

Unit-04 Network Security Applications

Kerberos:

It is the authentication service developed as part of Project Athena at MIT.

In Open Distributed Network Following Threats are Possible:

- A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
- A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
- A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

Unit-04 Network Security Applications

Motivation:-

In a distributed architecture consisting of dedicated user workstations (clients) and distributed or centralized servers.

In this environment, three approaches to security can be envisioned:

- Rely on each individual client workstation to assure the identity of its user or users and rely on each server to enforce a security policy based on user identification (ID).
- Require that client systems authenticate themselves to servers, but trust the client system concerning the identity of its user.
- Require the user to prove his or her identity for each service invoked. Also require that servers prove their identity to clients.

A Simple Authentication Dialogue

(1) $C \rightarrow AS: ID_C \parallel P_C \parallel ID_V$

(2) $AS \rightarrow C: Ticket$

(3) $C \rightarrow V: ID_C \parallel Ticket$

$Ticket = E(K_v, [ID_C \parallel AD_C \parallel ID_V])$

where

C = client

AS = authentication server

V = server

ID_C = identifier of user on C

ID_V = identifier of V

P_C = password of user on C

A Simple Authentication Dialogue

Problems with Simple Authentication Dialogue

1. User needs different ticket for every different service. Such as if a user wished to access a print server, file server, mail server and so on. So need a different ticket for each logon service.
2. Need to avoid password transmitted as plaintext, otherwise eavesdropper could capture the password and use it for server accessibility.

A more Secure Authentication Dialogue

Once per user logon session:

(1) $C \rightarrow AS: ID_C \parallel ID_{tgs}$

(2) $AS \rightarrow C: E(K_C, Ticket_{tgs})$

Once per type of service:

(3) $C \rightarrow TGS: ID_C \parallel ID_V \parallel Ticket_{tgs}$

(4) $TGS \rightarrow C: Ticket_v$

Once per service session:

(5) $C \rightarrow V: ID_C \parallel Ticket_v$

$Ticket_{tgs} = E(K_{tgs}, [ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_1 \parallel Lifetime_1])$

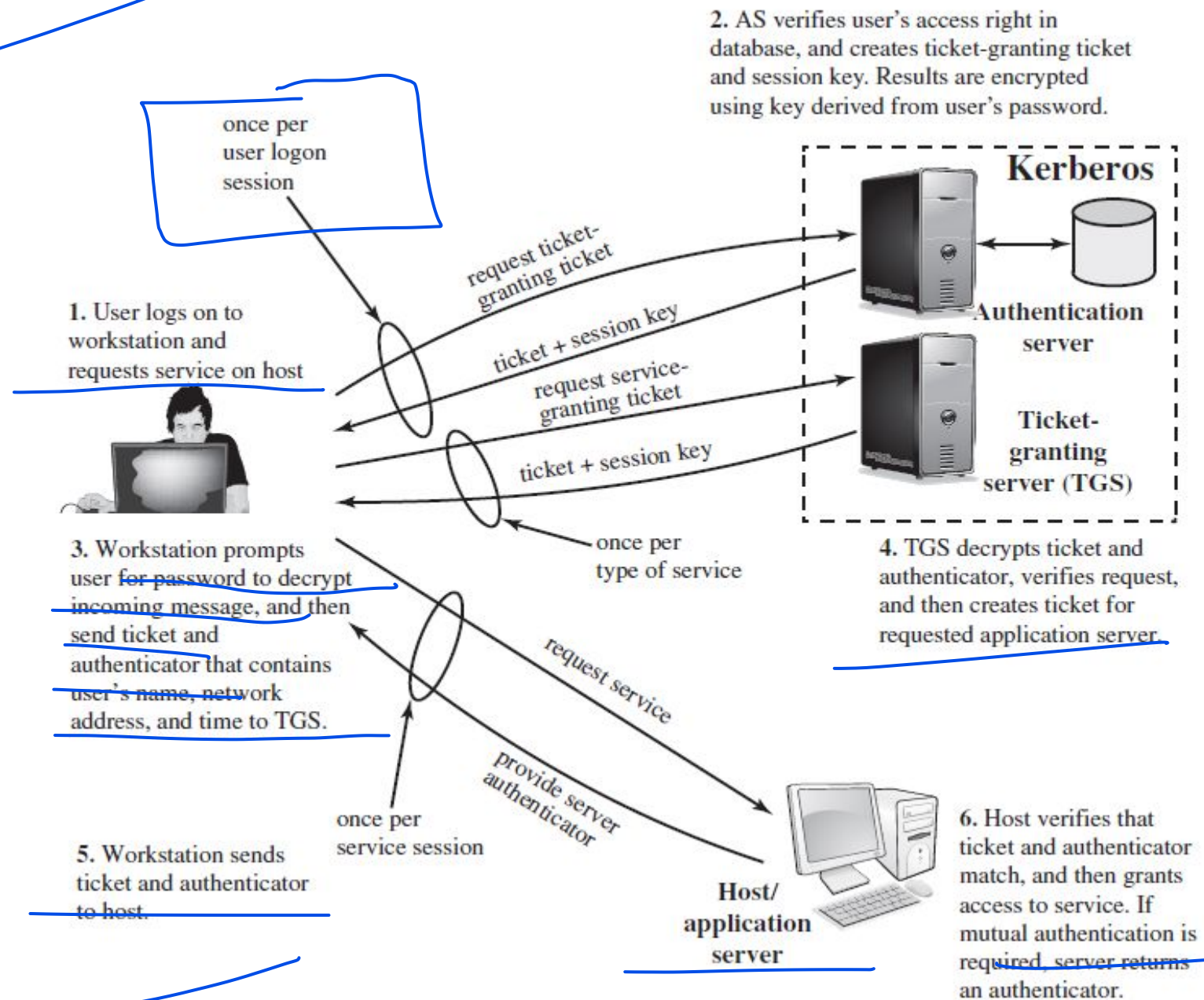
$Ticket_v = E(K_v, [ID_C \parallel AD_C \parallel ID_v \parallel TS_2 \parallel Lifetime_2])$

More Secure Authentication Dialogue

Problems with More Secure Authentication Dialogue

1. The lifetime is associated with ticket granting server-ticket.
 - If the lifetime is large , the opponent has greater opportunity to replay.
 - If the lifetime is very short, then the user will repeatedly ask for password.
2. There may be requirement for servers to authenticate themselves to users.

Overview of Kerberos



Kerberos 4 Authentication Dialogue

Table 15.1 Summary of Kerberos Version 4 Message Exchanges

(1) $C \rightarrow AS$ $ID_c \parallel ID_{tgs} \parallel TS_1$

(2) $AS \rightarrow C$ $E(K_c, [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS$ $ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4) $TGS \rightarrow C$ $E(K_{c,tgs}, [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_c \parallel AD_c \parallel TS_3])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V$ $Ticket_v \parallel Authenticator_c$

(6) $V \rightarrow C$ $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$Ticket_v = E(K_v, [K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4])$

$Authenticator_c = E(K_{c,v}, [ID_c \parallel AD_c \parallel TS_5])$

(c) Client/Server Authentication Exchange to obtain service

Kerberos Realms and Multiple Kerber...

A full-service Kerberos environment consisting of a Kerberos server, no of clients and no of application servers requires the following:

1. Kerberos server must have user ID and hashed passwords of all participating users in its database. All users registered with Kerberos server.
2. The Kerberos server must share a secret key with each server.
3. The Kerberos server shares a secret key with the server in other realm.

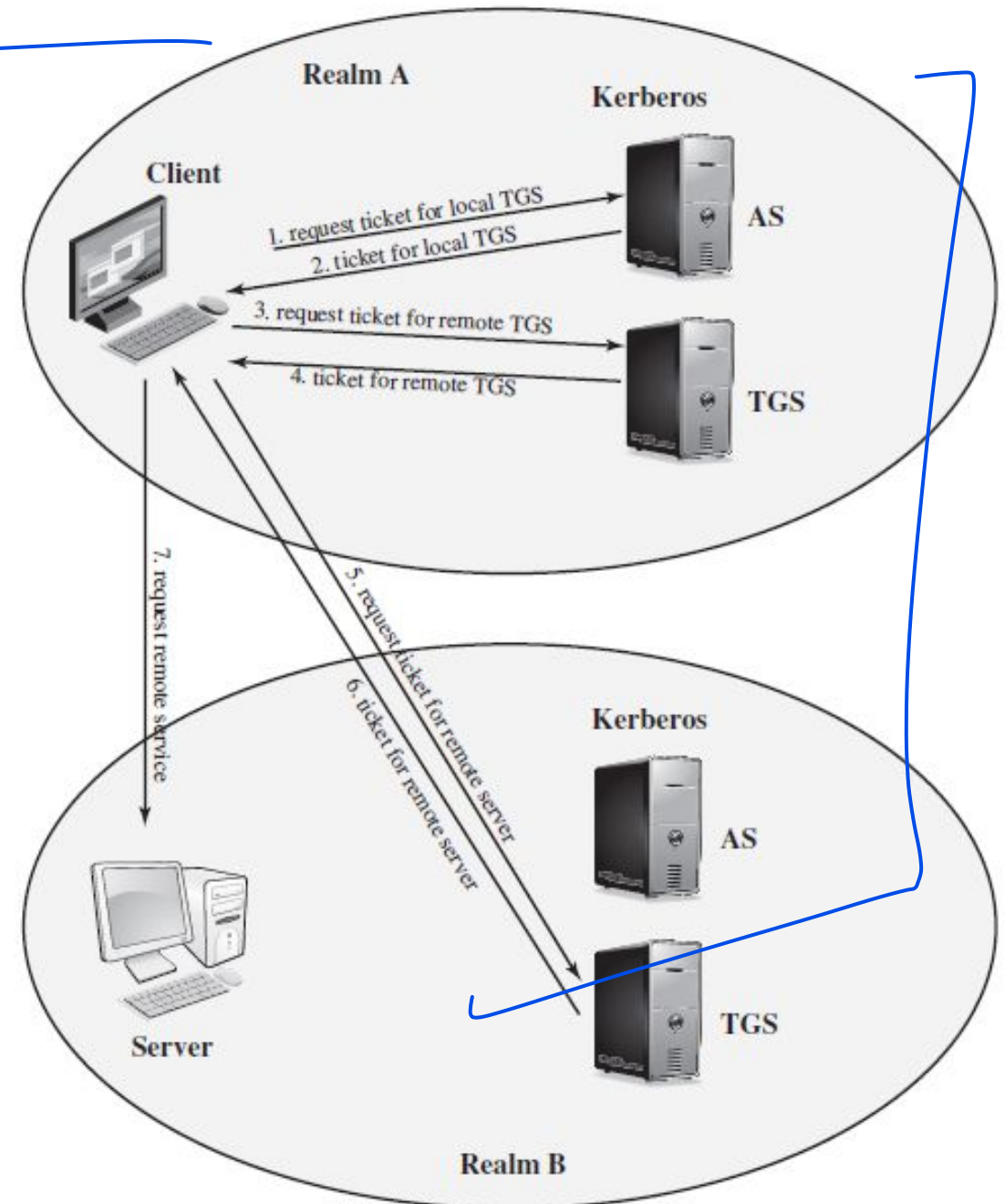
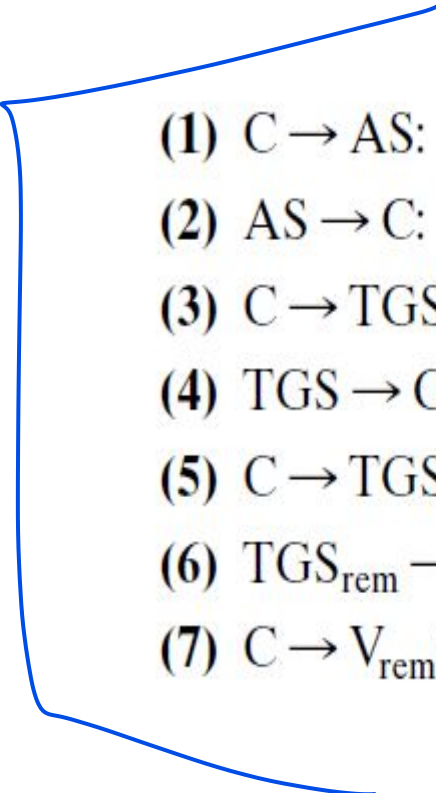
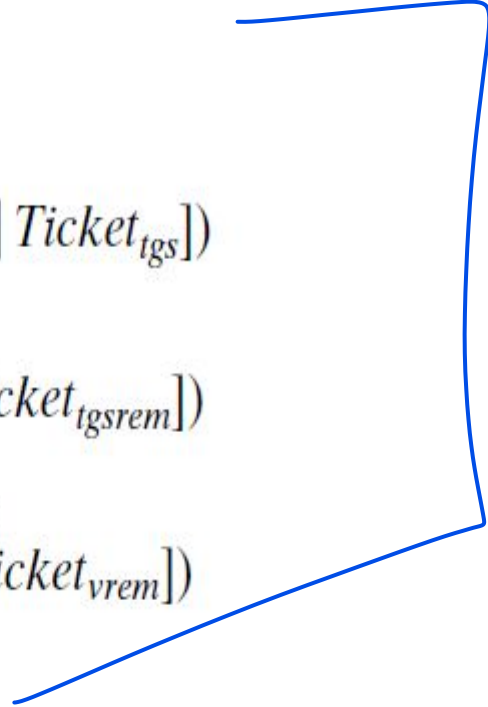


