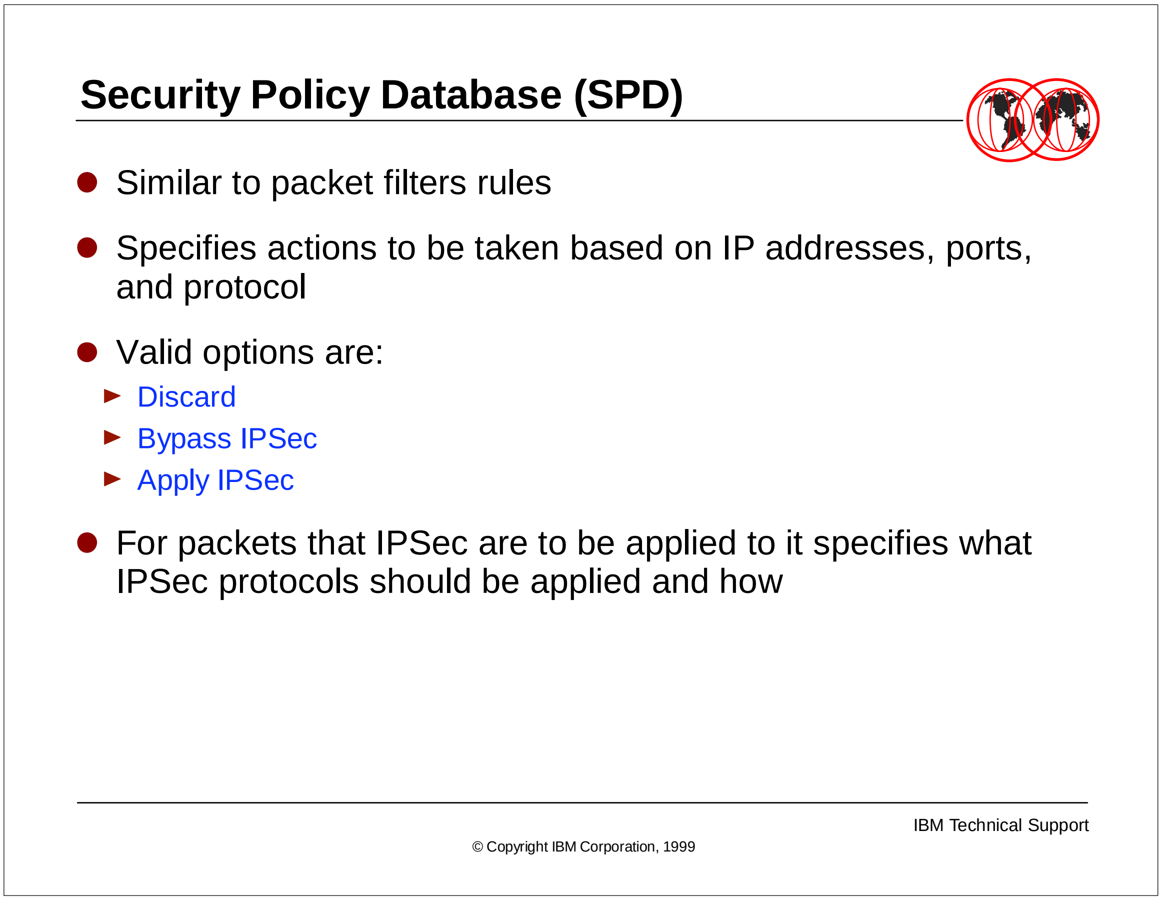


Security Parameter Index (SPI)

- A 32 bit value
- Used to identify different IPSec SAs terminating at the same destination address using the same IPSec Protocol
- The combination SPI, IP destination address, and Security Protocol uniquely identifies an IPSec SA

IBM Technical Support

- As seen on the previous page the SPI is part of an SA. An SPI is a 32 bit value. The SPI is one of 3 parameters that together uniquely identify an IPSec SA. The other 2 parameters are the destination IP address and IPSec protocol. As such the SPI's purpose in life is to uniquely identify IPSec SAs that terminate at the same destination IP address and use the same IPSec protocol.
- At this point it may not be obvious why a device might have multiple IPSec SAs terminating at the same destination IP address and using the same IPSec protocol. An example of this would be that a device wants to use the same IPSec protocol (e.g. ESP) for both UDP and TCP traffic destined for a particular IP address; however, it wants to use different values for a particular parameter in the SA (e.g. a different encryption algorithm)
- SPI values are sent unencrypted as part of an IPSec protected packet. This will be more evident when the AH and ESP protocols are discussed. When an IPSec protected packet arrives at the device it is destined for it uses the triplet of <destination IP address as specified in the IP header, IPSec protocol, SPI> to locate the SA containing the information needed to decapsulate the packet.

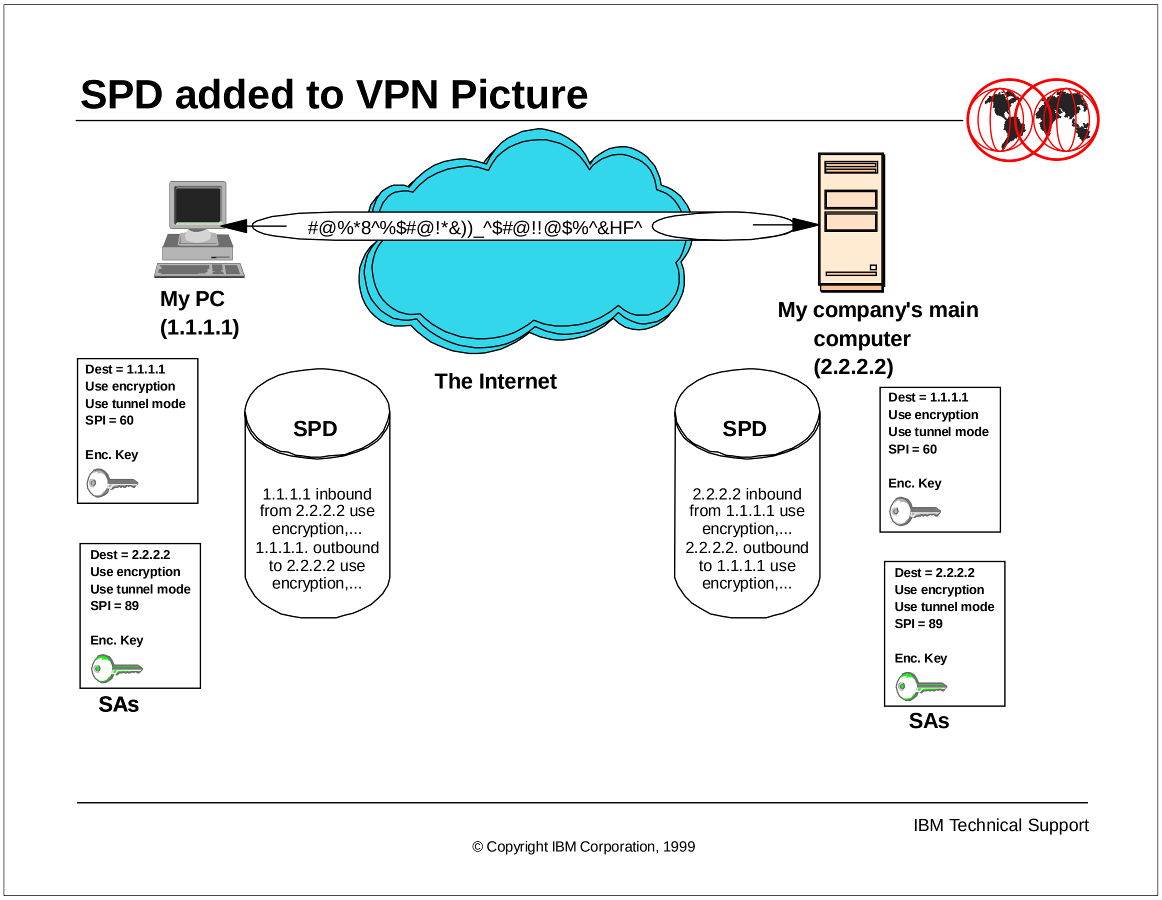


Security Policy Database (SPD)

- Similar to packet filters rules
- Specifies actions to be taken based on IP addresses, ports, and protocol
- Valid options are:
  - Discard
  - Bypass IPSec
  - Apply IPSec
- For packets that IPSec are to be applied to it specifies what IPSec protocols should be applied and how

IBM Technical Support

- The term database is misleading. I would have called the SPD the SP (Security Policy). This policy does not have to be stored in a physical database like DB2.
- The SPD is similar to packet filter rules. The SPD contains an ordered list of policy entries. These entries have selector values similar to those found on filter rules (i.e. source IP address, source port, source protocol, destination IP address, destination port, and destination protocol). These entries also indicate what action should be taken when an IP packet matches the selector value. The allowable actions are discard the packet, bypass IPSec, and apply IPSec. For the apply IPSec action the SPD entry also specifies what IPSec protocols are to be applied and how. The SPD provides the information needed to create instances of SAs.



SPD added to VPN Picture

My PC
(1.1.1.1)

#@%*8^%$#@!*&))_^$#@!!@$%^&HF^

The Internet

My company's main
computer
(2.2.2.2)

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAs

SPD

1.1.1.1 inbound from 2.2.2.2 use encryption,... 1.1.1.1. outbound to 2.2.2.2 use encryption,...

SPD

2.2.2.2 inbound from 1.1.1.1 use encryption,... 2.2.2.2. outbound to 1.1.1.1 use encryption,...

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

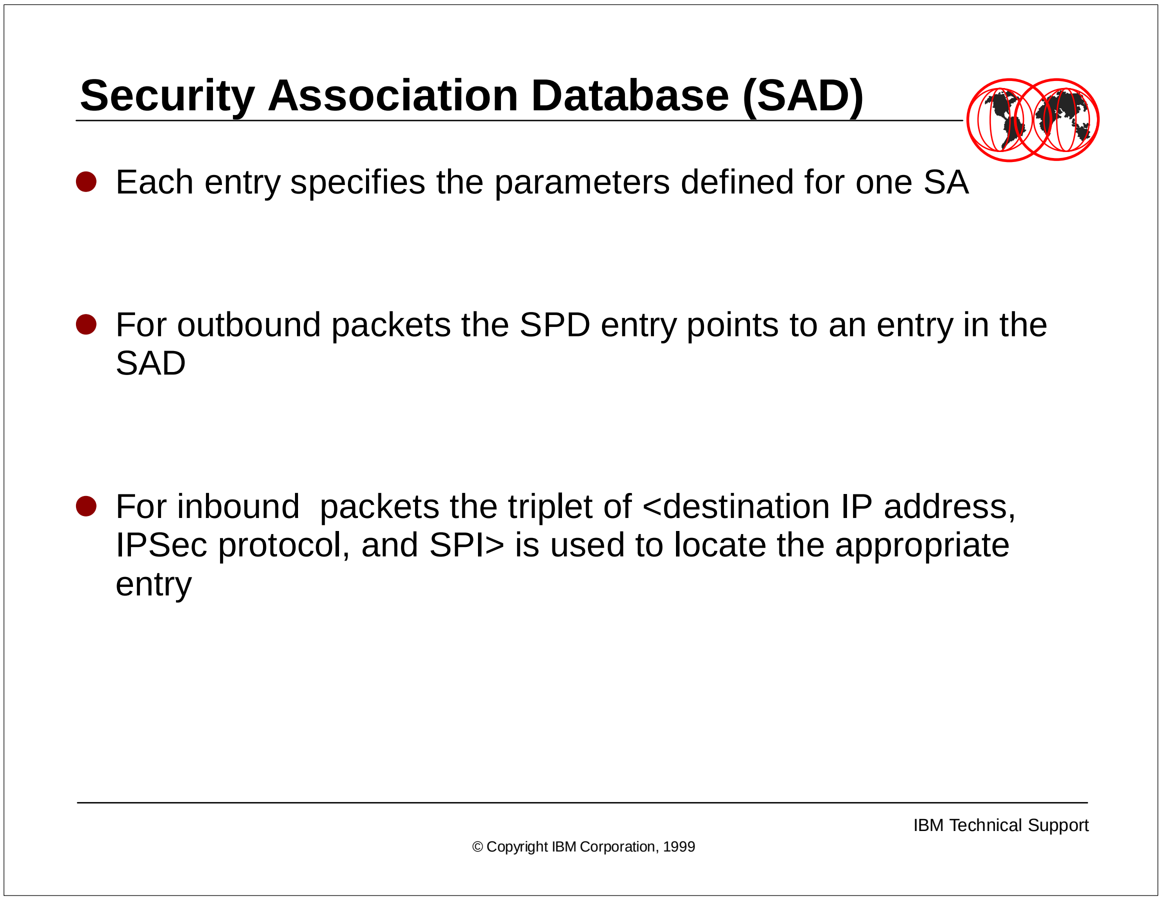
Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAs

IBM Technical Support

- This is a diagram to show the details of the last page. Note that this is a simplified SPD.

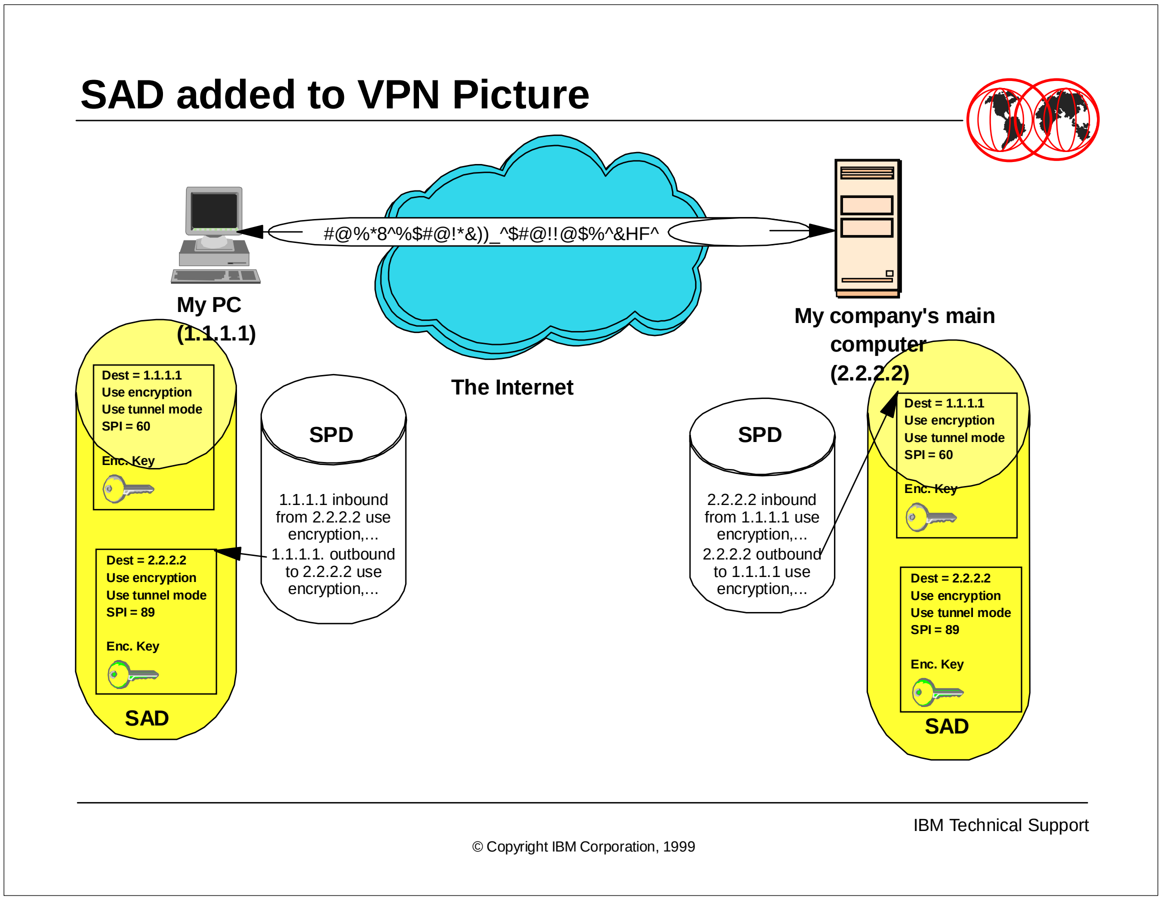


Security Association Database (SAD)

- Each entry specifies the parameters defined for one SA
- For outbound packets the SPD entry points to an entry in the SAD
- For inbound packets the triplet of <destination IP address, IPSec protocol, and SPI> is used to locate the appropriate entry

IBM Technical Support

- Once again the term database here is misleading. When I think of database I think of a physical database like DB2. In this case the term database is a logical database. I would have called the SAD the Active Security Associations. In reality that is all the SAD is. Each entry in the SAD represents a specific instance of an SA.
- SAs are created from the policy defined in the SPD. The creation of the SA could be packet driven or it could be predefined.
- The SPD is consulted for each outbound IP packet to see what action should be taken (the possible actions were discussed on a previous page). If IPSec processing should be applied to this packet the SPD will point to the corresponding active SA entry in the SAD in the case of the predefined SAs. If the creation of SAs is allowed to be packet driven then the SPD might not point to an active SA. In this case one would be created and the SPD would be updated to point to this new SA. The SA will contain all the information needed to encapsulate the IPSec packet.
- For each inbound IP packet the SAD is checked looking for a matching triplet of <destination IP address, IPSec protocol,SPI>. If such an SA exists, then this SA will contain the information required to decapsulate the IPSec packet.



SAD added to VPN Picture

#@%*8^%$#@!*&))_^$#@!!@$%^&HF^

My PC
(1.1.1.1)

The Internet

My company's main
computer
(2.2.2.2)

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAD

SPD

1.1.1.1 inbound from 2.2.2.2 use encryption,...
1.1.1.1. outbound to 2.2.2.2 use encryption,...

SPD

2.2.2.2 inbound from 1.1.1.1 use encryption,...
2.2.2.2 outbound to 1.1.1.1 use encryption,...

