MODULE V

- Network security: Electronic Mail Security: Pretty good privacy- S/MIME. IP Security: Architecture- authentication Header Encapsulating Security payload- Combining Security associations- Key management.

PRETTY GOOD PRIVACY(PGP)

- PGP was developed by Phil Zimmermann.
- provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

Notations in PGP

K_s	=	session key used in conventional encryption scheme
KR_a	=	private key of user A, used in public-key encryption scheme
KU_a	=	public key of user A, used in public-key encryption scheme
EP	=	public-key encryption
DP	=	public-key decryption
EC	=	conventional encryption
DC	=	conventional decryption
H	=	hash function
$\|\|$	=	concatenation
Z	=	compression using ZIP algorithm
R64	=	conversion to radix 64 ASCII format

operation of PGP consists of five services:

- Authentication
- Confidentiality
- Compression
- E-mail compatibility
- Segmentation

Summary of PGP Services

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.

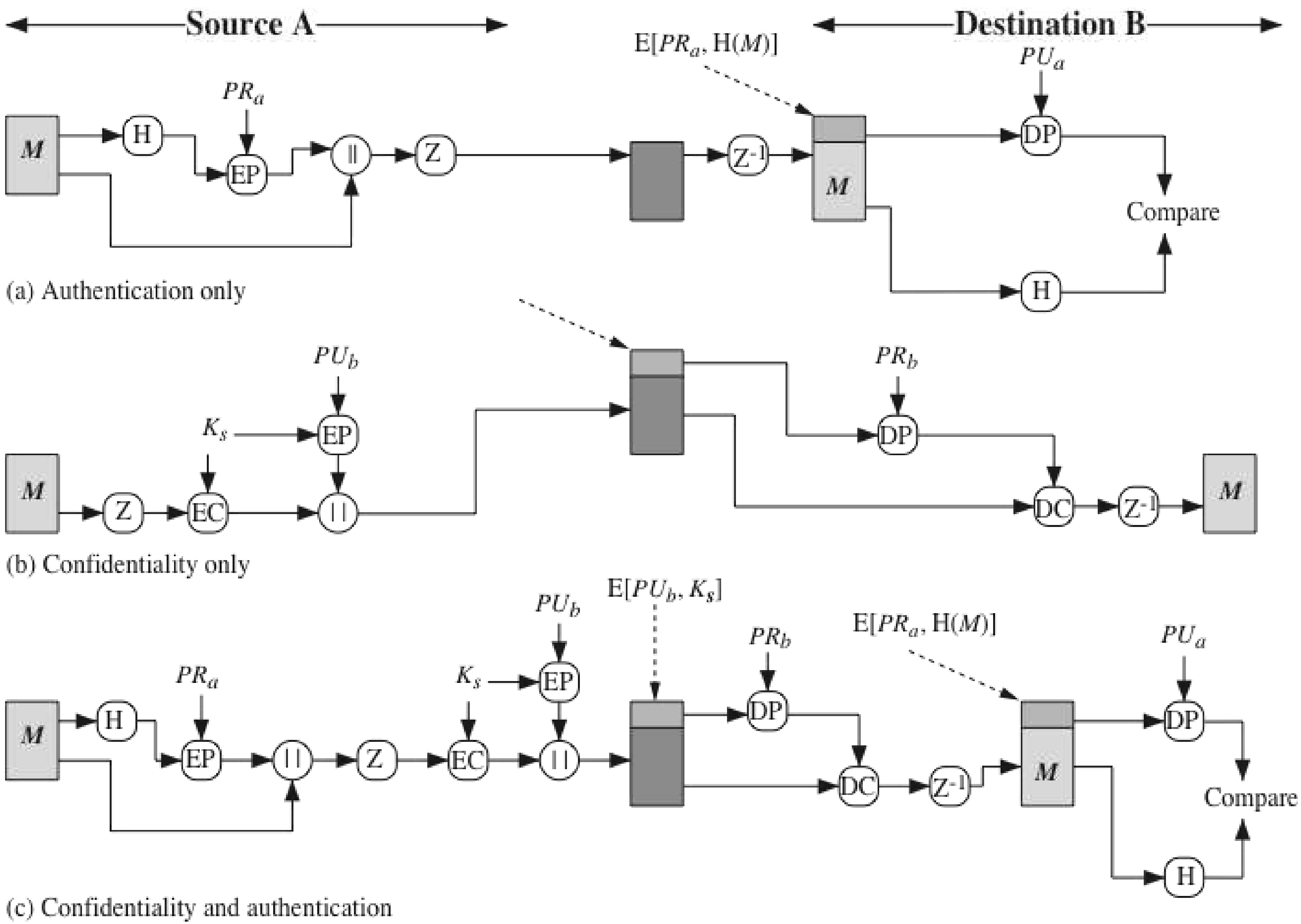
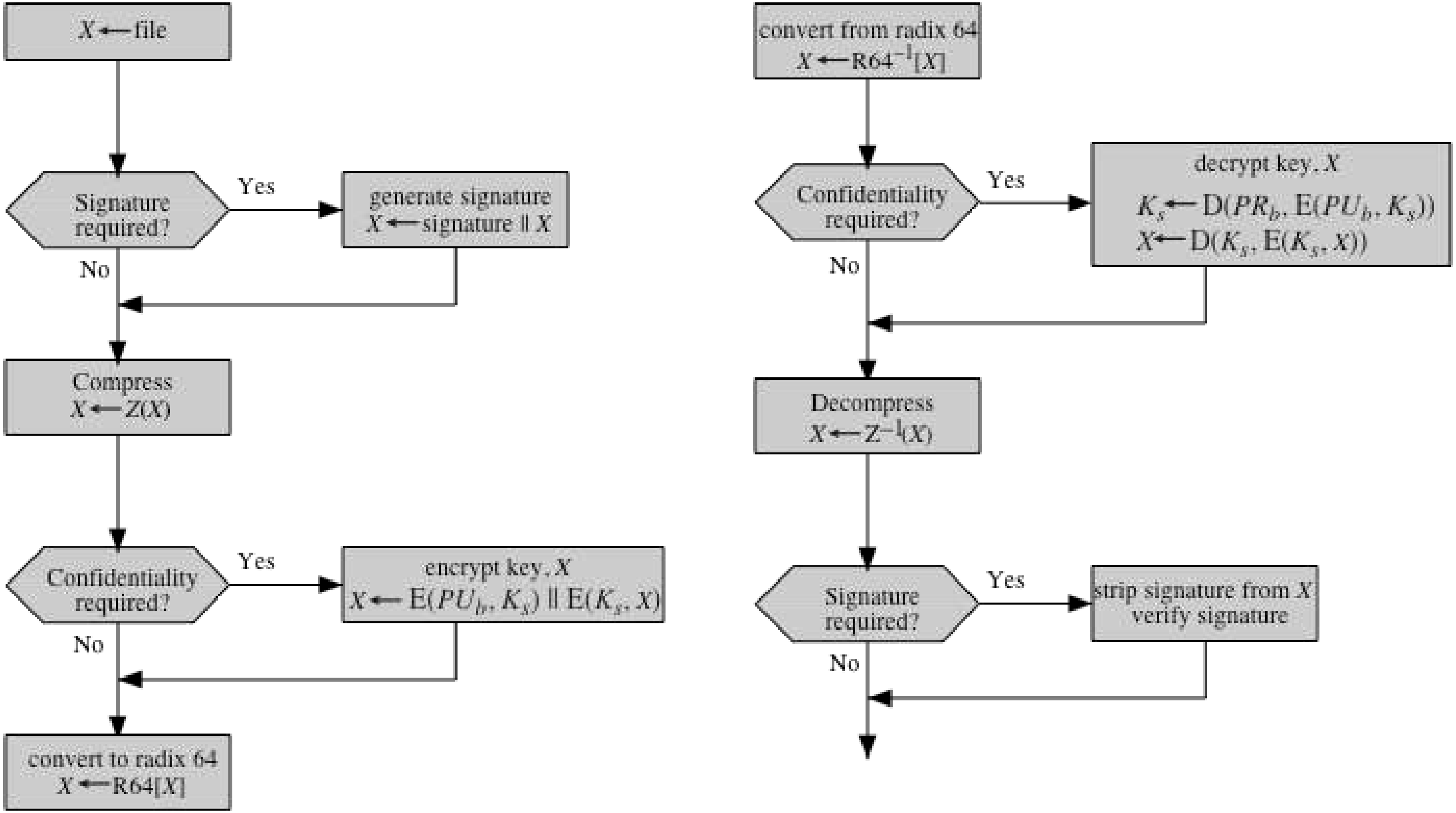


