# THEME 3

# IP Security

**Telecommunication systems department**

**Lecturer:** assistant professor Persikov Anatoliy Valentinovich

# IP SECURITY AND VIRTUAL PRIVATE NETWORKS

**IPsec** is a framework of open standards for ensuring private communications over public networks. It has become the most common network layer security control, typically used to create a virtual private network (VPN).

A **VPN** is a virtual network built on top of existing physical networks that can provide a secure communications mechanism for data and control information transmitted between networks.

VPNs are used most often **to protect communications carried over public networks** such as the Internet.

A VPN can provide several types of data protection, including **confidentiality**, **integrity**, **data origin authentication**, **replay protection** and **access control**.

Although VPNs can reduce the risks of networking, they cannot totally eliminate them. For example, a VPN implementation may have flaws in algorithms or software, or a VPN may be set up with insecure configuration settings and values. Both of these flaws can be exploited by attackers.

# MODELS OF VPN ARCHITECTURES

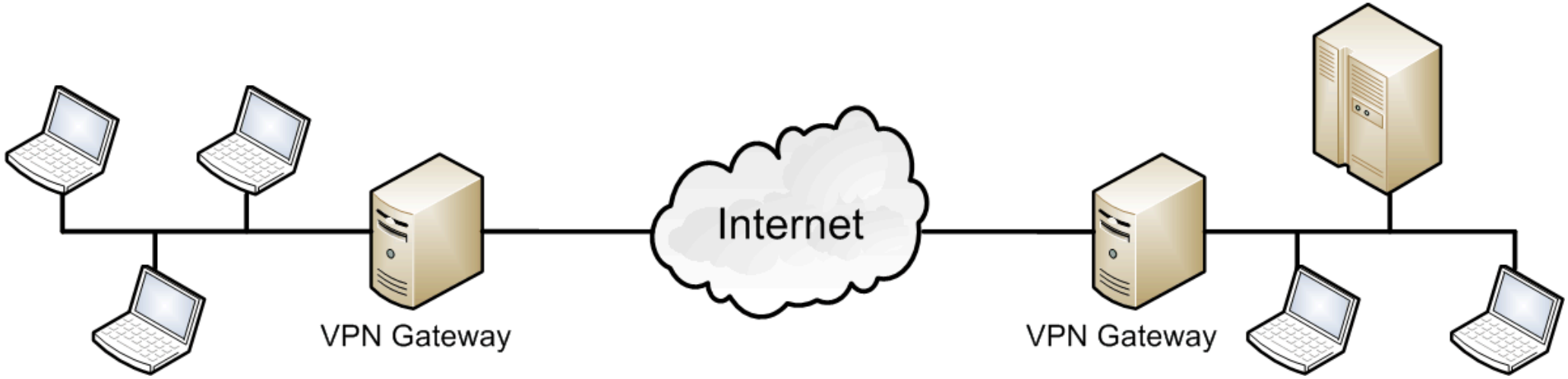
There are three primary models for VPN architectures:

**Gateway-to-gateway.** This model protects communications between two specific networks, such as an organization's main office network and a branch office network, or two business partners' networks.

**Host-to-gateway.** This model protects communications between one or more individual hosts and a specific network belonging to an organization. The host-to-gateway model is most often used to allow hosts on unsecured networks, such as traveling employees and telecommuters, to gain access to internal organizational services, such as the organization's e-mail and Web servers.

**Host-to-host.** A host-to-host architecture protects communication between two specific computers. It is most often used when a small number of users need to use or administer a remote system that requires the use of inherently insecure protocols.

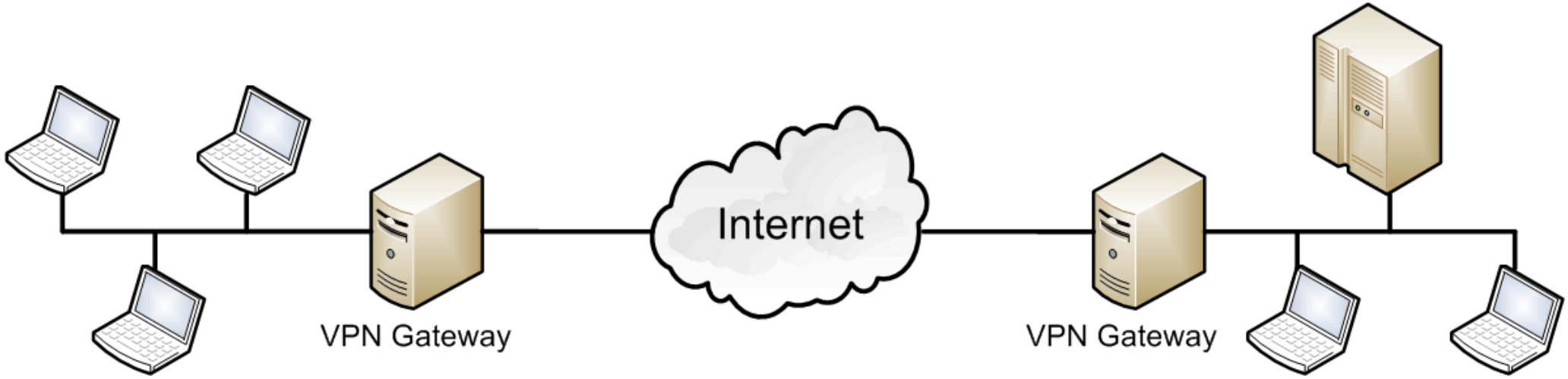
# GATEWAY-TO-GATEWAY ARCHITECTURE




IPsec-based VPNs are often used to provide secure network communications between two networks. This is typically done by deploying a VPN gateway onto each network and establishing a VPN connection between the two gateways. Traffic between the two networks that needs to be secured passes within the established VPN connection between the two VPN gateways.

The VPN gateway may be a dedicated device that only performs VPN functions, or it may be part of another network device, such as a firewall or router.