Figure 15.3 Request for Service in Another Realm

Kerberos Realms and Multiple Kerber

- 
- 
- (1) $C \rightarrow AS:$ $ID_c \parallel ID_{tgs} \parallel TS_1$
 - (2) $AS \rightarrow C:$ $E(K_c, [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}])$
 - (3) $C \rightarrow TGS:$ $ID_{tgsrem} \parallel Ticket_{tgs} \parallel Authenticator_c$
 - (4) $TGS \rightarrow C:$ $E(K_{c,tgs}, [K_{c,tgsrem} \parallel ID_{tgsrem} \parallel TS_4 \parallel Ticket_{tgsrem}])$
 - (5) $C \rightarrow TGS_{rem}:$ $ID_{vrem} \parallel Ticket_{tgsrem} \parallel Authenticator_c$
 - (6) $TGS_{rem} \rightarrow C:$ $E(K_{c,tgsrem}, [K_{c,vrem} \parallel ID_{vrem} \parallel TS_6 \parallel Ticket_{vrem}])$
 - (7) $C \rightarrow V_{rem}:$ $Ticket_{vrem} \parallel Authenticator_c$

Difference Between Kerberos 4 and Kerberos 5

Environmental Shortcomings of Kerberos 4

- a. Encryption System Dependence
- b. Internet protocol Dependence
- c. Message byte Ordering
- d. Ticket Lifetime: Max lifetime is 1280 minutes
- e. Authentication forwarding
- f. Interrealm authentication

Technical Deficiencies of Kerberos 4

- a. Double Encryption
- b. PCBC encryption
- c. Session Keys
- d. Password attacks

Kerberos 5 Authentication Dialogue

- (1) $C \rightarrow AS$ Options $\parallel ID_C \parallel Realm_c \parallel ID_{TGS} \parallel Times \parallel Nonce_1$
(2) $AS \rightarrow C$ $Realm_c \parallel ID_C \parallel Ticket_{TGS} \parallel E(K_c, [K_{c,TGS} \parallel Times \parallel Nonce_1 \parallel Realm_{TGS} \parallel ID_{TGS}])$
 $Ticket_{TGS} = E(K_{TGS}, [Flags \parallel K_{c,TGS} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

- (3) $C \rightarrow TGS$ Options $\parallel ID_V \parallel Times \parallel Nonce_2 \parallel Ticket_{TGS} \parallel Authenticator_c$
(4) $TGS \rightarrow C$ $Realm_c \parallel ID_C \parallel Ticket_V \parallel E(K_{c,TGS}, [K_{c,V} \parallel Times \parallel Nonce_2 \parallel Realm_V \parallel ID_V])$
 $Ticket_{TGS} = E(K_{TGS}, [Flags \parallel K_{c,TGS} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$
 $Ticket_V = E(K_V, [Flags \parallel K_{c,V} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$
 $Authenticator_c = E(K_{c,TGS}, [ID_C \parallel Realm_c \parallel TS_1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

- (5) $C \rightarrow V$ Options $\parallel Ticket_V \parallel Authenticator_c$
(6) $V \rightarrow C$ $E_{K_{C,V}} [TS_2 \parallel Subkey \parallel Seq\#]$
 $Ticket_V = E(K_V, [Flags \parallel K_{c,V} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$
 $Authenticator_c = E(K_{c,V}, [ID_C \parallel Realm_c \parallel TS_2 \parallel Subkey \parallel Seq\#])$

(c) Client/Service Authentication Exchange to obtain service

Pretty Good Privacy

- Pretty Good Privacy (PGP) is an encryption software program designed to ensure the confidentiality, integrity, and authenticity of virtual communications and information.

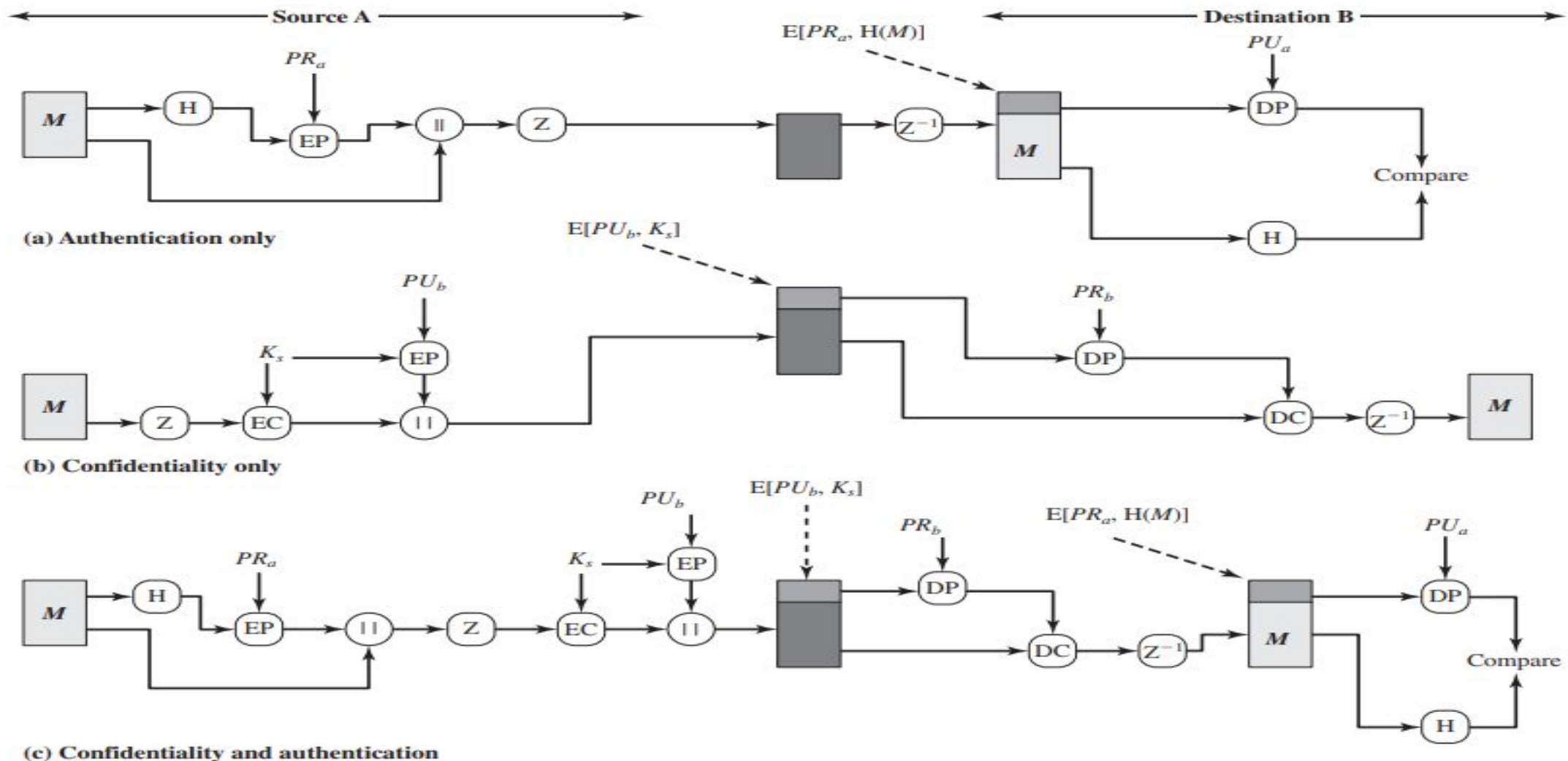


Figure 19.1 PGP Cryptographic Functions

Pretty Good Privacy

- Pretty Good Privacy (PGP) is an encryption software program designed to ensure the confidentiality, integrity, and authenticity of virtual communications and information.
- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

PGP Services

Pretty Good Privacy

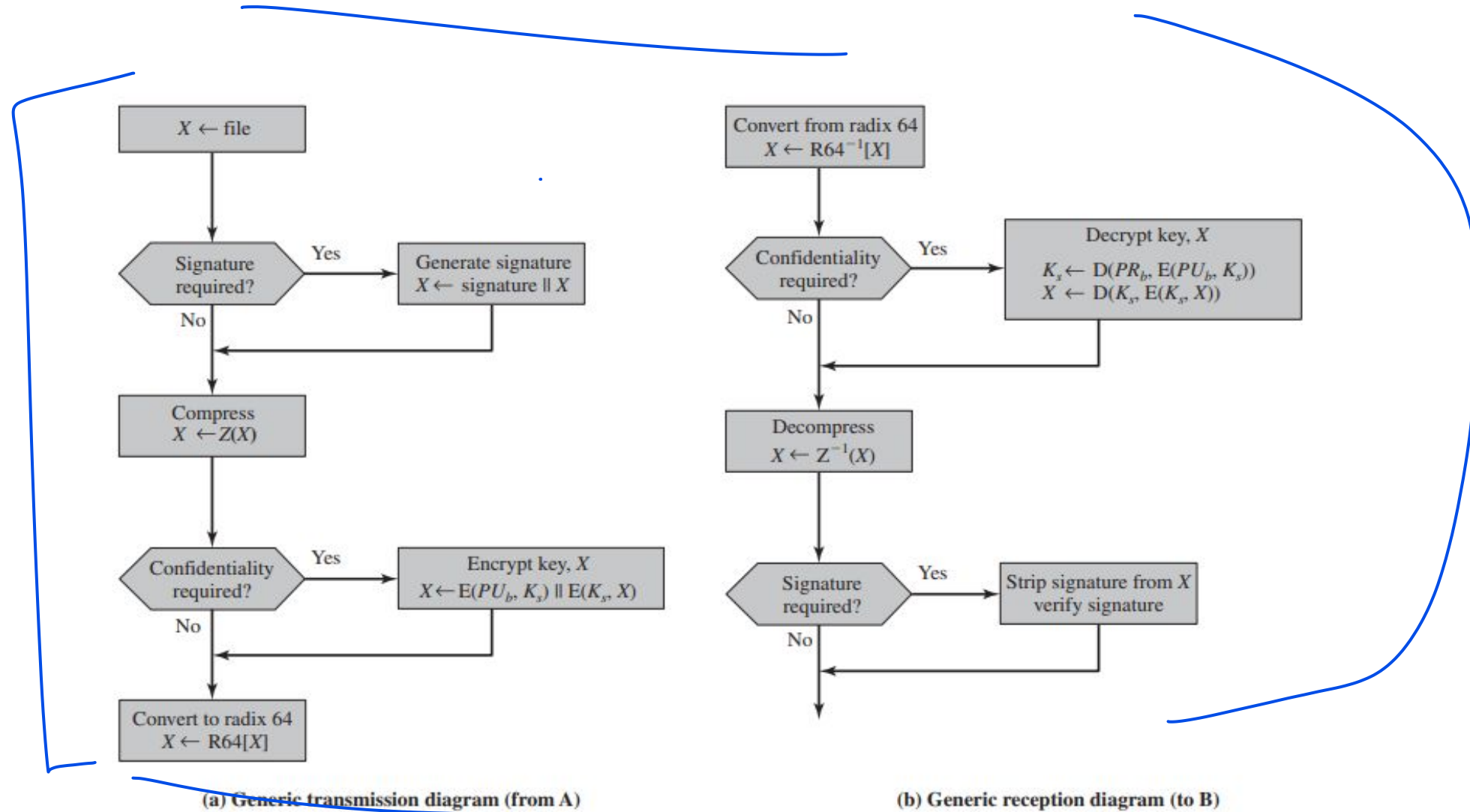


Figure 19.2 Transmission and Reception of PGP Messages

X.509 certificates

- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- X.509 is based on the use of public-key cryptography and digital signatures.

