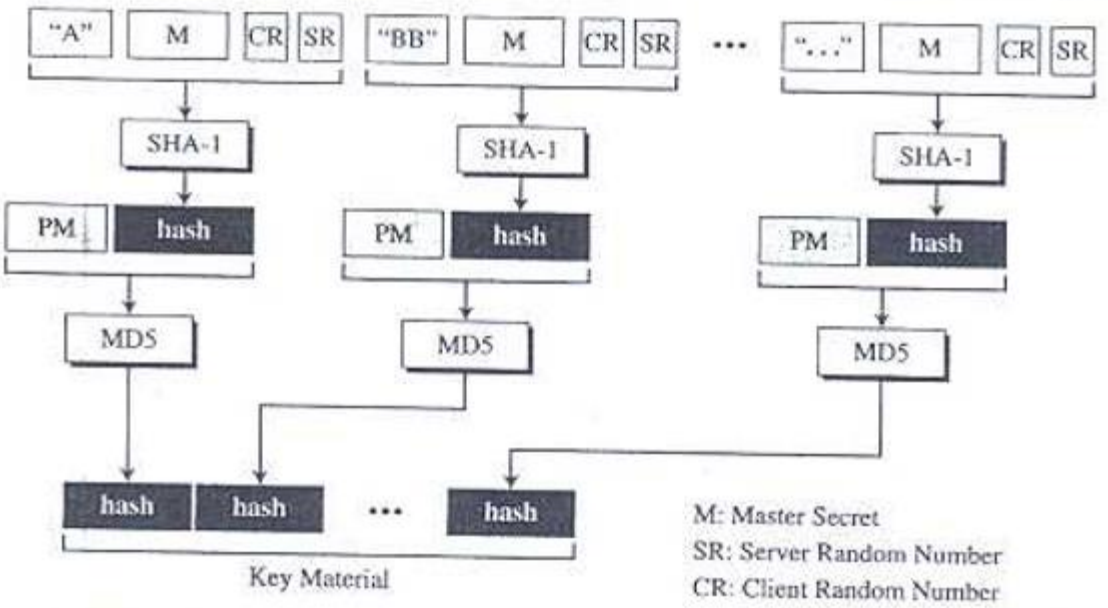
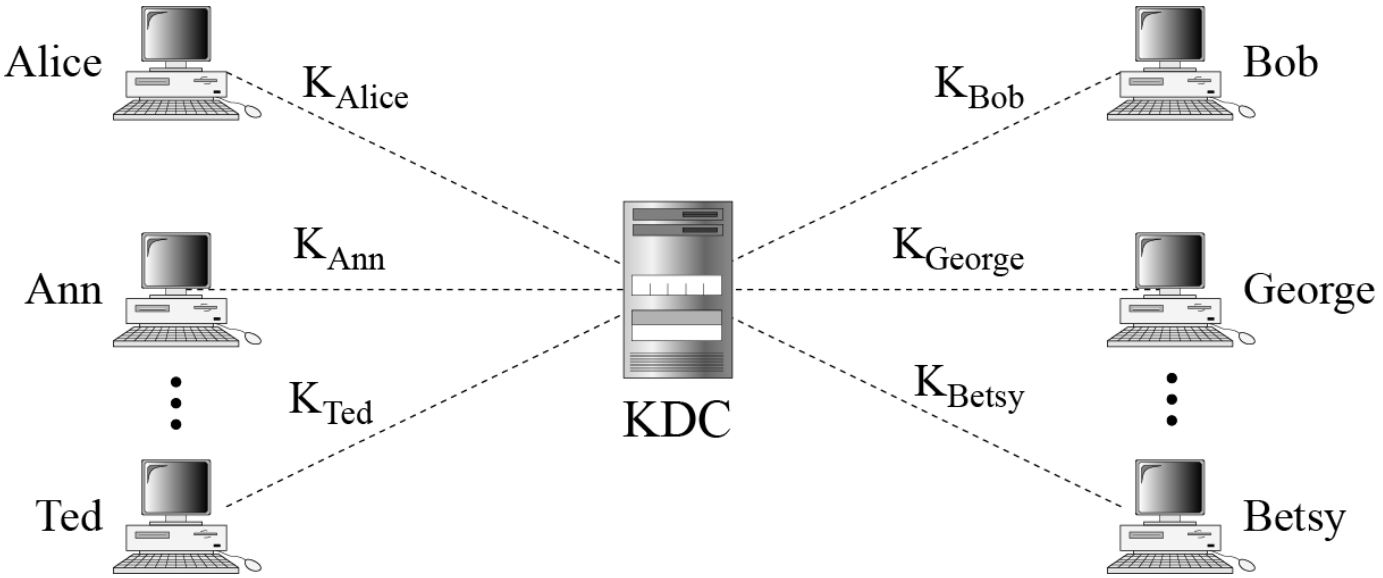
Figure 17.9 *Calculation of key material from master secret*