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

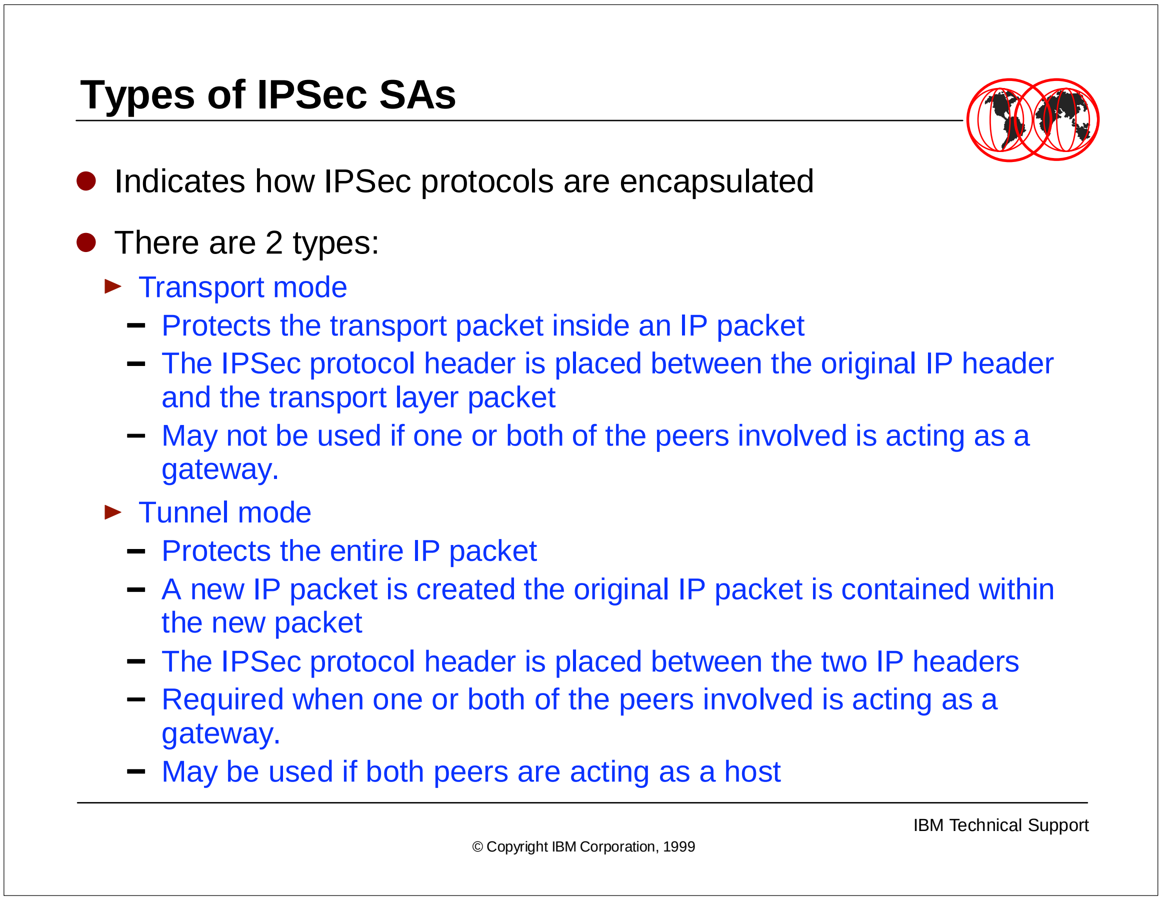
Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAD

IBM Technical Support

- This is a diagram to the show the details of the last page.

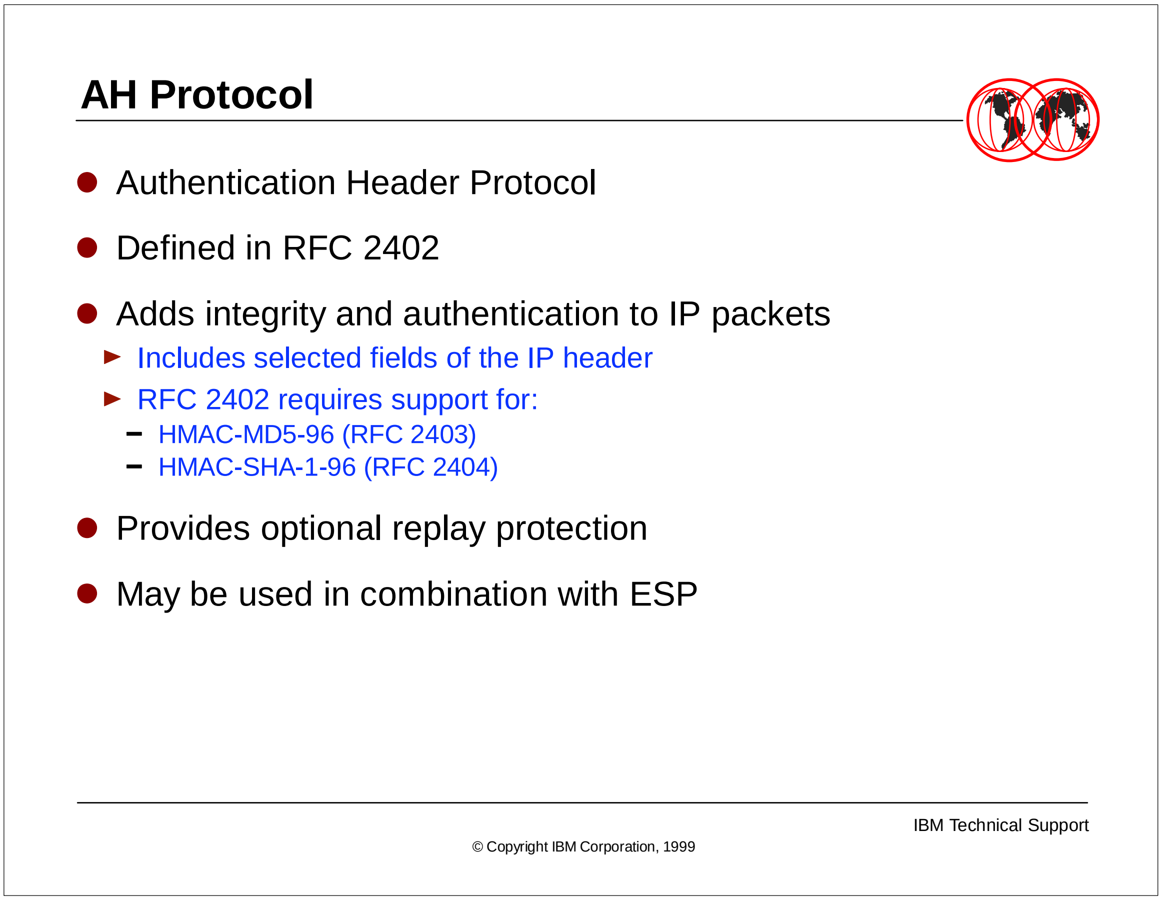


Types of IPSec SAs

- Indicates how IPSec protocols are encapsulated
- There are 2 types:
  - Transport mode
    - Protects the transport packet inside an IP packet
    - The IPSec protocol header is placed between the original IP header and the transport layer packet
    - May not be used if one or both of the peers involved is acting as a gateway.
  - Tunnel mode
    - Protects the entire IP packet
    - A new IP packet is created the original IP packet is contained within the new packet
    - The IPSec protocol header is placed between the two IP headers
    - Required when one or both of the peers involved is acting as a gateway.
    - May be used if both peers are acting as a host

IBM Technical Support

- There are two types of IPSec SAs. They are classified based on the mode of encapsulation being used. The two types are transport mode SAs and tunnel mode SAs.
- An IP packet consists of an IP header and data. Usually a higher level transport such as TCP or UDP is being used. In this case the data portion of the IP packet is a transport layer packet, which by the way has its own header.
- In transport mode the IP header is separated from the transport packet. The IPSec protocol is then applied to the transport packet (note: part of the IP header is also used in the case of the AH protocol). The result is an IPSec packet that contains the original transport packet . The original IP header is then attached to the IPSec packet. In the process of doing this the length, protocol, and checksum fields of the original IP header are modified. The other fields of the original IP header are not modified (e.g. source and destination address).
- In tunnel mode a new IP header is created. The IPSec protocol is applied to the entire IP packet creating an IPSec packet that contains the original IP packet (note: part of the new IP header is also used in the case of the AH protocol). The new IP header is then updated and attached to the IPSec packet. It is possible for the new IP header's source and destination addresses to be different than in the original IP header. This provides the ability to hide private IP addresses.
- The details of tunnel and transport mode should become clearer when the AH and ESP protocols are discussed in the next few pages.
- Recall that an SA is an agreement made between two IPSec devices. In the case where one or both of these devices are acting as a gateway, the SA must be a tunnel mode SA. When both devices are acting as a host either transport mode SAs or tunnel mode SAs may be used.

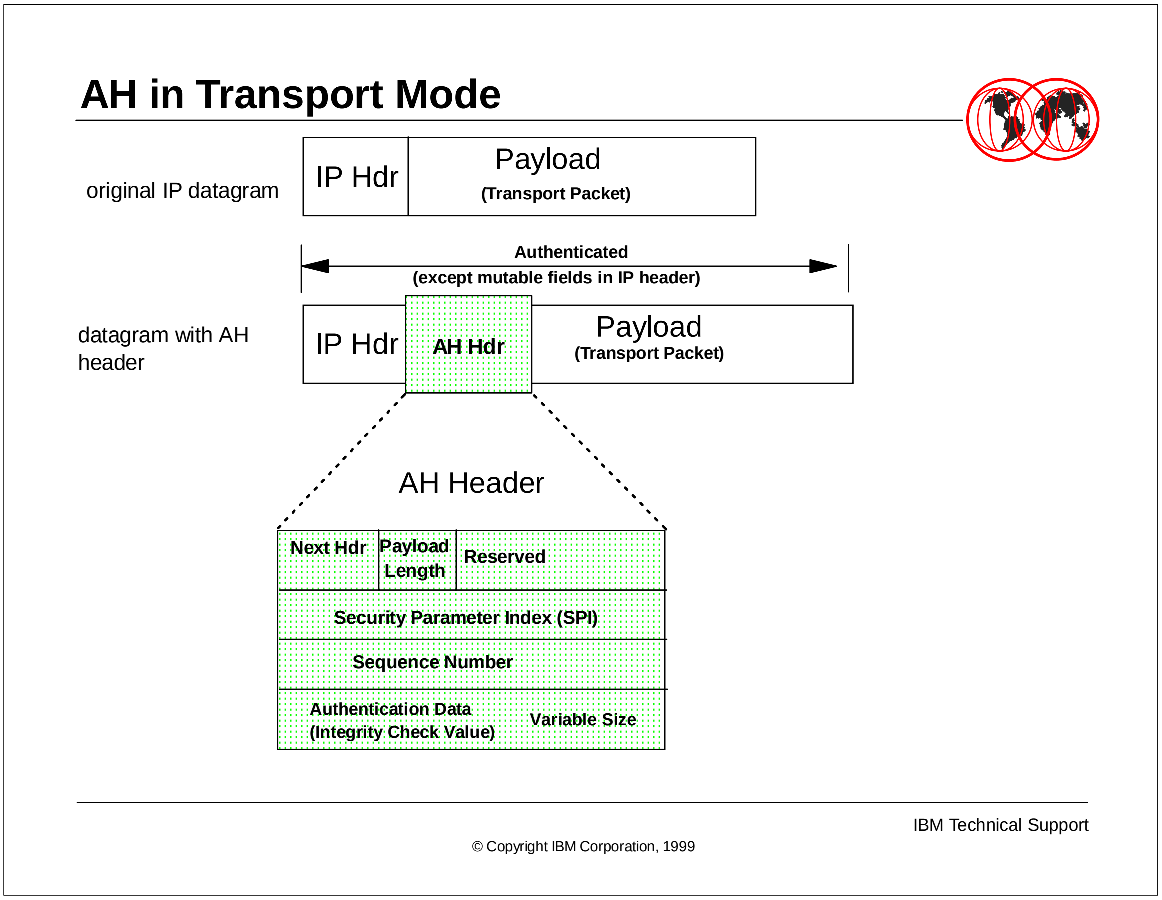


AH Protocol

- Authentication Header Protocol
- Defined in RFC 2402
- Adds integrity and authentication to IP packets
  - Includes selected fields of the IP header
  - RFC 2402 requires support for:
    - HMAC-MD5-96 (RFC 2403)
    - HMAC-SHA-1-96 (RFC 2404)
- Provides optional replay protection
- May be used in combination with ESP

IBM Technical Support

- As mentioned earlier, one way IPSec improves the security of IP is defined in RFC 2402. RFC 2402 defines the Authentication Header Protocol. This protocol adds integrity to an IP packet by adding an Integrity Check Value (ICV). An ICV is a cryptographic checksum. Part of an IPSec SA contains the information needed create and validate the ICV. Input to the ICV calculation is either the transport packet or IP packet (depending on the mode of the SA), the AH IPSec header, and selected fields in the outermost IP header. The inclusion of the ICV makes it computationally infeasible for someone to modify the IPSec packet.
- The term outermost IP header is refers to the new IP header in the case of tunnel mode and the original IP header in the case of transport mode.
- Mutable fields of the outer most IP header are not included in the ICV calculation. The mutable fields are those fields that can change in route. These fields include the Type of Service field (TOS), flags, Fragment offset, Time to Live (TTL), and header checksum.
- The source address contained in the outer IP header is included as part of the ICV calculation. This means that the origin IP address can be authenticated. In addition the ICV is cryptographically generated. The ability to properly process the ICV provides another method to authenticate the of data origin (i.e. the data had to come from somebody that possessed the proper cryptographic key).
- A sequence number is included in the AH header. The AH header is included in the ICV calculation. The sequence number provides a mechanism to detect replays. The sender of an IPSec packet is required to send this field, but the receiver is not required to process it. To that extent the replay protection is considered optional.
- Both the AH protocol and the ESP protocol maybe be used to protect the same IP packet.
- There are additional RFCs relative to the algorithms that can be used to calculate the ICV.

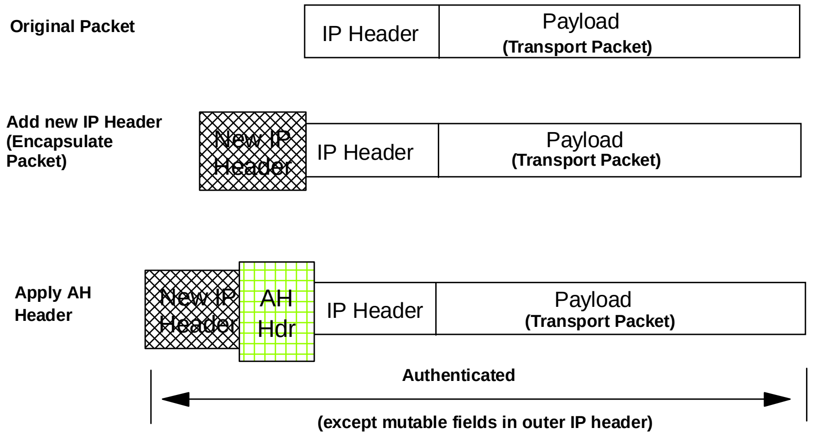


AH in Transport Mode

IBM Technical Support

- This is what AH in transport mode looks like. Note the placement of the AH header between the IP header and the transport packet.
- Also note the that every thing is authenticated except the mutable fields in the IP header.
- One final observation is that the ICV is part of the AH header. An earlier page indicated that the ICV uses the AH header as input to the ICV calculation. This seems to be a chicken and the egg problem, but it is not. When the ICV is calculated, the it uses an AH header with a value of 0 contained in the ICV field.

AH in Tunnel Mode



Original Packet

IP Header | Payload (Transport Packet)

Add new IP Header (Encapsulate Packet)

New IP Header | IP Header | Payload (Transport Packet)

Apply AH Header

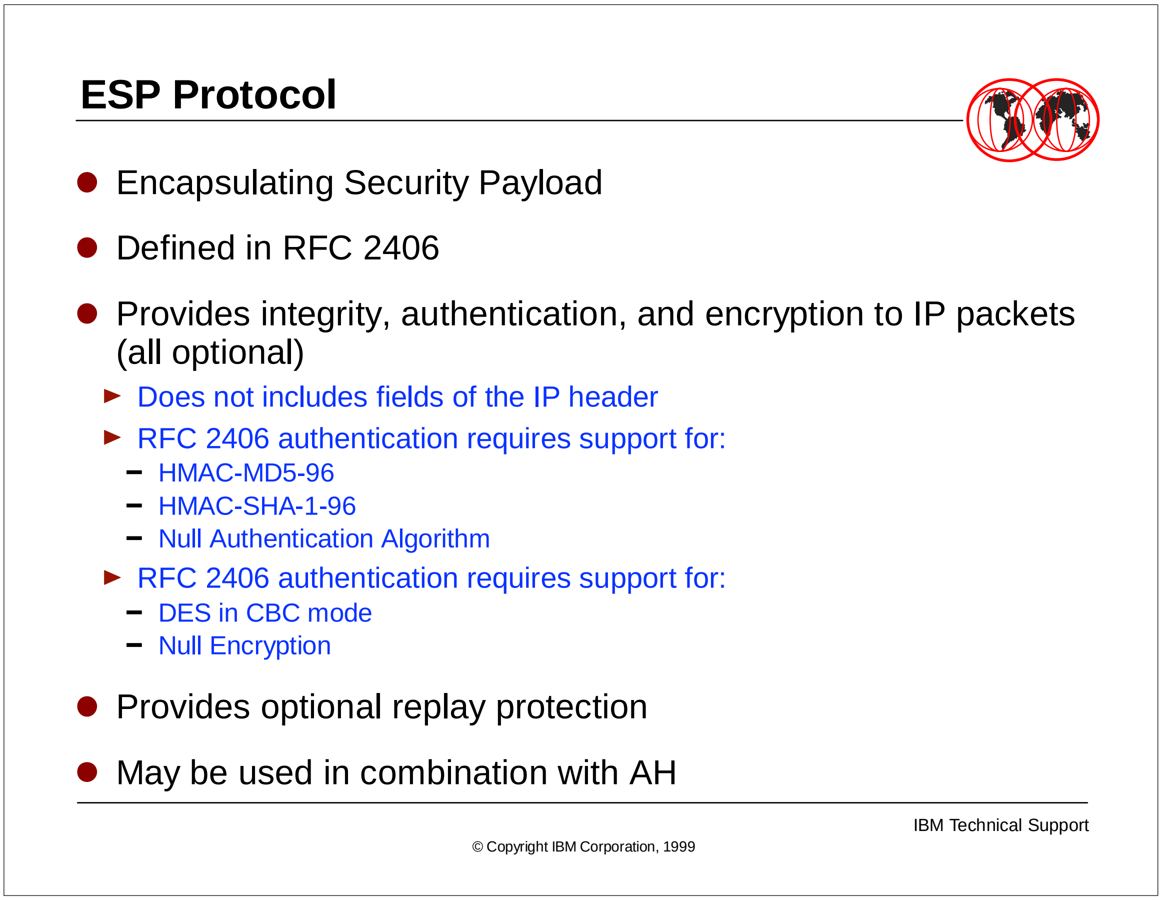
New IP Header | AH Hdr | IP Header | Payload (Transport Packet)

Authenticated

(except mutable fields in outer IP header)

IBM Technical Support

- This is what AH in tunnel mode looks like. Note the placement of the AH header between a new IP header and the original IP packet.
- Also note the that everything is authenticated except the mutable fields in the outer IP header.