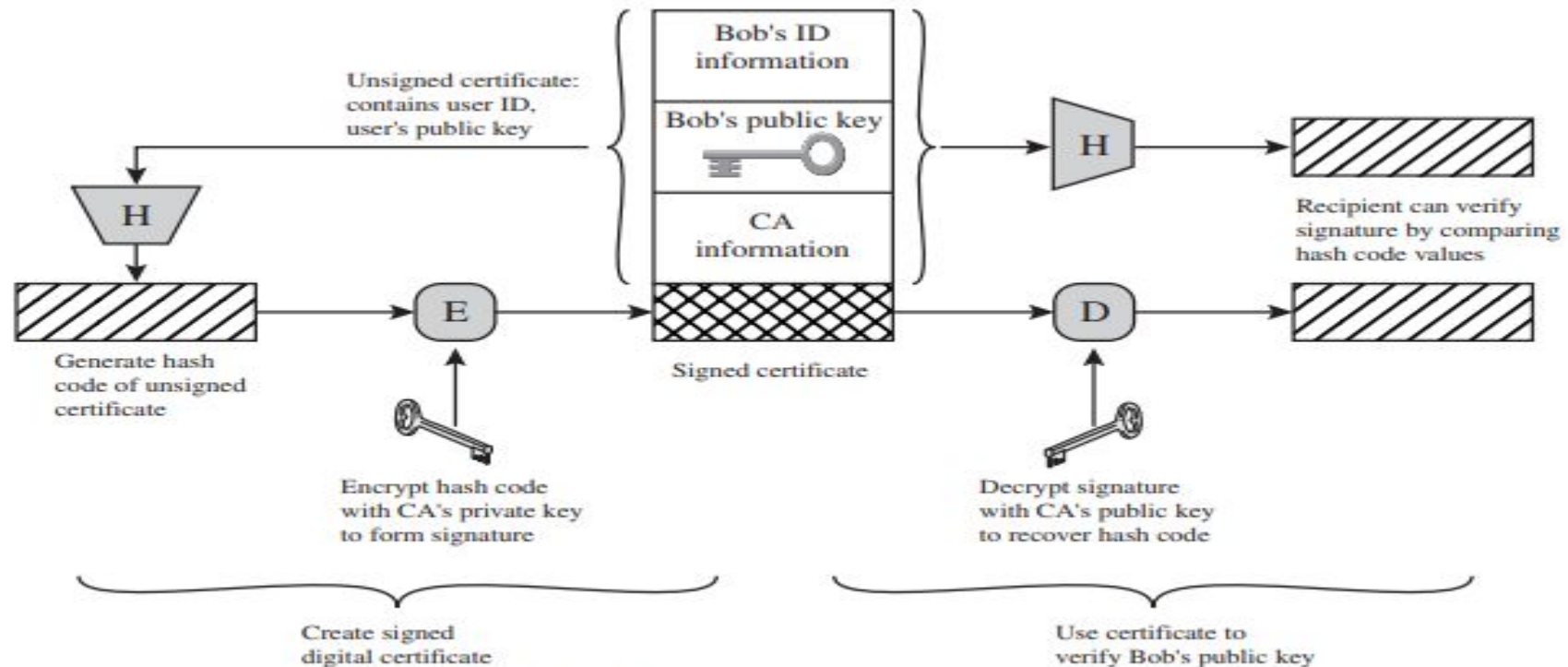
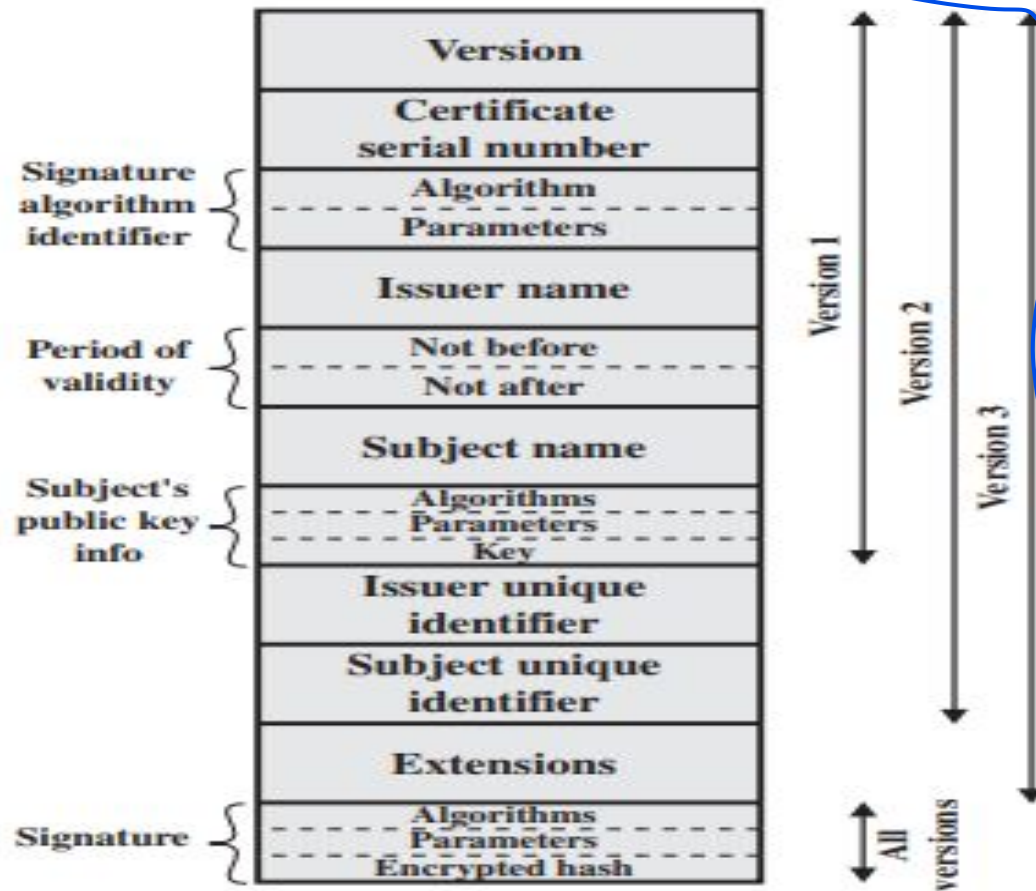


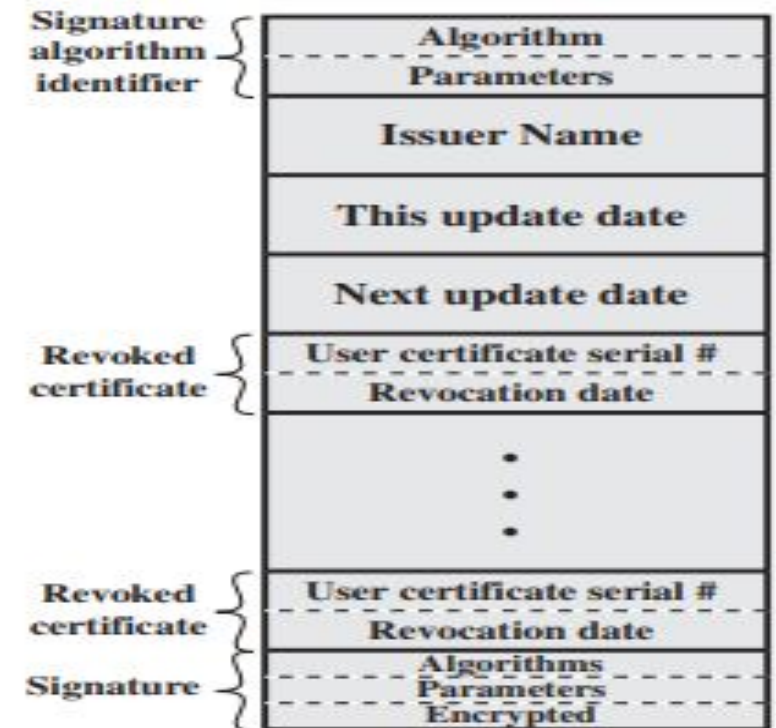
Figure 14.14 Public-Key Certificate Use

Certificates



(a) X.509 certificate

Figure 14.15 X.509 Formats



(b) Certificate revocation list

X.509 Hierarchy: A Hypothetical Example

In this example, user A can acquire the following certificates from the directory to establish a certification path to B:

$$X \ll W \gg W \ll V \gg V \ll Y \gg Y \ll Z \gg Z \ll B \gg$$

When A has obtained these certificates, it can unwrap the certification path in sequence to recover a trusted copy of B's public key. Using this public key, A can send encrypted messages to B. If A wishes to receive encrypted messages back from B, or to sign messages sent to B, then B will require A's public key, which can be obtained from the following certification path:

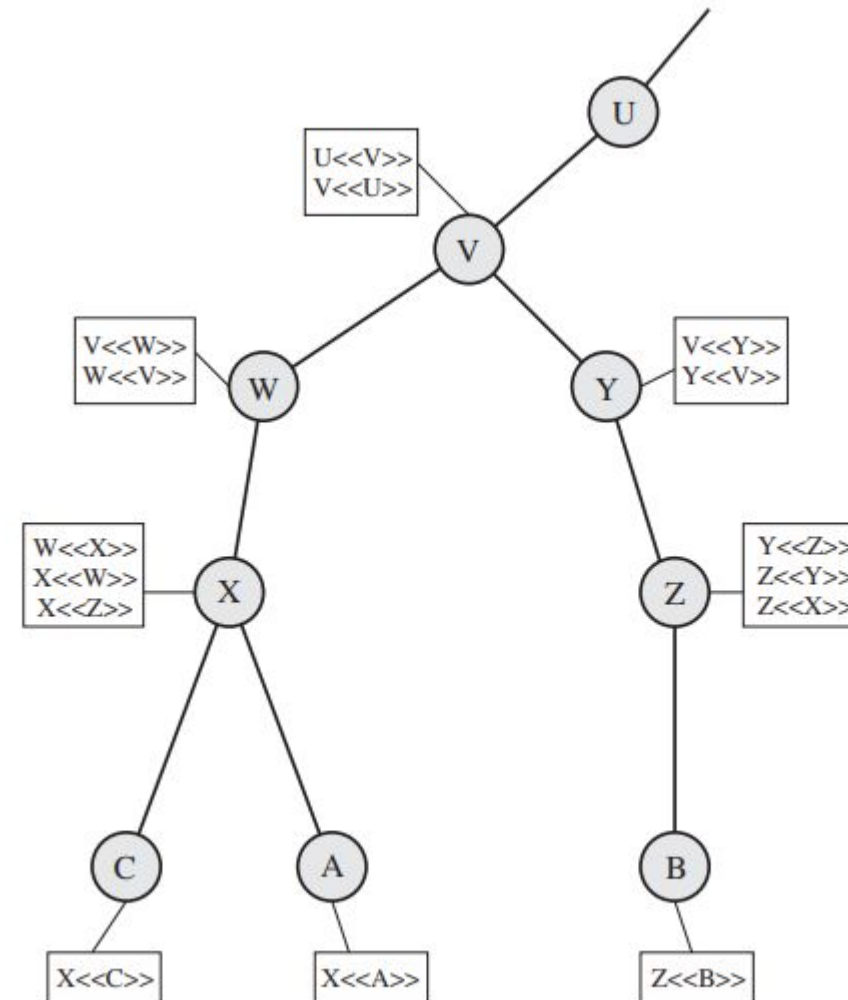
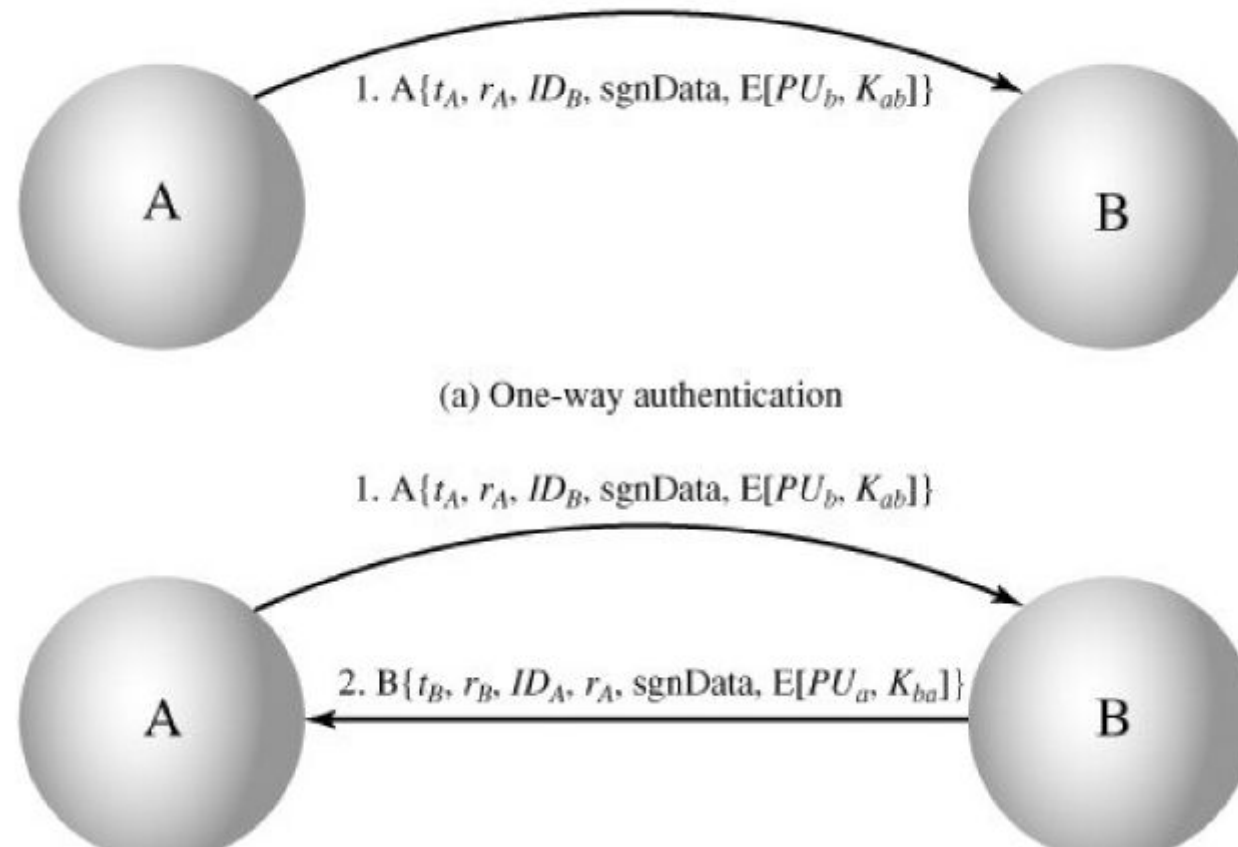
$$Z \ll Y \gg Y \ll V \gg V \ll W \gg W \ll X \gg X \ll A \gg$$