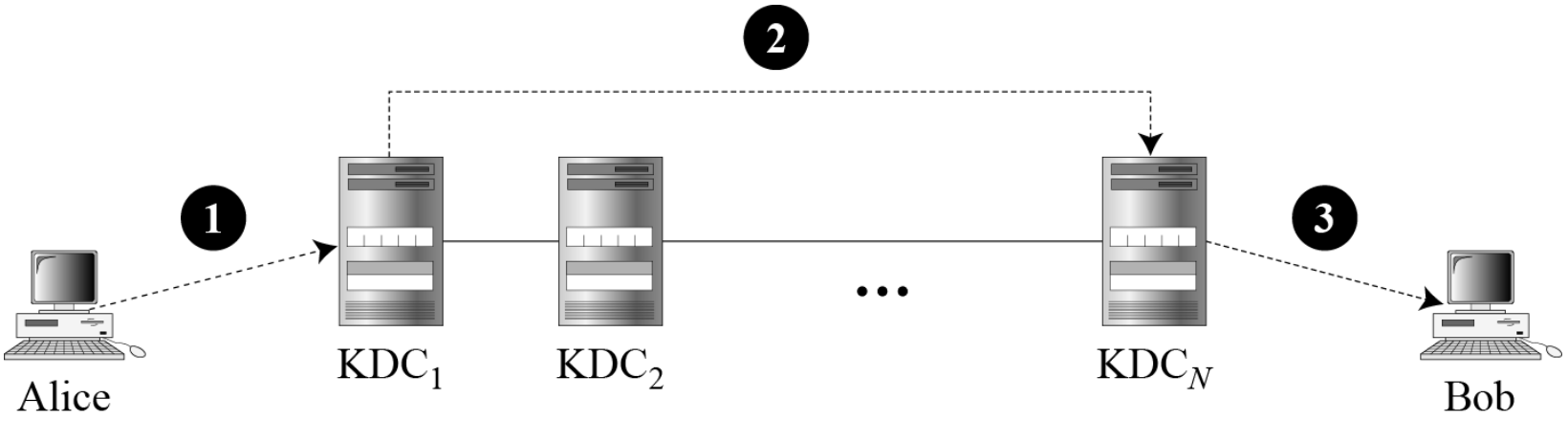
15.1.1 Key-Distribution Center: KDC

Figure 15.1 *Key-distribution center (KDC)*



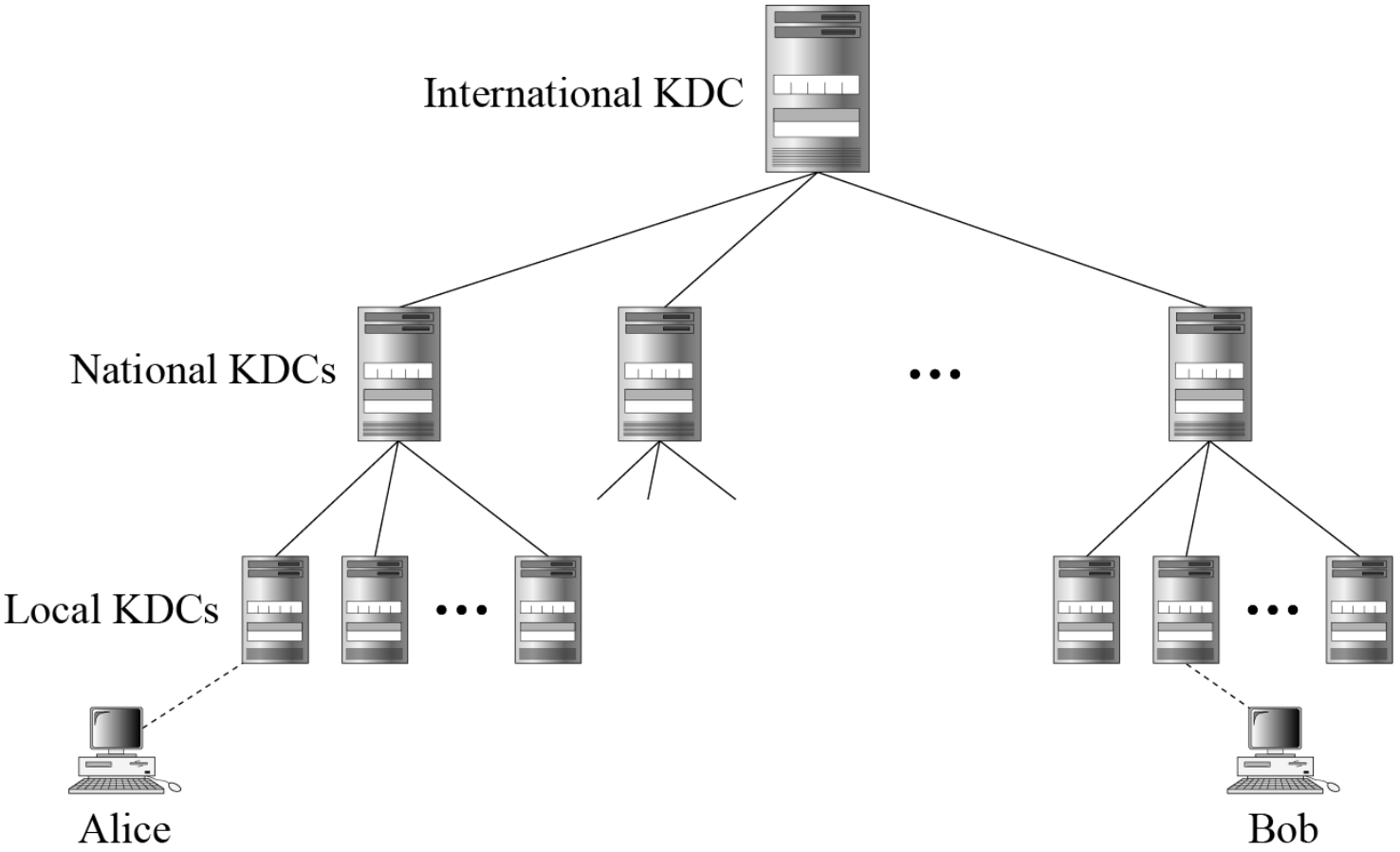
Flat Multiple KDCs.