Figure 19.1 PGP Cryptographic Functions

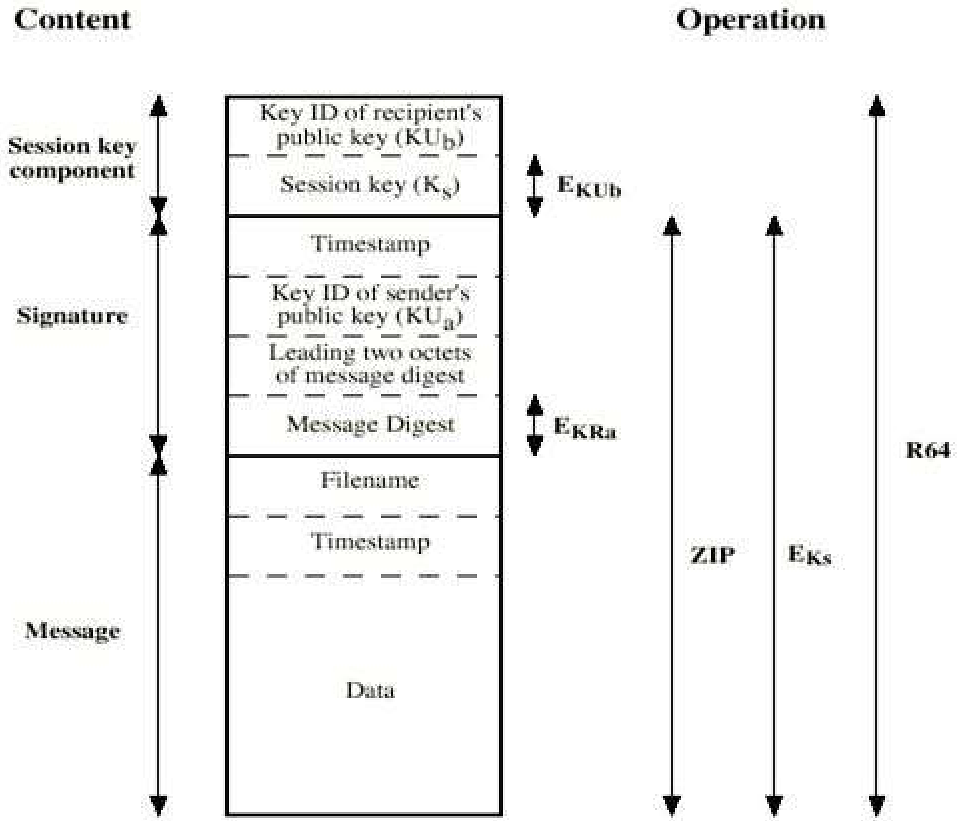


(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

Figure 19.2 Transmission and Reception of PGP Messages

General Format of PGP Message



Notation:

E_{KUb} = encryption with user b's public key
E_{KRa} = encryption with user a's private key
E_{Ks} = encryption with session key
ZIP = Zip compression function
R64 = Radix-64 conversion function

PGP Key ID concept

- since a user may have many public/private keys in use, there is a need to identify which is actually used to encrypt session key in a message
  - PGP uses a key identifier which is least significant 64-bits of the public key
- Key IDs are used in signatures too
- Key IDs are sent together with messages

PGP Key Rings

- each PGP user has a pair of key rings to store public and private keys
  - public-key ring contains all the public-keys of other PGP users known to this user

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

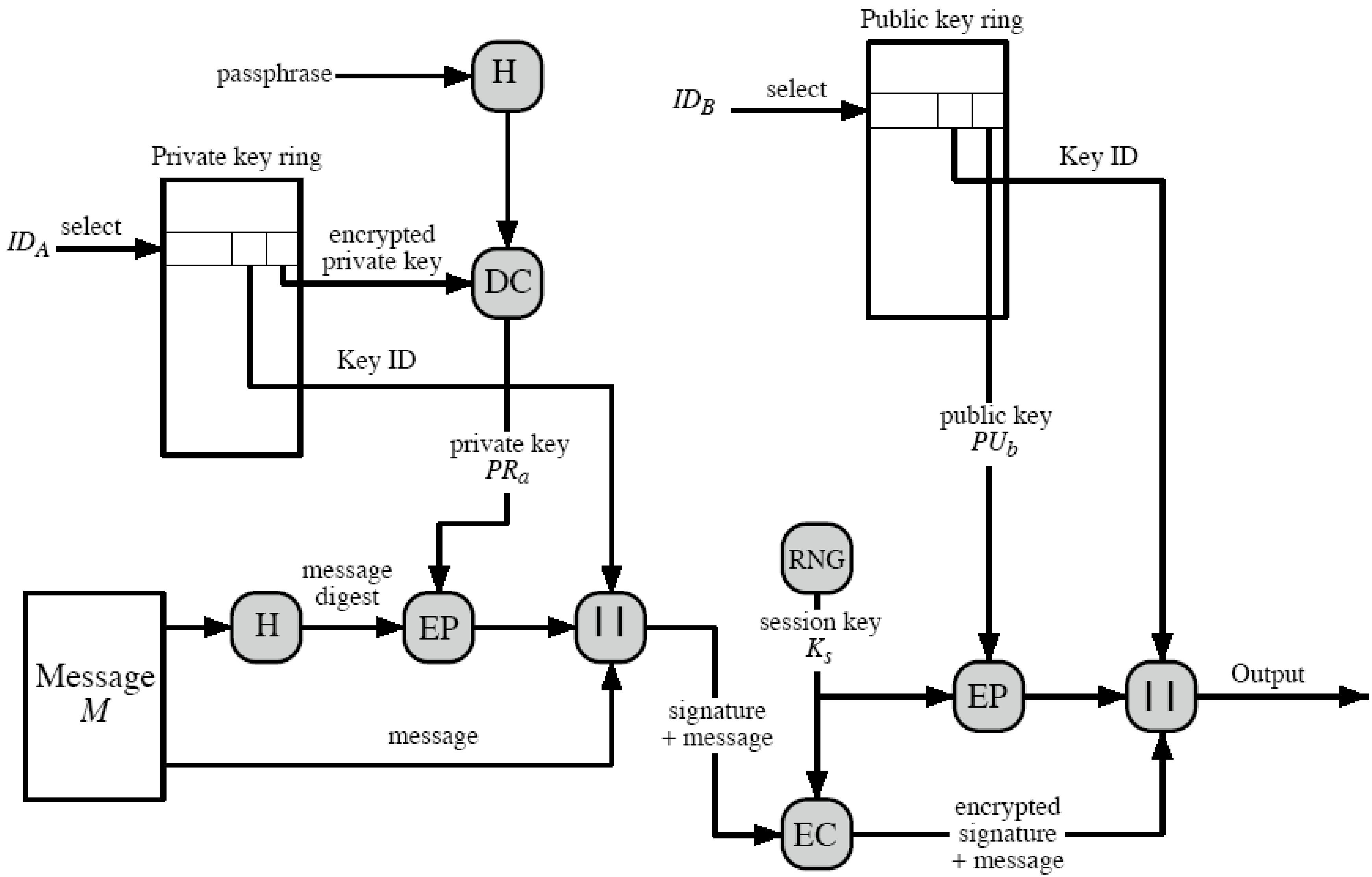
PGP Key Rings

- private-key ring contains the public/private key pair(s) for this user,
- private keys are encrypted using a key derived from a hashed passphrase