Figure 14.16 X.509 Hierarchy: A Hypothetical Example

X.509 Strong Authentication Procedures

Figure 14.6. X.509 Strong Authentication Procedures



Public Key Infrastructure

Public-key infrastructure (PKI) is the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

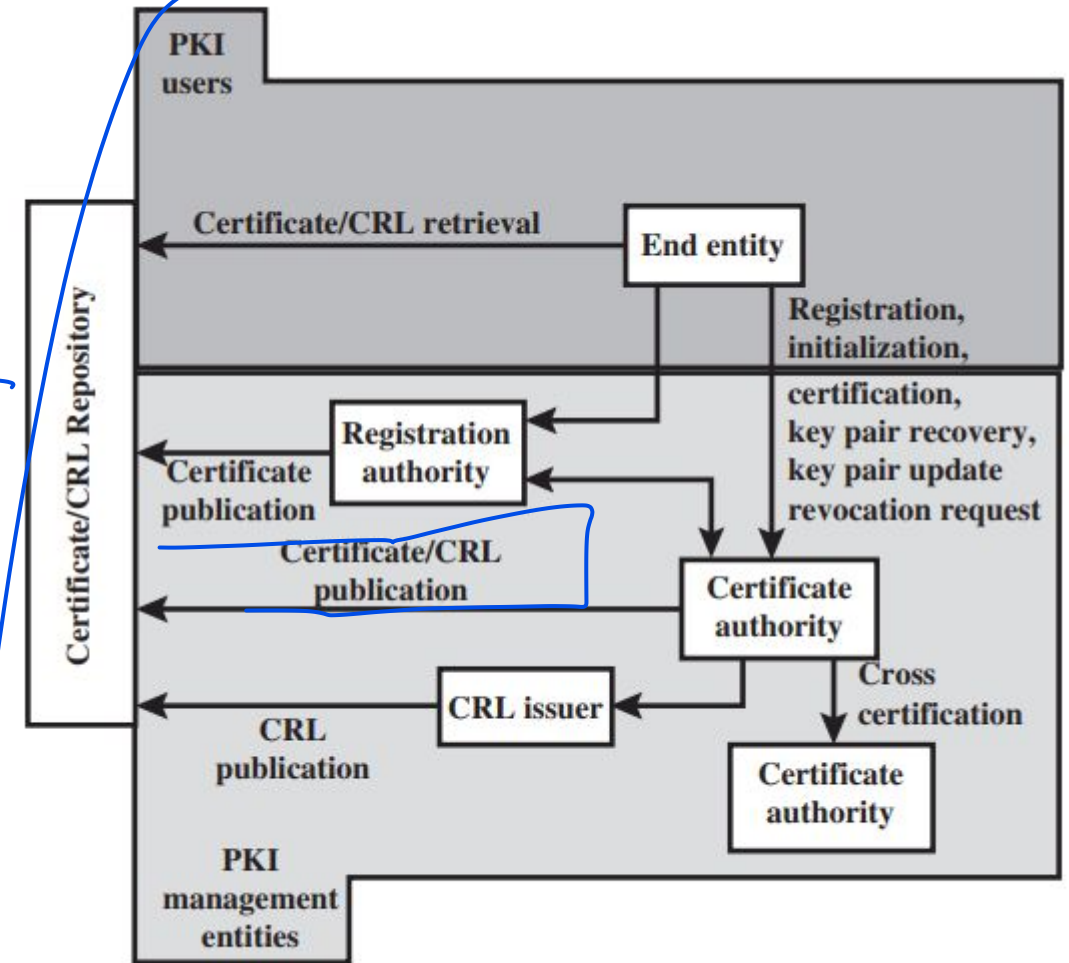


Figure 14.17 PKIX Architectural Model

Public Key Infrastructure

The elements of PKIX Model are

End entity: A generic term used to denote end users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public key certificate. End entities typically consume and/or support PKI-related services.

Certification authority (CA): The issuer of certificates and (usually) certificate revocation lists (CRLs). It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.

Registration authority (RA): An optional component that can assume a number of administrative functions from the CA. The RA is often associated with the End Entity registration process, but can assist in a number of other areas as well.

CRL issuer: An optional component that a CA can delegate to publish CRLs.

Repository: A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by End Entities.

Public Key Infrastructure

PKIX Management Functions:

Registration: This is the process whereby a user first makes itself known to a CA (directly, or through an RA), prior to that CA issuing a certificate or certificates for that user. Registration begins the process of enrolling in a PKI.

Initialization: Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure. For example, the client needs to be securely initialized with the public key and other assured information of the trusted CA(s), to be used in validating certificate paths.

Certification: This is the process in which a CA issues a certificate for a user's public key, and returns that certificate to the user's client system and/or posts that certificate in a repository.

Key pair recovery: Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both.

Key pair update: All key pairs need to be updated regularly (i.e., replaced with a new key pair) and new certificates issued. Update is required when the certificate lifetime expires and as a result of certificate revocation.

Revocation request: An authorized person advises a CA of an abnormal situation requiring certificate revocation. Reasons for revocation include private key compromise, change in affiliation, and name change.

S/MIME

- S/MIME stands for Secure/Multipurpose Internet Mail Extensions.
- Through encryption, S/MIME offers protection for business emails.
- S/MIME is a protocol used for encrypting or decrypting digitally signed E-mails. This means that users can digitally sign their emails as the owner(sender) of the e-mail.

Earlier Standards:

1. RFC5322:

RFC 5322 defines a format for text messages that are sent using electronic mail.

In the RFC 5322 context, messages are viewed as having an envelope and contents.

2. MIME

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP)

Problems with SMTP---RFC 5322

1. SMTP cannot transmit executable files or other binary objects.
2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle non textual data included in X.400 messages.

Header fields in MIME

- **MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- **Content-Type:** Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.
- **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
- **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description:** A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

Table 19.2 MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript format.
	octet-stream	General binary data consisting of 8-bit bytes.

example.

From: Nathaniel Borenstein <nsb@bellcore.com>

To: Ned Freed <ned@innosoft.com>

Subject: Formatted text mail

MIME-Version: 1.0

Content-Type: multipart/alternative;
boundary=boundary42

--boundary42

Content-Type: text/plain; charset=us-ascii

... plain text version of message goes here....

--boundary42

Content-Type: text/enriched

.... RFC 1896 text/enriched version of same message
goes here ...

boundary42

Table 19.3 MIME Transfer Encodings

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present, but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

S/MIME Functionality

Enveloped data: This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.

- **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.

- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64

- **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

Cryptographic Algorithms

Table 19.5 Cryptographic Algorithms Used in S/MIME

Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility.
Encrypt message digest to form a digital signature.	Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with a message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with a one-time session key.	Sending and receiving agents MUST support encryption with tripleDES. Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code.	Receiving agents MUST support HMAC with SHA-1. Sending agents SHOULD support HMAC with SHA-1.

Advantages of S/MIME

1. It is available in various modern mail agents like Netscape, MS outlook etc.
2. It is utilized in commercial or Industrial settings.
3. It ensures the message's validity and security.
4. The digital signature protects the email by using email spoofing

How does S/MIME Certificate Works



S/MIME Content Types

Table 15.7. S/MIME Content Types

(This item is displayed on page 468 in the print version)

Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs 7-mime	signedData	A signed S/MIME entity.
	pkcs 7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs 7-mime	degenerate signedData	An entity containing only public- key certificates.
	pkcs 7-mime	CompressedData	A compressed S/MIME entity
	pkcs 7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

IP Security

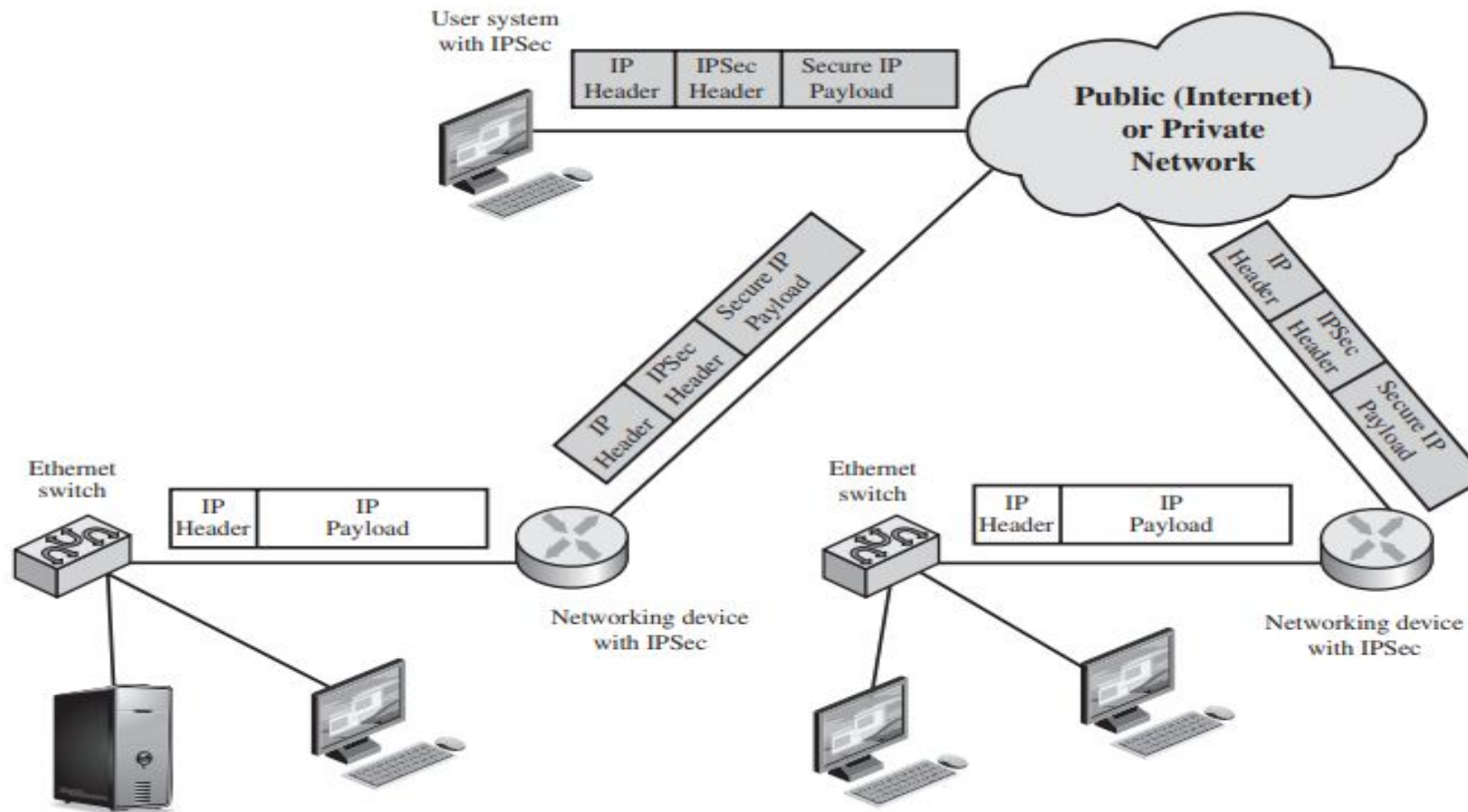


Figure 20.1 An IP Security Scenario

IP Security

- ❑ There need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end user-to-end-user traffic using authentication and encryption mechanism.
- ❑ IPSec provides the capability to secure communications across LAN, across private and public WANs and across internet.