Figure 15.2 *Flat multiple KDCs*



Hierarchical Multiple KDCs

Figure 15.3 *Hierarchical multiple KDCs*



15.1.2 Session Keys

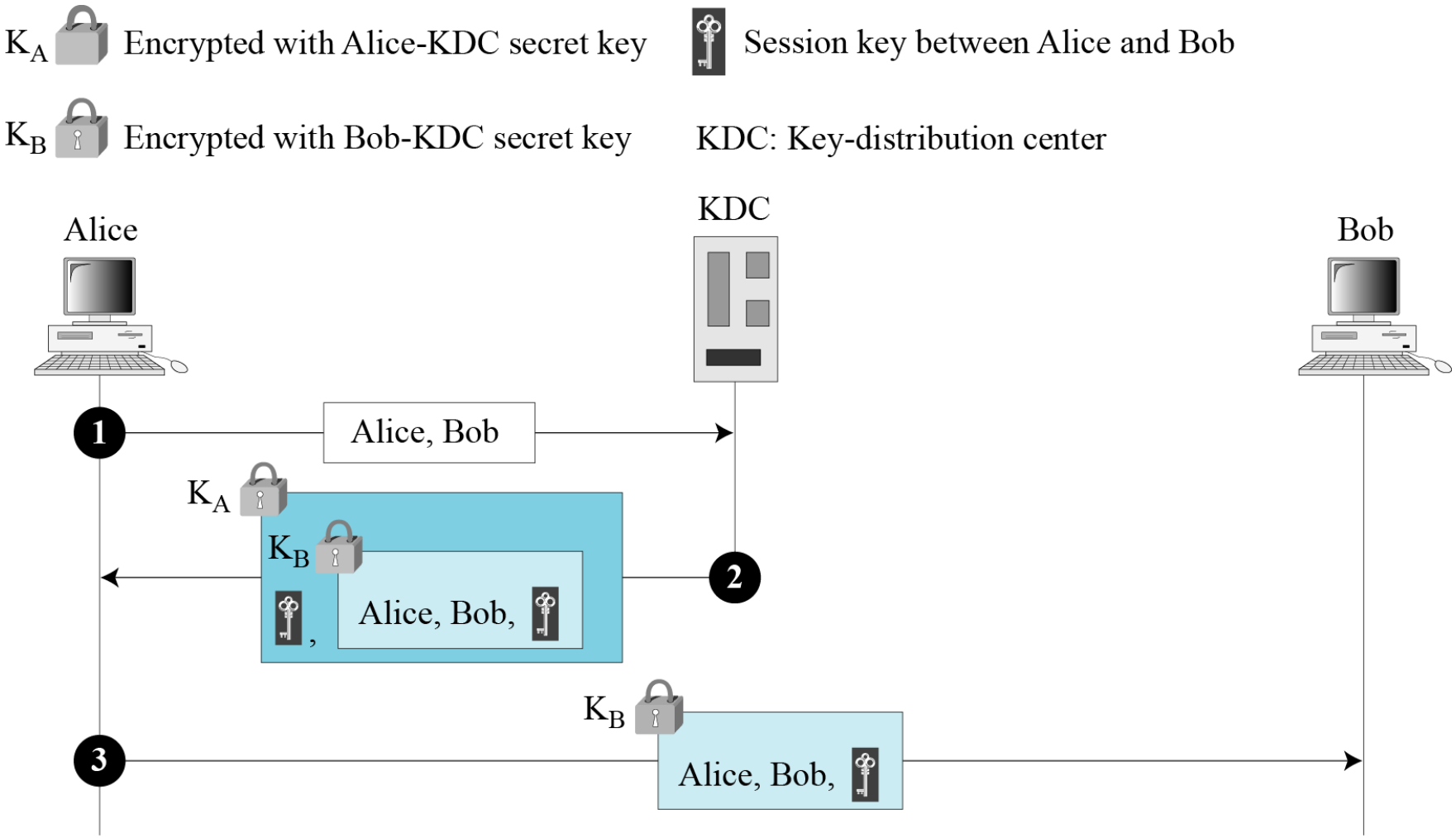
A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.

Note

A session symmetric key between two parties is used only once.