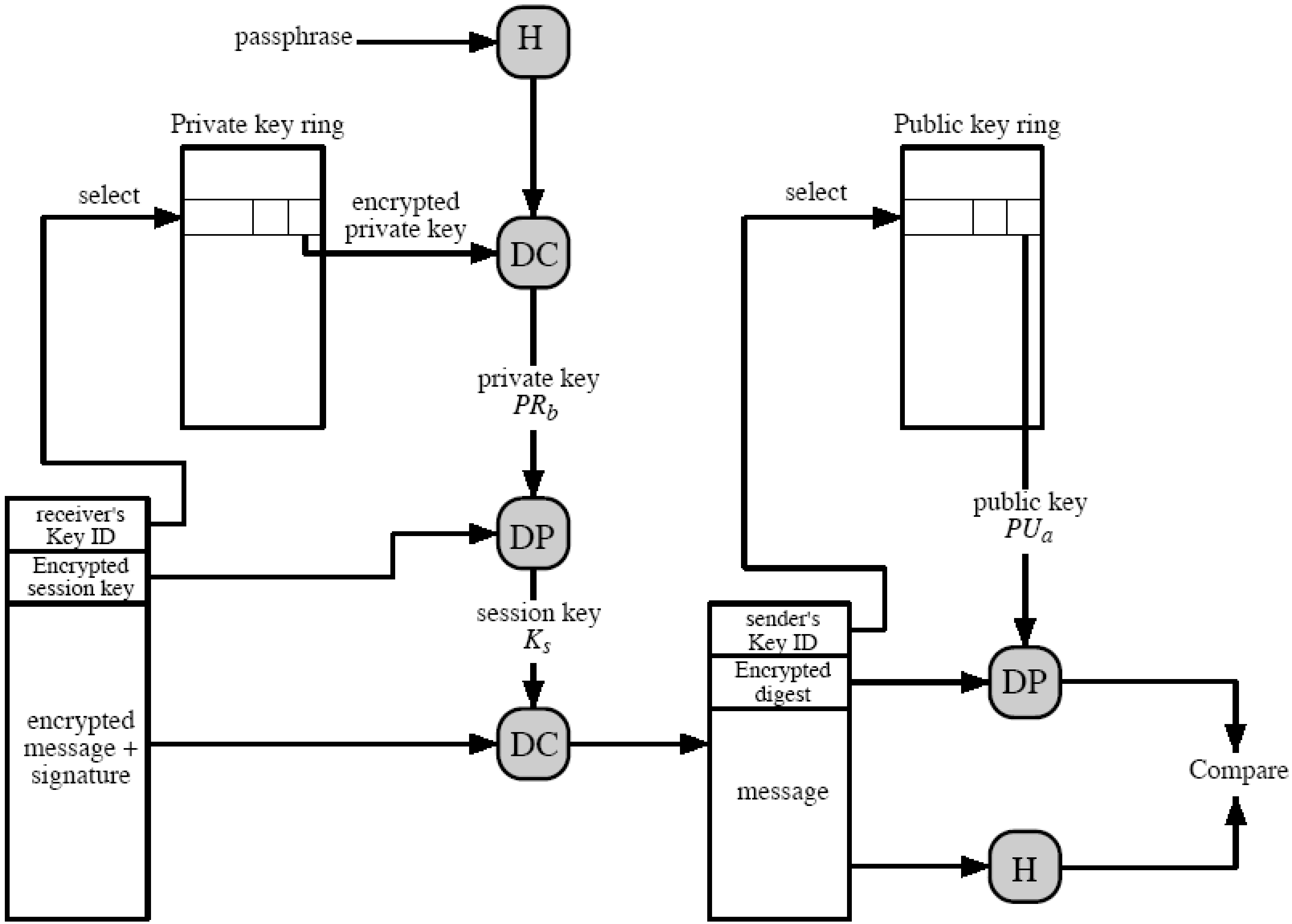
Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Key rings and message generation

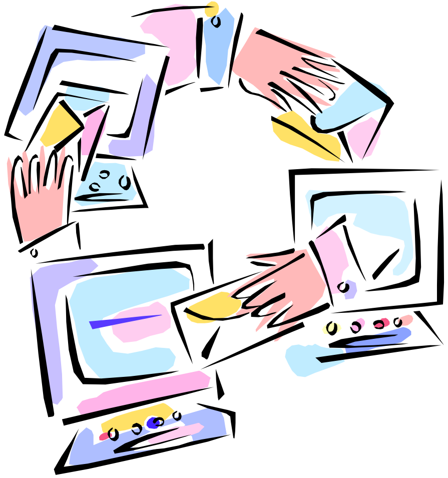


Key rings and message reception



Secure/Multipurpose Internet Mail Extension (S/MIME)

- A security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security
- Defined in:
  - RFCs 3370, 3850, 3851, 3852



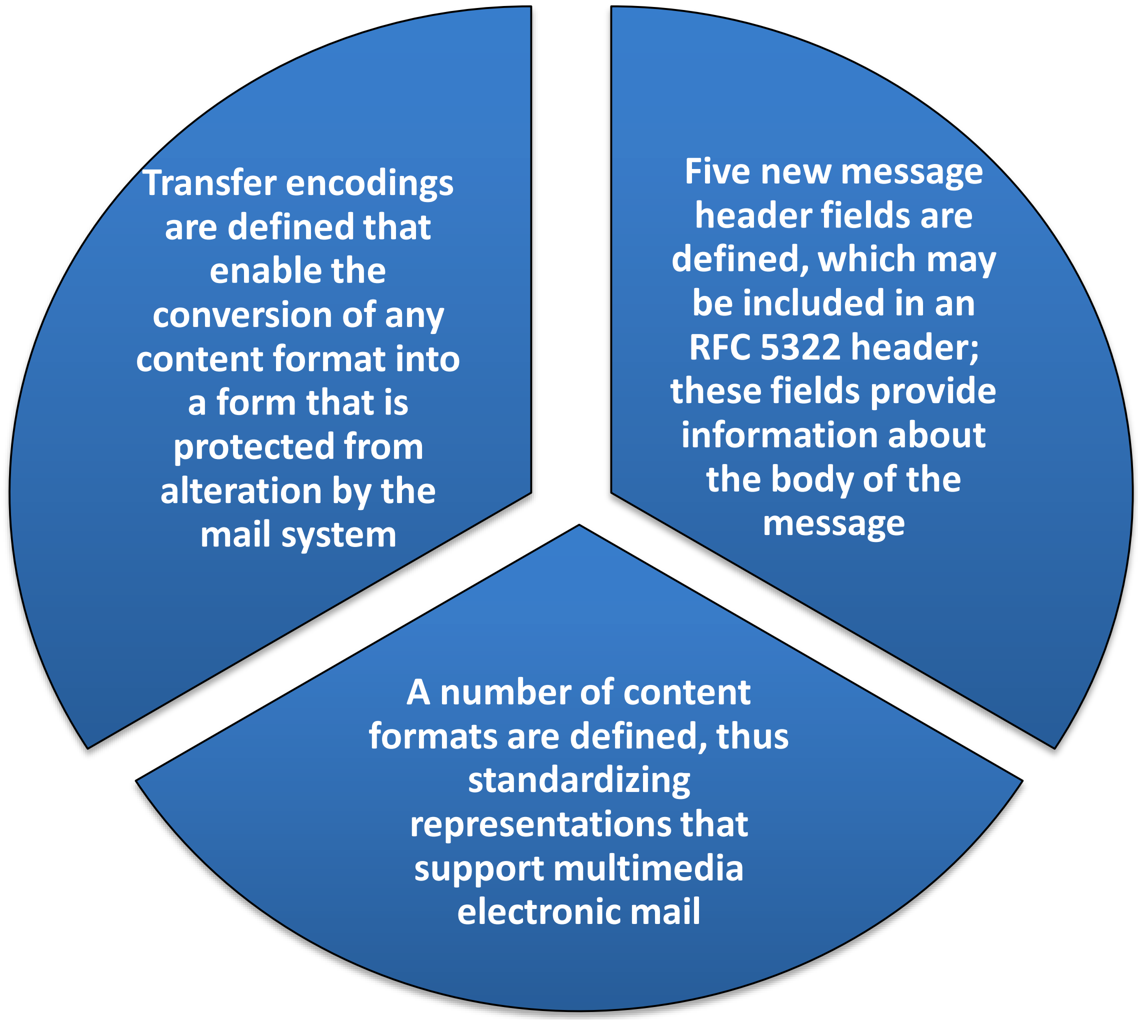
RFC 5322

- Defines a format for text messages that are sent using electronic mail
- Messages are viewed as having an envelope and contents
  - The envelope contains whatever information is needed to accomplish transmission and delivery
  - The contents compose the object to be delivered to the recipient
  - RFC 5322 standard applies only to the contents
- The content standard includes a set of header fields that may be used by the mail system to create the envelope

Multipurpose Internet Mail Extensions (MIME)

- 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)
  - Is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations
  - The specification is provided in RFCs 2045 through 2049

MIME specification includes the following elements:



The Five Header Fields Defined 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

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.

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

S/MIME

Enveloped data

- 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

- Only the digital signature is encoded using base64
- As a result recipients without S/MIME capability can view the message content, although they cannot verify the signature

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

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.