Applications of IPSec

- ❑ Secure Branch office connectivity over the internet
- ❑ Secure Remote access over the internet
- ❑ Establishing extranet and intranet connectivity with partners
- ❑ Enhancing electronic commerce security
- ❑ IPsec that enables it to support these varied applications is that it can encrypt and/or authenticate all traffic at the IP level.
- ❑ Thus, all distributed applications (including remote logon, client/server, e-mail, file transfer, Web access, and so on) can be secured.

Benefits of IPSec

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPsec is implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization

Routing Applications

IPSec can ensure that:

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbor advertisement (a router seeks to establish or maintain a neighbor relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial IP packet was sent.
- A routing update is not forged.

IP Security Architecture

IPSec Documents:

The IPSec specification consists of numerous documents.

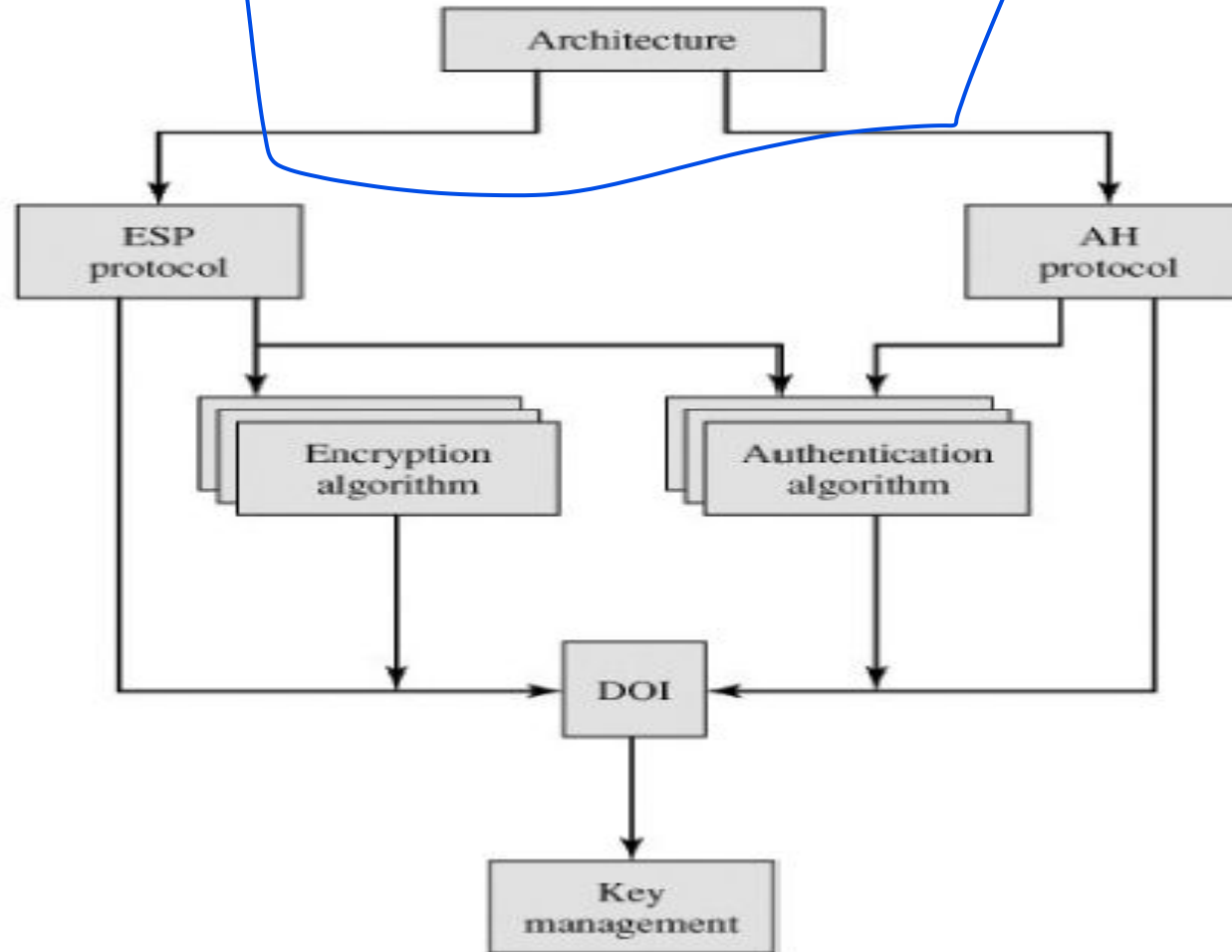
The most important of these, issued in November of 1998, are RFCs 2401, 2402, 2406, and 2408:

- RFC 2401: An overview of a security architecture
- RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- RFC 2408: Specification of key management capabilities

IP Security Architecture

IPSec Documents:

Figure 16.2. IPSec Document Overview
(This item is displayed on page 488 in the print version)



IP Security Architecture– IPSec Documents

Architecture: Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.

Encapsulating Security Payload (ESP): Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.

Authentication Header (AH): Covers the packet format and general issues related to the use of AH for packet authentication.

Encryption Algorithm: A set of documents that describe how various encryption algorithms are used for ESP.

Authentication Algorithm: A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.

Key Management: Documents that describe key management schemes.

Domain of Interpretation (DOI): Contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime.

IP Security Architecture– IPSec Services

- ❑ Access control
- ❑ Connectionless integrity
- ❑ Data origin authentication
- ❑ Rejection of replayed packets (a form of partial sequence integrity)
- ❑ Confidentiality (encryption)
- ❑ Limited traffic flow confidentiality

Table 16.1. IPSec Services

(This item is displayed on page 490 in the print version)

[View full size image](#)

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✓	✓	✓
Connectionless integrity	✓		✓
Data origin authentication	✓		✓
Rejection of replayed packets	✓	✓	✓
Confidentiality		✓	✓
Limited traffic flow confidentiality		✓	✓

IP Security Architecture–Security Associations

A security association is uniquely identified by three parameters:

Security Parameters Index (SPI): A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.

IP Destination Address: Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end user system or a network system such as a firewall or router.

Security Protocol Identifier: This indicates whether the association is an AH or ESP security association.

SA Parameters

Sequence Number Counter: A 32-bit value used to generate the Sequence Number field in AH or ESP headers.

Sequence Counter Overflow: A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA (required for all implementations).

Anti-Replay Window: Used to determine whether an inbound AH or ESP packet is a replay.

AH Information: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.

IP Security Architecture—Security Associations

SA Parameters

ESP Information: Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations).

Lifetime of This Security Association: A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur (required for all implementations).

IPSec Protocol Mode: Tunnel, transport, or wildcard (required for all implementations).

Path MTU: Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations).

Security Selectors

Security Policy Database (SPD) contains entries, each of which defines subset of IP traffic and points to an SA for that traffic.

Each SPD entry is defined by a set of IP and upper-layer protocol field values, called *selectors*. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA. Outbound processing obeys the following general sequence for each IP packet:

- Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs.
- Determine the SA if any for this packet and its associated SPI.
- Do the required IPsec processing (i.e., AH or ESP processing).

Security Associations (SA) Selectors

The following selectors determine SPD entry

Destination IP Address: This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address.

Source IP Address: This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address.

UserID: A user identifier from the operating system. This is not a field in the IP or upper-layer headers but is available if IPSec is running on the same operating system as the user.

Data Sensitivity Level: Used for systems providing information flow security (e.g., Secret or Unclassified).

Transport Layer Protocol: Obtained from the IPv4 Protocol or IPv6 Next Header field. This may be an individual protocol number, a list of protocol numbers, or a range of protocol numbers.

Source and Destination Ports: These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

IP Security Architectures- Transport and Tunnel modes.

Both AH and ESP supports 2 modes.