A Simple Protocol Using a KDC

Figure 15.4 *First approach using KDC*



15.1.2 Continued

Needham-Schroeder Protocol

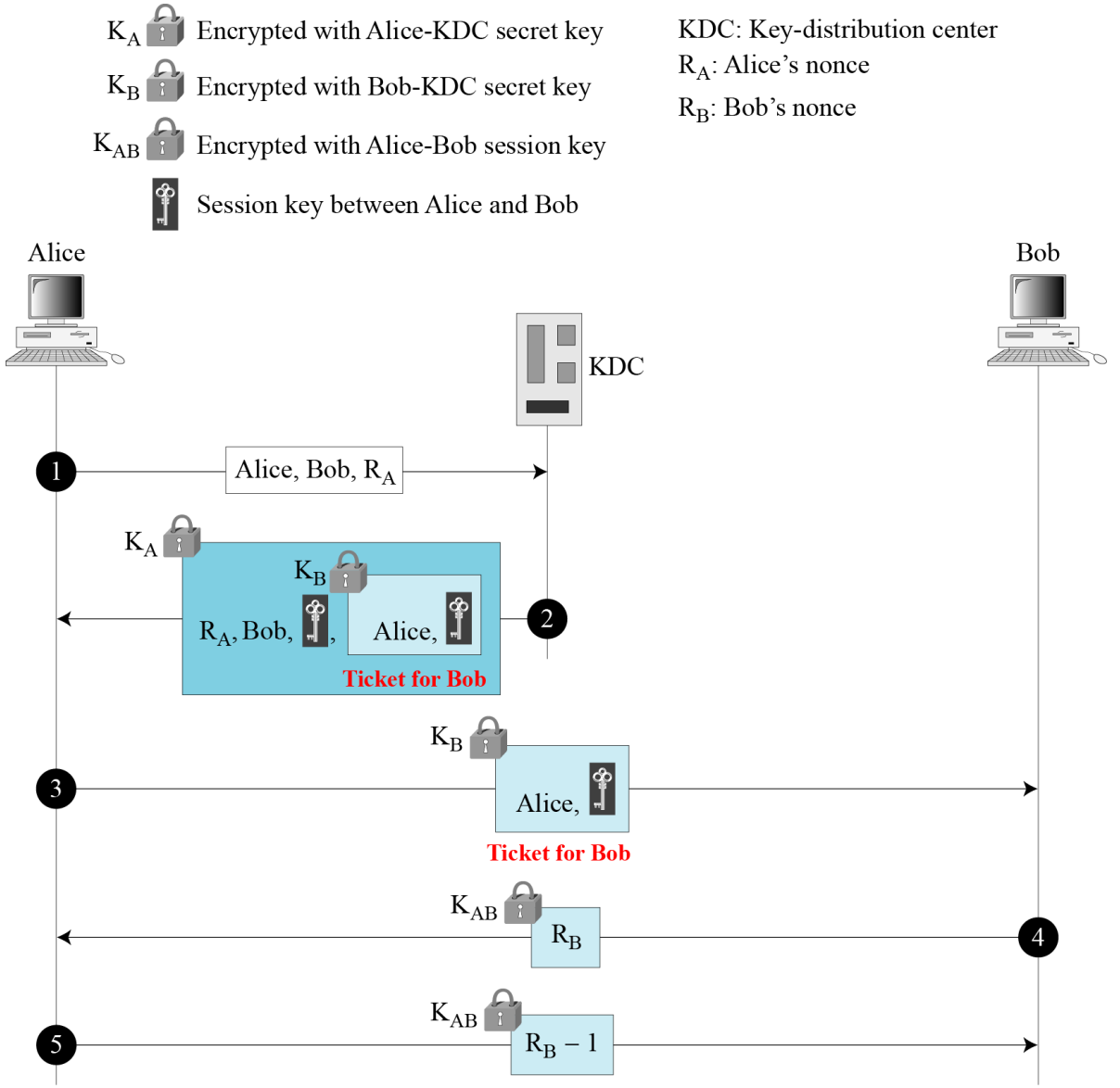


Figure 15.5

Needham-Schroeder protocol

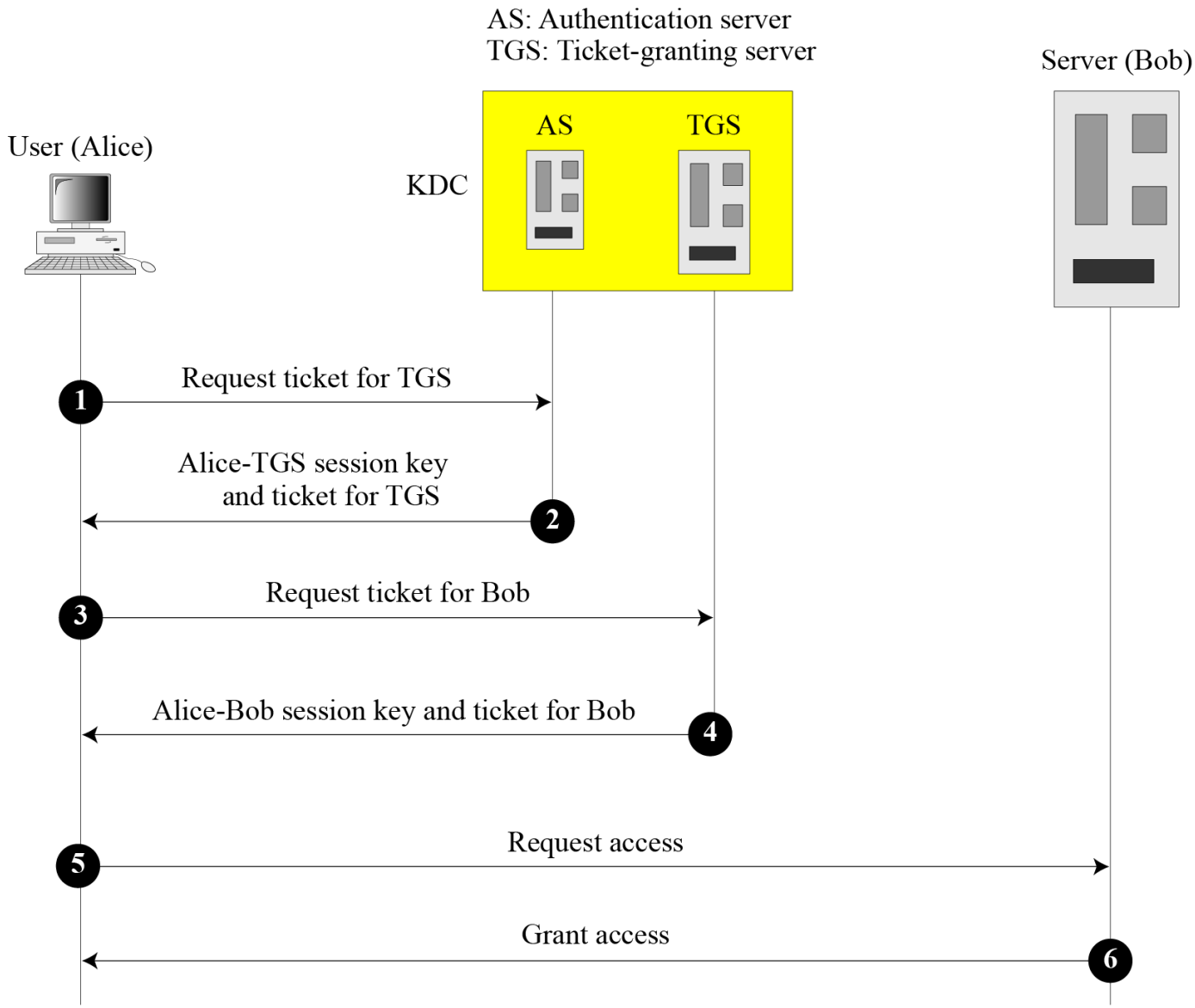
15-2 KERBEROS

Kerberos is an authentication protocol, and at the same time a KDC, that has become very popular. Several systems, including Windows 2000, use Kerberos. Originally designed at MIT, it has gone through several versions.

Topics discussed in this section:

15.2.1 Servers

Figure 15.7 *Kerberos servers*



15.2.1 Continued

Authentication Server (AS)

The authentication server (AS) is the KDC in the Kerberos protocol.

Ticket-Granting Server (TGS)