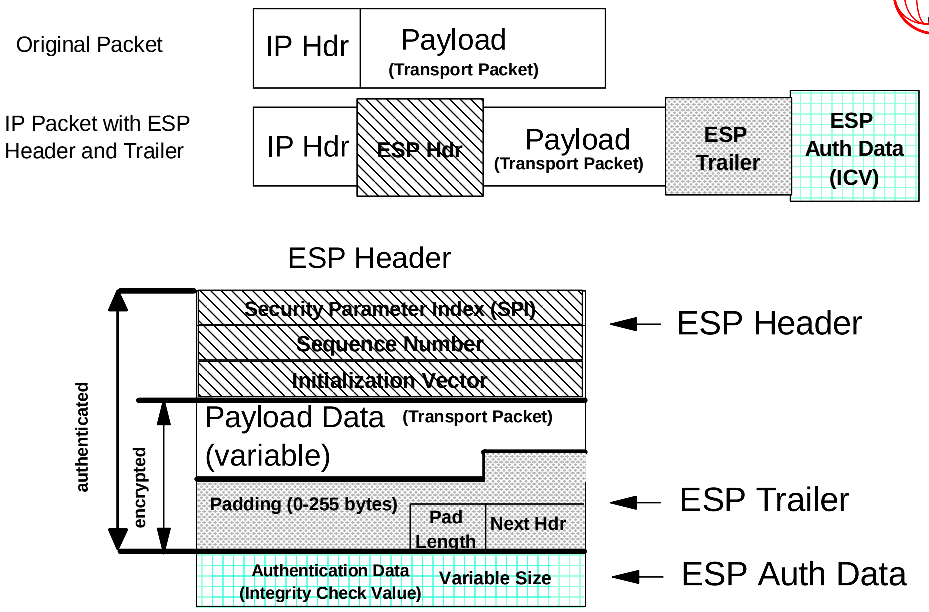
ESP Protocol

- Encapsulating Security Payload
- Defined in RFC 2406
- Provides integrity, authentication, and encryption to IP packets (all optional)
  - Does not includes fields of the IP header
  - RFC 2406 authentication requires support for:
    - HMAC-MD5-96
    - HMAC-SHA-1-96
    - Null Authentication Algorithm
  - RFC 2406 authentication requires support for:
    - DES in CBC mode
    - Null Encryption
- Provides optional replay protection
- May be used in combination with AH

IBM Technical Support

- Another way that IPSec improves the security of IP is defined in RFC 2406. RFC 2406 defines the Encapsulating Security Payload Protocol. This protocol adds integrity to an IP packet by adding an Integrity Check Value (ICV). An ICV is a cryptographic checksum. Part of an IPSec SA contains the information needed create and validate the ICV. Input to the ICV calculation is either the transport packet or IP packet (depending on the mode of the SA) and the ESP IPSec header. The inclusion of such a value makes it computationally infeasible for someone to modify the IPSec packet.
- A sequence number is included in the ESP header. The ESP header is included in the ICV calculation. The sequence number in the ESP header provides a mechanism to detect replays. The sender of an IPSec packet is required to send this field, but the receiver of an IPSec packet is not required to process it. To that extent the replay protection is considered optional.
- The ICV is cryptographically generated. The ability to properly process the ICV provides a method of data origin authentication (i.e. the data had to come from somebody that possessed the proper cryptographic key).
- Encryption is applied to the portion of the original IP packet that is being encapsulated. That is in tunnel mode the entire IP packet is encrypted. In transport mode only the transport packet of the original IP packet is encrypted.
- The security services to be provided by the ESP protocol are selectable with a few restrictions. Integrity and authentication must either both be selected or not selected. If replay protection is selected, then authentication and integrity must also be selected.
- Both the AH protocol and the ESP protocol maybe be used to protect the same IP packet.
- There are additional RFCs relative to the encryption algorithms that can be used. The two

ESP in Transport Mode



Original Packet

IP Hdr | Payload (Transport Packet)

IP Packet with ESP Header and Trailer

IP Hdr | ESP Hdr | Payload (Transport Packet) | ESP Trailer | ESP Auth Data (ICV)

ESP Header

Security Parameter Index (SPI)

Sequence Number

Initialization Vector

Payload Data (Transport Packet)
(variable)

Padding (0-255 bytes)

Pad Length

Next Hdr

Authentication Data
(Integrity Check Value)

Variable Size

authenticated

encrypted

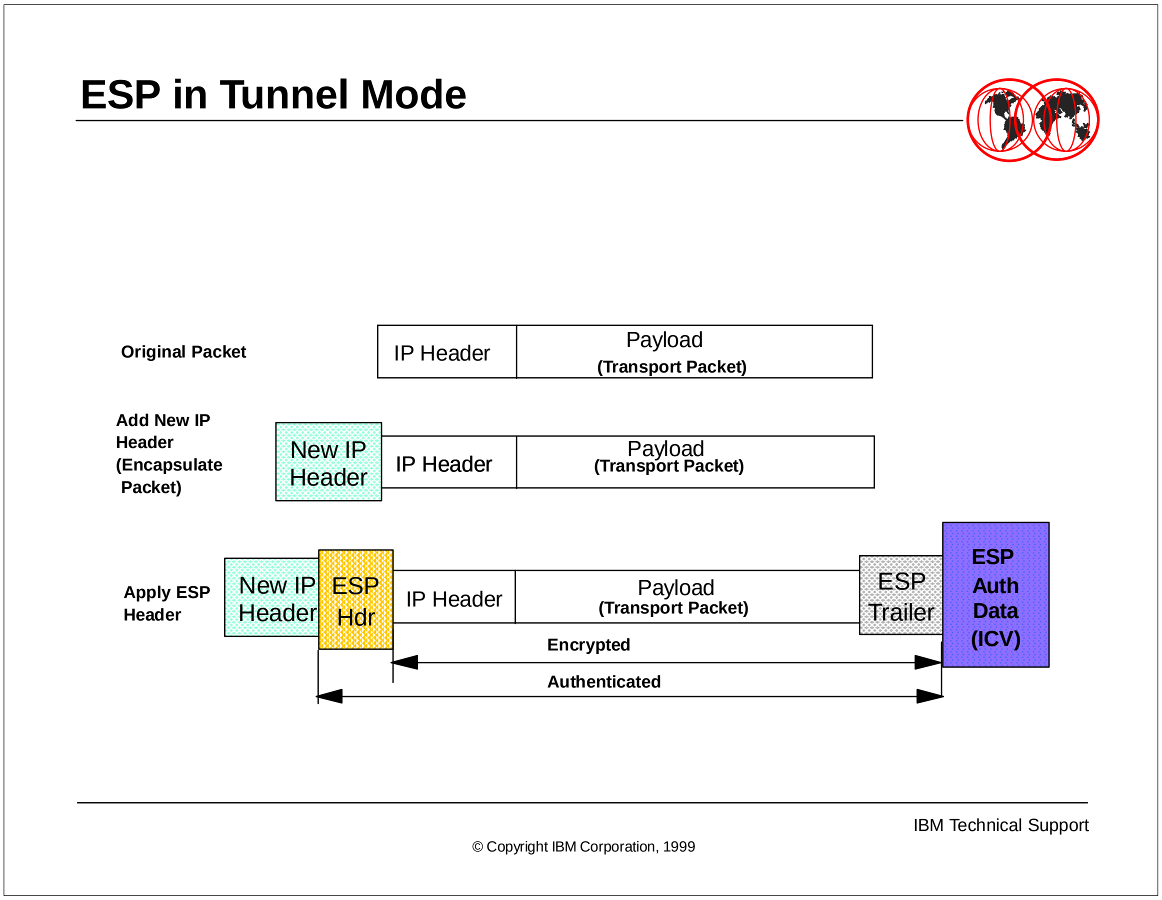
ESP Header

ESP Trailer

ESP Auth Data

IBM Technical Support

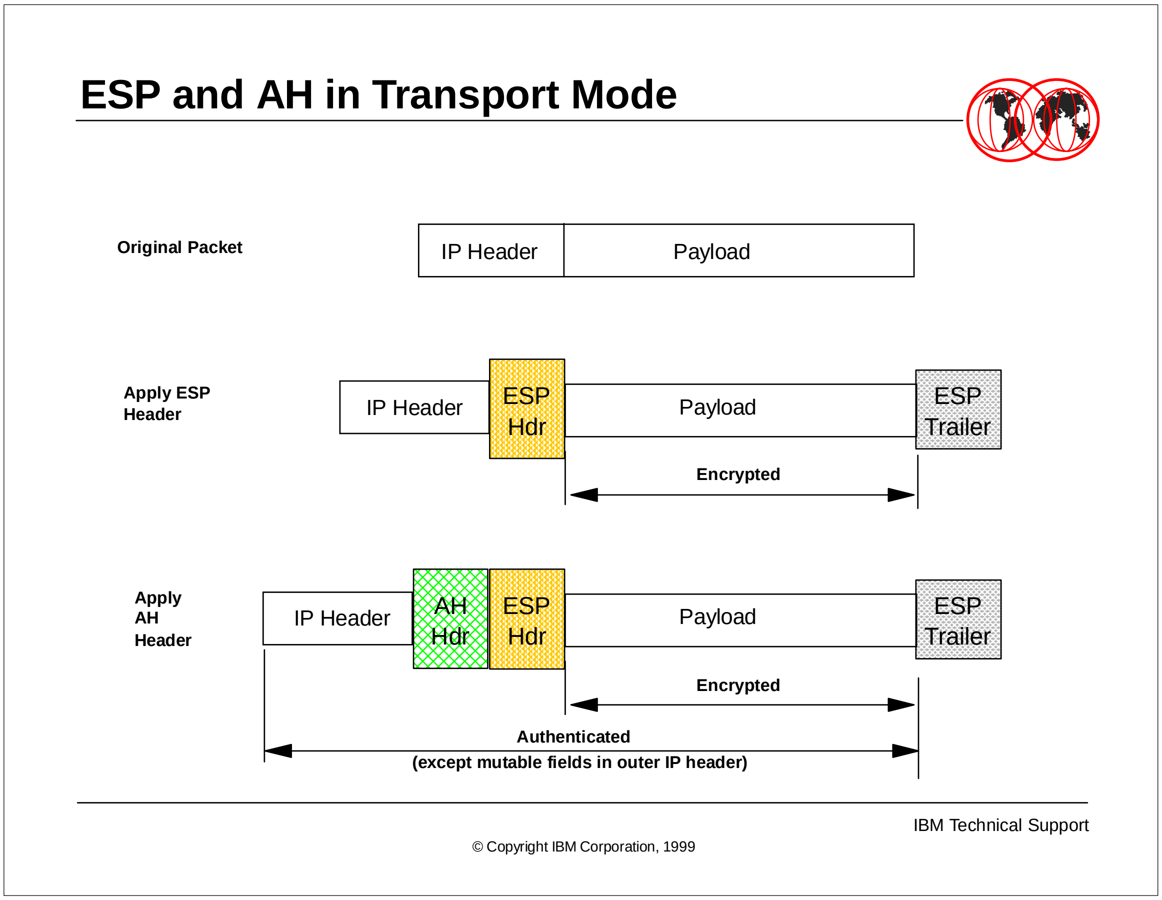
- This is what ESP in transport mode looks like. Note the placement of the ESP header between the IP header and the transport packet.
- In addition to a header, the ESP protocol has a trailer followed by the ICV. The next header and ICV are similar to the next header and ICV fields in the AH header. The padding field is there to satisfy encryption algorithms that require a fixed size block of input data. That is why the ESP trailer is included are part of the encrypted data.
- Notice the IP header is not authenticated in this case.



ESP in Tunnel Mode

IBM Technical Support

- This is what ESP in tunnel mode looks like. Note the placement of the ESP header between a new IP header and the original IP packet.
- Also note that in this case the original IP address is both authenticated and encrypted.



ESP and AH in Transport Mode

Original Packet

IP Header | Payload

Apply ESP Header

IP Header | ESP Hdr | Payload | ESP Trailer

Encrypted

Apply AH Header

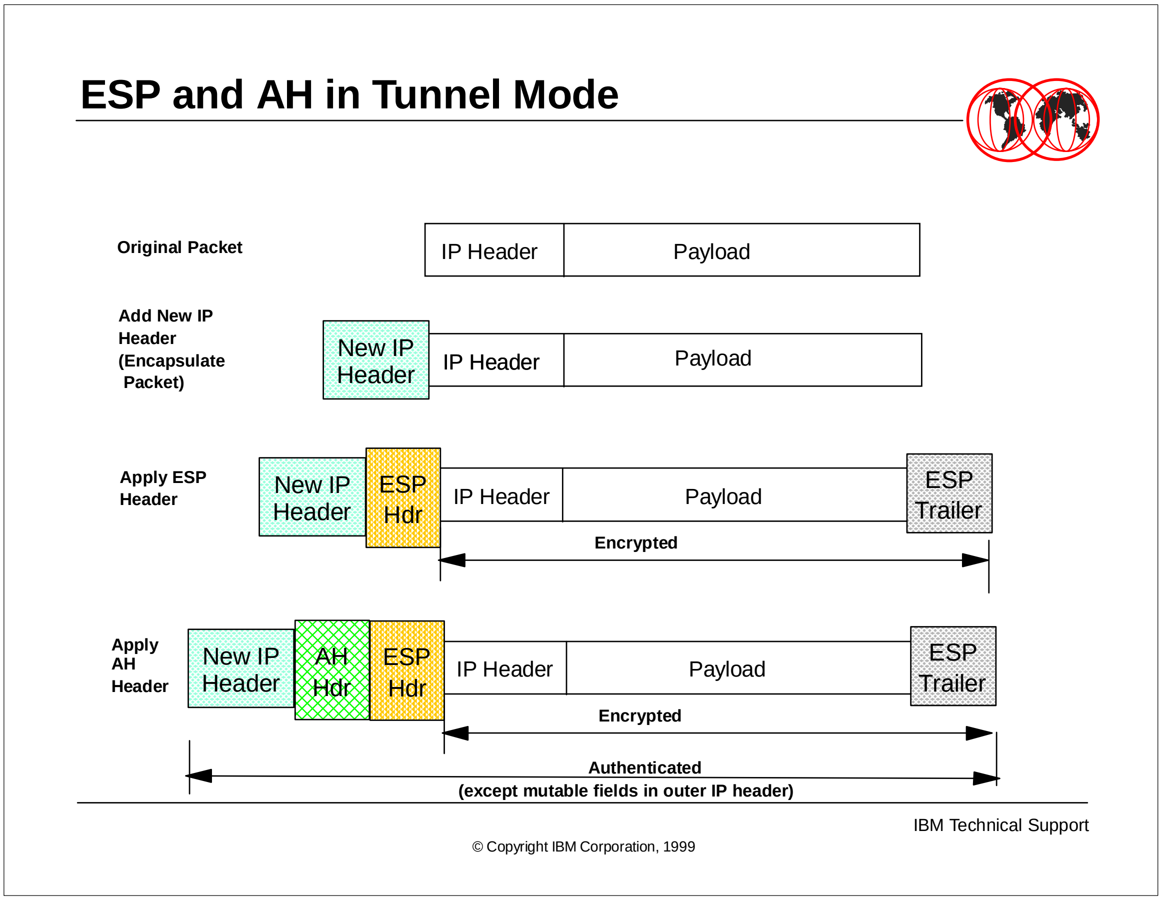
IP Header | AH Hdr | ESP Hdr | Payload | ESP Trailer

Encrypted

Authenticated
(except mutable fields in outer IP header)

IBM Technical Support

- The use of ESP and AH can be combined. In this case it does not make sense to use the authentication feature of the ESP protocol. That is why there is no ICV following the ESP trailer.
- This combination of ESP and AH represents a policy which requests that authentication be performed after encryption in transport mode.



ESP and AH in Tunnel Mode

Original Packet

IP Header | Payload

Add New IP Header (Encapsulate Packet)

New IP Header | IP Header | Payload

Apply ESP Header

New IP Header | ESP Hdr | IP Header | Payload | ESP Trailer

Encrypted

Apply AH Header

New IP Header | AH Hdr | ESP Hdr | IP Header | Payload | ESP Trailer

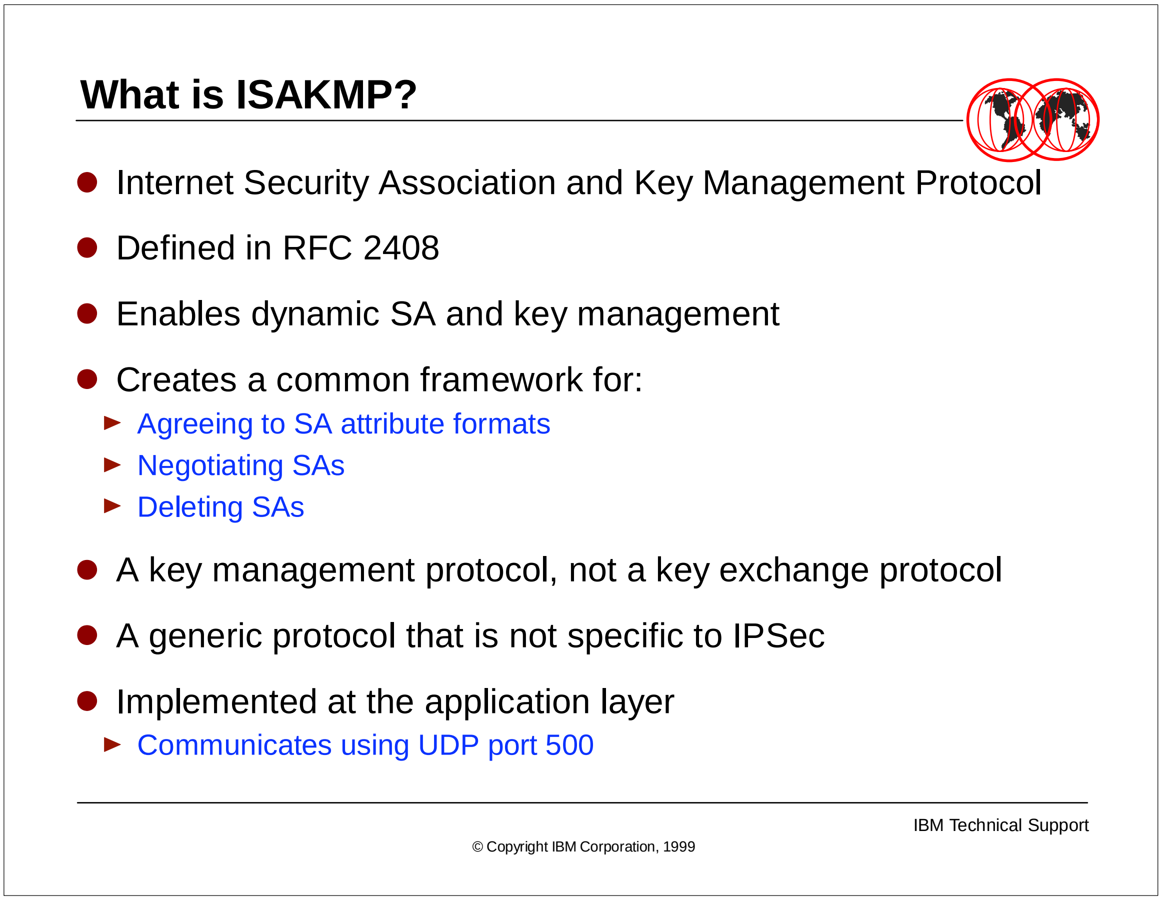
Encrypted

Authenticated
(except mutable fields in outer IP header)

IBM Technical Support

© Copyright IBM Corporation, 1999

- This combination of ESP and AH represents a policy which requests that authentication be performed after encryption in tunnel mode.