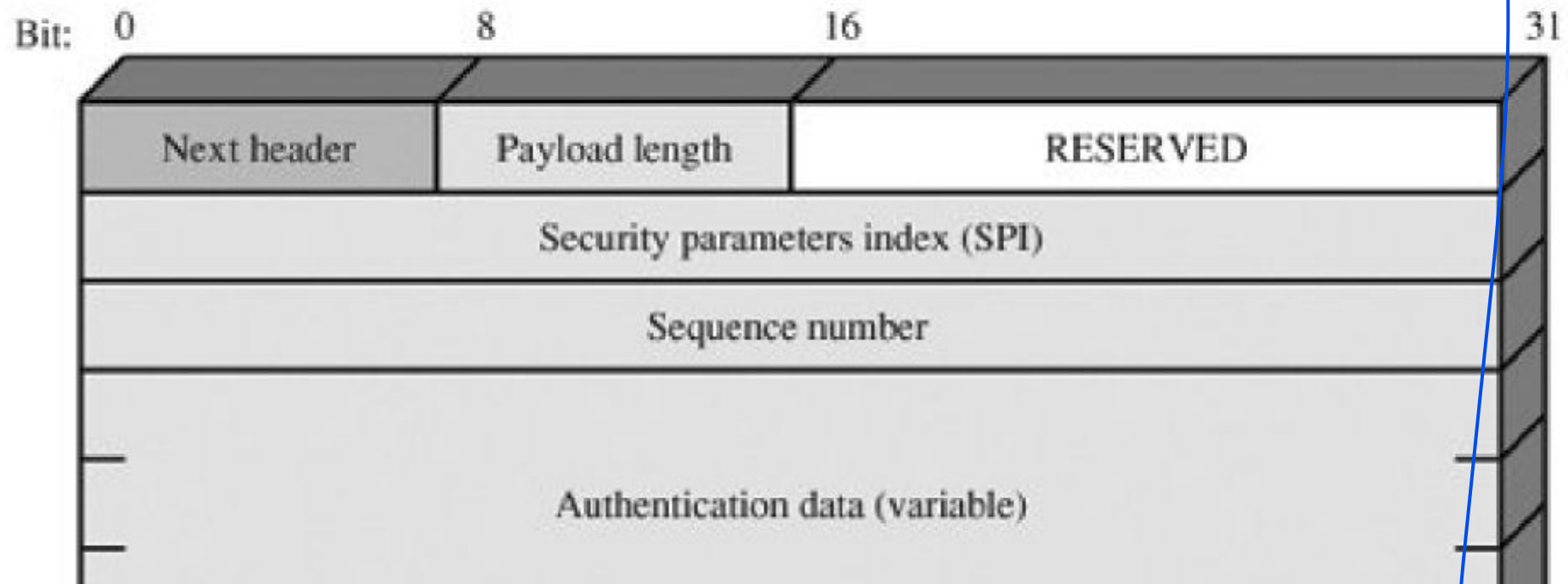
- Transport Mode
- Tunnel Mode

Table 16.2. Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

Authentication Header

Figure 16.3. IPSec Authentication Header



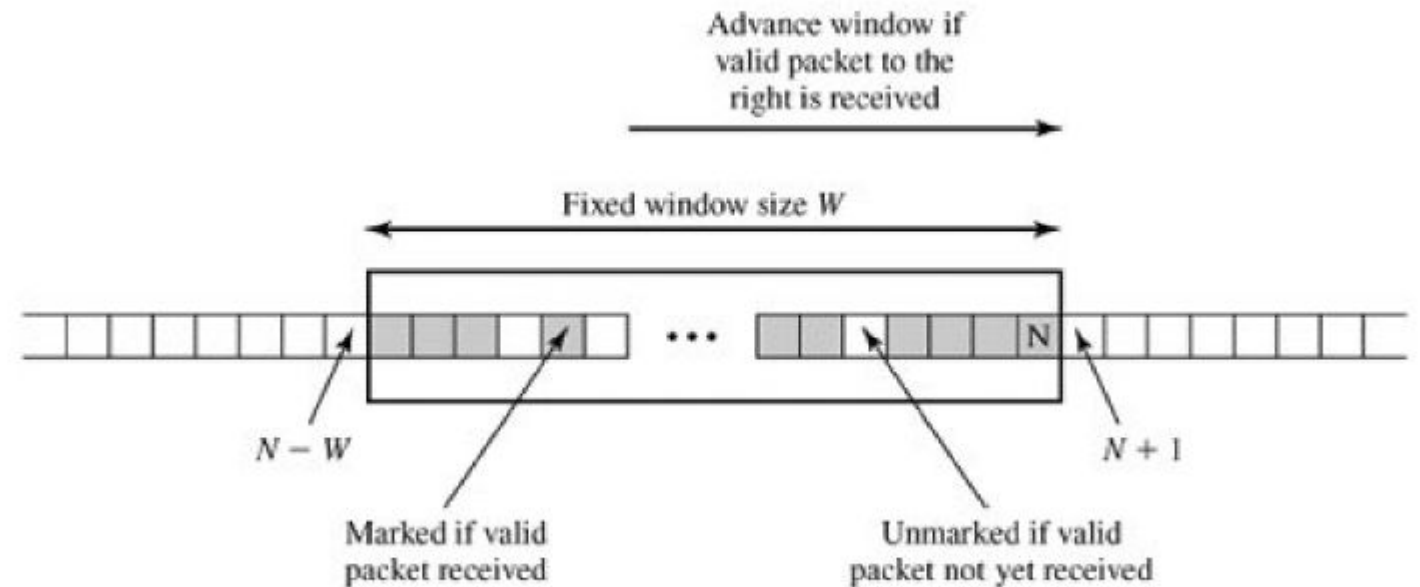
Authentication Header

Authentication Header consists of following fields.

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Payload Length (8 bits):** Length of Authentication Header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC, for this packet.

Anti Replay Mechanism

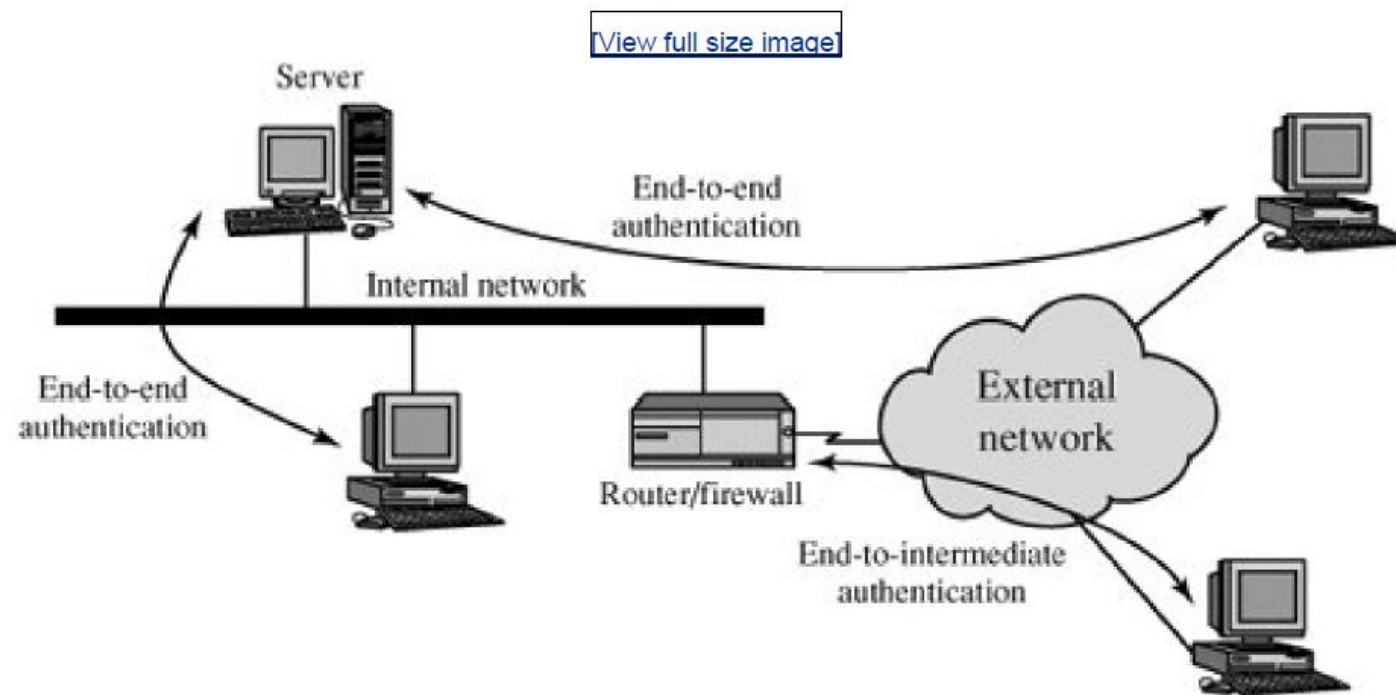
- A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination.
- The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence.
- The Sequence Number field is designed to thwart such attacks



Authentication Header

Transport Mode and Tunnel Mode in Authentication

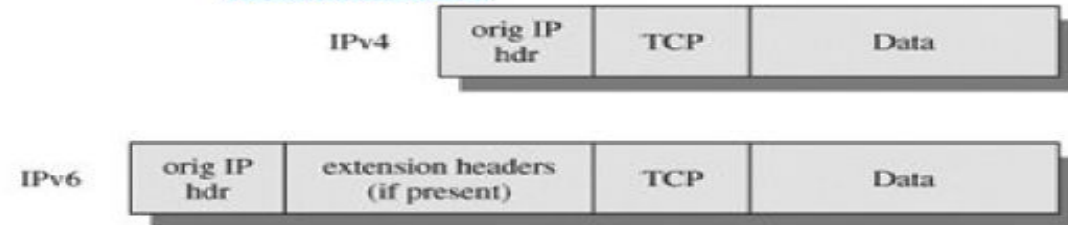
Figure 16.5. End-to-End versus End-to-Intermediate Authentication



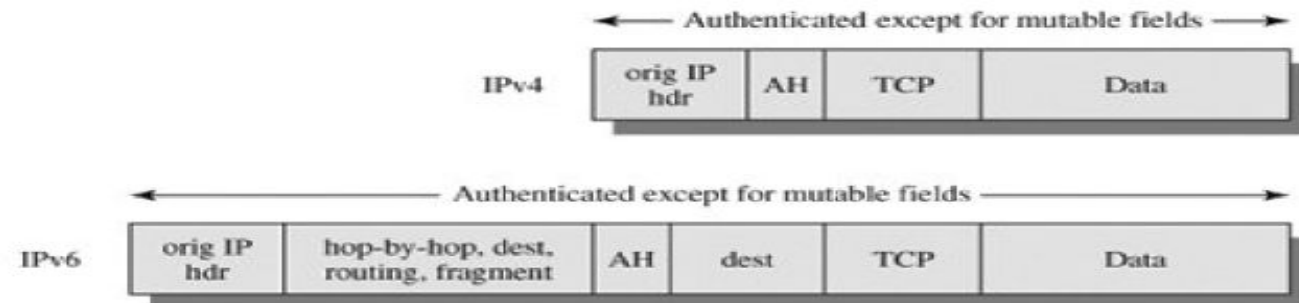
Tunnel Mode and Transport Mode in Authentication

Figure 16.6. Scope of AH Authentication

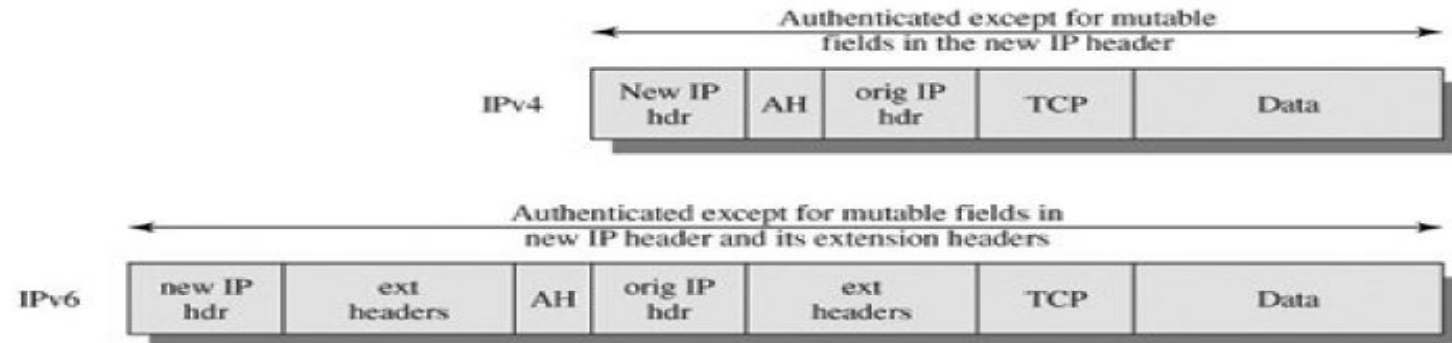
[View full size image](#)