Canonical Form

- Important concept in MIME & S/MIME
- Is a format , appropriate to the content type , that is standardized for use between systems
- Contrast to native form, which is a format that may be peculiar to a particular system

Table 19.6
S/MIME Content Types

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	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-mime	CompressedData	A compressed S/MIME entity.
	pkcs7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

Securing a MIME Entity

- S/MIME secures a MIME entity with a signature, encryption, or both
- The MIME entity is prepared according to the normal rules for MIME message preparation
  - The MIME entity plus some security-related data, such as algorithm identifiers and certificates, are processed by S/MIME to produce what is known as a PKCS object
  - A PKCS object is then treated as message content and wrapped in MIME
- In all cases the message to be sent is converted to canonical form

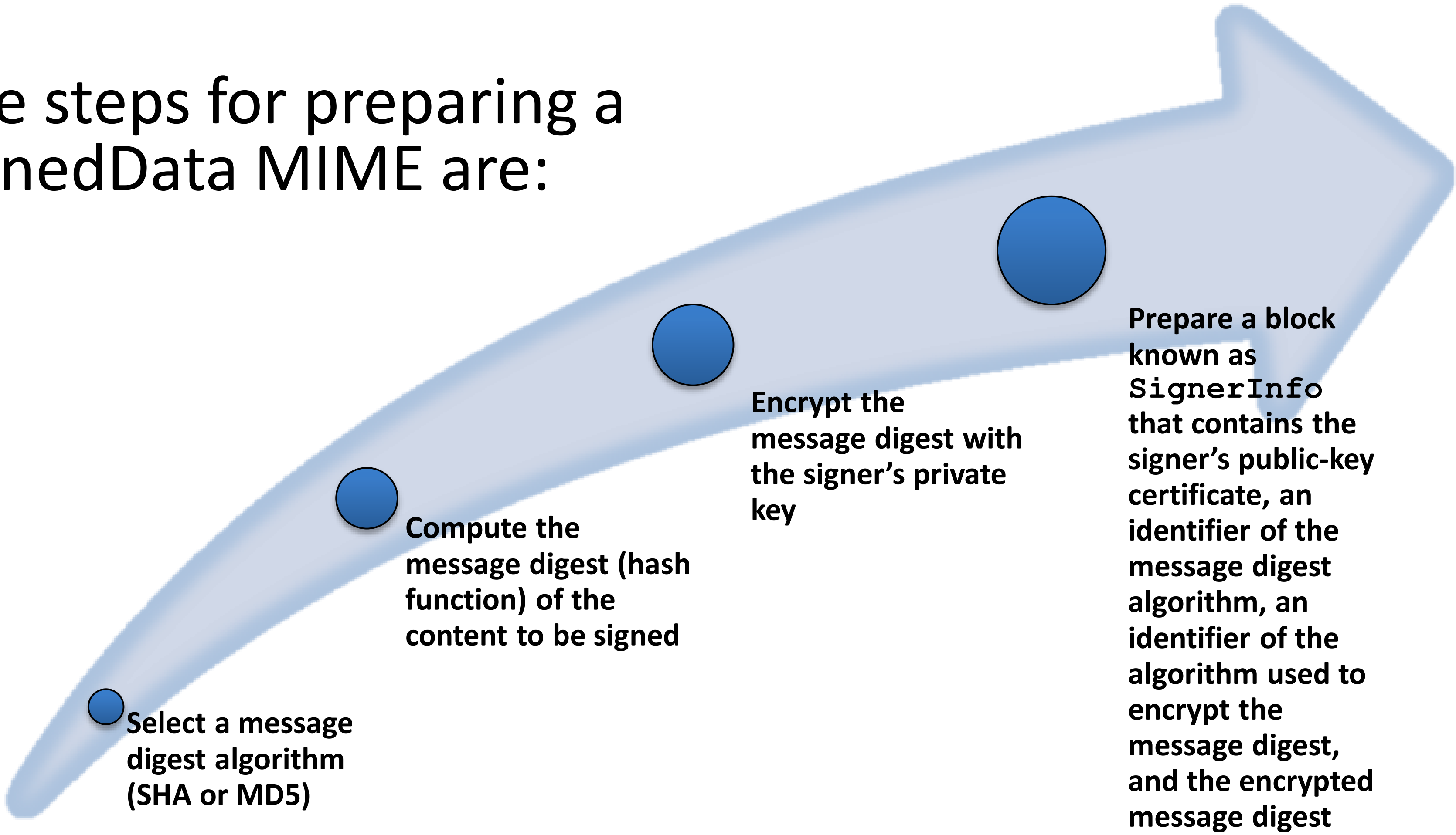
EnvelopedData

- The steps for preparing an envelopedData MIME are:

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm
2. For each recipient, encrypt the session key with the recipient's public RSA key
3. For each recipient, prepare a block known as `RecipientInfo` that contains an identifier of the recipient's public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key
4. Encrypt the message content with the session key

SignedData

- The steps for preparing a signedData MIME are:



Clear Signing

- Achieved using the multipart content type with a signed subtype
- This signing process does not involve transforming the message to be signed
- Recipients with MIME capability but not S/MIME capability are able to read the incoming message

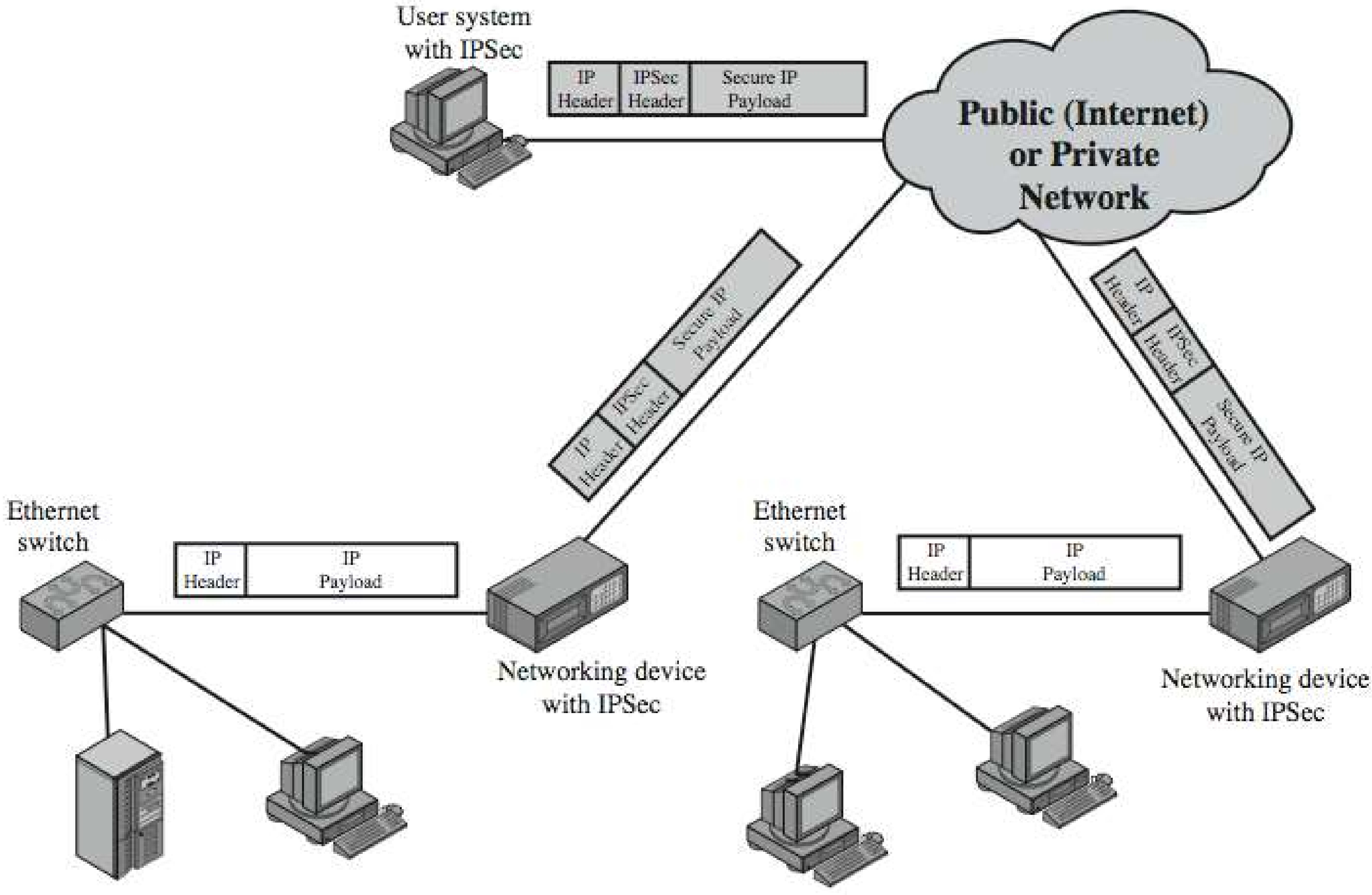
IP Security

- have a range of application specific security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- however there are security concerns that cut across protocol layers
- would like security implemented by the network for all applications

IP Security

- general IP Security mechanisms
- provides
  - authentication
  - confidentiality
  - key management
- applicable to use over LANs, across public & private WANs, & for the Internet
- need identified in 1994 report
  - need authentication, encryption in IPv4 & IPv6

IP Security Uses



Benefits of IPSec

- in a firewall/router provides strong security to all traffic crossing the perimeter
- in a firewall/router is resistant to bypass
- is below transport layer, hence transparent to applications
- can be transparent to end users
- can provide security for individual users
- secures routing architecture