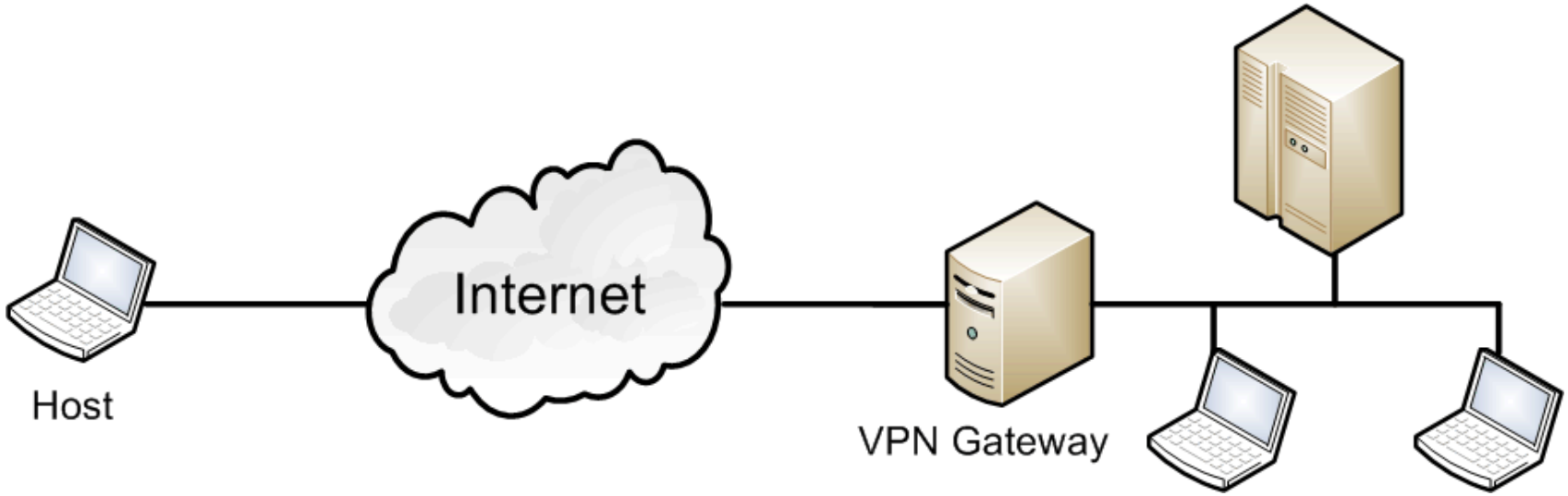
# GATEWAY-TO-GATEWAY ARCHITECTURE




This model is relatively simple to understand. To facilitate VPN connections, one of the VPN gateways issues a request to the other to establish an IPsec connection. The two VPN gateways exchange information with each other and create an IPsec connection.

Routing on each network is configured so that as hosts on one network need to communicate with hosts on the other network, their network traffic is automatically routed through the IPsec connection, protecting it appropriately. A single IPsec connection establishing a tunnel between the gateways can support all communications between the two networks, or multiple IPsec connections can each protect different types or classes of traffic.

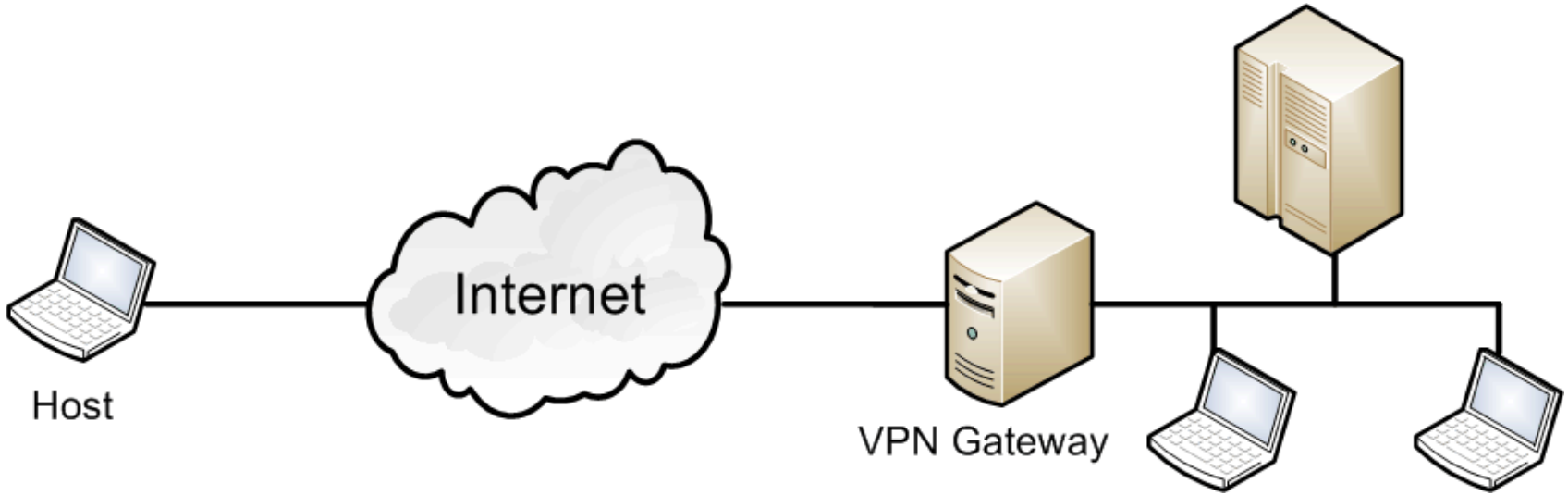
# HOST-TO-GATEWAY ARCHITECTURE




An increasingly common VPN model is the **host-to-gateway model**, which is most often used to provide **secure remote access**.

The organization deploys a VPN gateway onto their network; each remote access user then establishes a VPN connection between the local computer (host) and the VPN gateway. As with the gateway-to-gateway model, the VPN gateway may be a dedicated device or part of another network device.

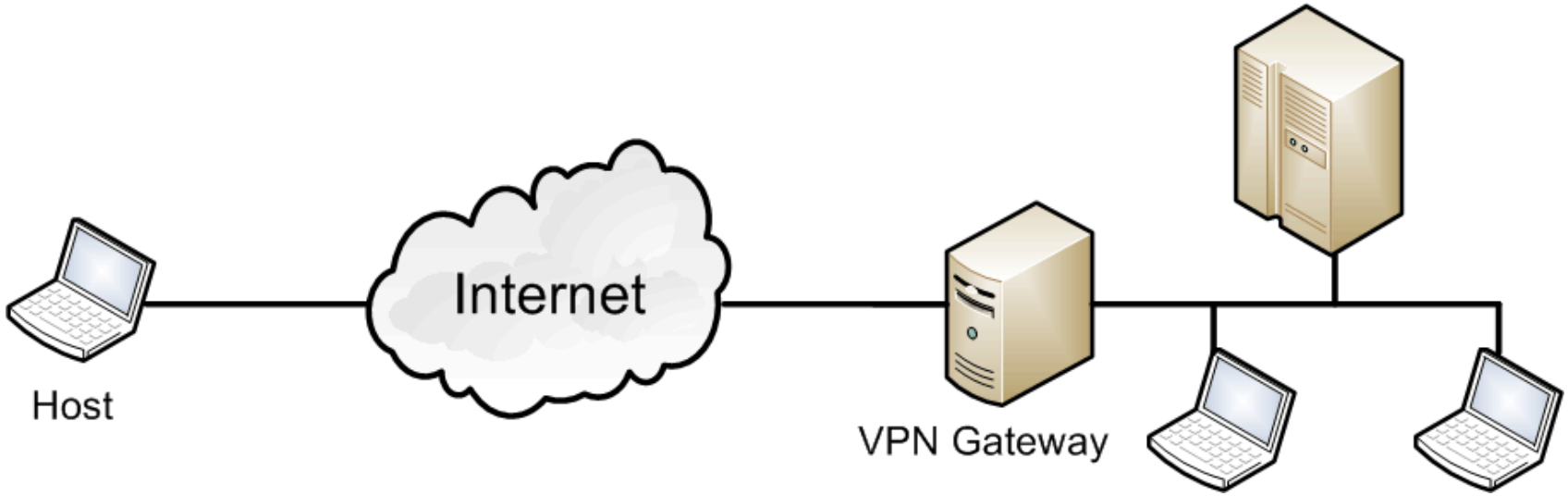
# HOST-TO-GATEWAY ARCHITECTURE




IPsec connections are created as needed for each individual VPN user.