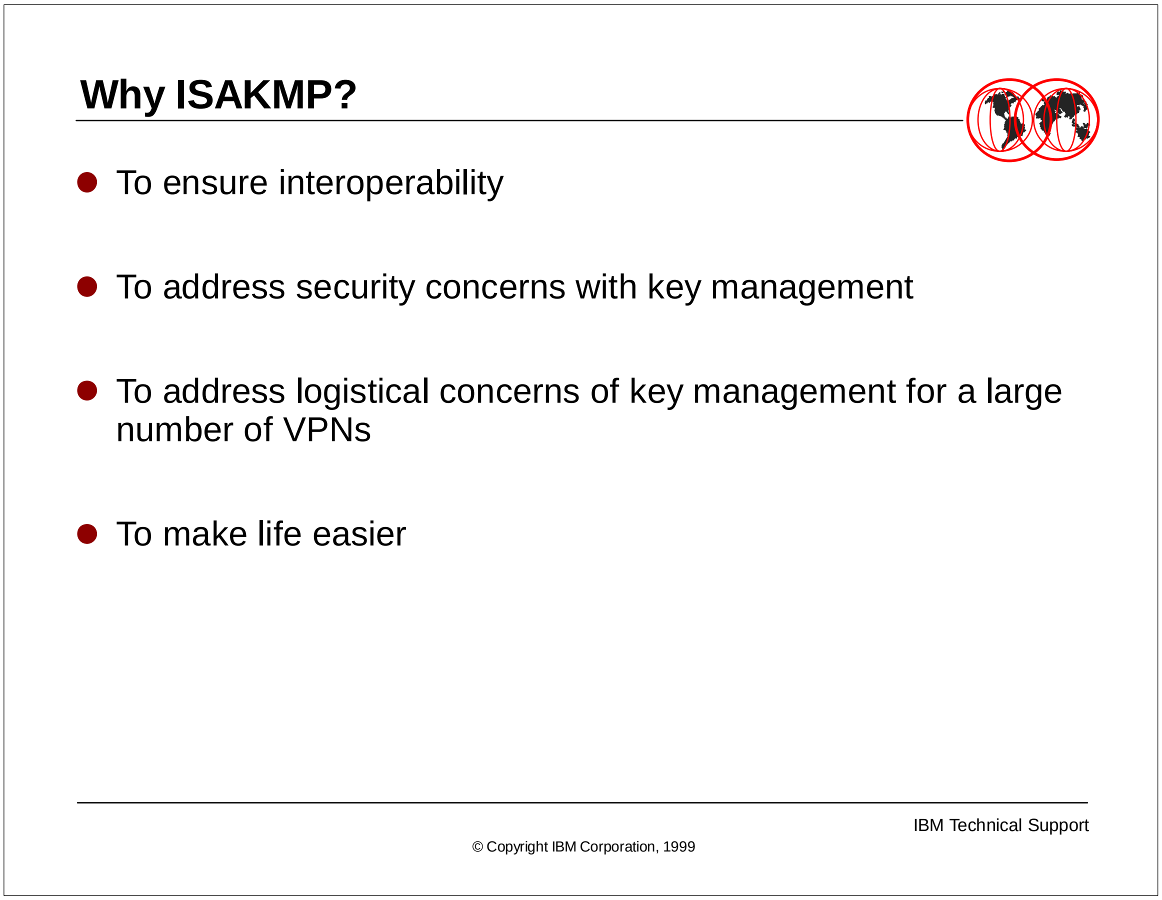


What is ISAKMP?

- Internet Security Association and Key Management Protocol
- Defined in RFC 2408
- Enables dynamic SA and key management
- Creates a common framework for:
  - Agreeing to SA attribute formats
  - Negotiating SAs
  - Deleting SAs
- A key management protocol, not a key exchange protocol
- A generic protocol that is not specific to IPSec
- Implemented at the application layer
  - Communicates using UDP port 500

IBM Technical Support

- ISAKMP is an acronym for the Internet Security Association and Key Management protocol. ISAKMP is defined in RFC 2408.
- ISAKMP does as the name suggests. It enables SA and key management procedures. It defines message formats and message flows that will allow two devices to dynamically agree to the contents of an SA that will be shared between them (including keys). ISAKMP provides the ability to authenticate the agreeing parties.
- Now is a good time introduces some ISAKMP terminology. Message formats are called payloads. The process of agreeing to an SA and dynamically generated keys is called negotiation. The defined message flows between two devices is called an exchange.
- ISAKMP does not define how to exchange keys or authenticate peers. It is a framework that facilitates the definition of SA attributes, negotiating SAs, modifying SAs, deleting SAs, authenticating peers, and exchanging keys. This means that ISAKMP provides the building blocks needed to do this, but does not define the specifics. That is left up to other RFCs such as RFC 2407 (The Internet IP Domain of Interpretation for ISAKMP) and RFC 2409 (The Internet Key Exchange). Both of these RFCs will be discussed in more detail.
- ISAKMP is generic or extensible. It intended to support negotiations of SAs for all layers of the TCP/IP stack. It is not specific to IPSec.
- The ISAKMP protocol is implemented at the application layer of the TCP/IP protocol. ISAKMP communications occur over UDP port 500.

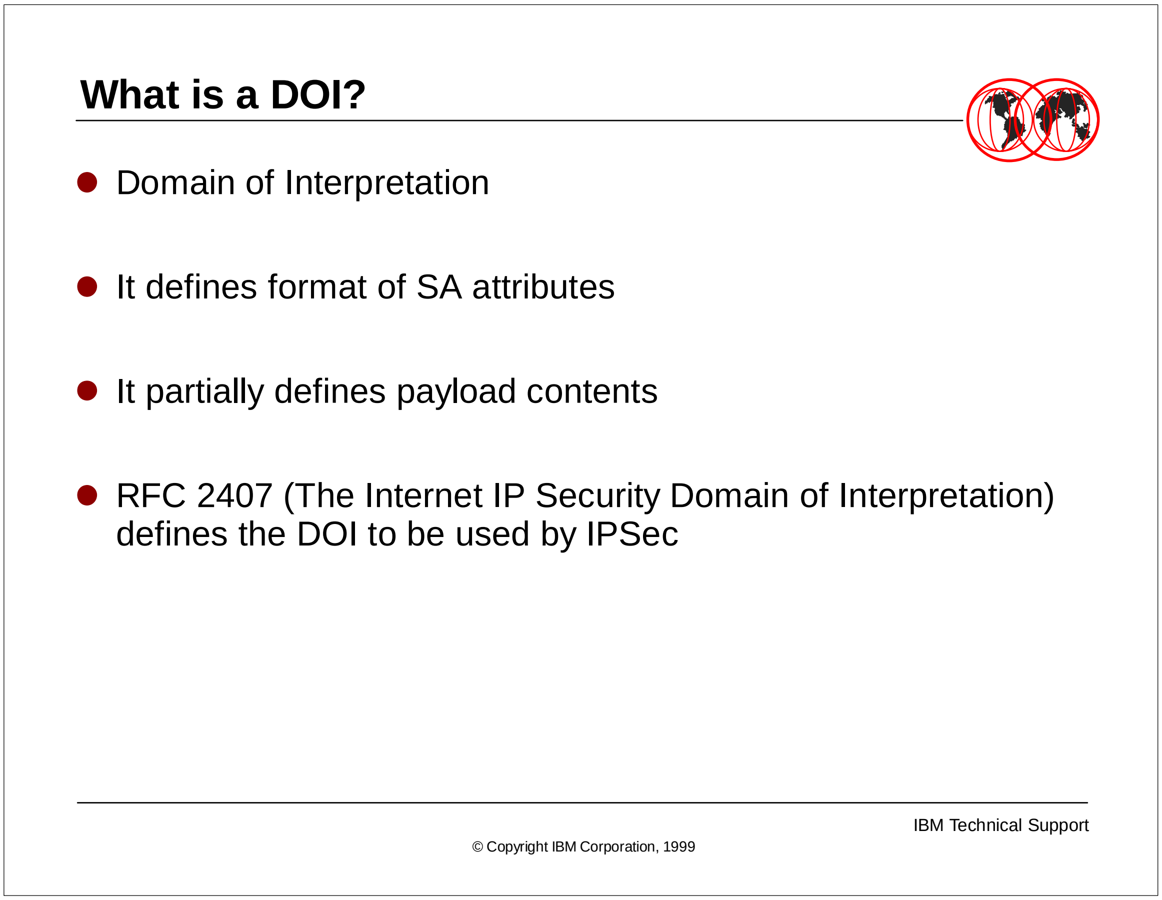


Why ISAKMP?

- To ensure interoperability
- To address security concerns with key management
- To address logistical concerns of key management for a large number of VPNs
- To make life easier

IBM Technical Support

- There are more vendors offering VPNs solutions today than there was 6 months ago. This trend is expected to continue in the future. As vendors implement their VPN solutions they will need tackle many of the same questions being addressed by ISAKMP. As the number of players in the VPN arena increases, so does the concern over ability of each vendor's implementation to work together. ISAKMP provides a framework that can insure interoperability.
- In order to use the IPSec AH and ESP protocols to create a VPN two devices must have knowledge of the cryptographic keys involved. If these keys are compromised then the entire VPN is vulnerable to attack. ISAKMP provides the framework needed to implement interoperable key exchange protocols in a secure manner.
- Manual procedures for distributing keys and SA information might be acceptable when dealing with a small number of VPNs, but as the number of VPNs being managed increases manual procedures become time consuming, costly, and error prone. Compare managing 3 VPNs where keys are to be refreshed daily to managing 100 VPNs where key are refreshed at different intervals.
- ISAKMP should make life easier. ISAKMP should facilitate the reuse of existing SA policy, meaning that once a VPN policy is setup, other VPNs should be able to use it. This policy should be reusable no matter whose ISAKMP solution your peer is using. Have to ever tried to configure a VPN where each device is using a different VPN implementation? This is not a trivial task.

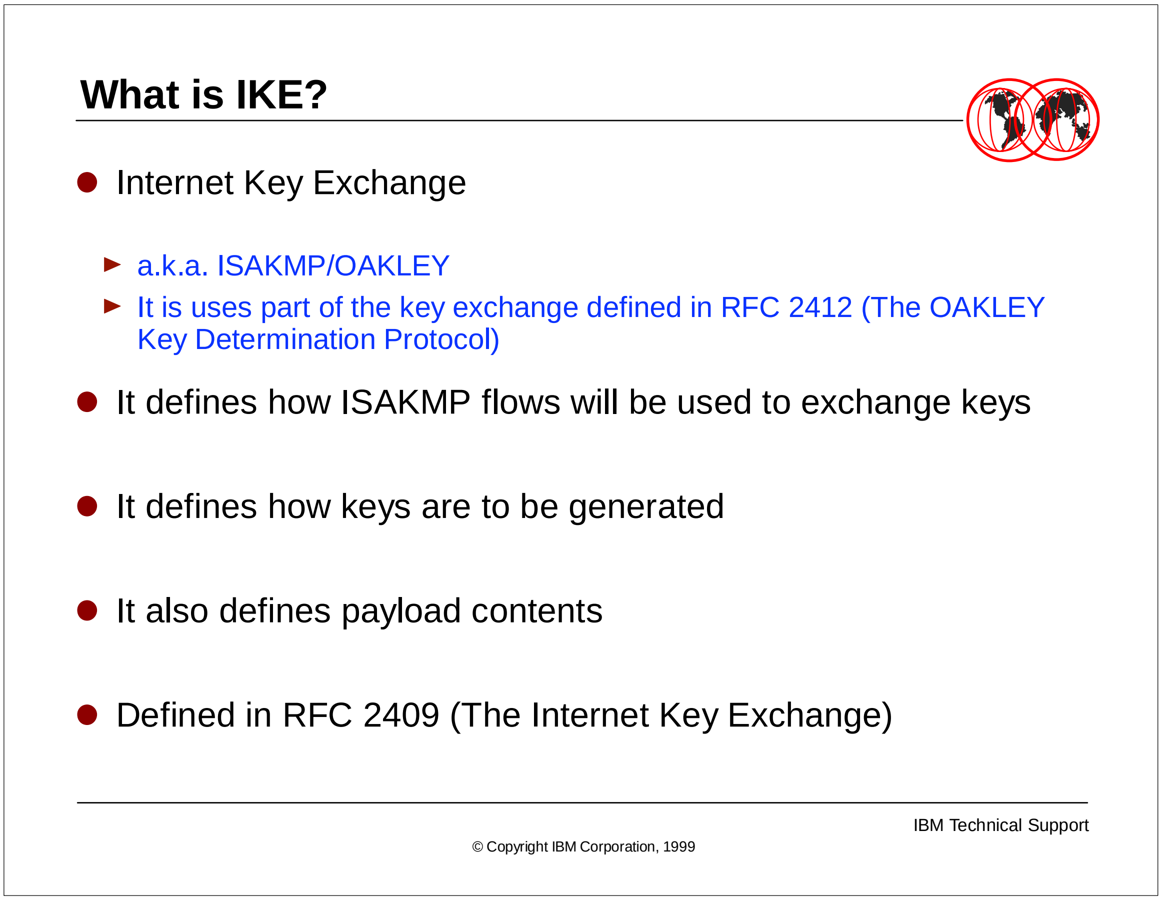


What is a DOI?

- Domain of Interpretation
- It defines format of SA attributes
- It partially defines payload contents
- RFC 2407 (The Internet IP Security Domain of Interpretation) defines the DOI to be used by IPSec

IBM Technical Support

- As mentioned previously, ISAKMP is a framework that is not specific to IPSec. The previous page indicated that ISAKMP could be used to create interoperability between IPSec implementations. How does a generic protocol such as ISAKMP help? The answer is by incorporating the concept of a DOI (Domain of Interpretation) within the protocol.
- ISAKMP provides a general framework with large holes that need to be filled by an RFC whose domain is more specific than that of ISAKMP. Think of these holes as an exercise left to the writer of the more specific RFC. The RFC written to fill these holes is called a Domain of Interpretation.
- Currently there is only one DOI defined. It is the IPSec DOI and it is documented in RFC 2407.
- One of the exercises left to the new RFC writer is to define domain specific SA attributes. Another exercise is to further define the contents of specific payloads.

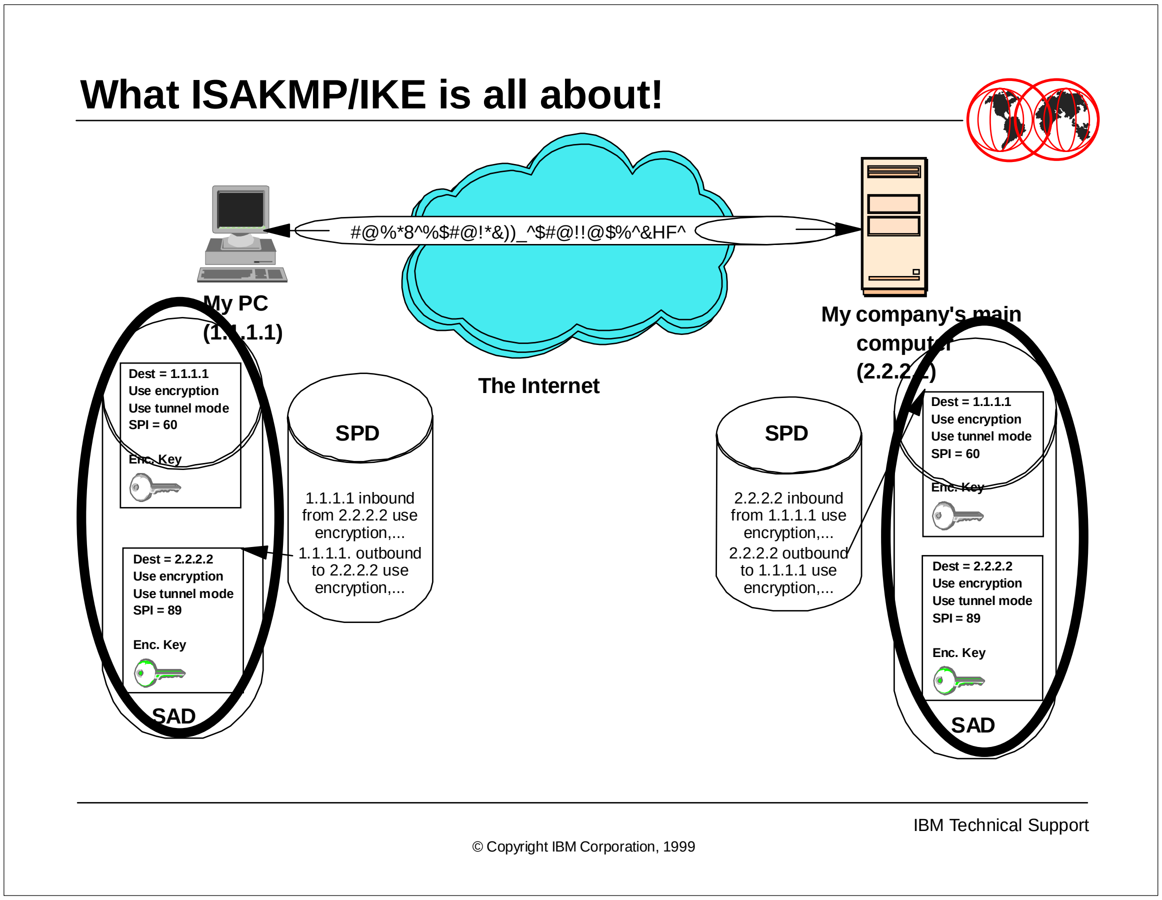


What is IKE?

- Internet Key Exchange
  - a.k.a. ISAKMP/OAKLEY
  - It is uses part of the key exchange defined in RFC 2412 (The OAKLEY Key Determination Protocol)
- It defines how ISAKMP flows will be used to exchange keys
- It defines how keys are to be generated
- It also defines payload contents
- Defined in RFC 2409 (The Internet Key Exchange)

IBM Technical Support

- As mentioned previously, ISAKMP is a framework. It does not define how keys are exchanged or how peers are authenticated. RFC 2409 (The Internet Key Exchange) defines a protocol that is used in conjunction with ISAKMP to obtain authenticated keying material for use with ISAKMP, AH, and ESP. Think of IKE as a specific implementation of ISAKMP.
- IKE is also known as ISAKMP/OAKLEY. OAKLEY itself is yet another IPSec RFC (RFC 2412). OAKLEY is a key exchange protocol. IKE uses a subset of the OAKLEY RFC and was originally called ISAKMP/OAKLEY. The name was later changed to IKE, presumably to reflect the fact that it is not a full blown OAKLEY implementation.
- IKE defines which ISAKMP flows are valid and adds payloads required to support key exchange and authentication.



What ISAKMP/IKE is all about!

#@%*8^%$#@!*&))_^$#@!!@$%^&HF^

My PC (1.1.1.1)

The Internet

My company's main computer (2.2.2.2)

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAD

SPD

1.1.1.1 inbound from 2.2.2.2 use encryption,...
1.1.1.1. outbound to 2.2.2.2 use encryption,...

SPD

2.2.2.2 inbound from 1.1.1.1 use encryption,...
2.2.2.2 outbound to 1.1.1.1 use encryption,...