(a) Before applying AH

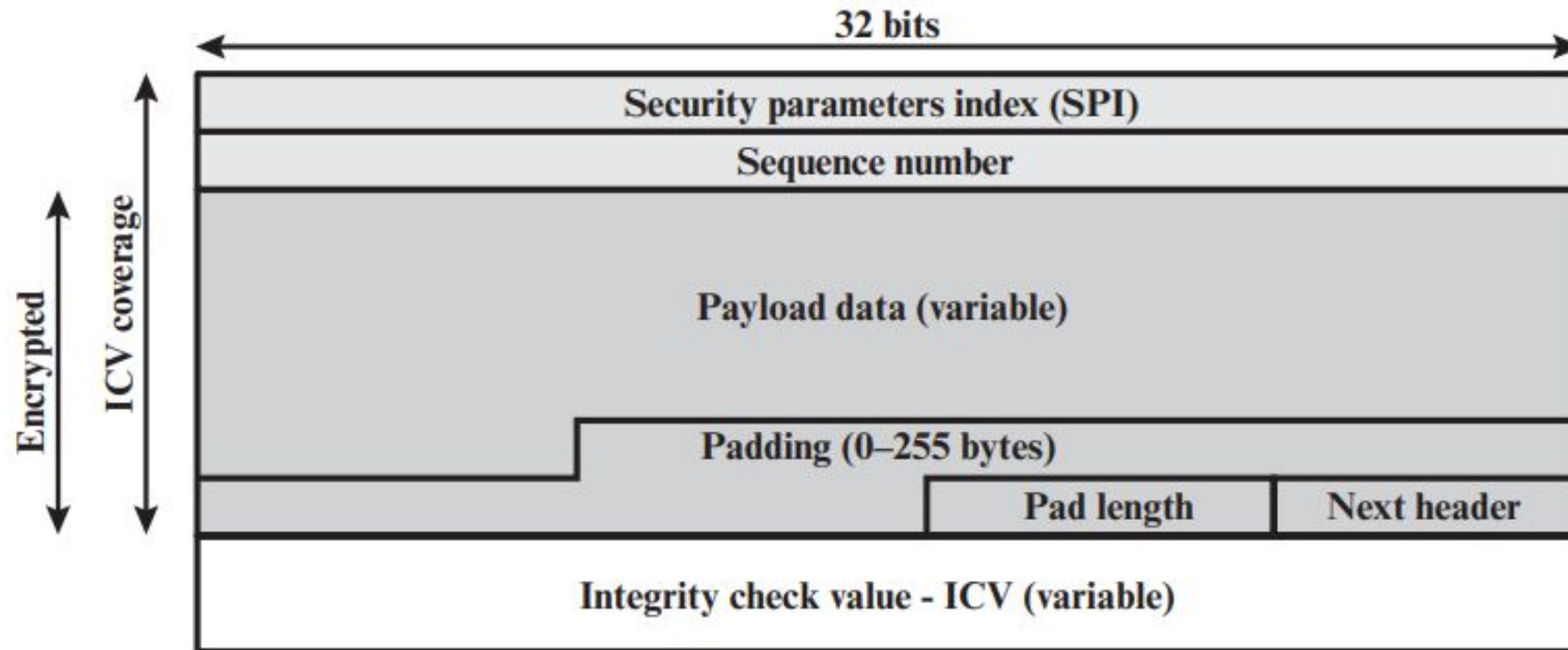


(b) Transport mode



(c) Tunnel mode

Encapsulating Security Payload



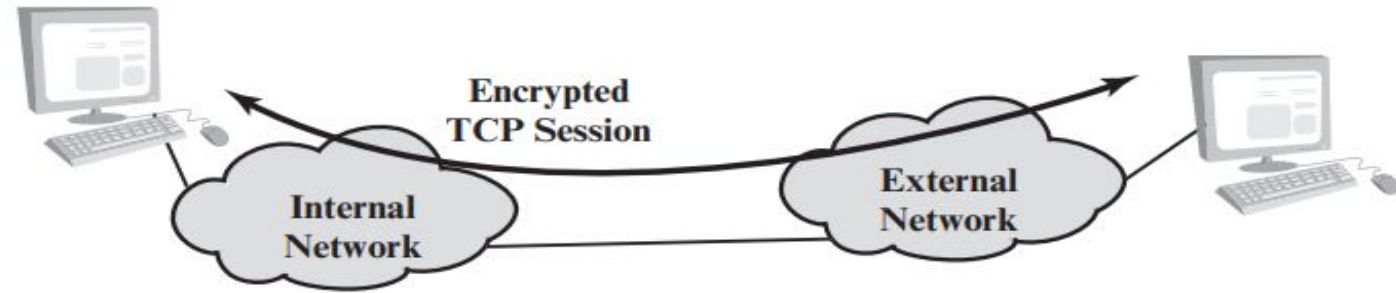
(a) Top-level format of an ESP Packet

Encapsulating Security Payload

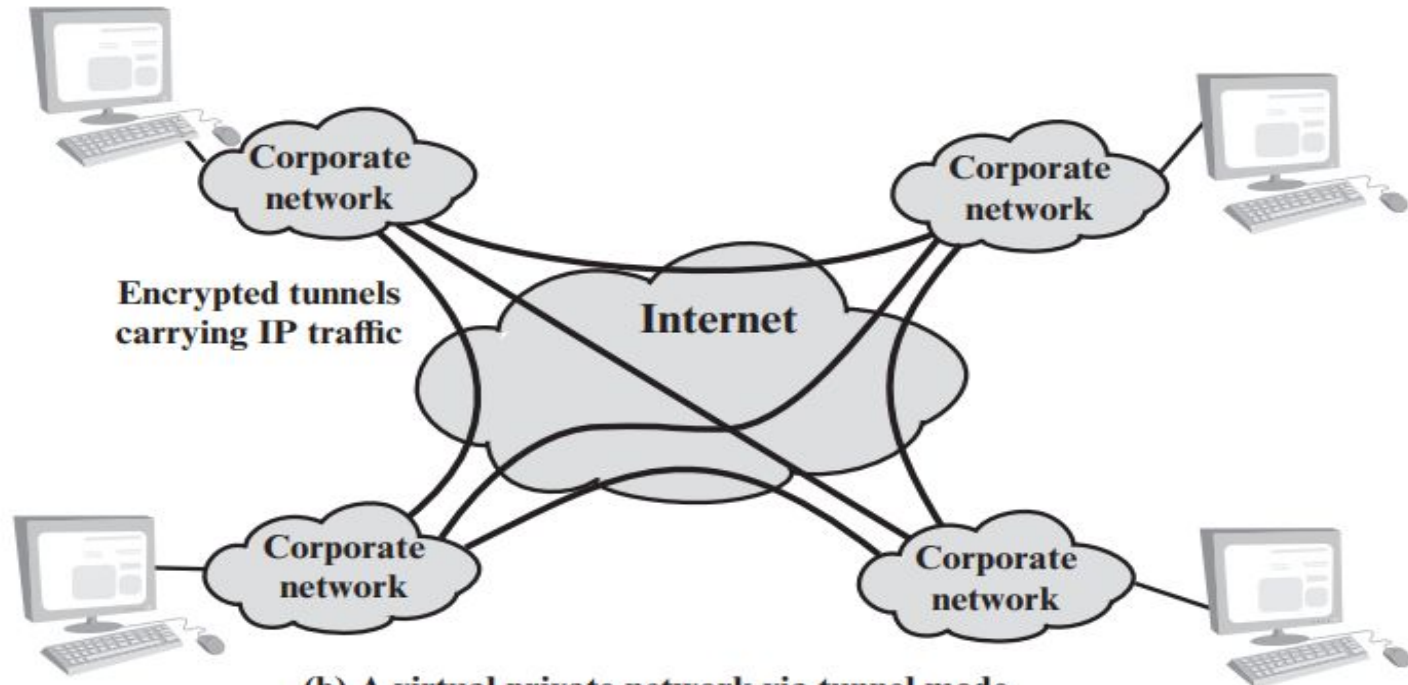
Encryption and Authentication Algorithms

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext.

Transport Mode Versus Tunnel Mode Encryption



(a) Transport-level security

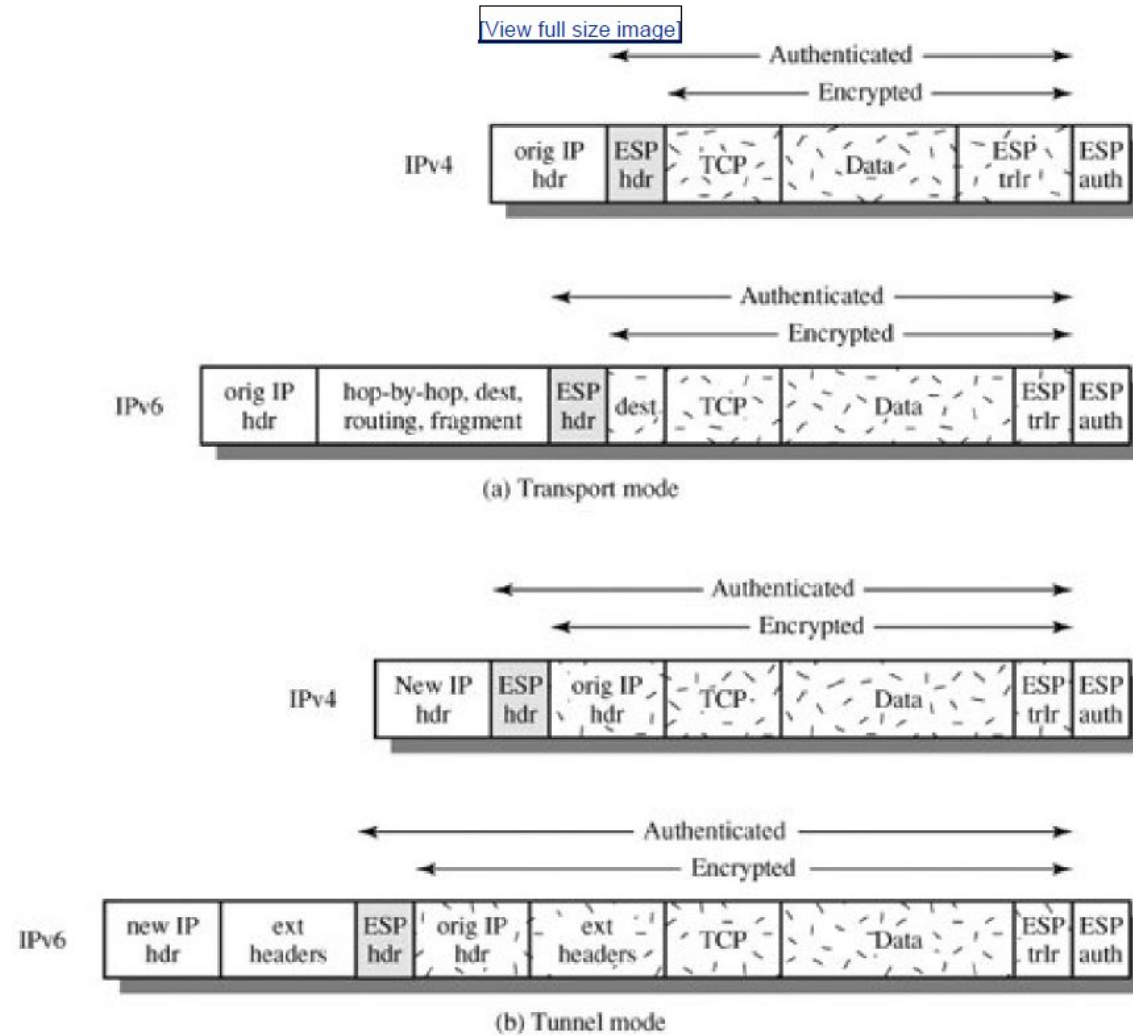


(b) A virtual private network via tunnel mode

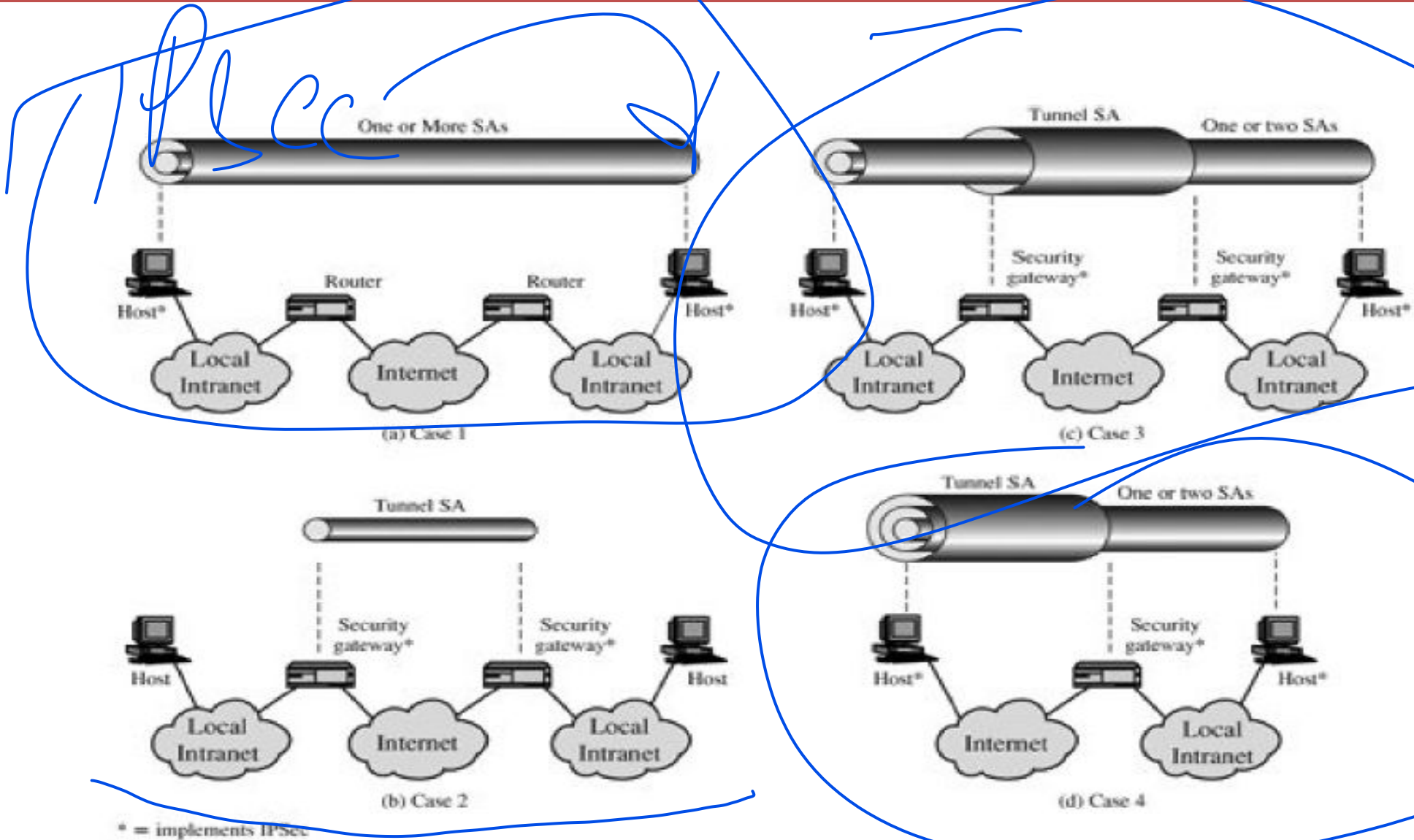
Figure 20.7 Transport-Mode versus Tunnel-Mode Encryption

Transport Mode Versus Tunnel Mode Encryption

Figure 16.9. Scope of ESP Encryption and Authentication



Combining Security Associations



Combining Security Associations

Security associations may be combined into bundles in two ways:

Transport adjacency: Refers to applying more than one security protocol to the same IP packet, without invoking tunneling.