Remote users' hosts have been configured to act as IPsec clients with the organization's IPsec gateway. When a remote user wishes to use computing resources through the VPN, the host initiates communications with the VPN gateway. The user is typically asked by the VPN gateway to authenticate before the connection can be established. The VPN gateway can perform the authentication itself or consult a dedicated authentication server.

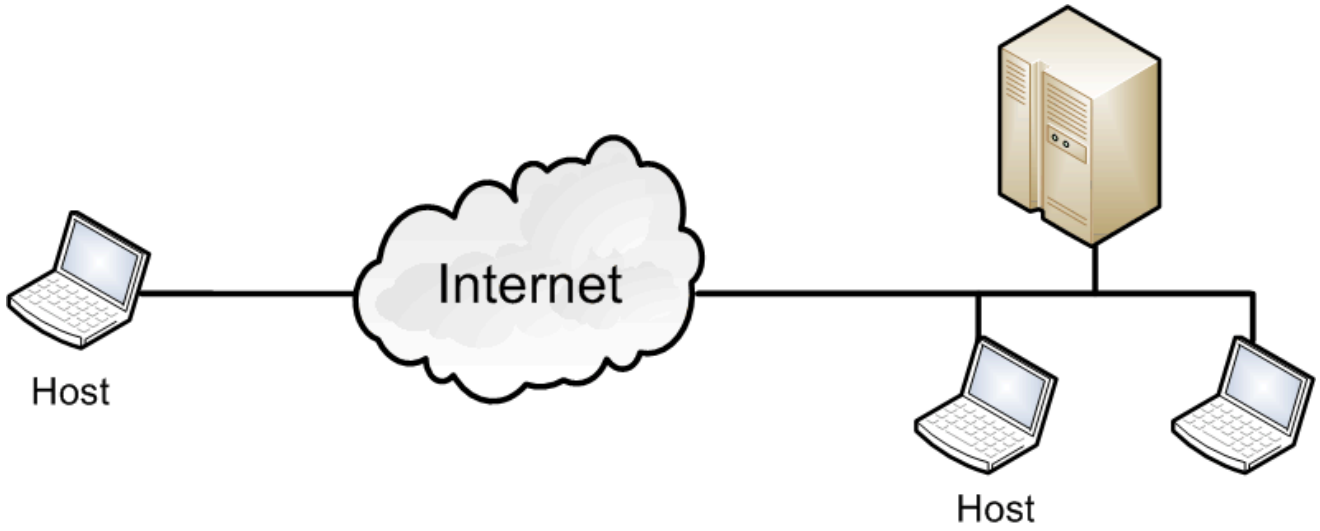
# HOST-TO-GATEWAY ARCHITECTURE




The client and gateway exchange information, and the IPsec connection is established. The user can now use the organization's computing resources, and the network traffic between the user's host and the VPN gateway will be protected by the IPsec connection.

Traffic between the user and systems not controlled by the organization can also be routed through the VPN gateway; this allows IPsec protection to be applied to this traffic as well if desired.

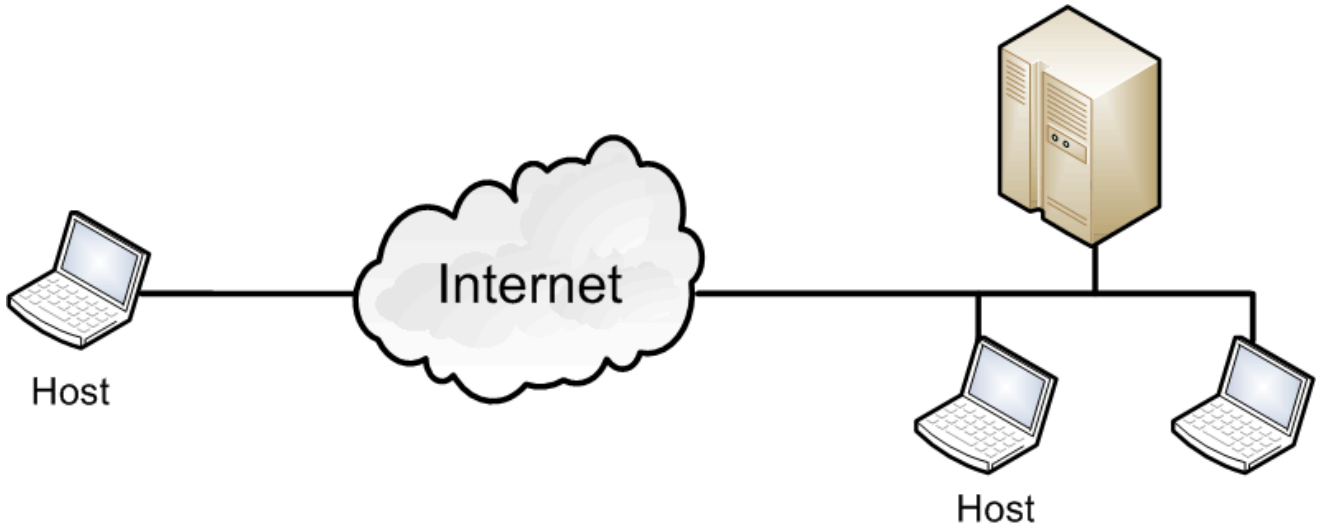
# HOST-TO-HOST ARCHITECTURE




The least commonly used VPN architecture is the host-to-host model, which is typically used for special purpose needs, such as system administrators **performing remote management of a single server**.

In this case, the organization configures the server to provide VPN services and the system administrators' hosts to act as VPN clients. The system administrators use the VPN client when needed to establish encrypted connections to the remote server

# HOST-TO-HOST ARCHITECTURE




**IPsec connections are created as needed for each individual VPN user**.

Users' hosts have been configured to act as IPsec clients with the IPsec server. When a user wishes to use resources on the IPsec server, the user's host initiates communications with the IPsec server. The user is asked by the IPsec server to authenticate before the connection can be established. The client and server exchange information, and if the authentication is successful, the IPsec connection is established. The user can now use the server, and the network traffic between the user's host and the server will be protected by the IPsec connection.

# COMPARISON OF VPN ARCHITECTURE MODELS

| Feature | Gateway-to-gateway | Host-to-gateway | Host-to-host |
|---|---|---|---|
| **Provides protection between client and local gateway** | No | N/A (client is VPN endpoint) | N/A (client is VPN endpoint) |
| **Provides protection between VPN endpoints** | Yes | Yes | Yes |
| **Provides protection between remote gateway and remote server (behind gateway)** | No | No | N/A (server is VPN endpoint) |
| **Transparent to users** | Yes | No | No |
| **Transparent to users' systems** | Yes | No | No |
| **Transparent to servers** | Yes | Yes | No |

# IPSEC FUNDAMENTALS

IPsec is a collection of protocols that assist in protecting communications over IP networks. IPsec protocols work together in various combinations to provide protection for communications.