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60

Enc. Key

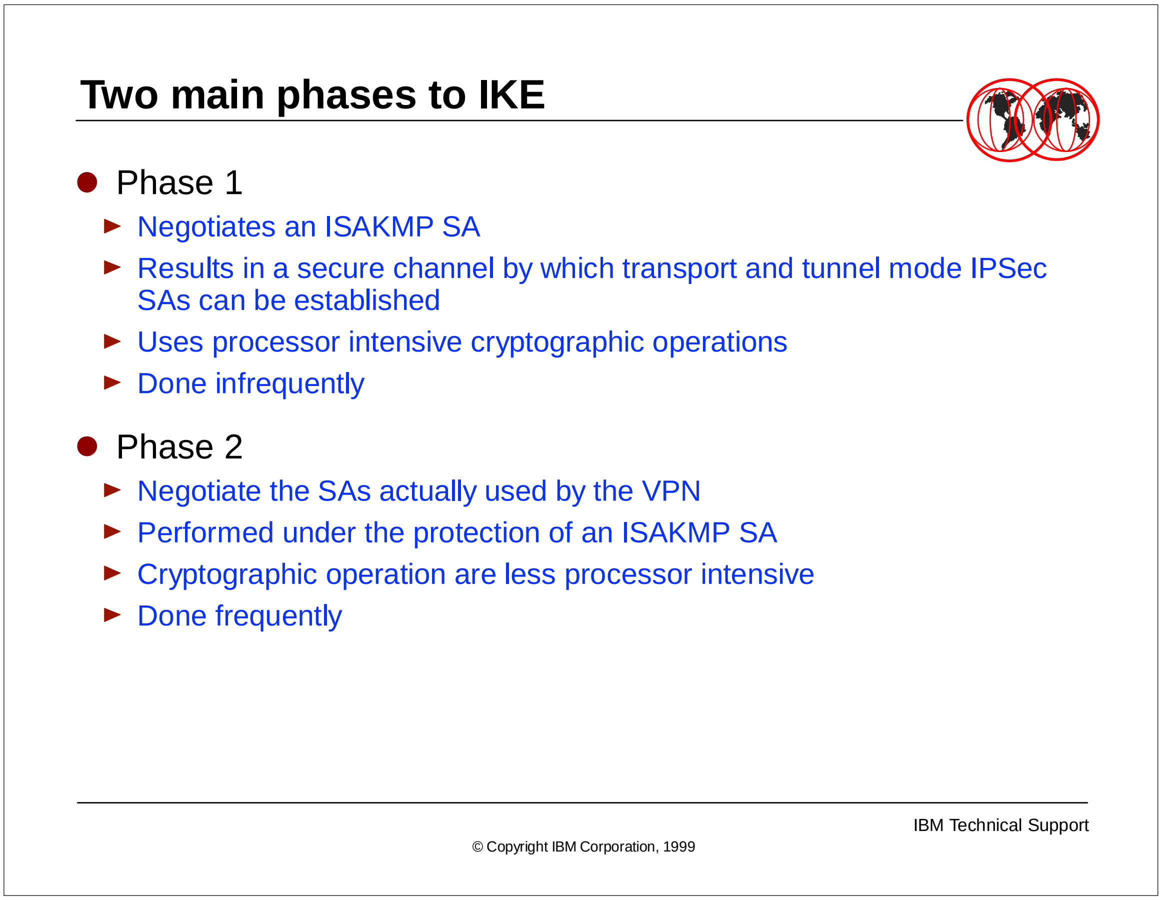
Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89

Enc. Key

SAD

IBM Technical Support

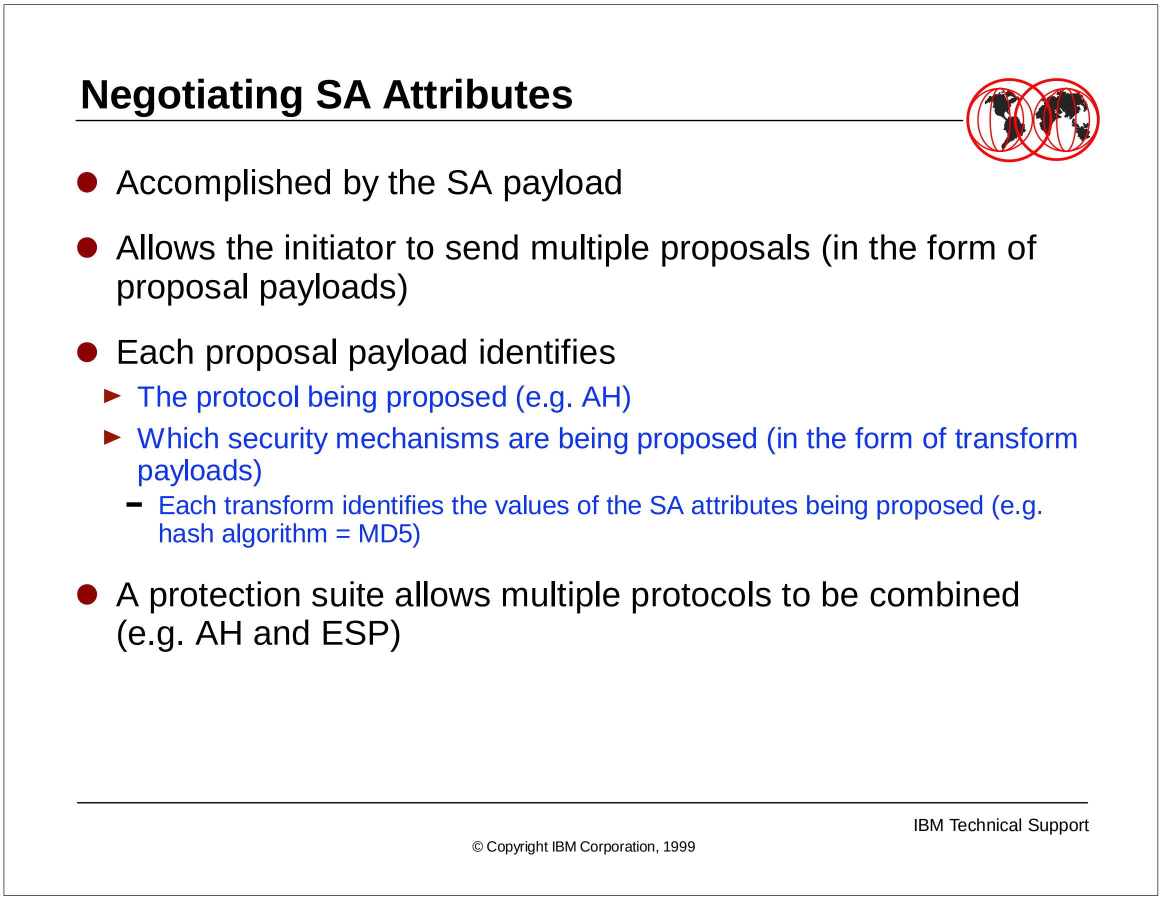
- This is a diagram to show what ISAKMP and IKE is all about. SA and keys!



Two main phases to IKE

- Phase 1
  - Negotiates an ISAKMP SA
  - Results in a secure channel by which transport and tunnel mode IPSec SAs can be established
  - Uses processor intensive cryptographic operations
  - Done infrequently
- Phase 2
  - Negotiate the SAs actually used by the VPN
  - Performed under the protection of an ISAKMP SA
  - Cryptographic operation are less processor intensive
  - Done frequently

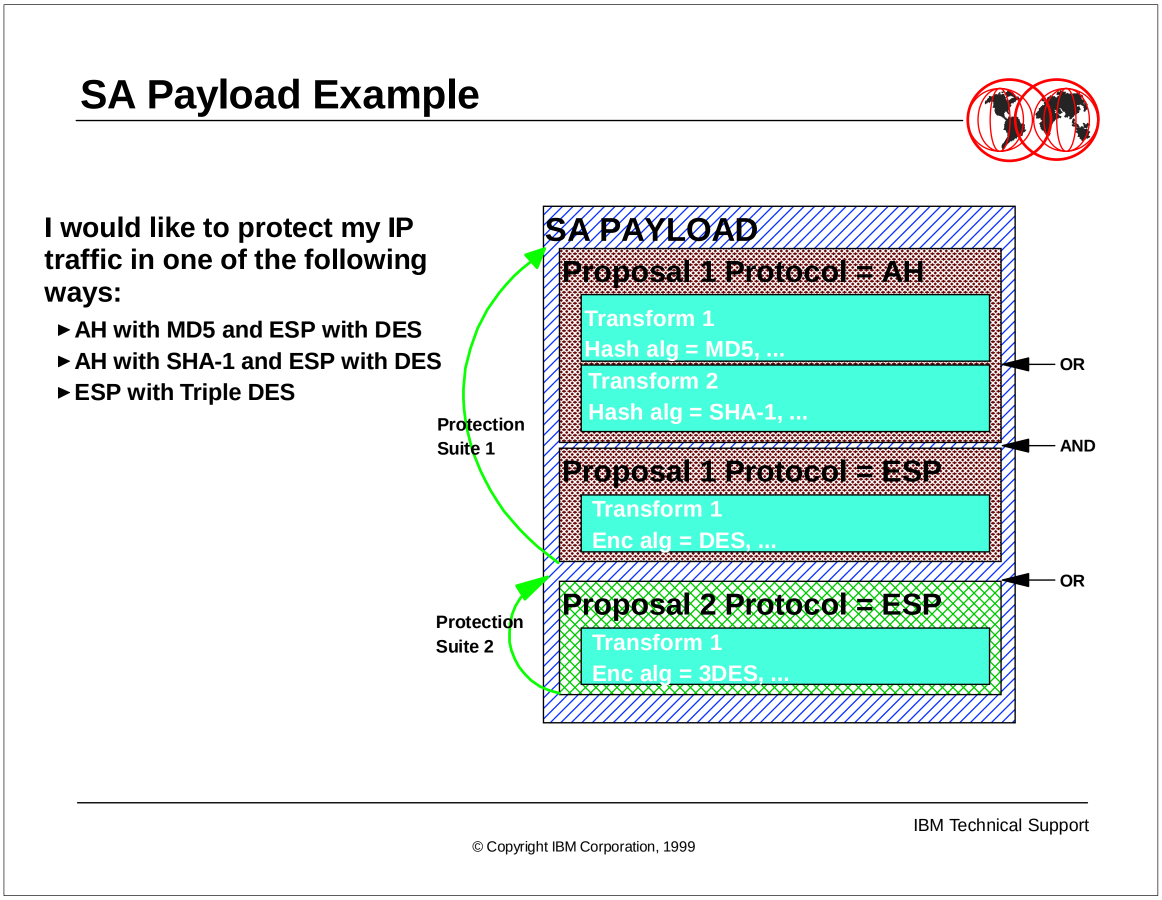
- The ISAKMP/IKE protocol uses a two phase approach. In the first phase an ISAKMP SA is created. The ISAKMP SA contains all the information needed to protect phase 2 negotiations. In a few pages the ISAKMP SA will be discussed in more detail.
- The cryptographic operations performed in phase 1 are processor intensive. A Diffie-Hellman exchange is performed and the result is used to generate keying material. If certificates are being used for authentication purposes addition public key encryption operation are performed.
- Phase 1 negotiations between 2 IPSec devices are performed infrequently. Perhaps once a day or once a week.
- In the second phase general purpose IPSec SAs are created. These are the SAs that will be used by IPSec to protect IP packets. Phase 2 exchanges are performed under the protection of the keying material generated during phase 1.
- Phase 2 exchanges are performed on a regular basis. Perhaps as frequently every 10 minutes.



Negotiating SA Attributes

- Accomplished by the SA payload
- Allows the initiator to send multiple proposals (in the form of proposal payloads)
- Each proposal payload identifies
  - The protocol being proposed (e.g. AH)
  - Which security mechanisms are being proposed (in the form of transform payloads)
    - Each transform identifies the values of the SA attributes being proposed (e.g. hash algorithm = MD5)
- A protection suite allows multiple protocols to be combined (e.g. AH and ESP)

- An SA payload is used to negotiate SA attributes. The device initiating the negotiation sends a message containing an SA payload. The responding device picks one of the offers. The responding device then sends back an SA payload with the offer chosen.
- Note that by sending more than 1 proposal the initiator allows the responder to chose the attributes of the SA. If the initiator wishes to control the choice of attributes, then the initiator should only send one choice.
- The SA payload is really made up of a series of smaller payloads. The SA payload contains proposal payloads. Proposal payloads contain transform payloads.
- A proposal payload identifies which security service(s) are being offered. Currently defined security services include the AH protocol, the ESP protocol, and the IKE protocol. Each security service has a set of attributes associated with them. These attributes are defined in transform payloads.
- Proposal payloads can be grouped to form a protection suite. Proposal payloads are grouped by using the same proposal number in multiple proposal payloads. A protection suite allows multiple protocols to be combined in an offer.



SA Payload Example

I would like to protect my IP traffic in one of the following ways:

- AH with MD5 and ESP with DES
- AH with SHA-1 and ESP with DES
- ESP with Triple DES

SA PAYLOAD

Proposal 1 Protocol = AH

Transform 1
Hash alg = MD5, ...

Transform 2
Hash alg = SHA-1, ...

Proposal 1 Protocol = ESP

Transform 1
Enc alg = DES, ...

Proposal 2 Protocol = ESP

Transform 1
Enc alg = 3DES, ...

OR

AND

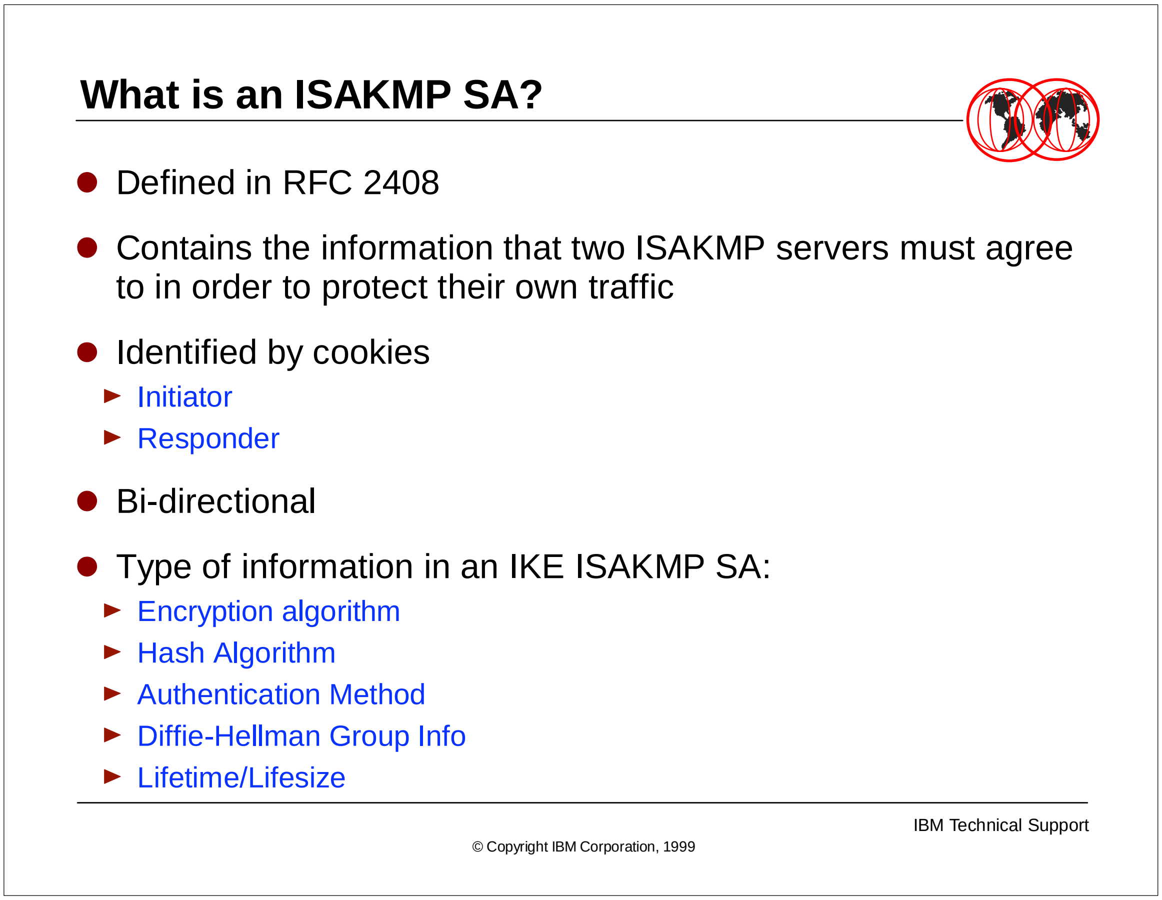
OR

Protection Suite 1

Protection Suite 2

IBM Technical Support

- This is an example of an SA payload.
- There are two proposal payloads identified as proposal 1. This forms a protection suite where the responder must pick one transform from each proposal payload in order to accept this proposal. The proposals in a protection suite are combined together using a logical AND.
- The first proposal 1 payload contains two transform payloads. Multiple transform payloads contained within a proposal payload are treated as a logical OR operation.
- The SA payload contains multiple protection suites. Multiple protection suites are also treated as a logical OR operation.



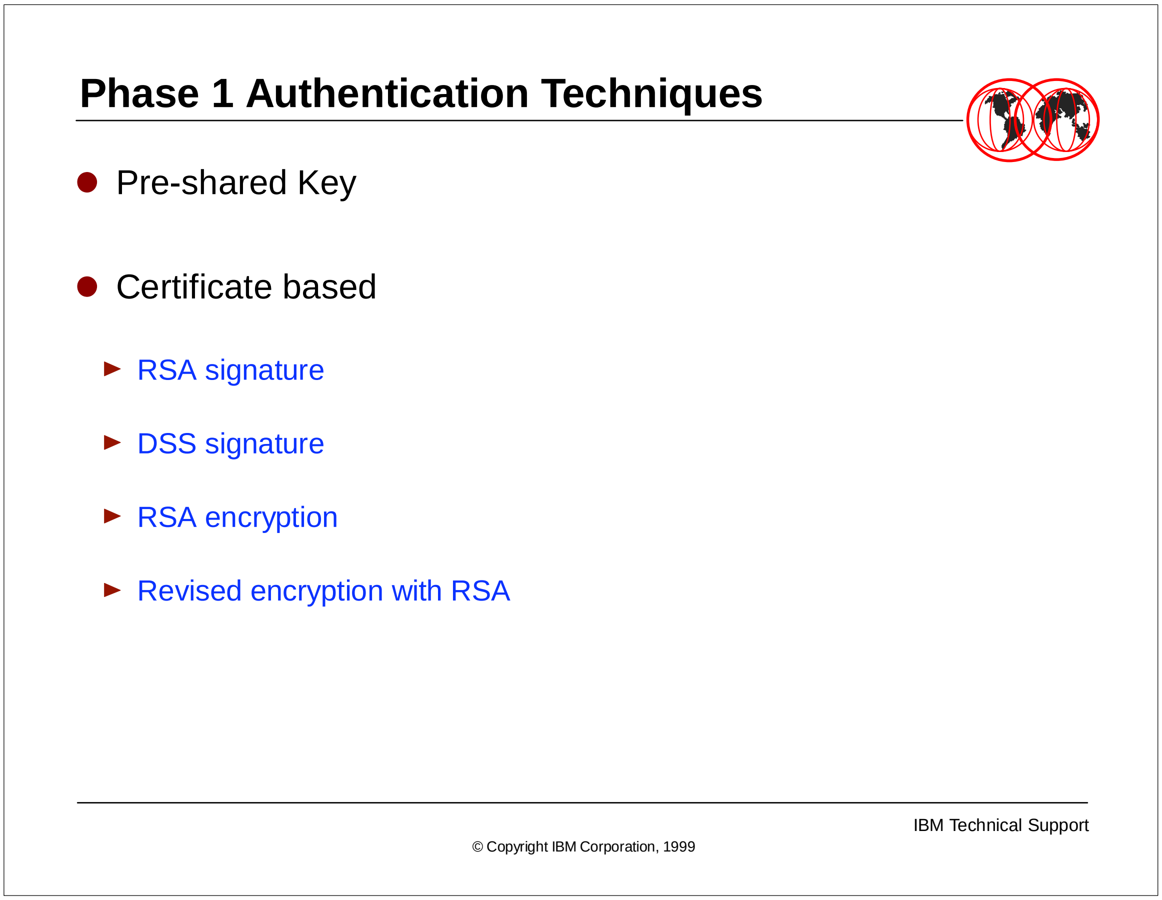
What is an ISAKMP SA?