IP Security Architecture

- specification is quite complex, with groups:
  - Architecture
    - RFC4301 *Security Architecture for Internet Protocol*
  - Authentication Header (AH)
    - RFC4302 *IP Authentication Header*
  - Encapsulating Security Payload (ESP)
    - RFC4303 *IP Encapsulating Security Payload (ESP)*
  - Internet Key Exchange (IKE)
    - RFC4306 *Internet Key Exchange (IKEv2) Protocol*
  - Cryptographic algorithms
  - Other

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

Security Associations

- a one-way relationship between sender & receiver that affords security for traffic flow
- defined by 3 parameters:
  - Security Parameters Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier
- has a number of other parameters
  - seq no, AH & EH info, lifetime etc
- have a database of Security Associations

SA Parameters

- Sequence Number Counter
- Sequence Counter Overflow
- Anti-Replay Window
- AH Information
- ESP Information
- Lifetime of this SA
- IPSec Protocol Mode
- Path MTU

Security Policy Database (SPD)

- relates IP traffic to specific SAs
  - match subset of IP traffic to relevant SA
  - based on local & remote IP addresses, next layer protocol, name, local & remote ports

SA Selectors

- Each SPD entry is defined by a set of IP and upper –layer protocol field values
- use selectors to filter outgoing traffic to map

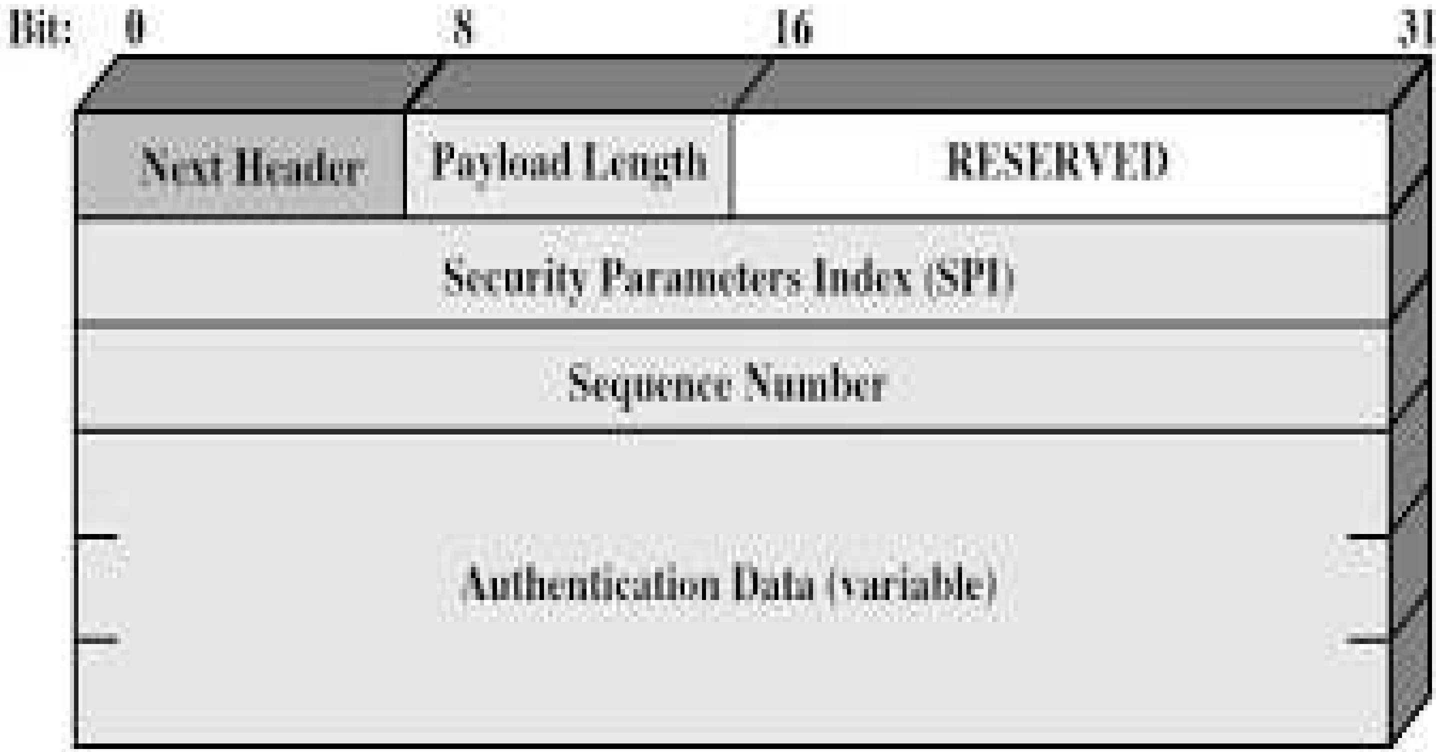
Transport and Tunnel Modes

- Transport Mode
  - to encrypt & optionally authenticate IP data
  - Provides protection primarily for upper layer protocols
  - can do traffic analysis but is efficient
  - good for ESP host to host traffic
- Tunnel Mode
  - encrypts entire IP packet
  - add new header for next hop
  - no routers on way can examine inner IP header
  - good for VPNs, gateway to gateway security

Authentication header

- Support for data integrity & authentication of IP packets
- Authentication – MAC code
- Data integrity – undetected modification not possible

IPSec Authentication Header



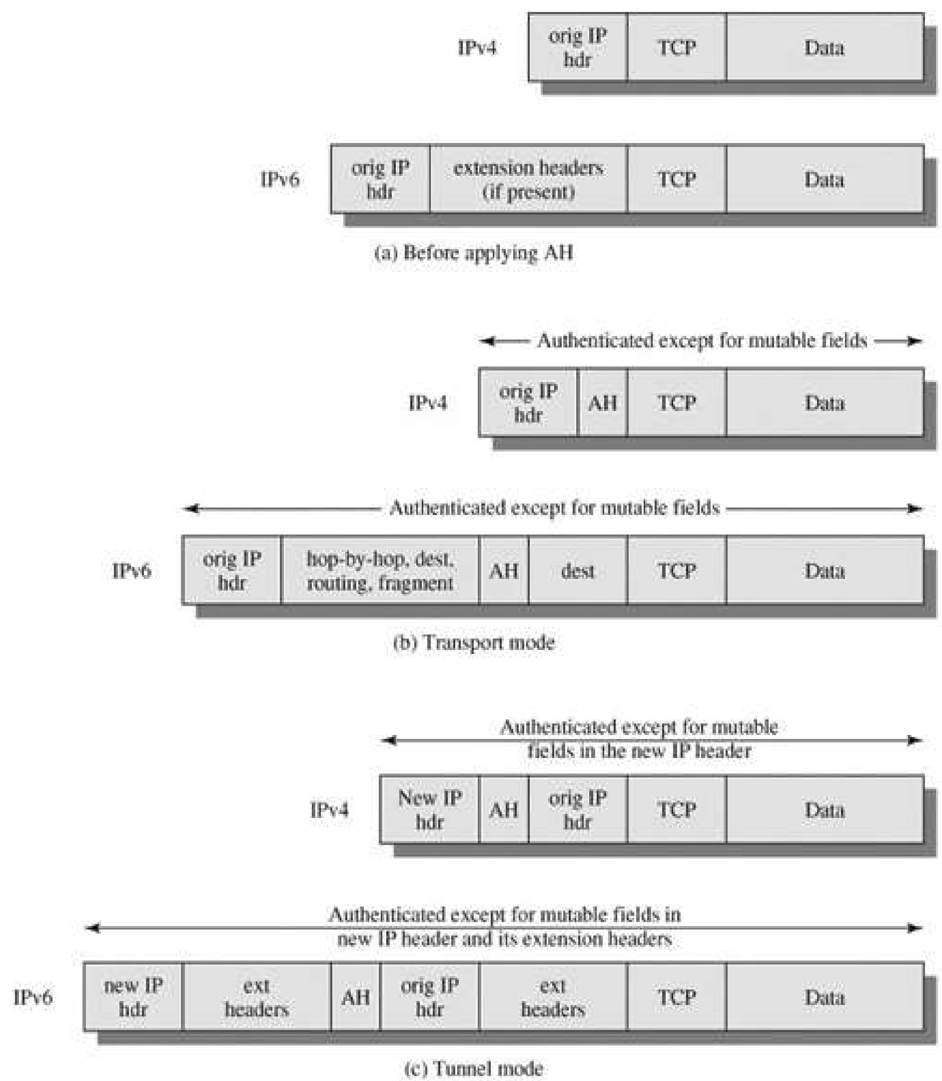
Anti-Replay Service

- replay is when attacker resends a copy of an authenticated packet
- use sequence number to thwart this attack
- sender initializes sequence number to 0 when a new SA is established
  - increment for each packet
  - must not exceed limit of $2^{32} - 1$
- receiver then accepts packets with seq no within window of ($N - W + 1$)
- W- window size
- N- sequence number

Integrity Check Value

- Authentication data holds a value referred to as ICV
- Is a message authentication code or a truncated version of a code produced by MAC algorithm.