Our lecture will focus on the three primary components — the **Encapsulating Security Payload** (ESP), **Authentication Header** (AH), and **Internet Key Exchange** (IKE) protocols — explaining the purpose and function of each protocol, and showing how they work together to create IPsec connections.

Also, this section will discuss the value of using the **IP Payload Compression Protocol** (IPComp) as part of an IPsec implementation.

# AUTHENTICATION HEADER

AH, one of the IPsec security protocols, provides integrity protection for packet headers and data, as well as user authentication. It can optionally provide replay protection and access protection. AH cannot encrypt any portion of packets.

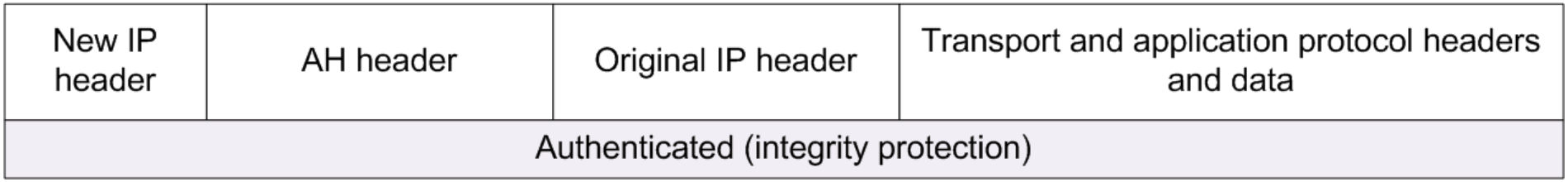
In the initial version of IPsec, the ESP protocol could provide only encryption, not authentication, so AH and ESP were often used together to provide both confidentiality and integrity protection for communications.

Because authentication capabilities were added to ESP in the second version of IPsec, AH has become less significant; in fact, some IPsec software no longer supports AH. However, AH is still of value because AH can authenticate portions of packets that ESP cannot. Also, many existing IPsec implementations are using AH.

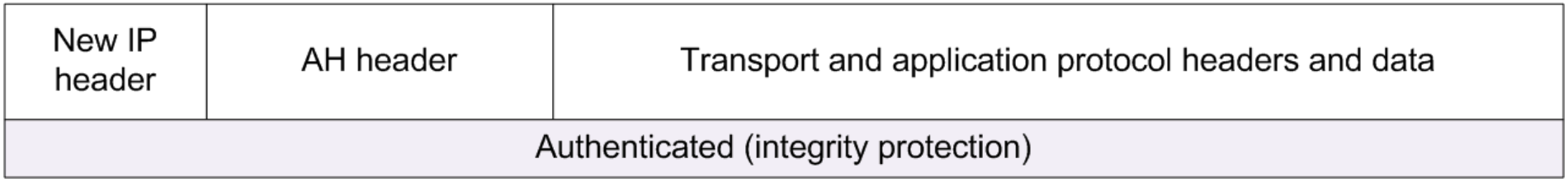
# AUTHENTICATION HEADER MODES

AH has two modes: **transport** and **tunnel**. In tunnel mode, AH creates a new IP header for each packet; in transport mode, AH does not create a new IP header. In IPsec architectures that use a gateway, the true source or destination IP address for packets must be altered to be the gateway's IP address. Because transport mode cannot alter the original IP header or create a new IP header, transport mode is generally used in host-to-host architectures. AH provides integrity protection for the entire packet, regardless of which mode is used.

**TUNNEL MODE**




**TRANSPORT MODE**



# AH INTEGRITY PROTECTION PROCESS

The first step of integrity protection is to create a hash by using a keyed hash algorithm, also known as a **message authentication code** (MAC) algorithm.

A standard hash algorithm generates a hash based on a message, while a keyed hash algorithm creates a hash based on both a message and a secret key shared by the two endpoints. The hash is added to the packet, and the packet is sent to the recipient.

The recipient can then regenerate the hash using the shared key and confirm that the two hashes match, which provides integrity protection for the packet.

IPsec uses **hash message authentication code** (HMAC) algorithms, which perform two keyed hashes.Examples of keyed hash algorithms are HMAC-MD5 and HMAC-SHA-1.

Another common MAC algorithm is AES Cipher Block Chaining MAC (AES-XCBC-MAC-96).

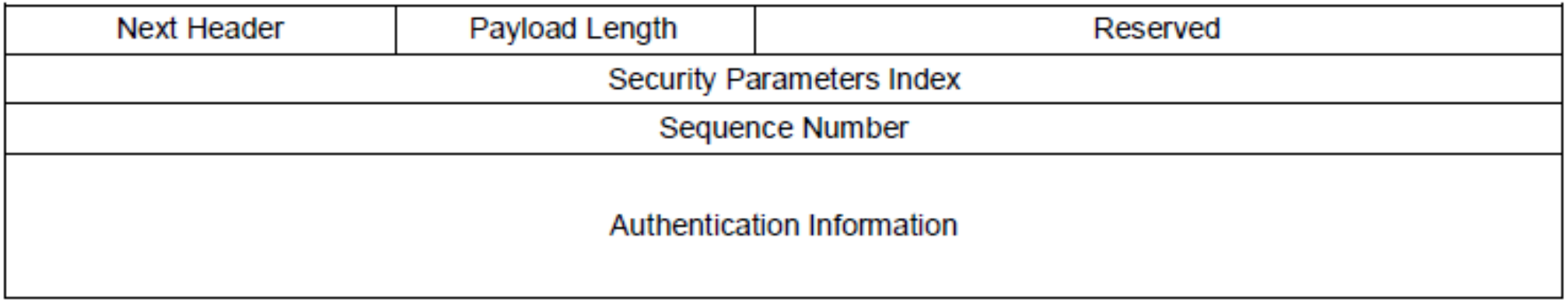
# AH INTEGRITY PROTECTION PROCESS

Technically, figures are somewhat misleading because **it is not possible to protect the integrity of the entire IP header**. Certain IP header fields, such as time to live (TTL) and the IP header checksum, are dynamic and may change during routine communications.

If the hash is calculated on all the original IP header values, and some of those values legitimately change in transit, the recalculated hash will be different. The destination would conclude that the packet had changed in transit and that its integrity had been violated. To avoid this problem, IP header fields that may legitimately change in transit in an unpredictable manner are excluded from the integrity protection calculations.

This same principle explains why AH is often incompatible with **network address translation** (NAT) implementations. The IP source and destination address fields are included in the AH integrity protection calculations. If these addresses are altered by a NAT device (e.g., changing the source address from a private to a public address), the AH integrity protection calculation made by the destination will not match.