- Defined in RFC 2408
- Contains the information that two ISAKMP servers must agree to in order to protect their own traffic
- Identified by cookies
  - Initiator
  - Responder
- Bi-directional
- Type of information in an IKE ISAKMP SA:
  - Encryption algorithm
  - Hash Algorithm
  - Authentication Method
  - Diffie-Hellman Group Info
  - Lifetime/Lifesize

IBM Technical Support

- The concept of an SA was defined in RFC 2401. Up to this point in the presentation discussions about SAs were primarily focused on IPSec SAs. The term IPSec SA was my own term that I was using when discussing SAs that are used to represent security information pertaining to the AH and ESP protocols. My definition of IPSec SA is really inaccurate. ISAKMP is another IPSec protocol and as such the ISAKMP SA is technically an IPSec SA; however, this just confuses the issue. I will continue to use the term IPSec SA to mean an SA that relates to AH and ESP SAs.
- The concept of the ISAKMP SA is defined in RFC 2408. An ISAKMP SA is the information that two ISAKMP servers must agree to in order to protect their own traffic.
- ISAKMP SAs are identified by cookie pairs. A cookie is an 8 octet value that uniquely identifies an SA. Both ISAKMP servers generate their own cookie. Both server's cookies are contained in the header of each message exchanged. The server that initiated a phase 1 exchange is known as the initiator. The server with which the exchange was initiated is known as the responder.
- ISAKMP is a generic protocol. The contents of the ISAKMP SA is dependent on things like the DOI and Key Exchange being implemented. The following is some of the negotiable information in an IKE ISAKMP SA:

❑ Encryption algorithm
❑ Hash Algorithm
❑ Authentication Method
❑ Diffie-Hellman Group Info
❑ Lifetime/Lifesize

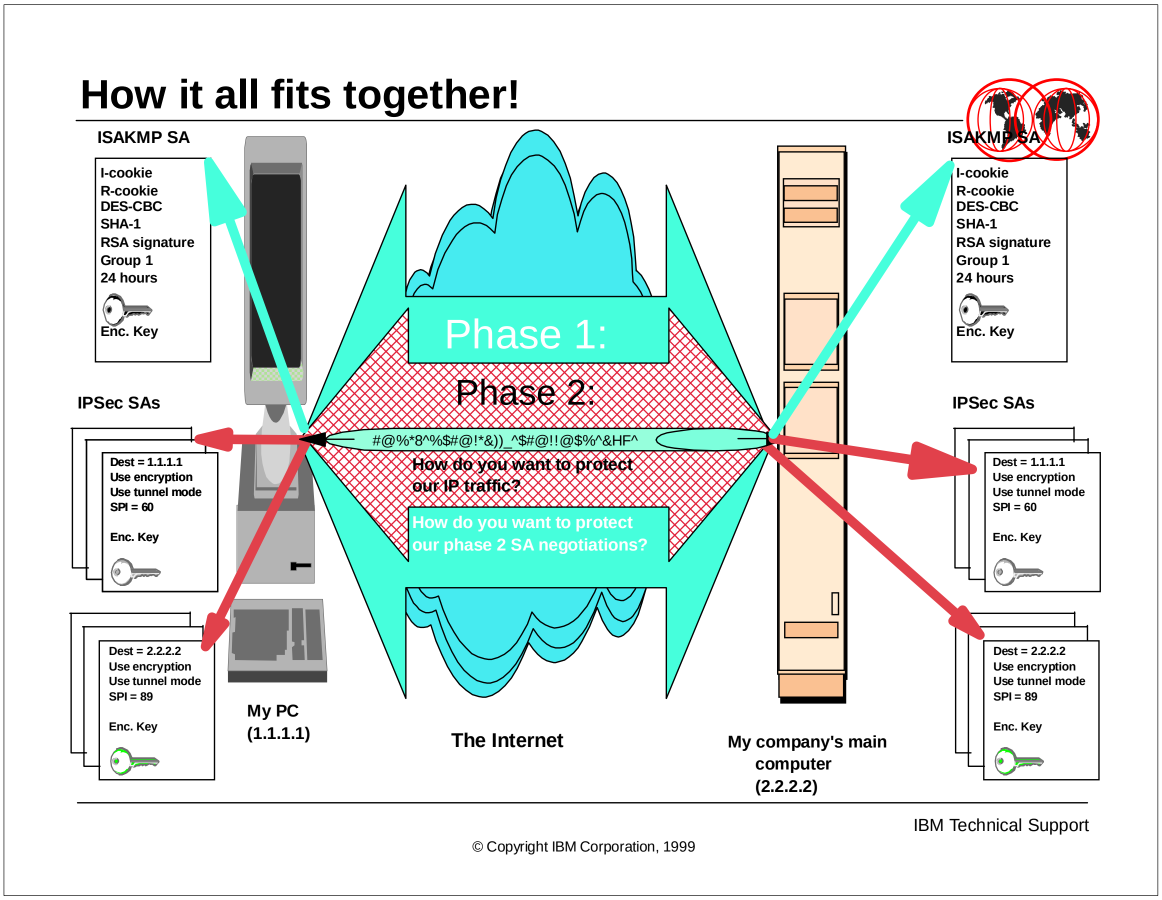


Phase 1 Authentication Techniques

- Pre-shared Key
- Certificate based
  - RSA signature
  - DSS signature
  - RSA encryption
  - Revised encryption with RSA

IBM Technical Support

- Authentication between ISAKMP servers is performed during phase 1 processing. Authentication is either pre-shared key based or certificate based.
- Pre-shared key authentication should really be called pre-shared value authentication. The basic concept is that both sides agree to a value. This value is included in the calculation that dynamically creates the secret encryption key that will be used to encrypt messages between the two servers. The ability to successfully send a message encrypted with this secret key proves the pre-shared value is known.
- A certificate is a binding between an entity and the public key portion of an asymmetrical cryptographic key pair. A third party, called a certificate authority, certifies the identify of the entity involved and that the entity does possess the associated private key.
- Certificate based authentication comes in two forms: signature based and encryption based.
- In signature based authentication each ISAKMP server signs a piece of information with their private key. If the peer can successfully verify the signature with the public key contained in the ISAKMP server's certificate, then that ISAKMP server is the entity identified by the certificate.
- The principle behind encryption based authentication is basically the same. The difference is that instead of signing a piece of information, each ISAKMP server encrypts a piece (or pieces) of information. If the peer can successfully decrypt the piece(s) of information with the public key contained in the ISAKMP server's certificate, then that ISAKMP server is the entity identified by the certificate.
- There are two encryption based authentication schemes. The main difference between them is how many public key operations are required. Normal RSA encryption requires each device to perform two public key encryption operations and two public key decryption



How it all fits together!

ISAKMP SA

I-cookie
R-cookie
DES-CBC
SHA-1
RSA signature
Group 1
24 hours
Enc. Key

IPSec SAs

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60
Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89
Enc. Key

My PC
(1.1.1.1)

Phase 1:

Phase 2:

#@%*8^%$#@!*&))_^$#@!!@$%^&HF^

How do you want to protect our IP traffic?

How do you want to protect our phase 2 SA negotiations?

The Internet

My company's main computer
(2.2.2.2)

ISAKMP SA

I-cookie
R-cookie
DES-CBC
SHA-1
RSA signature
Group 1
24 hours
Enc. Key

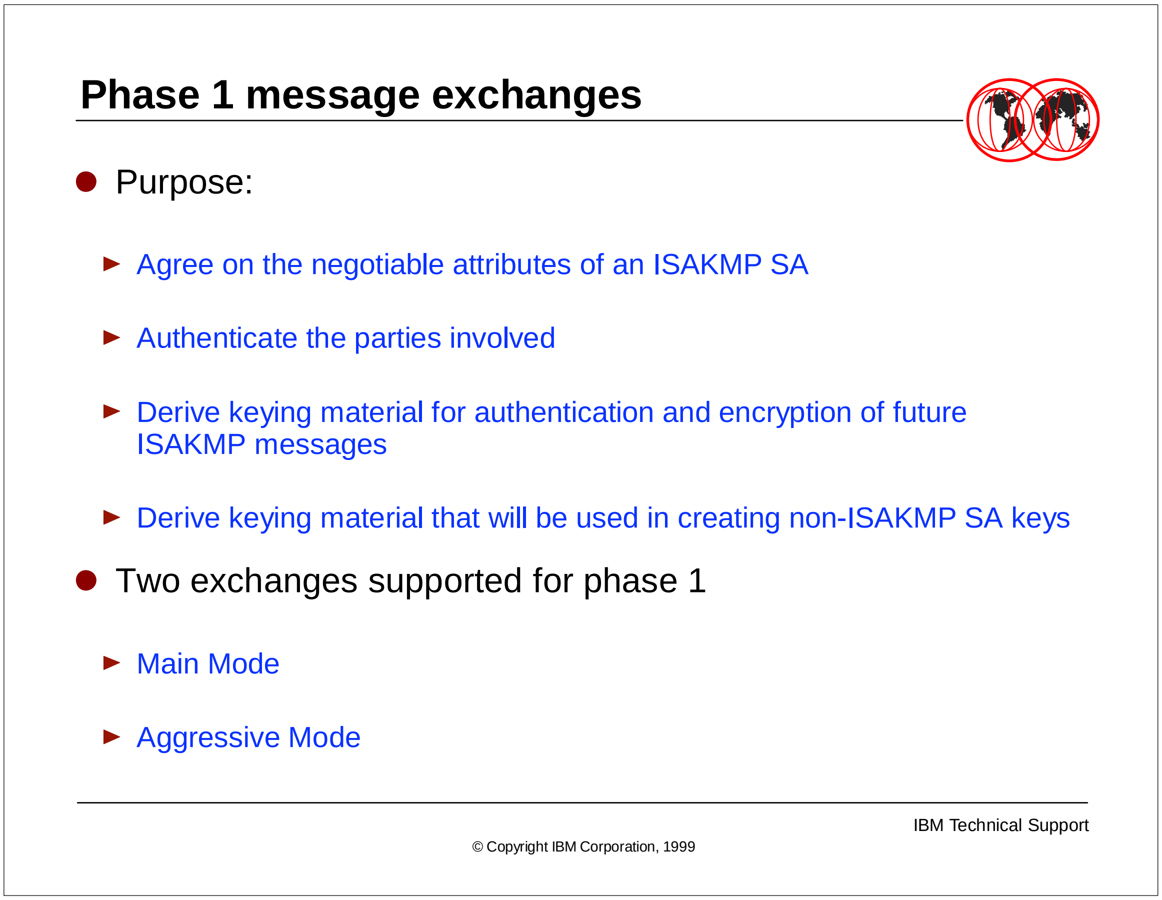
IPSec SAs

Dest = 1.1.1.1
Use encryption
Use tunnel mode
SPI = 60
Enc. Key

Dest = 2.2.2.2
Use encryption
Use tunnel mode
SPI = 89
Enc. Key

IBM Technical Support

- This diagram shows how ISAKMP and IPSec protocols such as ESP and AH all fit together. Note, this is the same example that was used in many of the previous diagrams.
- ISAKMP SAs are created as a result of phase 1 negotiations. IPSec SAs are created as a results of phase 2 negotiations.
- The ISAKMP SA has been added to the diagram. Phase 1 ISAKMP SAs protect the phase 2 SA negotiations. That is why the phase 2 arrow is contained in the phase 1 arrow. Likewise, the actual IP packets are protected using the IPSec SAs created during phase 2. This is why the VPN is contained in the phase 2 arrow.
- Many phase 2 SAs may be created under the protection of one phase 1 SA. That is why multiple copies of the IPSec SAs are shown.

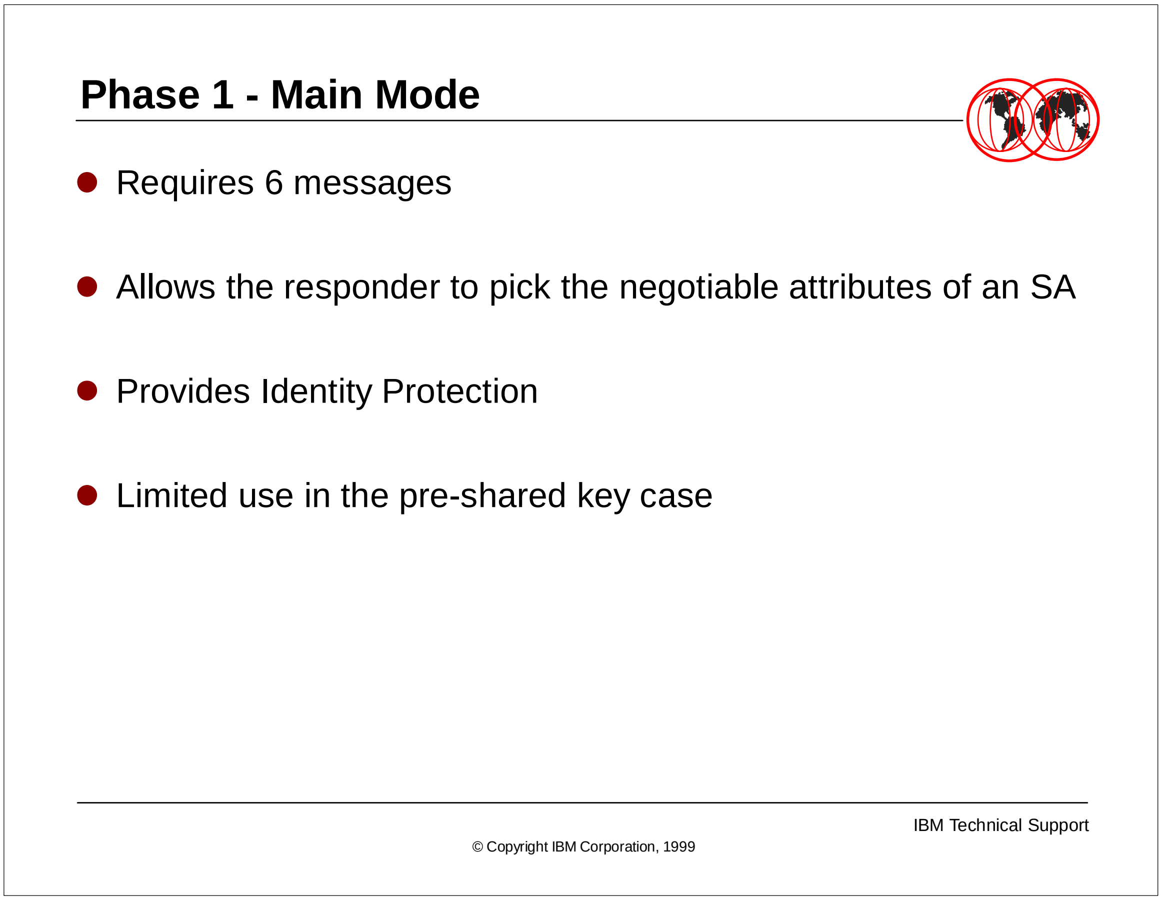


Phase 1 message exchanges

- Purpose:
  - Agree on the negotiable attributes of an ISAKMP SA
  - Authenticate the parties involved
  - Derive keying material for authentication and encryption of future ISAKMP messages
  - Derive keying material that will be used in creating non-ISAKMP SA keys
- Two exchanges supported for phase 1
  - Main Mode
  - Aggressive Mode

IBM Technical Support

- The purpose of a phase 1 exchange is to create an ISAKMP SA and to authenticate the identities involved in the negotiation.
- There are certain attributes of an SA that are negotiable. The encryption algorithm, the hash algorithm, the authentication method, Diffie-Hellman group, and lifetime/lifesize are all negotiable.
- The phase 1 exchange derives the keying material associated with the SA in a secure manner.
- There are three values calculated for use as keying material during a phase 1 negotiation. One value is used to authenticate future messages. Another value is used to encrypt future messages. The final value is used to derive keys for non-IPSec SAs.
- There are two types of phase 1 exchanges. They are main mode and aggressive mode.

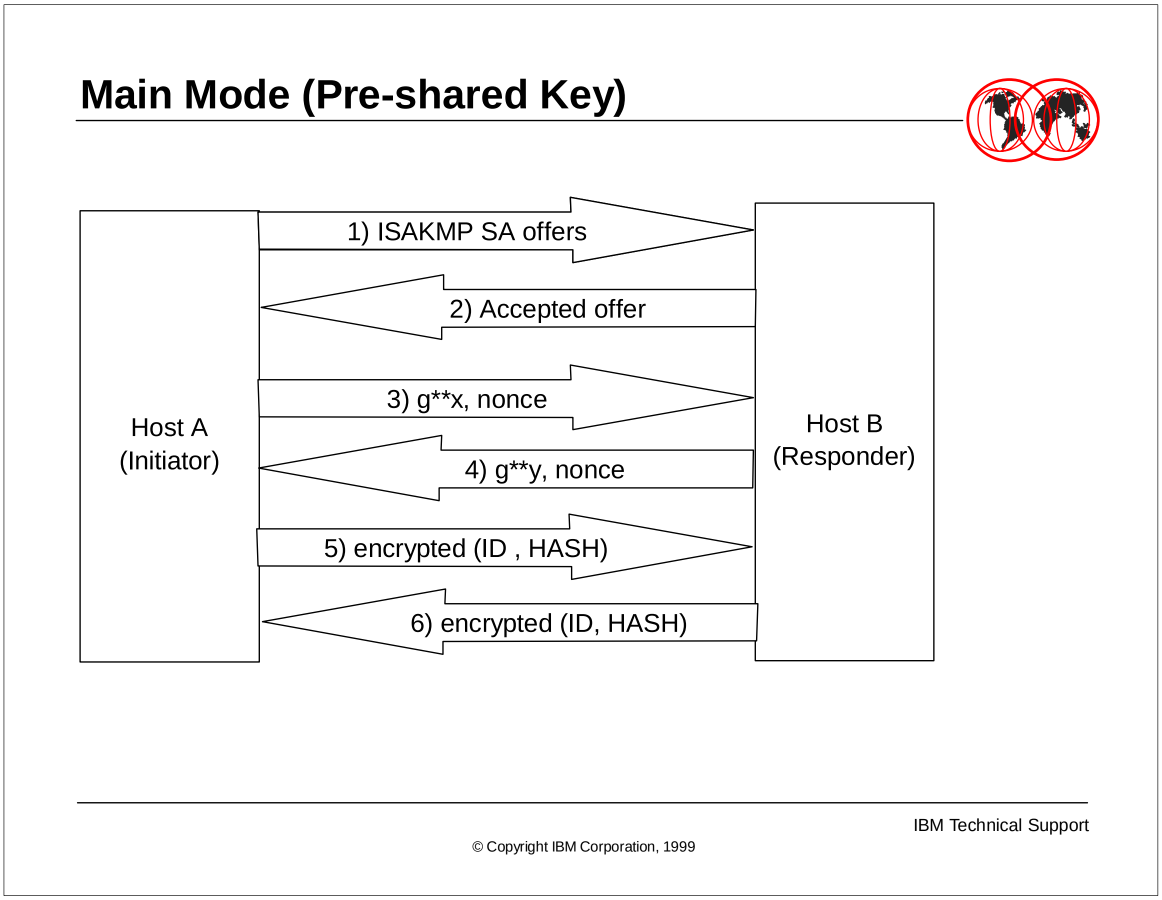


Phase 1 - Main Mode

- Requires 6 messages
- Allows the responder to pick the negotiable attributes of an SA
- Provides Identity Protection
- Limited use in the pre-shared key case

IBM Technical Support

- In a main mode exchange each ISAKMP device sends its peer 3 messages for a total of 6 messages. In main mode the identities of the parties involved are protected (i.e. sent encrypted).
- There is a chicken and the egg scenario with main mode. Phase 1 policy must be picked based on the identity of the peer's device. Policy is agreed to in messages 2 and 3. The identity of the devices is not sent until messages 5 and 6. The net is that the policy picked during message 2 of main mode processing must be rechecked when the real identity becomes known.
- A similar, but worse problem exists in pre-shared key mode. Pre-shared keys are specific to devices. The nature of pre-shared key main mode is such that the pre-shared key must be identified prior to messages 5 and 6. Unlike policy, deferring a decision until message 5 or 6 is not an option. If pre-shared main mode is to be used, the devices involved must be identified by their IP addresses.