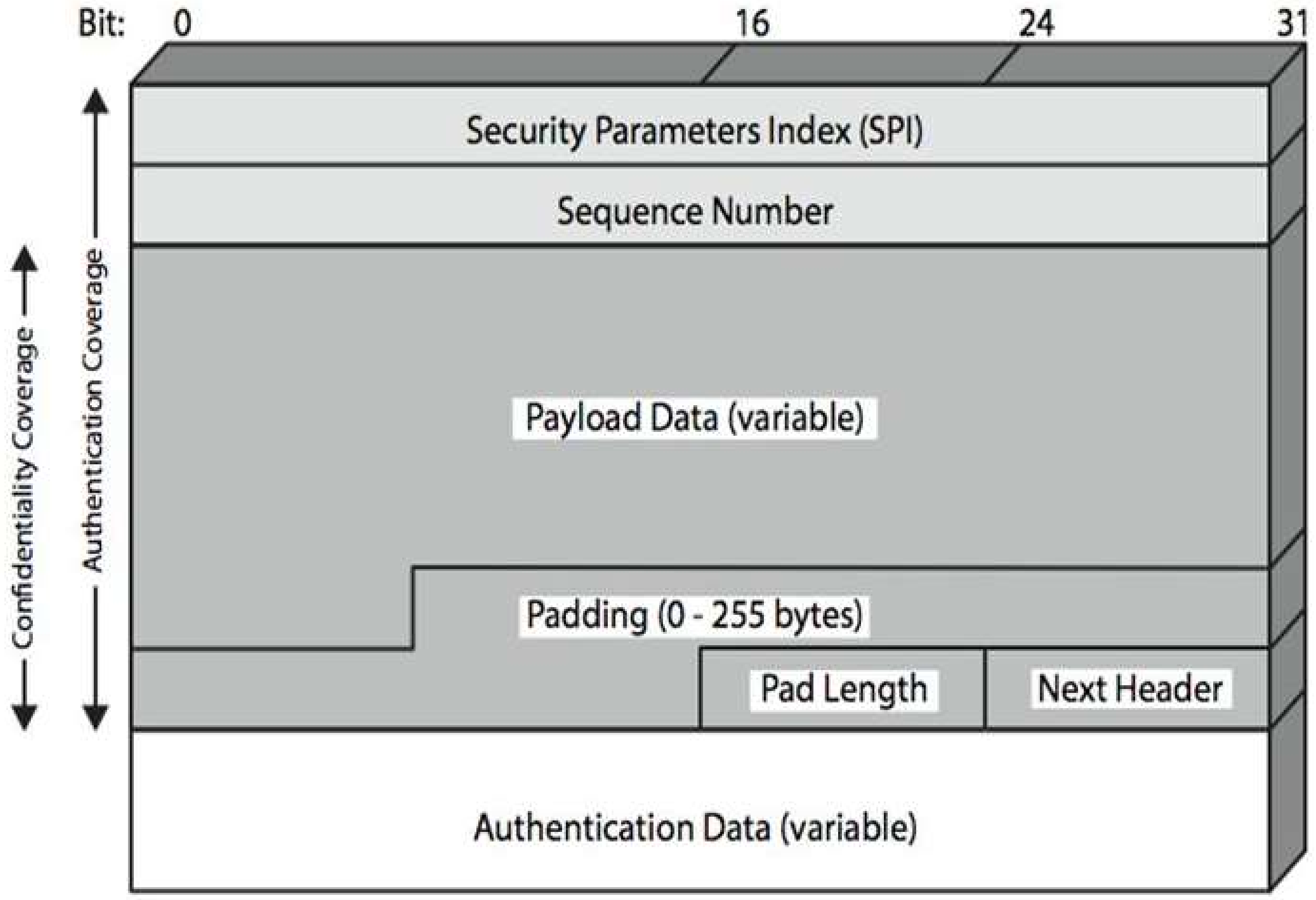
Scope of AH authentication



Encapsulating Security Payload (ESP)

- provides message content confidentiality, data origin authentication, connectionless integrity, an anti-replay service, limited traffic flow confidentiality
- services depend on options selected when establish Security Association (SA), net location
- can use a variety of encryption & authentication algorithms

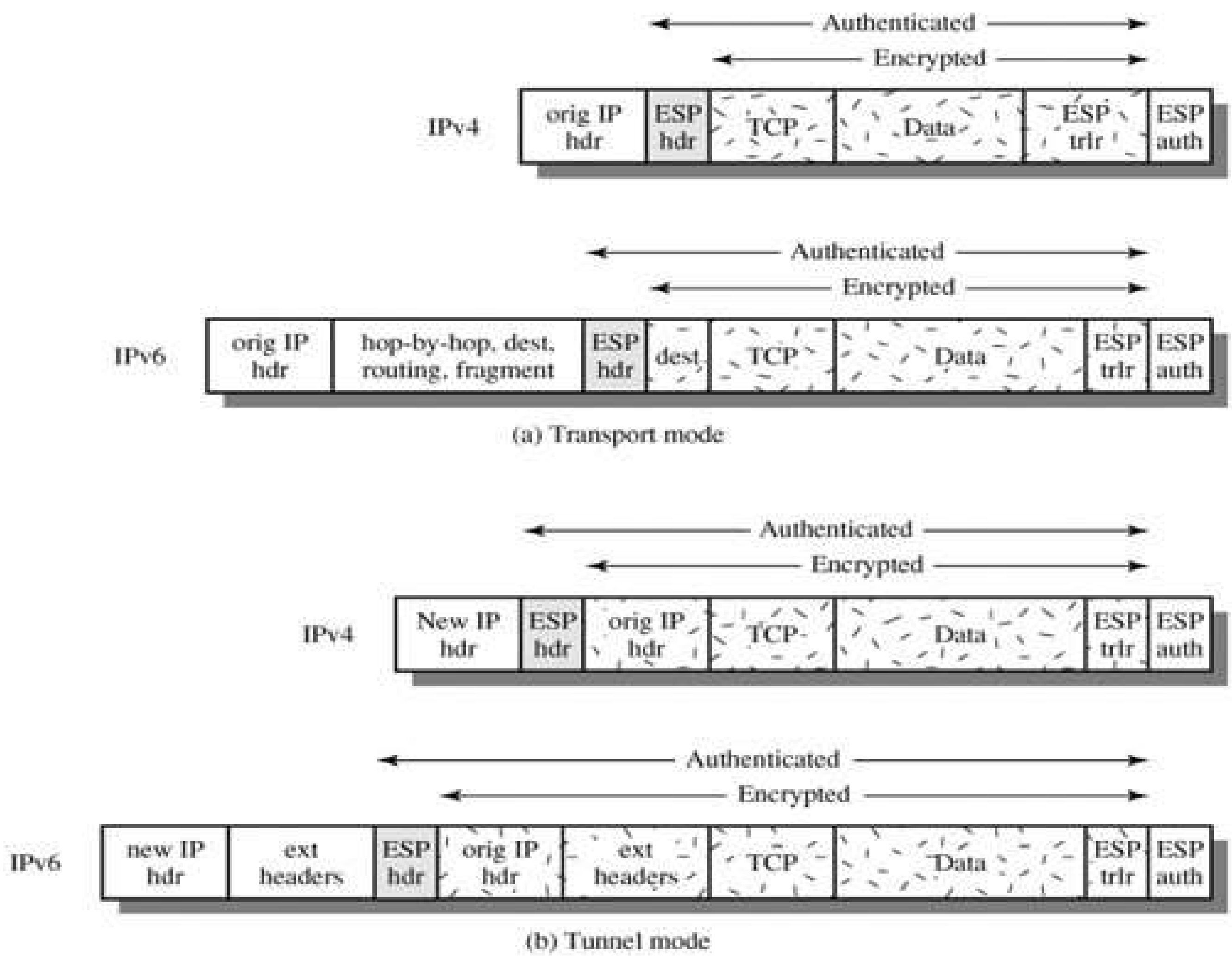
IPsec ESP format



Encryption & Authentication Algorithms & Padding

- ESP can encrypt payload data, padding, pad length, and next header fields
  - if needed have IV at start of payload data
- ESP can have optional ICV (integrity check value) for integrity
  - is computed after encryption is performed
- ESP uses padding
  - to expand plaintext to required length
  - to align pad length and next header fields
  - to provide partial traffic flow confidentiality

scope of ESP encryption & authentication



Combining Security Associations

- SA's can implement either AH or ESP
- to implement both need to combine SA's
  - form a security association bundle
  - may terminate at different or same endpoints
- combining authentication & encryption
  - ESP with authentication, bundled inner ESP & outer AH, bundled inner transport & outer ESP