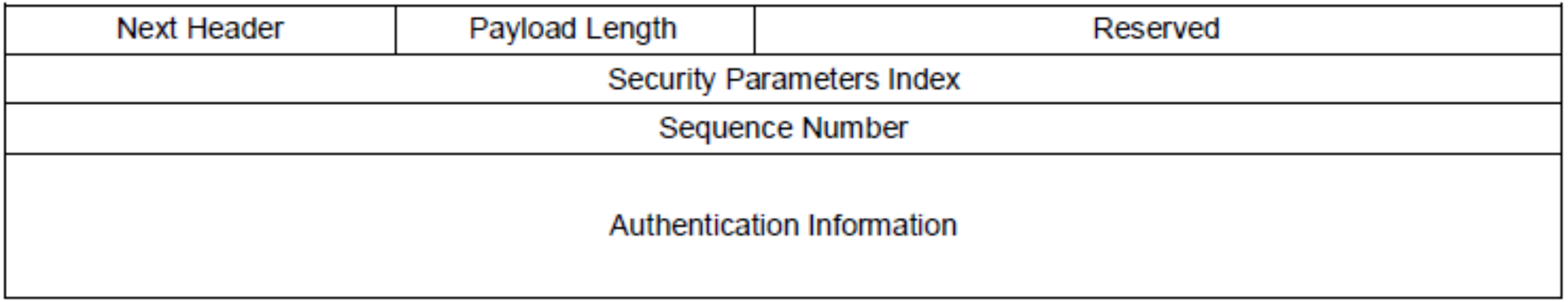
# AH HEADER




Each AH header is composed of six fields:

- **Next Header**. This field contains the IP protocol number for the next packet payload. In tunnel mode, the payload is an IP packet, so the Next Header value is set to 4 for IP-in-IP. In transport mode, the payload is usually a transport-layer protocol, often TCP (protocol number 6) or UDP (protocol number 17).
- **Payload Length**. This field contains the length of the payload in 4-byte increments, minus 2.
- **Reserved**. This value is reserved for future use, so it should be set to 0.

# AH HEADER




- **Security Parameters Index (SPI)**. Each endpoint of each IPsec connection has an arbitrarily chosen SPI value, which acts as a unique identifier for the connection. The recipient uses the SPI value, along with the destination IP address and (optionally) the IPsec protocol type (in this case, AH), to determine which Security Association (SA) is being used.
- **Sequence Number**. Each packet is assigned a sequential sequence number, and only packets within a sliding window of sequence numbers are accepted. This provides protection against replay attacks because duplicate packets will use the same sequence number.
- **Authentication Information**. The recipient of the packet can recalculate the MAC to confirm that the packet has not been altered in transit.

# ENCAPSULATING SECURITY PAYLOAD (ESP)

ESP is the second core IPsec security protocol.

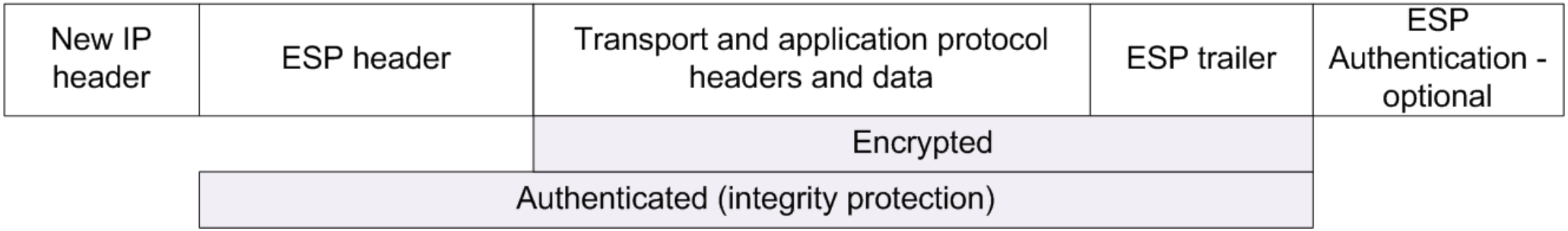
In the initial version of IPsec, ESP provided only encryption for packet payload data. Integrity protection was provided by the AH protocol if needed.

In the second version of IPsec, ESP became more flexible. It can perform authentication to provide integrity protection, although not for the outermost IP header. Also, ESP's encryption can be disabled through the Null ESP Encryption Algorithm. Therefore, in all but the oldest IPsec implementations, ESP can be used to provide only encryption; encryption and integrity protection; or only integrity protection.

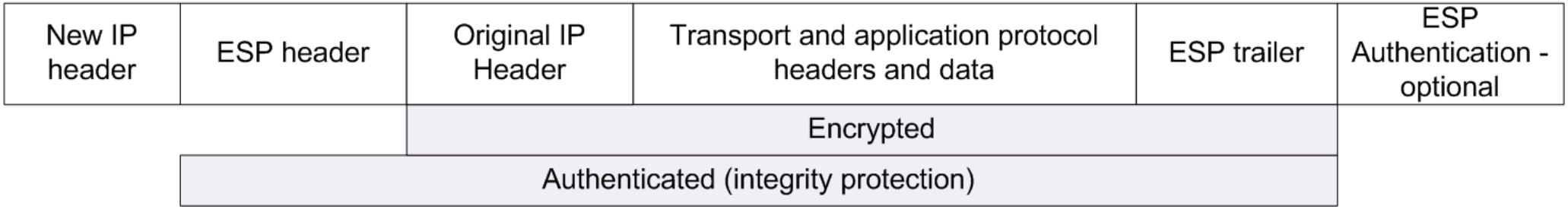
# ESP MODES

ESP has two modes: transport and tunnel. In tunnel mode, ESP creates a new IP header for each packet. The new IP header lists the endpoints of the ESP tunnel (such as two IPsec gateways) as the source and destination of the packet. Because of this, tunnel mode can be used with all three VPN architecture models.

**TRANSPORT MODE**




**TUNNEL MODE**



# ESP MODES

Tunnel mode can encrypt and/or protect the integrity of both the data and the original IP header for each packet. Encrypting the data protects it from being accessed or modified by unauthorized parties; encrypting the IP header conceals the nature of the communications, such as the actual source or destination of the packet. If authentication is being used for integrity protection, each packet will have an ESP Authentication section after the ESP trailer.

ESP tunnel mode is used far more frequently than ESP transport mode. In transport mode, ESP uses the original IP header instead of creating a new one.

ESP can only encrypt and/or protect the integrity of packet payloads and certain ESP components, but not IP headers. As with AH, ESP transport mode is generally only used in host-to-host architectures.

Also, transport mode is incompatible with NAT.

# ENCRYPTION PROCESS

ESP uses **symmetric cryptography** to provide encryption for IPsec packets.

Accordingly, both endpoints of an IPsec connection protected by ESP encryption must use the same key to encrypt and decrypt the packets. When an endpoint encrypts data, it divides the data into small blocks (for the AES algorithm, 128 bits each), and then performs multiple sets of cryptographic operations (known as rounds) using the data blocks and key.

Encryption algorithms that work in this way are known as **block cipher algorithms**.

When the other endpoint receives the encrypted data, it performs decryption using the same key and a similar process, but with the steps reversed and the cryptographic operations altered.

Examples of encryption algorithms used by ESP are AES-Cipher Block Chaining (AES-CBC), AES Counter Mode (AES-CTR), and Triple DES (3DES).