Iterated tunneling: Refers to the application of multiple layers of security protocols effected through IP tunneling.

Combining Security Associations

In **Case 1**, all security is provided between end systems that implement IPsec. For any two end systems to communicate via an SA, they must share the appropriate secret keys.

Among the possible combinations:

- a. AH in transport mode
- b. ESP in transport mode
- c. ESP followed by AH in transport mode (an ESP SA inside an AH SA)
- d. Any one of a, b, or c inside an AH or ESP in tunnel mode

For **Case 2**, security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support.

Case 3 builds on Case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication or confidentiality or both for all traffic between end systems.

Case 4 provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall.

Key Management

- The key management portion of IPSec involves the determination and distribution of secret keys.
- A typical requirement is four keys for communication between two applications: transmit and receive pairs for both AH and ESP.

The IPSec Architecture document support for two types of key management:

- **Manual:** A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
- **Automated:** An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration.

Key Management

The default automated key management protocol for IPSec is referred to as ISAKMP/Oakley and consists of the following elements:

- **Oakley Key Determination Protocol:** Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats.
- **Internet Security Association and Key Management Protocol (ISAKMP):** ISAKMP provides a framework for internet key management and provides the specific protocol support, including formats, for negotiation of security attributes.

Key Management

Diffie Hellman Key Exchange algorithm:

- Alice and Bob agree on two large prime numbers, p and g , and a public key exchange algorithm.
- Alice chooses a secret integer, a , and computes $A = g^a \bmod p$. She sends A to Bob.
- Bob chooses a secret integer, b , and computes $B = g^b \bmod p$. He sends B to Alice.
- Alice computes $s = B^a \bmod p$. Bob computes $s = A^b \bmod p$.
- Alice and Bob now both have shared secret keys, which they can use to establish a secure communication channel.

Global Public Elements

q prime number

α $\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A $X_A < q$

Calculate public Y_A $Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B $X_B < q$

Calculate public Y_B $Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$$K = (Y_B)^{X_A} \bmod q$$

Calculation of Secret Key by User B

$$K = (Y_A)^{X_B} \bmod q$$

Key Management-- Oakley Key Determination Protocol

Attractive features of Diffie Hellman Key Exchange

1. Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability.
2. The exchange requires no preexisting infrastructure other than an agreement on the global parameters.

Weaknesses of Diffie Hellman Key Exchange Algorithm

1. It does not provide any information about the identities of the parties.
2. It is subject to a man-in-the-middle attack, in which a third party C impersonates B while communicating with A and impersonates A while communicating with B.
3. It is computationally intensive. As a result, it is vulnerable to a clogging attack, in which an opponent requests a high number of keys.

Oakley Key Determination Protocol---Features

1. It employs a mechanism known as cookies to thwart clogging attacks.
2. it enables the two parties to negotiate a *group*; this, in essence, specifies the global parameters of the Diffie-Hellman key exchange.
3. It uses nonces to ensure against replay attacks.
4. It enables the exchange of Diffie-Hellman public key values.
5. It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

ISAKMP (Internet Security Association and Key Management Protocol) mandates cookie generation satisfies 3 requirements

- The cookie must depend on the specific parties. This prevents an attacker from obtaining a cookie using a real IP address and UDP port and then using it to swamp the victim with requests from randomly chosen IP addresses or ports.
- It must not be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity.
- The cookie generation and verification methods must be fast to thwart attacks intended to sabotage processor resources.

Key Management--

There are 3 authentication methods used with oakley

- ❑ **Digital signatures:** The exchange is authenticated by signing a mutually obtainable hash; each party encrypts the hash with its private key. The hash is generated over important parameters, such as user IDs and nonces.
- ❑ **Public-key encryption:** The exchange is authenticated by encrypting parameters such as IDs and nonces with the sender's private key.
- ❑ **Symmetric-key encryption:** A key derived by some out-of-band mechanism can be used to authenticate the exchange by symmetric encryption of exchange parameters.

Key Management—Oakley Key Exchange

Figure 16.11. Example of Aggressive Oakley Key Exchange

[\[View full size image\]](#)

$I \rightarrow R$: $CKY_I, OK_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, S_{KI}[ID_I \parallel ID_R \parallel N_I \parallel GRP \parallel g^x \parallel EHAO]$

$R \rightarrow I$: $CKY_R, CKY_I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, S_{KR}[ID_R \parallel ID_I \parallel N_R \parallel N_I \parallel GRP \parallel g^y \parallel g^x \parallel EHAS]$

$I \rightarrow R$: $CKY_I, CKY_R, OK_KEYX, GRP, g^x, EHAS, NIDP, ID_I, ID_R, N_I, N_R, S_{KI}[ID_I \parallel ID_R \parallel N_I \parallel N_R \parallel GRP \parallel g^x \parallel g^y \parallel EHAS]$

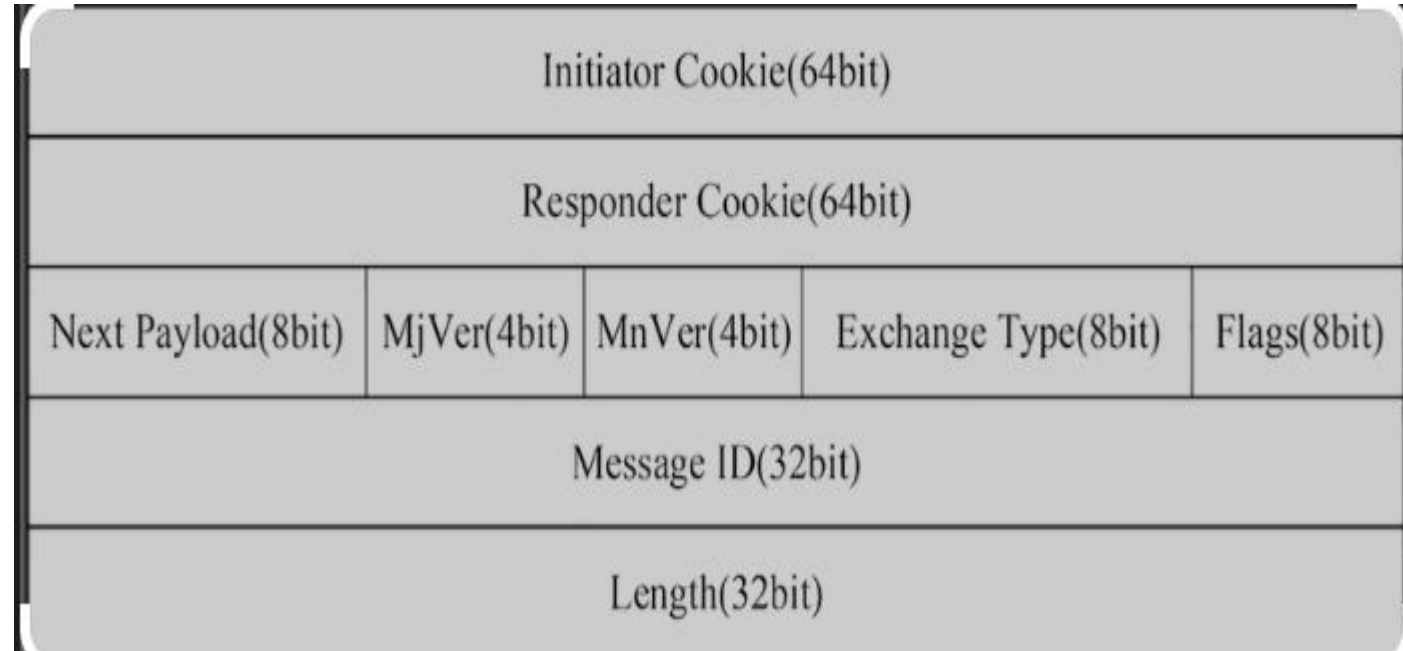
Notation:

I	=	Initiator
R	=	Responder
CKY_I, CKY_R	=	Initiator, responder cookies
OK_KEYX	=	Key exchange message type
GRP	=	Name of Diffie-Hellman group for this exchange
g^x, g^y	=	Public key of initiator, responder; g^{xy} = session key from this exchange
$EHAO, EHAS$	=	Encryption, hash authentication functions, offered and selected
$NIDP$	=	Indicates encryption is not used for remainder of this message
ID_I, ID_R	=	Identifier for initiator, responder
N_I, N_R	=	Random nonce supplied by initiator, responder for this exchange
$S_{KI}[X], S_{KR}[X]$	=	Indicates the signature over X using the private key (signing key) of initiator, responder

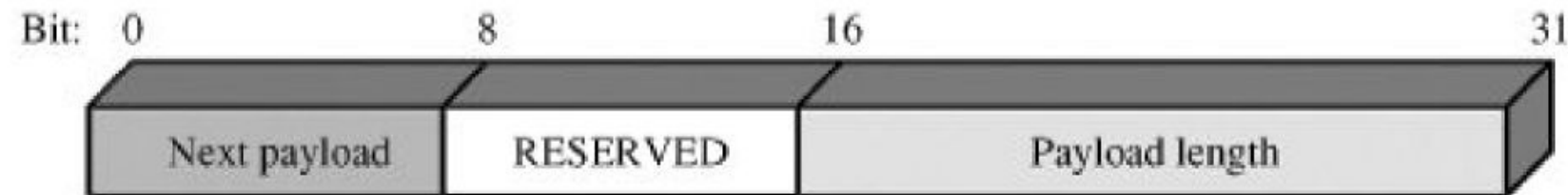
Key Management—ISAKMP

ISAKMP defines procedures and packet formats to establish, negotiate, modify, and delete security associations.

As part of SA establishment, ISAKMP defines payloads for exchanging key generation and authentication data.



ISAKMP Header



(b) Generic payload header

Key Management—ISAKMP Header Format

Initiator Cookie (64 bits): Cookie of entity that initiated SA establishment, SA notification, or SA deletion.

Responder Cookie (64 bits): Cookie of responding entity; null in first message from initiator.

Next Payload (8 bits): Indicates the type of the first payload in the message; payloads are discussed in the next subsection.

Major Version (4 bits): Indicates major version of ISAKMP in use.

Minor Version (4 bits): Indicates minor version in use.

Exchange Type (8 bits): Indicates the type of exchange; these are discussed later in this section.

Flags (8 bits): Indicates specific options set for this ISAKMP exchange. Two bits so far defined: The Encryption bit is set if all payloads following the header are encrypted using the encryption algorithm for this SA. The Commit bit is used to ensure that encrypted material is not received prior to completion of SA establishment.

Message ID (32 bits): Unique ID for this message.

Length (32 bits): Length of total message (header plus all payloads) in octets

ISAKMP Payload Types

The **Proposal payload** contains information used during SA negotiation. The payload indicates the protocol for this SA (ESP or AH) for which services and mechanisms are being negotiated.

The **Transform payload** defines a security transform to be used to secure the communications channel for the designated protocol.

The **Key Exchange payload** can be used for a variety of key exchange techniques, including Oakley, Diffie-Hellman, and the RSA-based key exchange used by PGP.

The **Identification payload** is used to determine the identity of communicating peers and may be used for determining authenticity of information.

The **Certificate payload** transfers a public-key certificate. The Certificate Encoding field indicates the type of certificate or certificate-related information: Ex: PGP certificate, DNS signed key, X.509 certificate signature, X.509 certificate key exchange, Kerberos tokens