Combining Security Associations

- An individual SA can implement either the AH or ESP protocol but not both
- *Security association bundle*
  - Refers to a sequence of SAs through which traffic must be processed to provide a desired set of IPsec services
  - The SAs in a bundle may terminate at different endpoints or at the same endpoint
- 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
- This approach allows for only one level of combination

Iterated tunneling

- Refers to the application of multiple layers of security protocols effected through IP tunneling
- This approach allows for multiple levels of nesting

Combining security associations

- **basic ESP-AH combination**
  1. **apply ESP in transport mode without authentication**
  2. **apply AH in transport mode**

original IP header	AH	ESP header	TCP/UDP header	data	ESP trailer

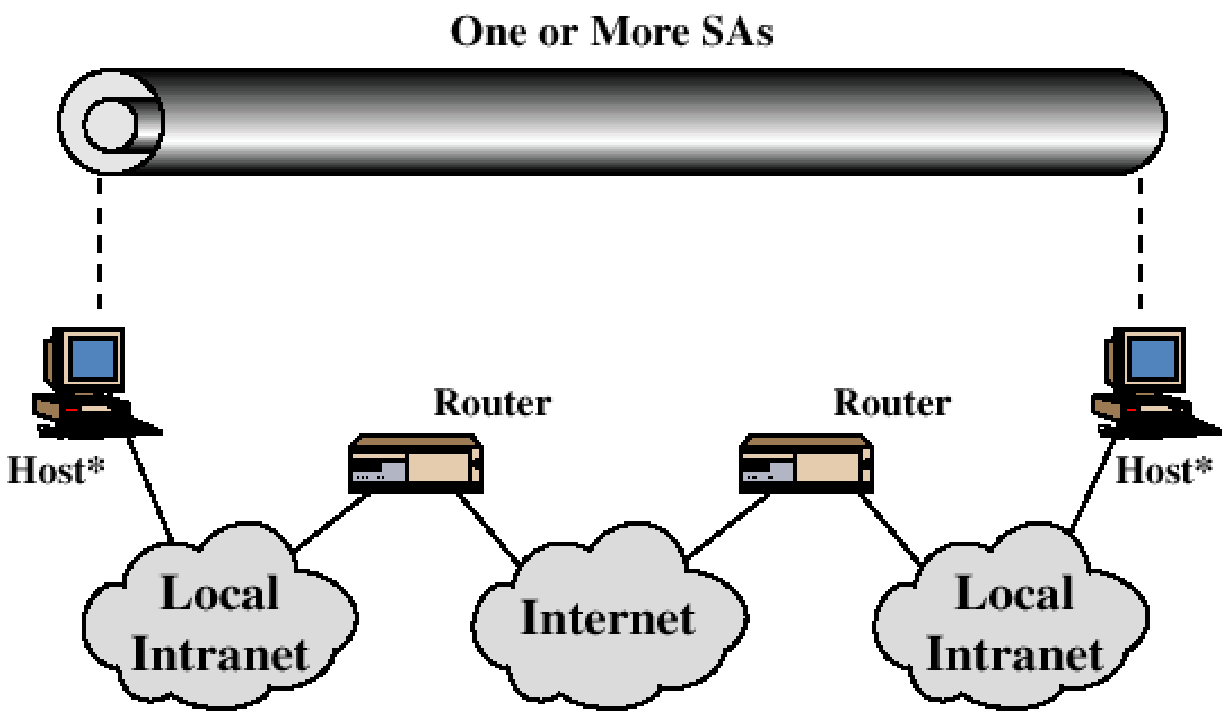
authenticated except for mutable fields in the IP header

- **basic AH-ESP combination**
  1. **apply AH in transport mode**
  2. **apply ESP in tunnel mode without authentication**

new IP header	ESP header	original IP header	AH	TCP/UDP header	data	ESP trailer

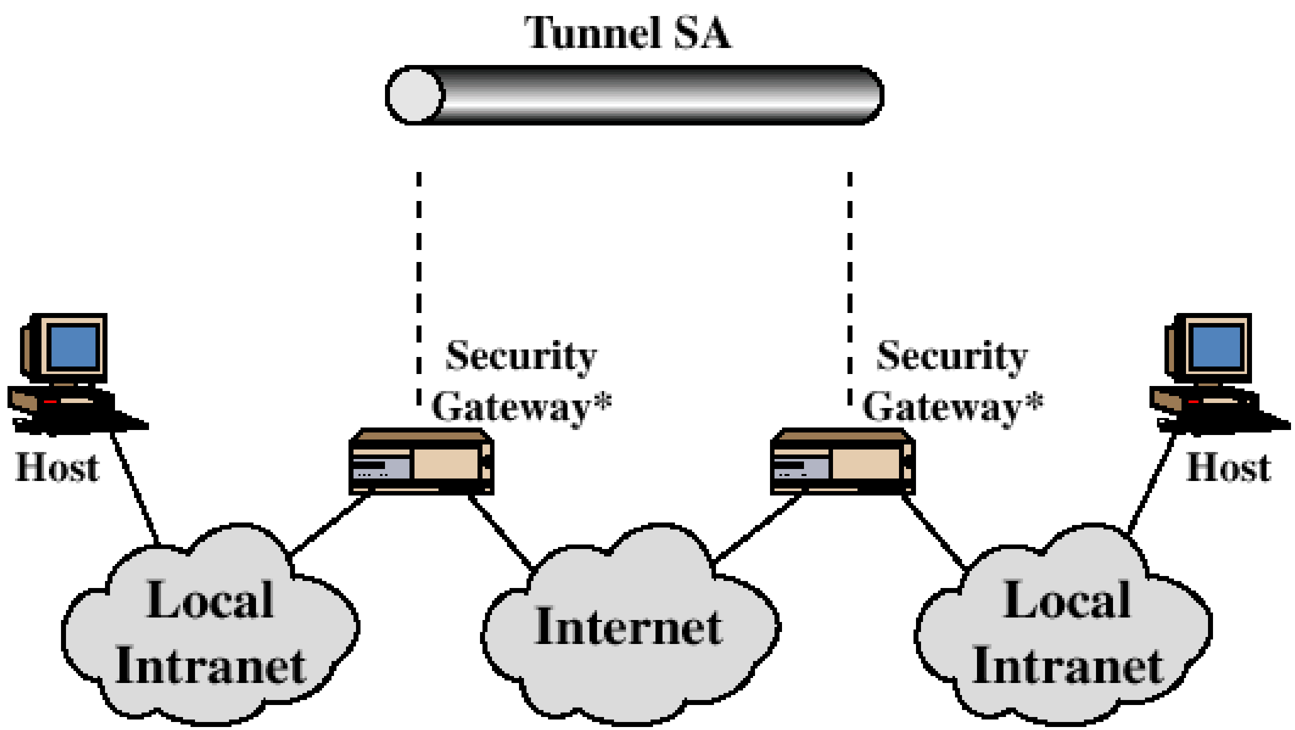
authenticated except for mutable fields in the inner IP header

Combinations of Security Associations



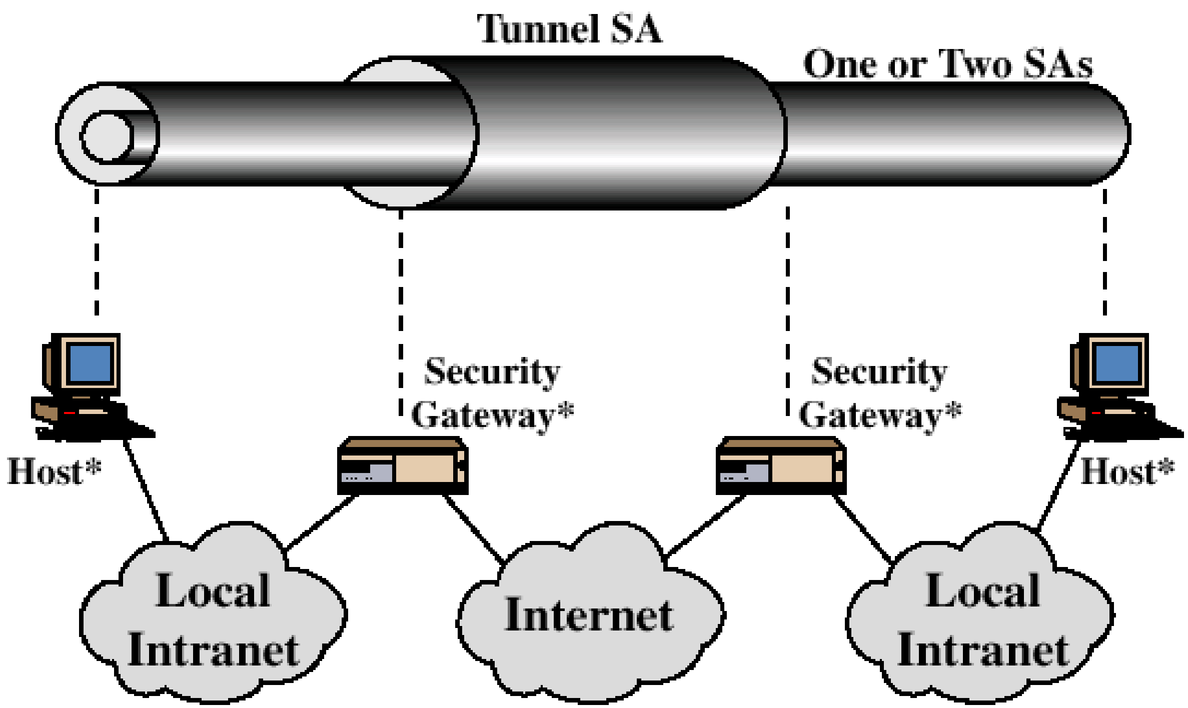
(a) Case 1

Combinations of Security Associations



(b) Case 2

Combinations of Security Associations



(c) Case 3

IPSec Key Management

- handles key generation & distribution
- typically need 2 pairs of keys
  - 2 for AH & ESP
- manual key management
  - System admin manually configures every system
- automated key management
  - automated system for on demand creation of keys for SA’s in large systems
  - has Oakley & ISAKMP elements