# ESP PACKET FIELDS

ESP adds a header and a trailer around each packet's payload. Each ESP header is composed of two fields:

**SPI.** Each endpoint of each IPsec connection has an arbitrarily chosen SPI value, which acts as a unique identifier for the connection. The recipient uses the SPI value, along with the destination IP address and (optionally) the IPsec protocol type (in this case, ESP), to determine which SA is being used.

**Sequence Number.** Each packet is assigned a sequential sequence number, and only packets within a sliding window of sequence numbers are accepted. This provides protection against replay attacks because duplicate packets will use the same sequence number. This also helps to thwart denial of service attacks because old packets that are replayed will have sequence numbers outside the window, and will be dropped immediately without performing any more processing.

The next part of the packet is the **payload**. It is composed of the payload data, which is encrypted, and the initialization vector (IV), which is not encrypted. The IV is used during encryption. Its value is different in every packet, so if two packets have the same content, the inclusion of the IV will cause the encryption of the two packets to have different results. This makes ESP less susceptible to cryptanalysis.

# ESP PACKET FIELDS

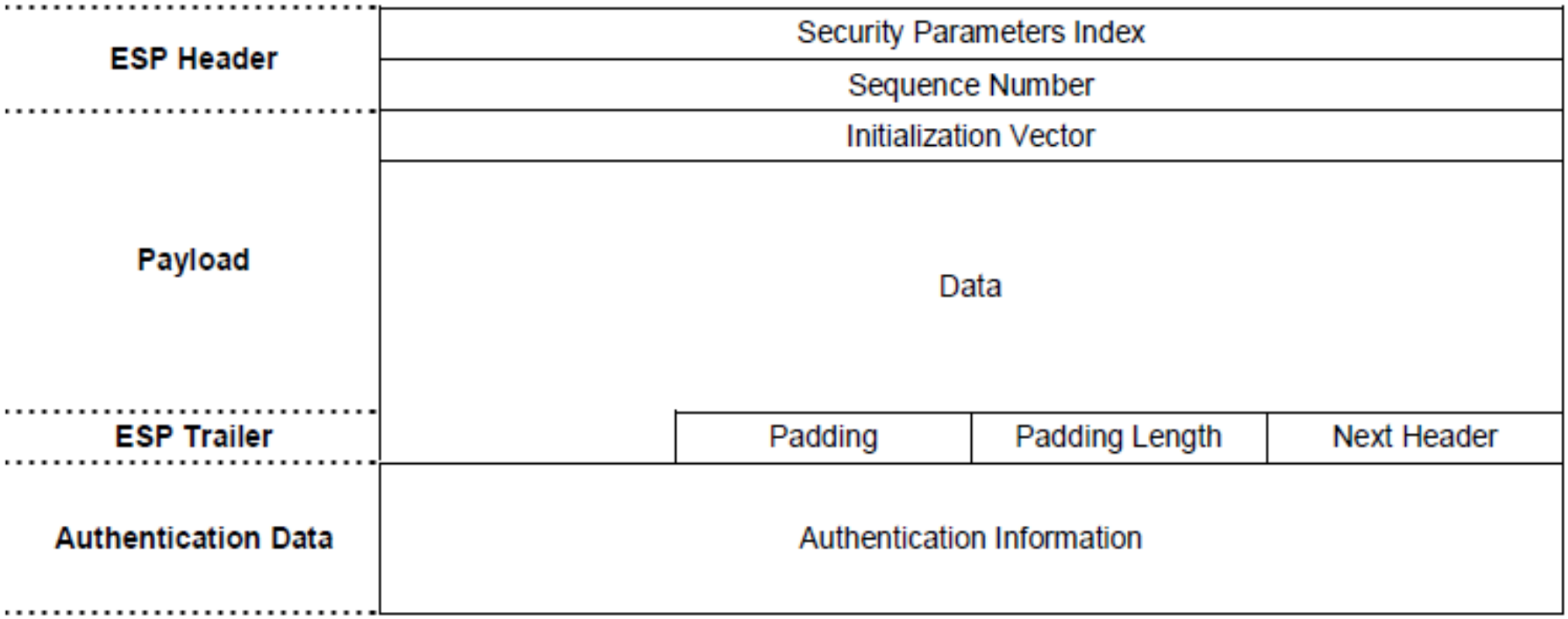
The third part of the packet is the ESP trailer, which contains at least two fields and may optionally include one more:

**Padding.** An ESP packet may optionally contain padding, which is additional bytes of data that make the packet larger and are discarded by the packet's recipient. Because ESP uses block ciphers for encryption, padding may be needed so that the encrypted data is an integral multiple of the block size. Padding may also be needed to ensure that the ESP trailer ends on a multiple of 4 bytes. Additional padding may also be used to alter the size of each packet, concealing how many bytes of actual data the packet contains. This is helpful in deterring traffic analysis.

**Padding Length.** This number indicates how many bytes long the padding is. The Padding Length field is mandatory.

**Next Header.** In tunnel mode, the payload is an IP packet, so the Next Header value is set to 4 for IP-in-IP. In transport mode, the payload is usually a transport-layer protocol, often TCP (protocol number 6) or UDP (protocol number 17). Every ESP trailer contains a Next Header value.

# ESP PACKET FIELDS



# ESP VERSION 3

A new standard for ESP, version 3, is currently in development. Based on the current standard draft, there should be several major functional differences between version 2 and version 3, including the following:

- The standard for ESP version 2 required ESP implementations to support using ESP encryption only (without integrity protection). The proposed ESP version 3 standard makes support for this optional.
- ESP can use an optional longer sequence number, just like the proposed AH version 3 standard.
- ESP version 3 supports the use of combined mode algorithms (e.g., AES Counter with CBC-MAC). Rather than using separate algorithms for encryption and integrity protection, a combined mode algorithm provides both encryption and integrity protection.

The version 3 standard draft also points to another standard draft that lists encryption and integrity protection cryptographic algorithm requirements for ESP. For encryption algorithms, the draft mandates support for the null encryption algorithm and 3DES-CBC, strongly recommends support for AES-CBC (with 128-bit keys), recommends support for AES-CTR, and discourages support for DES-CBC. For integrity protection algorithms, the draft mandates support for HMAC-SHA1-96 and the null authentication algorithm, strongly recommends support for AES-XCBC-MAC-96, and also recommends support for HMAC-MD5-96.

# INTERNET KEY EXCHANGE (IKE)

The purpose of the Internet Key Exchange (IKE) protocol is to negotiate, create, and manage security associations.

**Security association** (SA) is a generic term for a set of values that define the IPsec features and protections applied to a connection.

SAs can also be manually created, using values agreed upon in advance by both parties, but these SAs cannot be updated; this method does not scale for real-life large-scale VPNs.

IKE uses five different types of exchanges to create security associations, transfer status and error information, and define new Diffie-Hellman groups (main mode, aggressive mode, quick mode, informational, and group).

In IPsec, IKE is used to provide a secure mechanism for establishing IPsec-protected connections.

# IKE PHASE ONE EXCHANGE

The purpose of the IKE phase one exchange is for the two IPsec endpoints to successfully negotiate a secure channel through which an IPsec SA can be negotiated. The secure channel created during phase one is commonly known as an **IKE SA**.