Main Mode (Pre-shared Key)

Host A
(Initiator)

1) ISAKMP SA offers

2) Accepted offer

3) g**x, nonce

4) g**y, nonce

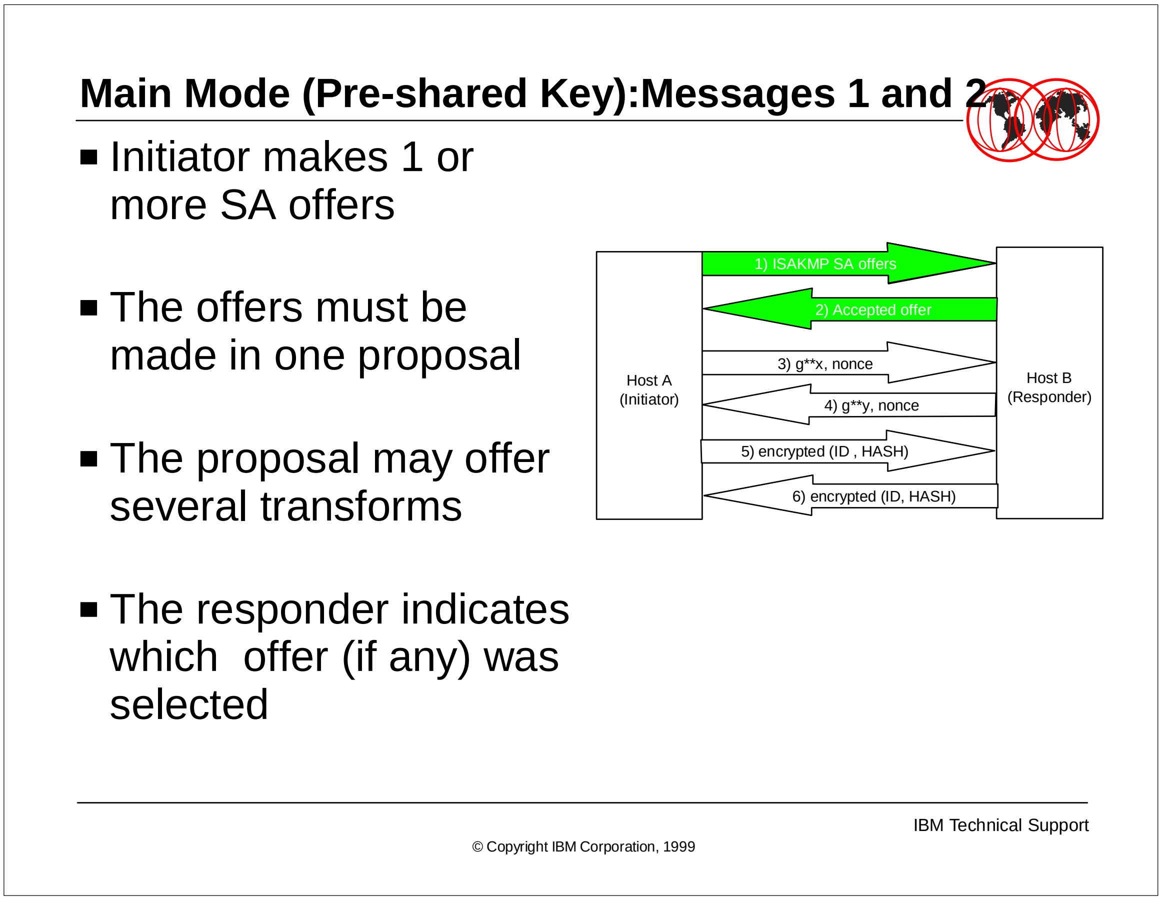
5) encrypted (ID , HASH)

6) encrypted (ID, HASH)

Host B
(Responder)

IBM Technical Support

- This is an indication of the flow of information that is sent between two ISAKMP devices during a main mode Phase 1 exchange using pre-shared key for authentication. The messages sent when a certificate based mode of authentication is used are similar.
- A detailed discussion of these messages is beyond the scope of this presentation. The next few pages will provide a high level description of each message.

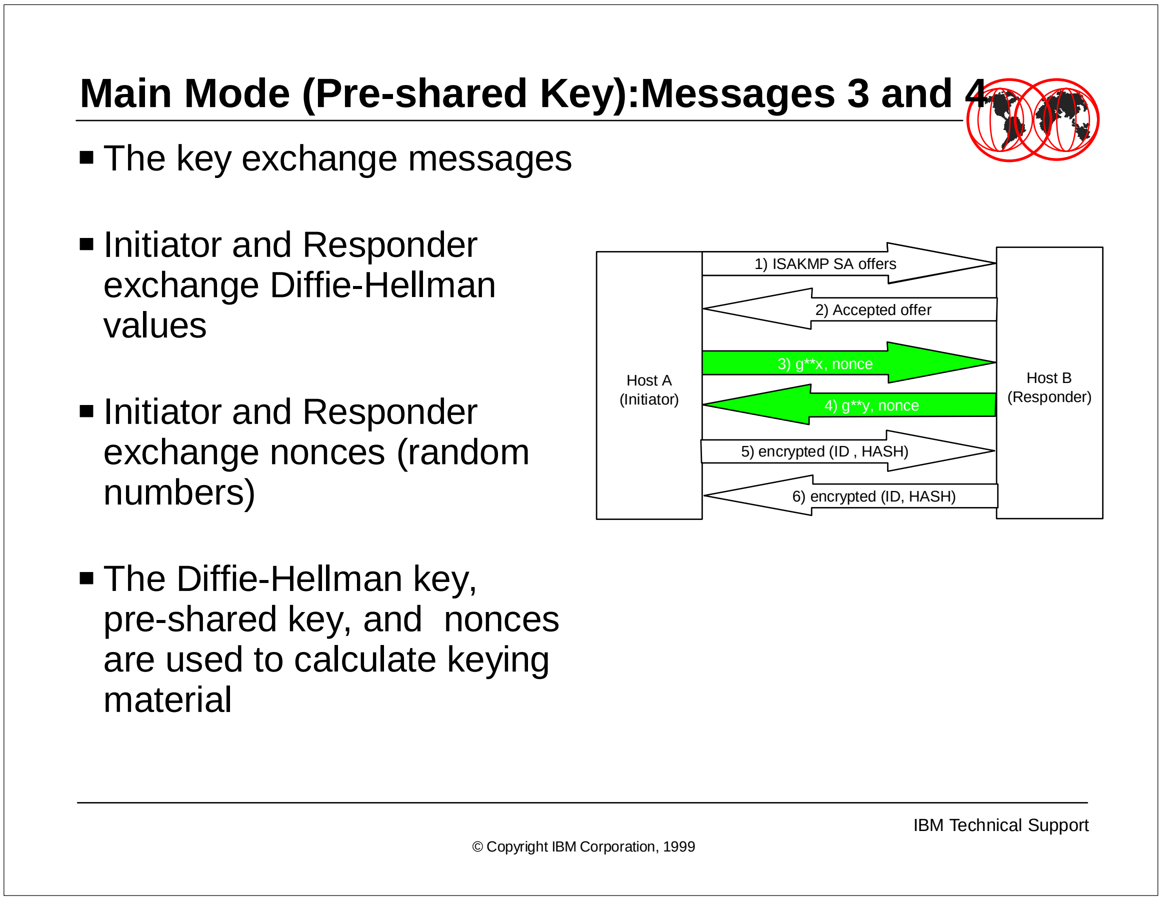


Main Mode (Pre-shared Key):Messages 1 and 2

- Initiator makes 1 or more SA offers
- The offers must be made in one proposal
- The proposal may offer several transforms
- The responder indicates which offer (if any) was selected

IBM Technical Support

- Message one kicks off the entire phase 1 negotiation. This message contains an SA payload. The SA payload must contain one proposal payload. Recall that the purpose of an IKE phase 1 negotiation is to create an IKE ISAKMP SA. By definition the protocol must be IKE and hence the justification of the one proposal restriction in the SA payload. Attribute values of the IKE ISAKMP SA can still be negotiated with the use of multiple transform payloads.
- The responder must pick one of the options or the negotiation fails. The responder may not modify the offers that were sent, with the exception of lifetime/lifesize. The responder returns the offer chosen in message 2.



Main Mode (Pre-shared Key):Messages 3 and 4

- The key exchange messages
- Initiator and Responder exchange Diffie-Hellman values
- Initiator and Responder exchange nonces (random numbers)
- The Diffie-Hellman key, pre-shared key, and nonces are used to calculate keying material

Host A
(Initiator)
1) ISAKMP SA offers
2) Accepted offer
3) g**x, nonce
4) g**y, nonce
5) encrypted (ID , HASH)
6) encrypted (ID, HASH)
Host B
(Responder)

IBM Technical Support

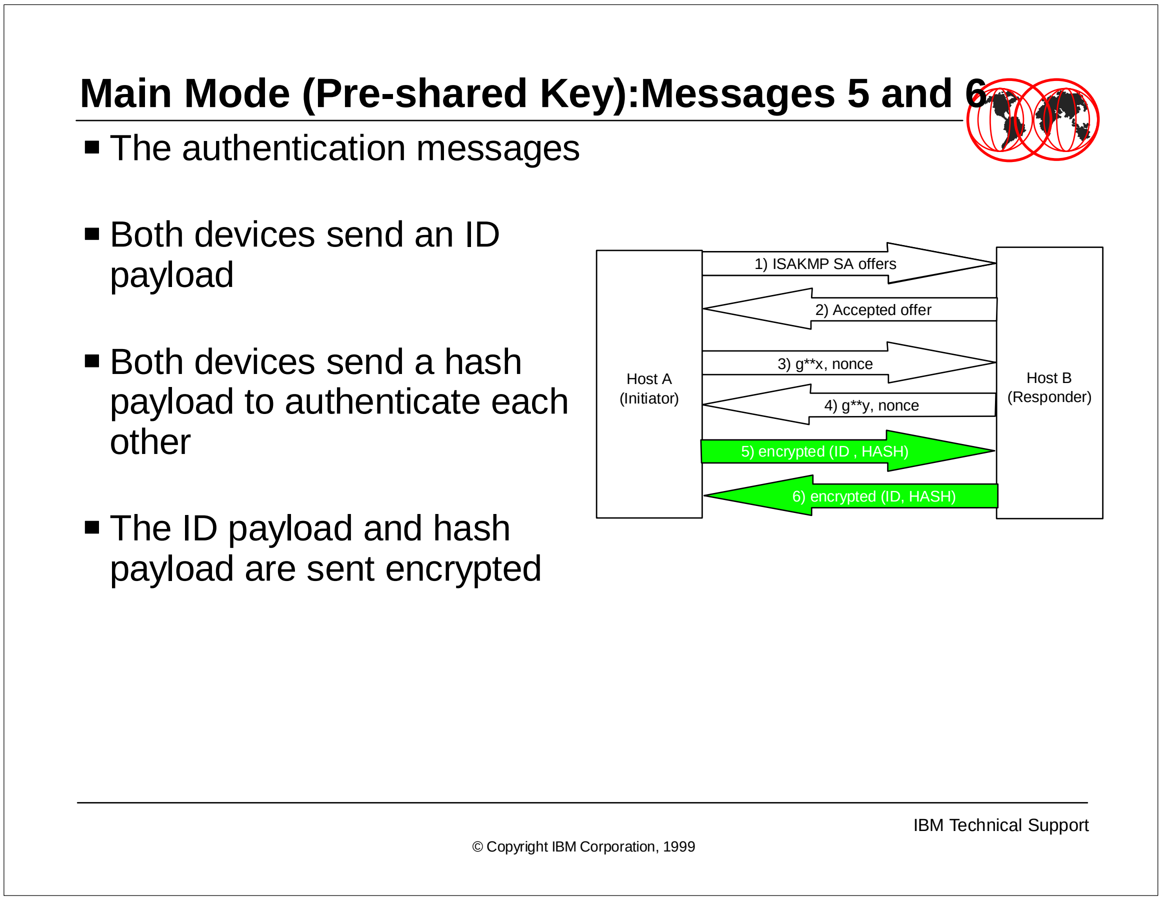
- Message 3 and 4 is where the information needed to generate keying material is exchanged. The g**x and g**y values are part of the Diffie-Hellman exchange.
- Diffie-Hellman is a public key exchange algorithm. This algorithm allows two parties to derive a shared secret key.
- The following is a quick description of the algorithm:

1. Host A and B agree to on two numbers g and n. The value for n should be a large prime number.
2. Host A picks a large random integer x and sends Host B a value A (where A = g**x mod n).
3. Host B picks a large random integer y and sends Host A a value B (where B = g**y mod n).
4. Host A calculates the Diffie-Hellman key as B**x mod n.
5. Host B calculates the Diffie-Hellman key as A**y mod n.

- This even works when small integers are used (its just easier to use a brute force attack to find the key). See the example below:

1. Pick g=10 and n=19
2. Host A picks x=3 and sends to Host B the value 10**3 mod 19 which equals 1000 mod 19 or 12.
3. Host B picks y=4 and sends to Host A the value10**4 mod 19 which equals 10000 mod 19 or 6
4. Host A calculates the Diffie-Hellman key as 6**3 mod 19 which equals 216 mod 19 or 7.
5. Host B calculates the Diffie-Hellman key as 12**4 mod 19 which equals 20736 mod 19 or 7.

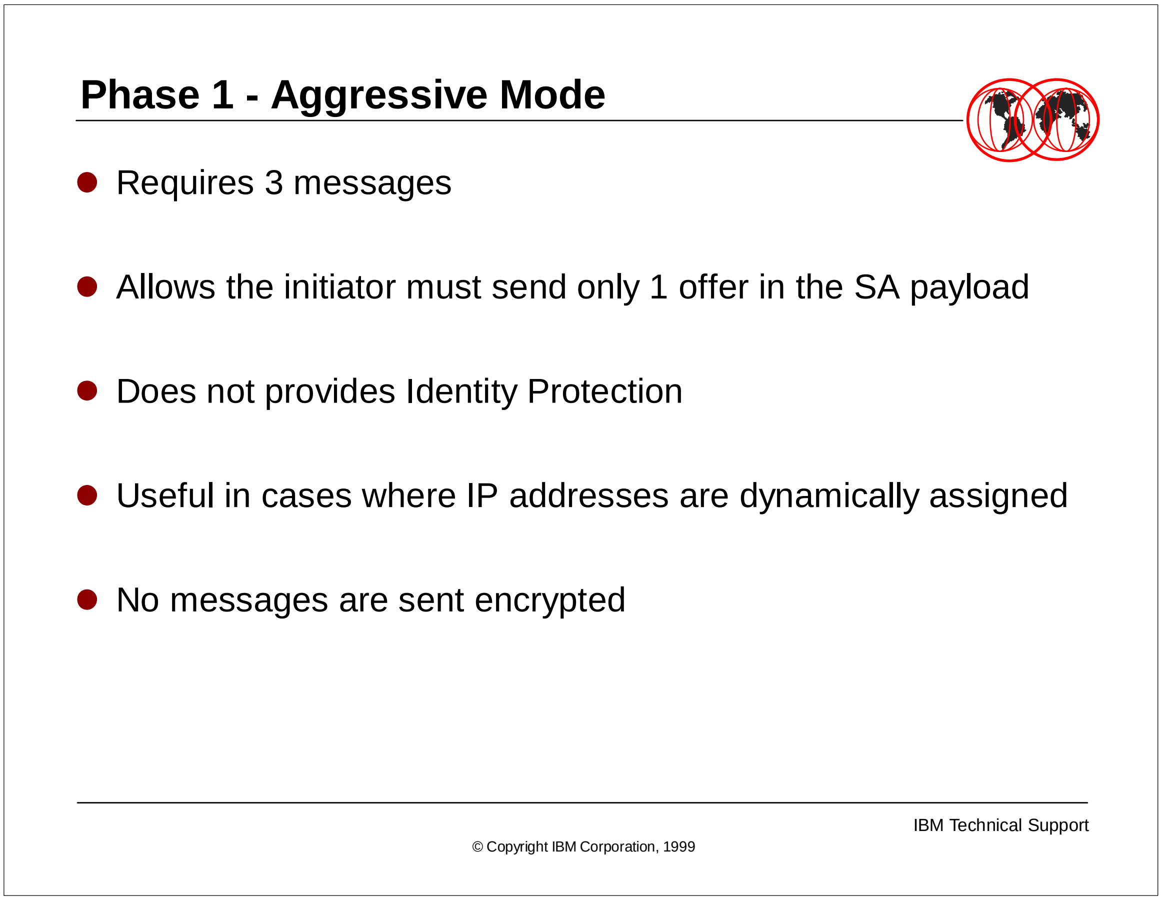
- This example might seem silly. It is. The value picked for n was a small prime number. In reality n is a very large number (768 bit or 1024 bit number).
- The values used for g and n are determined in message 1 and 2.



Main Mode (Pre-shared Key):Messages 5 and 6

- The authentication messages
- Both devices send an ID payload
- Both devices send a hash payload to authenticate each other
- The ID payload and hash payload are sent encrypted

- Messages 5 and 6 is where the information needed to identify and authenticate each device is exchanged. Both devices send an ID payload and a hash payload. The ID payload contains the identity of the device. The hash payload contains a hash of information that authenticates the identity of the device.
- At this time all the keying material has been generated. The payloads of messages 5 and 6 and are encrypted. This is how main mode provides identity protection.