Oakley

- a key exchange protocol
- based on Diffie-Hellman key exchange
- adds features to address weaknesses
  - no info on parties, man-in-middle attack, cost
  - so adds cookies, groups (global params), nonces, DH key exchange with authentication

Oakley

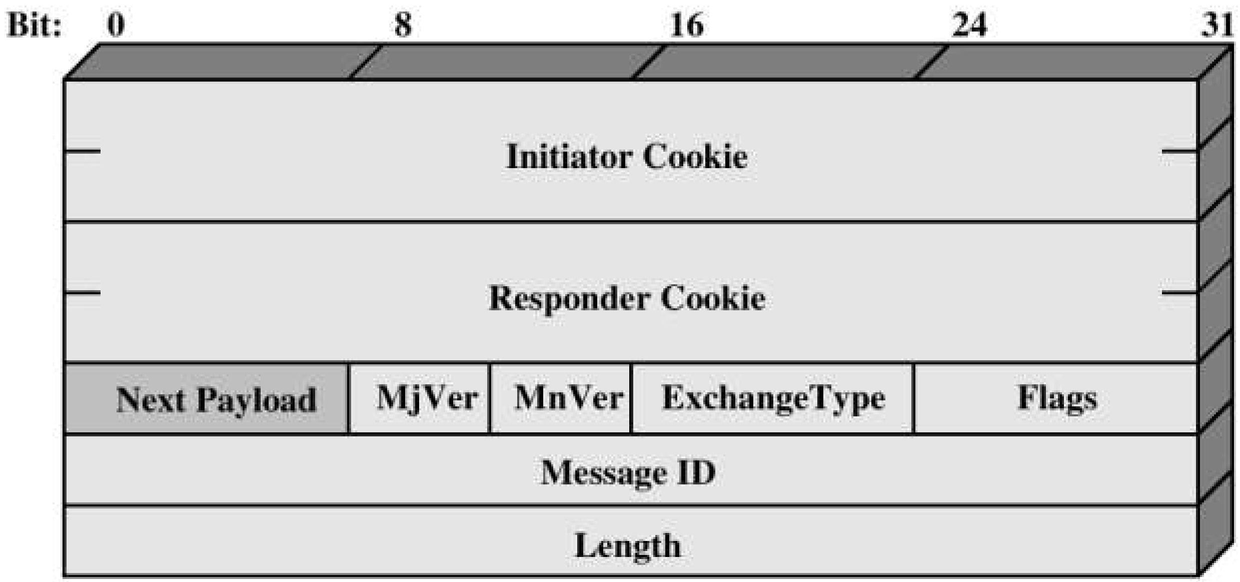
- Three authentication methods:
  - Digital signatures
  - Public-key encryption
  - Symmetric-key encryption

ISAKMP

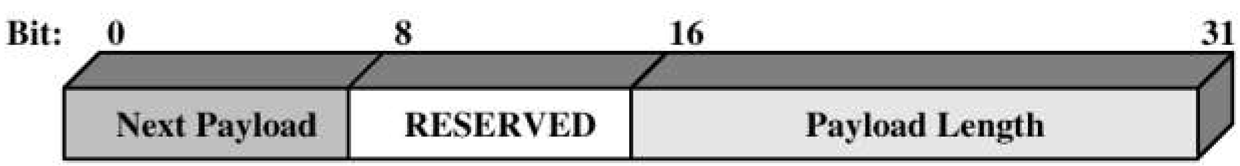
Internet Security Association and Key Management Protocol

- Provides framework for key management
- Defines procedures and packet formats to establish, negotiate, modify, & delete SAs
- Independent of key exchange protocol, encryption alg., & authentication method
- **Phase 1:** ISAKMP peers establish bi-directional secure channel using *main mode* or *aggressive mode*
- **Phase 2:** negotiation of security services for IPSec (maybe for several SAs) using *quick mode;* can have multiple Phase 2 exchanges, e.g., to change keys

ISAKMP



(a) ISAKMP Header



(b) Generic Payload Header

Figure 6.12 ISAKMP Formats

ISAKMP Payload Types

Type	Parameters	Description
Security Association (SA)	Domain of Interpretation, Situation	Used to negotiate security attributes and indicate the DOI and Situation under which negotiation is taking place.
Proposal (P)	Proposal #, Protocol-ID, SPI Size, # of Transforms, SPI	Used during SA negotiation; indicates protocol to be used and number of transforms.
Transform (T)	Transform #, Transform-ID, SA Attributes	Used during SA negotiation; indicates transform and related SA attributes.
Key Exchange (KE)	Key Exchange Data	Supports a variety of key exchange techniques.
Identification (ID)	ID Type, ID Data	Used to exchange identification information.
Certificate (CERT)	Cert Encoding, Certificate Data	Used to transport certificates and other certificate-related information.
Certificate Request (CR)	# Cert Types, Certificate Types, # Cert Auths, Certificate Authorities	Used to request certificates; indicates the types of certificates requested and the acceptable certificate authorities.
Hash (HASH)	Hash Data	Contains data generated by a hash function.
Signature (SIG)	Signature Data	Contains data generated by a digital signa wture function.
Nonce (NONCE)	Nonce Data	Contains a nonce.
Notification (N)	DOI, Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data	Used to transmit notification data, such as an error condition.
Delete (D)	DOI, Protocol-ID, SPI Size, # of SPIs, SPI (one or more)	Indicates an SA that is no longer valid.

ISAKMP Exchange Types

Exchange	Note
(a) Base Exchange	
(1) **I $\rightarrow$ R:** SA; NONCE	Begin ISAKMP-SA negotiation
(2) **R $\rightarrow$ I:** SA; NONCE	Basic SA agreed upon
(3) **I $\rightarrow$ R:** KE; ID_I; AUTH	Key generated; Initiator identity verified by responder
(4) **R $\rightarrow$ I:** KE; ID_R; AUTH	Responder identity verified by initiator; Key generated; SA established
(b) Identity Protection Exchange	
(1) **I $\rightarrow$ R:** SA	Begin ISAKMP-SA negotiation
(2) **R $\rightarrow$ I:** SA	Basic SA agreed upon
(3) **I $\rightarrow$ R:** KE; NONCE	Key generated
(4) **R $\rightarrow$ I:** KE; NONCE	Key generated
(5)* **I $\rightarrow$ R:** ID_I; AUTH	Initiator identity verified by responder
(6)* **R $\rightarrow$ I:** ID_R; AUTH	Responder identity verified by initiator; SA established
(c) Authentication Only Exchange	
(1) **I $\rightarrow$ R:** SA; NONCE	Begin ISAKMP-SA negotiation
(2) **R $\rightarrow$ I:** SA; NONCE; ID_R; AUTH	Basic SA agreed upon; Responder identity verified by initiator
(3) **I $\rightarrow$ R:** ID_I; AUTH	Initiator identity verified by responder; SA established
(d) Aggressive Exchange	
(1) **I $\rightarrow$ R:** SA; KE; NONCE; ID_I	Begin ISAKMP-SA negotiation and key exchange
(2) **R $\rightarrow$ I:** SA; KE; NONCE; ID_R; AUTH	Initiator identity verified by responder; Key generated; Basic SA agreed upon
(3)* **I $\rightarrow$ R:** AUTH	Responder identity verified by initiator; SA established
(e) Informational Exchange	
(1)* **I $\rightarrow$ R:** N/D	Error or status notification, or deletion

Notation:
- I = initiator
- R = responder
- * = signifies payload encryption after the ISAKMP header