The purpose of the IKE SA is to provide bidirectional encryption and authentication for other IKE exchanges: the negotiations that comprise phase two, the transfer of status and error information, and the creation of additional Diffie-Hellman groups.

In fact, a phase one IKE exchange must be successfully completed before any of the other IKE exchange types can be performed.

An IKE SA can be established through one of two modes: **main mode** and **aggressive mode**.

# IKE PHASE ONE EXCHANGE (MAIN MODE)

Main mode negotiates the establishment of the IKE SA through three pairs of messages. In the first pair of messages, each endpoint proposes parameters to be used for the SA. Four of the parameters are mandatory, and are collectively referred to as the protection suite:

**Encryption Algorithm.** This specifies the algorithm to be used to encrypt data. Examples of encryption algorithms are DES, 3DES, CAST, RC5, IDEA, Blowfish, and AES.

**Integrity Protection Algorithm.** This indicates which keyed hash algorithm should be used for integrity protection. HMAC-MD5 and HMAC-SHA-1 are commonly used keyed hash algorithms.

**Authentication Method.** There are several possible methods for authenticating the two endpoints to each other, including:

- pre-shared keys
- digital signatures
- public key encryption
- external authentication and
- Diffie-Hellman (DH) Group.

# IKE PHASE ONE EXCHANGE (MAIN MODE)

**Pre-shared Keys**. Each endpoint has been given the same secret key in advance. The endpoints use the key to generate a value that is then used to create the secret keys that will be used to protect the phase 1 secure channel, as well as the eventual IPsec SA. Successful completion of the phase 1 IKE negotiation constitutes proof that each peer possesses the pre-shared secret key, which serves to authenticate the peers to each other.

**Digital Signatures.** Each endpoint has its own digital certificate that contains a public key. The endpoint uses the corresponding private key to digitally sign data before sending it to the other endpoint, which verifies the signature using the peer's public key. The digital signature algorithm choices are RSA and the Digital Signature Standard (DSS).

**Public Key Encryption.** Instead of using the public/private key pair for signing data, each peer encrypts data with its own private key and decrypts data with the peer's public key. The algorithm typically used for public key encryption is RSA. Public key encryption-based authentication typically relies upon the establishment of a public key infrastructure (PKI) implementation and the issuance of digital certificates. This authentication method is defined in the IKE standard, but it is not commonly implemented or used.

# IKE PHASE ONE EXCHANGE (MAIN MODE)

**External Authentication.** Although not specified by the current IKE standard, some IPsec implementations support the use of external authentication servers and services such as Kerberos v5. In the Kerberos method, a Kerberos server maintains all of the keys for all devices within its domain. Kerberos may also be used to authenticate the hosts; however, the identity of the endpoints will not be concealed until the third set of messages. (When some authentication methods are used, such as pre-shared keys or digital signatures, the identity of the endpoints is protected during all three sets of messages.)

**Diffie-Hellman (DH) Group.** Diffie-Hellman is used to generate a shared secret for the endpoints in a secure manner, so that an observer of the IKE phase one exchange cannot determine the shared secret. This shared secret is then used to generate a value that is used as input to the calculations for the phase 1 and 2 secret keys.

# IKE PHASE ONE EXCHANGE (AGGRESSIVE MODE)

Aggressive mode offers a faster alternative to main mode.

It negotiates the establishment of the IKE SA through three messages instead of three pairs of messages. The first two messages negotiate the IKE SA parameters and perform a key exchange; the second and third messages authenticate the endpoints to each other.

The following provides more detail on each message:

1. In the first message, endpoint A sends all the protection suite parameters, as well as its portion of the Diffie-Hellman key exchange, a nonce, and its identity.
2. In the second message, endpoint B sends the protection suite parameters, its portion of the Diffie-Hellman key exchange, a nonce, its identity, and its authentication payload (through digital signature or hash).
3. In the third message, endpoint A sends its authentication payload.

# IKE PHASE ONE EXCHANGE (AGGRESSIVE MODE)

Aggressive mode negotiates all the same parameters as main mode through fewer messages. Also, unlike main mode, aggressive mode can be used with pre-shared secret key authentication for hosts without fixed IP addresses.

**However, with the increased speed of aggressive mode comes decreased security.**

Since the Diffie-Hellman key exchange begins in the first packet, the two parties do not have an opportunity to negotiate the Diffie-Hellman parameters. Also, the identity information is not always hidden in aggressive mode, so an observer could determine which parties were performing the negotiation. (Aggressive mode can conceal identity information in some cases when public keys have already been exchanged.) Aggressive mode negotiations are also susceptible to pre-shared key cracking, which can allow user impersonation and man-in-the-middle attacks. Another potential issue is that while all IPsec devices must support main mode, aggressive mode support is optional. Unless there are performance issues, it is generally recommended to use main mode for the phase one exchange.

# IKE PHASE TWO EXCHANGE

The purpose of phase two is to establish an SA for the actual IPsec connection. This SA is referred to as the IPsec SA. Unlike IKE SA's, which are bidirectional, IPsec SA's are unidirectional. This means that an IPsec connection between two systems requires two security associations. The pair of IPsec SAs is created through a single mode, quick mode. Quick mode uses three messages to establish the SA. Remember that quick mode communications are encrypted by the method specified in the IKE SA created by phase one.

After endpoint B validates the third message, the IPsec SAs are established. All active SAs are stored in a Security Association Database (SAD).

# IKE PHASE TWO EXCHANGE (SAD)

The SAD includes the following information for each protected connection:

- Source IP address;
- Destination IP address;
- SPI;
- IPsec security protocol (AH or ESP);
- Mode (transport or tunnel);
- Encryption algorithm for ESP (e.g., AES-CBC);
- Integrity protection algorithm (e.g., HMAC-MD5, HMAC-SHA-1);
- Secret keys used by the selected algorithms;
- Key length, if any of the selected algorithms can use multiple key sizes;
- SA lifetime (described later in this section);
- Sequence number information;
- Anti-replay information;
- Types of traffic to which this SA should be applied (e.g., specific ports and/or protocols).

# IKE PHASE TWO EXCHANGE (SPD)

An SA can be uniquely identified by the combination of three parameters: the destination IP address, the SPI, and the IPsec security protocol. When an endpoint needs to know which SA applies to a particular packet, it looks it up in the SAD using these parameters. The SA describes the security measures that IPsec should use to protect communications; however, it does not fully describe what types of traffic should be protected, and under what circumstances.

That information is stored in the **Security Policy Database** (SPD), which classifies traffic as requiring IPsec protection (protect), not requiring IPsec protection (bypass), or being prohibited (discard).

The SPD typically contains the following information for each type of traffic that needs to be protected:

- source and destination IP address;
- IP protocol (e.g., TCP, UDP, all);
- TCP or UDP port number (optional);
- IPsec protections to be applied;
- pointer to the SA within the SAD, if an SA has already been negotiated for a particular type of traffic.