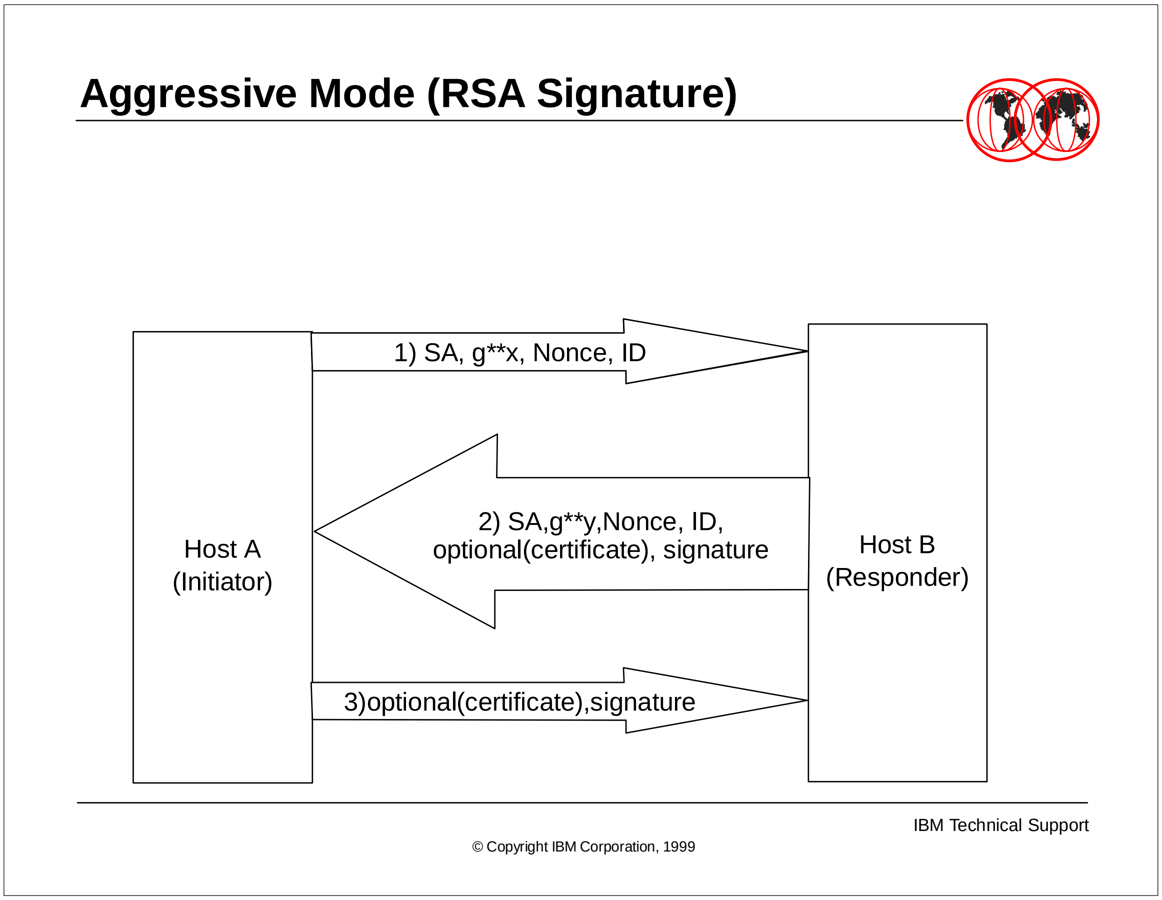


Phase 1 - Aggressive Mode

- Requires 3 messages
- Allows the initiator must send only 1 offer in the SA payload
- Does not provides Identity Protection
- Useful in cases where IP addresses are dynamically assigned
- No messages are sent encrypted

IBM Technical Support

- In an aggressive mode exchange the initiating ISAKMP device sends its peer 2 messages and the responding device sends 1 message for a total of 3 messages. In aggressive mode the identities of the parties involved are not protected (i.e. sent unencrypted).
- The identities of the devices are sent in the first two messages. This eliminates the problems dealing with locating the pre-shared key and policy that were discussed in main mode. This also allows ISAKMP devices to be known by ID types other than that of IPv4 addresses. We will discuss ID types later.
- No messages in an aggressive mode exchange are encrypted.



Aggressive Mode (RSA Signature)

Host A
(Initiator)

1) SA, g**x, Nonce, ID

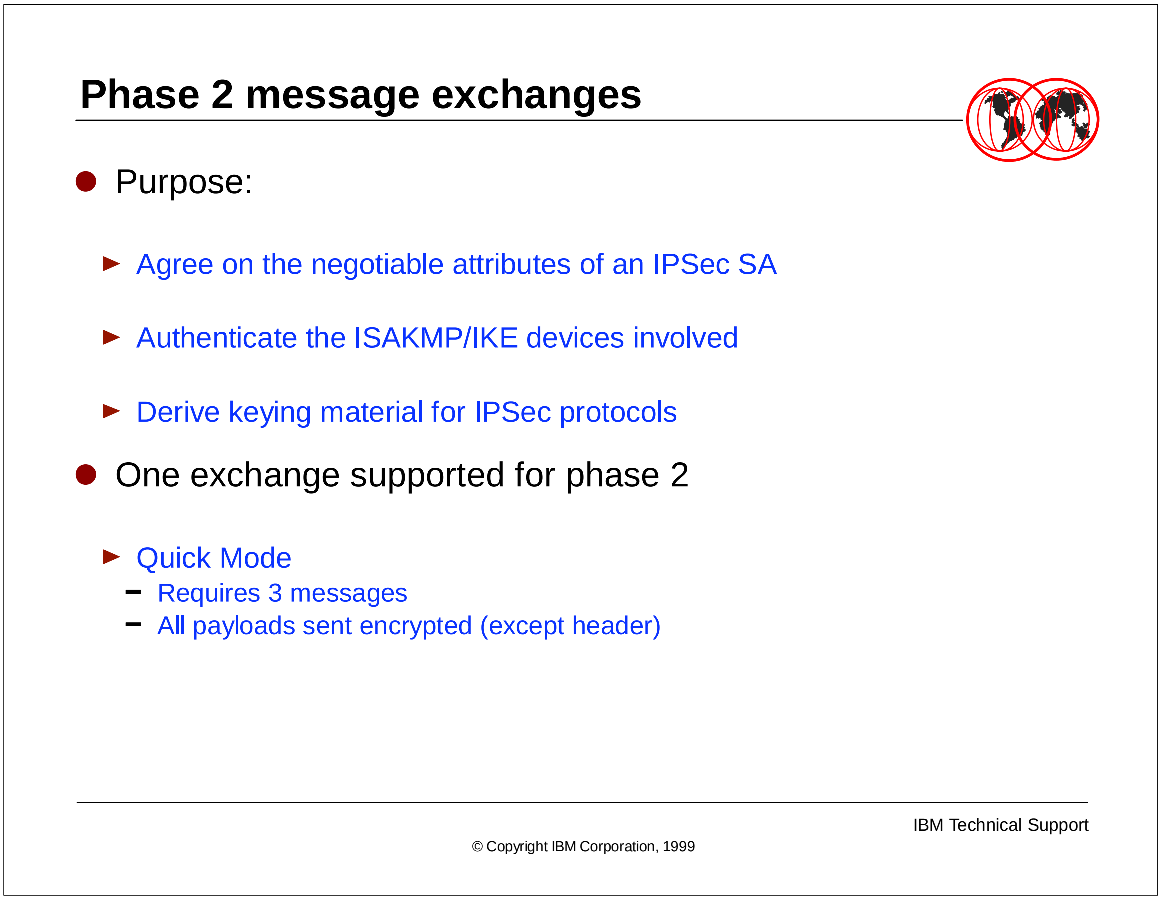
2) SA,g**y,Nonce, ID,
optional(certificate), signature

3)optional(certificate),signature

Host B
(Responder)

IBM Technical Support

- This is the flow of information sent during an aggressive mode exchange using RSA signature mode.
- Message 1 of an aggressive exchange contains all the information that was sent in message 1 and 3 of main mode plus some of message 6.
- Message 2 of an aggressive mode exchange contains all the information that was contained in messages 2, 4, and 6 of main mode. Note that this example contains an optional certificate payload and a signature payload instead of a hash payload. This is because the example shows the use of RSA signature mode.
- Message 3 of an aggressive mode exchange contains the remaining information from message 6 of main mode. Once again note that this example contains an optional certificate payload and a signature payload instead of a hash payload. This is because the example shows the use of RSA signature mode.
- As shown in this flow ISAKMP allows certificates to be optionally sent in an RSA signature mode exchange. What is not shown is that ISAKMP provides a payload to request that a peer sends its certificate. When this payload is sent the peer must send the appropriate certificate. The same payload can be used to help the peer determine which certificate is should use. In addition to requesting certificate information this payload can be used to request CRL (Certificate Revocation List) information.



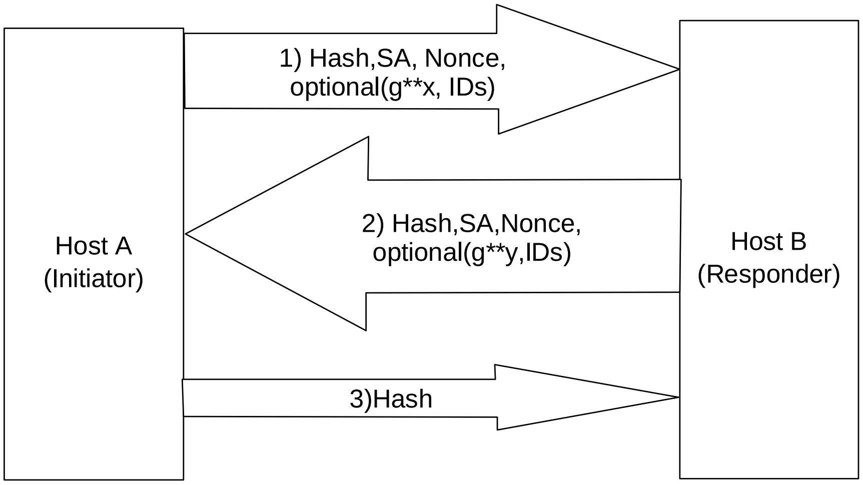
Phase 2 message exchanges

- Purpose:
  - Agree on the negotiable attributes of an IPSec SA
  - Authenticate the ISAKMP/IKE devices involved
  - Derive keying material for IPSec protocols
- One exchange supported for phase 2
  - Quick Mode
    - Requires 3 messages
    - All payloads sent encrypted (except header)

IBM Technical Support

- The purpose of a phase 2 exchange is to create an IPSec SA and to authenticate the identities involved in the negotiation.
- There are certain attributes of an SA that are negotiable. The exact attributes depend on the IPSec security service being used (i.e. AH or ESP).
- The phase 2 exchange is protected and authenticated by the phase 1 SA.
- There is only one type of phase 1 exchange. It is quick mode.
- In a quick mode exchange the initiating ISAKMP device sends its peer 2 messages and the responding device sends 1 message for a total of 3 messages. In a quick mode exchange all payloads with the exception of the header payload are sent encrypted using a key derived from the encryption keying material generated in phase 1.

Quick Mode



1) Hash,SA, Nonce, optional(g**x, IDs)

2) Hash,SA,Nonce, optional(g**y,IDs)

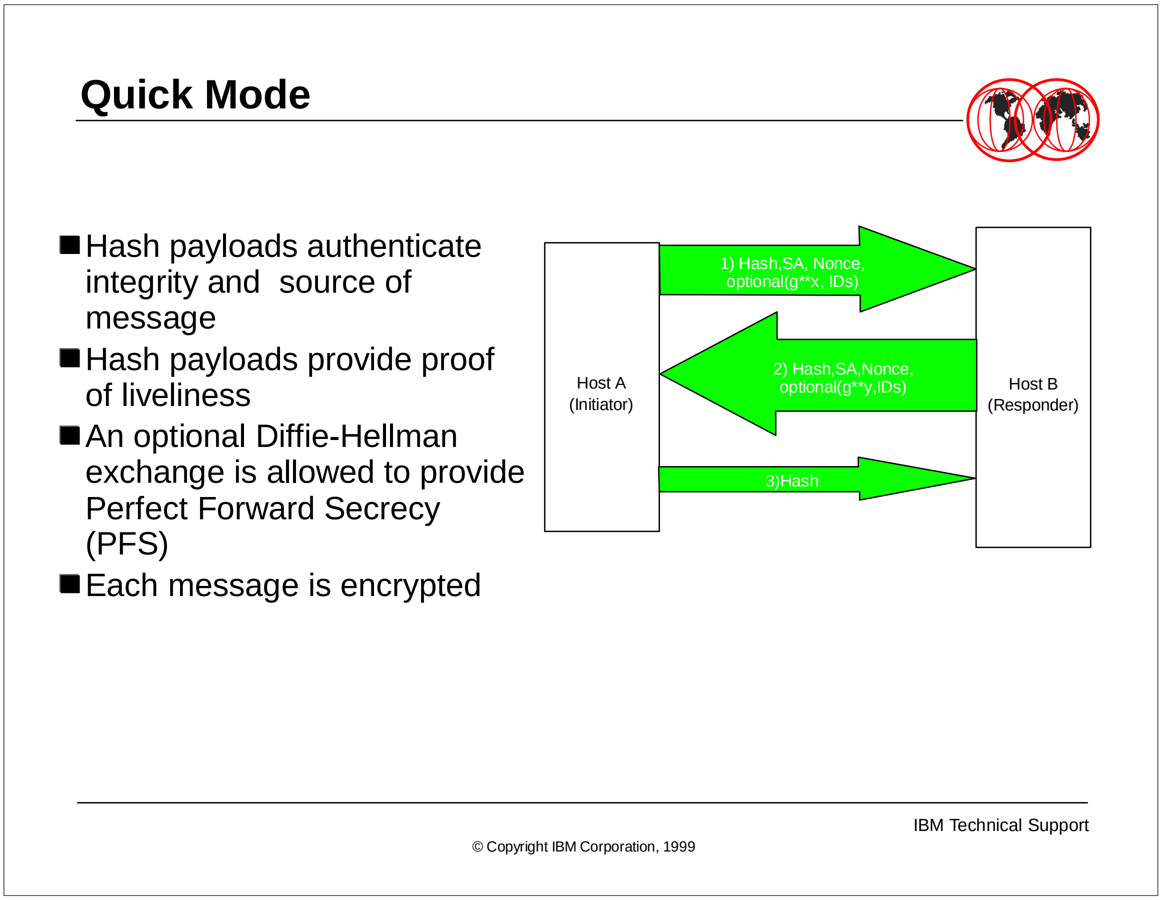
3)Hash

Host A (Initiator)

Host B (Responder)

IBM Technical Support

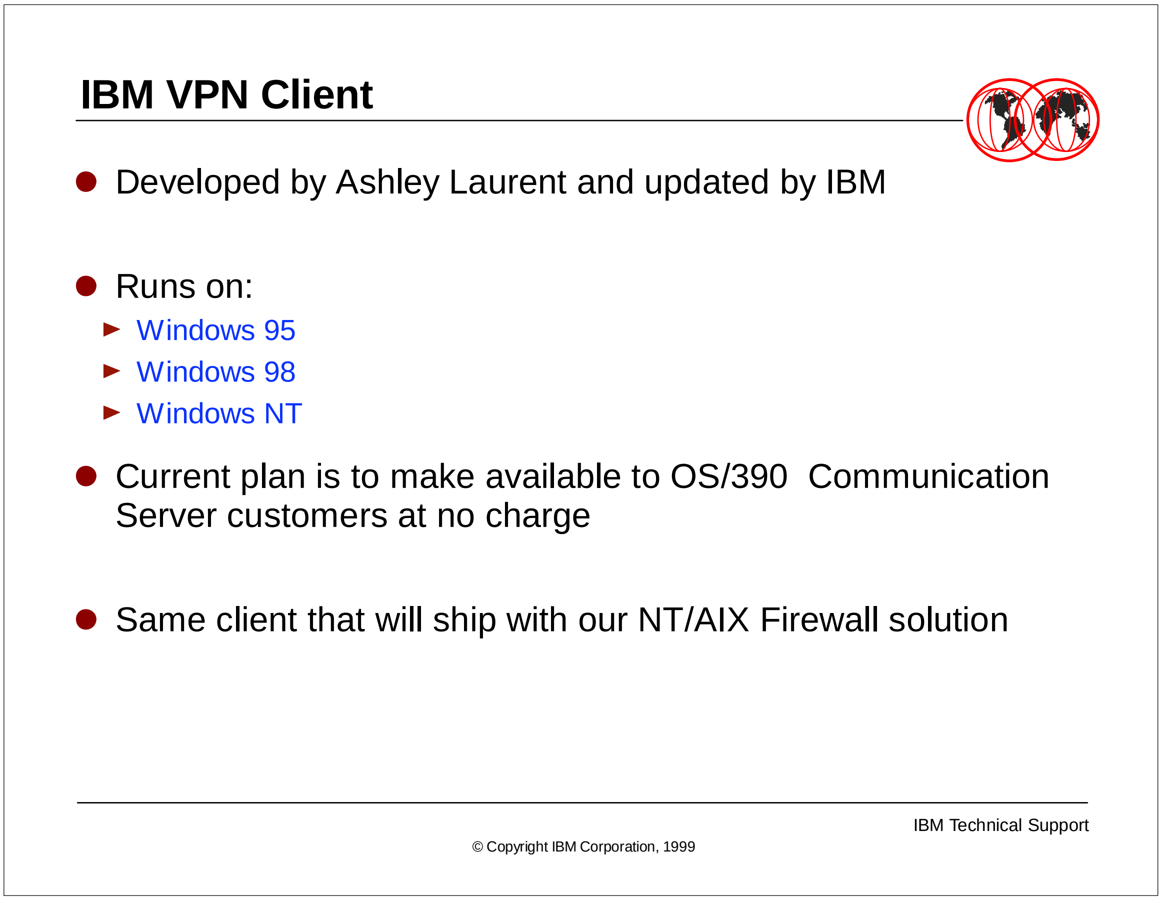
- This is the flow of information sent during a quick mode exchange.
- A detailed discussion of these messages is beyond the scope of this presentation. The next page will provide a high level description of each message.



Quick Mode

- Hash payloads authenticate integrity and source of message
- Hash payloads provide proof of liveliness
- An optional Diffie-Hellman exchange is allowed to provide Perfect Forward Secrecy (PFS)
- Each message is encrypted

- In quick mode each message contains a hash payload. The purpose of the hash is to authenticate the source of the message, authenticate the integrity of the message, and to prove liveliness.
- The key used to calculate the hash in all three messages is generated using the authentication keying material that was generated in phase 1. The source of the message is authenticated by verifying the hash sent. If the hash can be verified then the originator of the message possesses the authentication keying material.
- Part of the input to the hash function is a portion of the message itself. The integrity of the message is also authenticated by verifying the hash sent.
- The inclusion of nonce as input to the hash functions in message 2 and 3 proves liveliness of each side. In fact, message 3's only purpose in life is to prove that it successfully got message 2.
- In order to provide Perfect Forward Secrecy (PFS), quick mode allows an optional Diffie-Hellman exchange to take place. The key generated from this exchange will be used to generate the keying material to be used for the IPSec SA.
- PFS is the notion that compromise of a single key will only permit access to data encrypted with that key.
- Quick mode allows the IDs of the source and destination of the IP traffic to be optionally included in message 1 and 2. If no IDs are included then the source and destination of the traffic is assumed to be the IP addresses of the two devices that were identified during the phase 1 SA exchange.



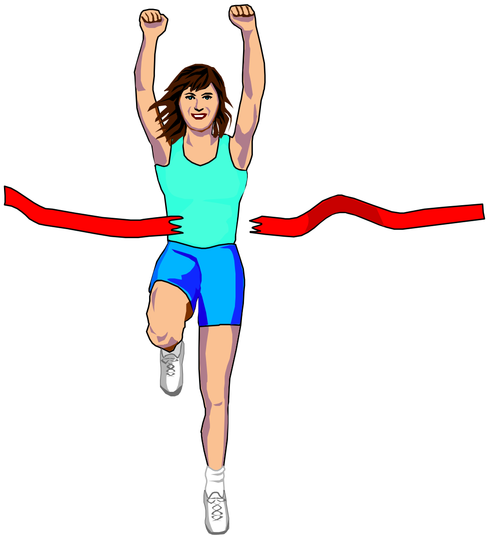
IBM VPN Client

- Developed by Ashley Laurent and updated by IBM
- Runs on:
  - Windows 95
  - Windows 98
  - Windows NT
- Current plan is to make available to OS/390 Communication Server customers at no charge
- Same client that will ship with our NT/AIX Firewall solution

IBM Technical Support

- IBM plans to offer a VPN client that was developed by Ashley Laurent and modified by IBM. This client will support both manual and dynamic VPNs. It will run on Windows 95, Windows 98, and Windows NT.
- The current plan is to make this client available to OS/390 Communication Server at no charge.

Time for a drink, I am burning !



IBM Technical Support