# IPSEC PLANNING AND IMPLEMENTATION

As with any new technology deployment, IPsec planning and implementation should be addressed in a phased approach. A successful deployment of IPsec can be achieved by following a clear, step-by-step planning and implementation process. The use of a phased approach for deployment can minimize unforeseen issues and identify potential pitfalls early in the process. This model also allows for the incorporation of advances in new technology, as well as adapting IPsec to the ever-changing enterprise.

IPsec planning and implementation phases:

**1. Identify Needs.** The first phase of the process involves identifying the need to protect network communications, determining which computers, networks, and data are part of the communications, and identifying related requirements (e.g., minimum performance). This phase also involves determining how that need can best be met (e.g., IPsec, SSL, SSH) and deciding where and how the security should be implemented.

**2. Design the Solution.** The second phase involves all facets of designing the IPsec solution. For simplicity, the design elements are grouped into four categories: architectural considerations, authentication methods, cryptography policy, and packet filters.

**3. Implement and Test a Prototype.** The next phase involves implementing and testing a prototype of the designed solution in a lab or test environment. The primary goals of the testing are to evaluate the functionality, performance, scalability, and security of the solution, and to identify any issues with the components, such as interoperability issues.

**4. Deploy the Solution.** Once the testing is completed and all issues are resolved, the next phase includes the gradual deployment of IPsec throughout the enterprise.

**5. Manage the Solution.** After the IPsec solution has been deployed, it is managed throughout its lifecycle. Management includes maintenance of the IPsec components and support for operational issues. The lifecycle process is repeated when enhancements or significant changes need to be incorporated into the solution.

# IPSEC PLANNING AND IMPLEMENTATION

Organizations should also implement other measures that support and complement IPsec implementations. These measures help to ensure that IPsec is implemented in an environment with the technical, management, and operational controls necessary to provide adequate security for the IPsec implementation. Examples of supporting measures are as follows:

- Establish and maintain control over all entry and exit points for the protected network, which helps to ensure its integrity.

- Ensure that all IPsec endpoints (gateways and hosts) are secured and maintained properly, which should reduce the risk of IPsec compromise or misuse.

- Revise organizational policies as needed to incorporate appropriate usage of the IPsec solution. Policies should provide the foundation for the planning and implementation of IPsec.

# IPSEC PLANNING AND IMPLEMENTATION (NEEDS IDENTIFICATION)

Assuming that IPsec is chosen as the solution's protocol, the Identify Needs phase should result in the following:

- Identification of all communications that need to be protected (e.g., servers, client hosts, networks, applications, data), and the protection that each type of communication needs (preferably encryption, integrity protection, and peer authentication)

- Selection of an IPsec architecture model (e.g., gateway-to-gateway, host-to-gateway, host-to-host)

- Specification of performance requirements (normal and peak loads).

# IPSEC PLANNING AND IMPLEMENTATION (DESIGN THE SOLUTION)

The next phase is to design a solution that meets the needs:

- **Architecture.** Designing the architecture of the IPsec implementation includes host placement (for host-to-host architectures) and gateway placement (for host-to-gateway and gateway-to-gateway architectures), IPsec client software selection (for host-to-host and host-to-gateway architectures), and host address space management considerations (for host-to-host and host-to-gateway architectures).

- **Authentication.** The IPsec implementation must have an authentication method selected, such as pre-shared key or digital signature.

- **Cryptography.** The algorithms for encryption and integrity protection must be selected, as well as the key strength for algorithms that support multiple key lengths.

- **Packet Filter.** The packet filter determines which types of traffic should be permitted and denied, and what protection and compression measures (if any) should be applied to each type of permitted traffic (e.g., ESP tunnel using AES for encryption and HMAC-SHA-1 for integrity protection; LZS for compression).

# IPSEC PLANNING AND IMPLEMENTATION (ARCHITECTURE)

The architecture of the IPsec implementation refers to the selection of devices and software to provide IPsec services and the placement of IPsec endpoints within the existing network infrastructure. These two considerations are often closely tied together; for example, a decision could be made to use the existing Internet firewall as the IPsec gateway.

# IPSEC PLANNING AND IMPLEMENTATION (GATEWAY PLACEMENT)

Due to the layered defense strategy used to protect enterprise networks, IPsec gateway placement is often a challenging task.

The gateway's placement has **security**, **functionality**, and **performance** implications.

Also, the gateway's placement may have an effect on other network devices, such as firewalls, routers, and switches. Incorporating an IPsec gateway into a network architecture requires strong overall knowledge of the network and security policy. The following are major factors to consider for IPsec gateway placement:

- **Device Performance.** IPsec can be computationally intensive, primarily because of encryption and decryption. Providing IPsec services from another device (e.g., firewall, router) may put too high of a load on the device during peak usage, causing service disruptions. A possible alternative is to offload the cryptography operations to a specialized hardware device, such as a card with built-in cryptography functions. Organizations should also review their network architecture to determine if bottlenecks are likely to occur due to network devices (e.g., routers, firewalls) that cannot sustain the processing of peak volumes of network traffic that includes IPsec-encapsulated packets.

# IPSEC PLANNING AND IMPLEMENTATION (GATEWAY PLACEMENT)

- **Traffic Examination.** If IPsec-encrypted traffic passes through a firewall, it cannot tell what protocols the packets' payloads contain, so it cannot filter the traffic based on those protocols. Intrusion detection systems encounter the same issue; they cannot examine encrypted traffic for attacks. It is generally recommended to design the IPsec architecture so that a firewall and intrusion detection software can examine the unencrypted traffic. Organizations most commonly address this by using their Internet firewalls as VPN gateways or placing VPN gateway devices just outside their Internet firewalls.

- **Traffic Not Protected by IPsec.** Organizations should consider carefully the threats against network traffic after it has been processed by the receiving IPsec gateway and sent without IPsec protection across additional network segments. For example, an organization that wants to place its VPN gateway outside its Internet firewalls should ensure that the traffic passing between the IPsec gateway and the Internet firewalls has sufficient protection against breaches of confidentiality and integrity.

# IPSEC PLANNING AND IMPLEMENTATION (GATEWAY PLACEMENT)

- **Gateway Outages.** The architecture should take into consideration the effects of IPsec gateway outages, including planned maintenance outages and unplanned outages caused by failures or attacks. For example, if the IPsec gateway is placed inline near the Internet connection point, meaning that all network traffic passes through it, a gateway failure could cause a loss of all Internet connectivity for the organization.

- **NAT.** There are known incompatibilities between IPsec and NAT because NAT modifies the IP addresses in the packet, which directly violates the packet integrity assurance provided by IPsec. However, there are a few solutions to this issue, as follows:
  - **Perform NAT before applying IPsec.** This can be accomplished by arranging the devices in a particular order, or by using an IPsec gateway that also performs NAT. For example, the gateway can perform NAT first and then IPsec for outbound packets.
  - **Use UDP encapsulation of ESP packets.** UDP encapsulation can be used with tunnel mode ESP or Layer 2 Tunneling Protocol (L2TP) over transport mode ESP.
  - At small or home offices, **configure cable and Digital Subscriber Line (DSL) routers** performing NAT to permit IPsec NAT pass-through for the IPsec client systems.