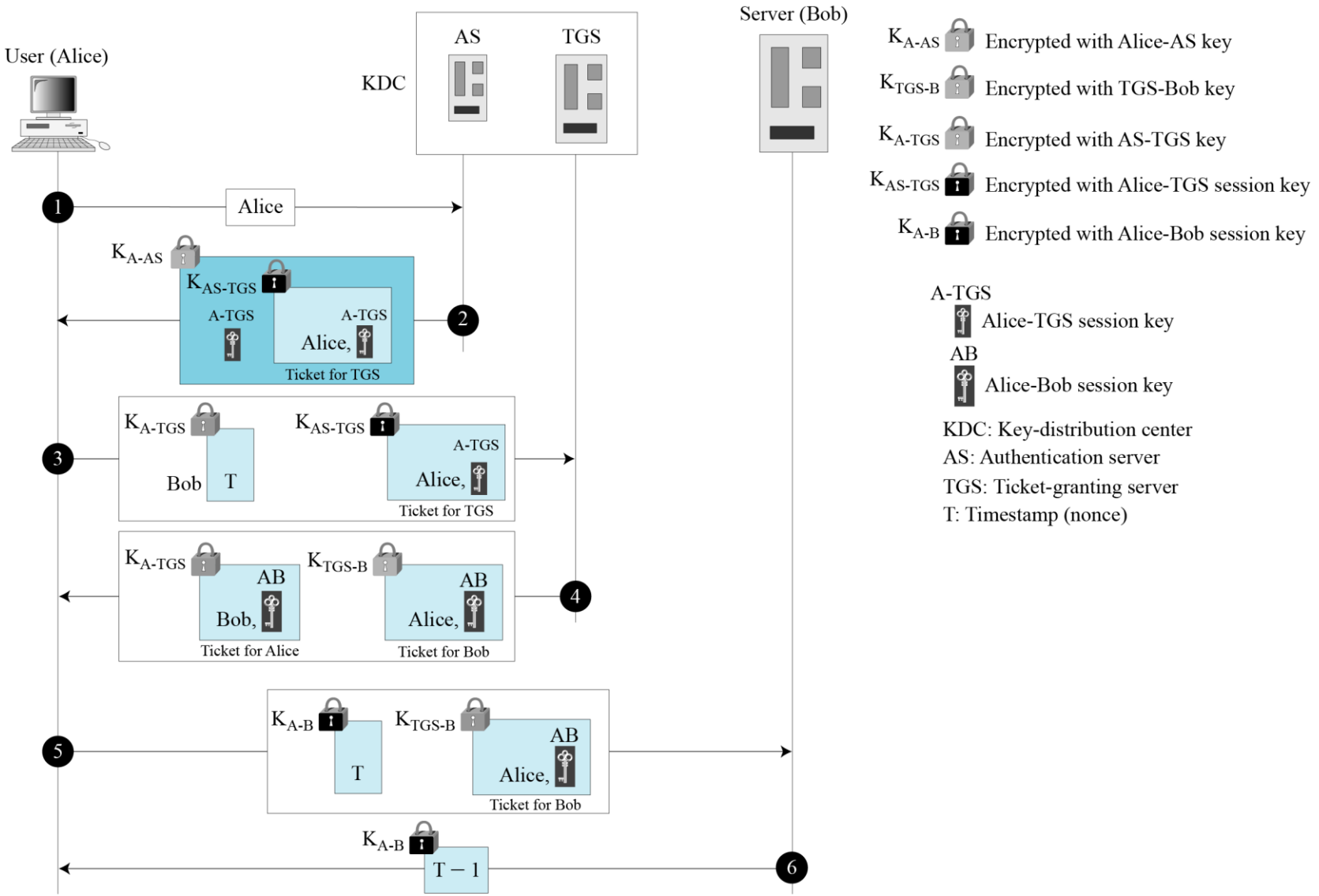
The ticket-granting server (TGS) issues a ticket for the real server (Bob).

Real Server

The real server (Bob) provides services for the user (Alice).

15.2.2 Operation

Figure 15.8 *Kerberos example*



15.2.3 Using Different Servers

Note that if Alice needs to receive services from different servers, she need repeat only the last four steps.

15.2.4 Kerberos Version 5

The minor differences between version 4 and version 5 are briefly listed below:

1) *Version 5 has a longer ticket lifetime.*
2) *Version 5 allows tickets to be renewed.*
3) *Version 5 can accept any symmetric-key algorithm.*
4) *Version 5 uses a different protocol for describing data types.*
5) *Version 5 has more overhead than version 4.*

15.2.5 Realms

Kerberos allows the global distribution of ASs and TGSs, with each system called a realm. A user may get a ticket for a local server or a remote server.

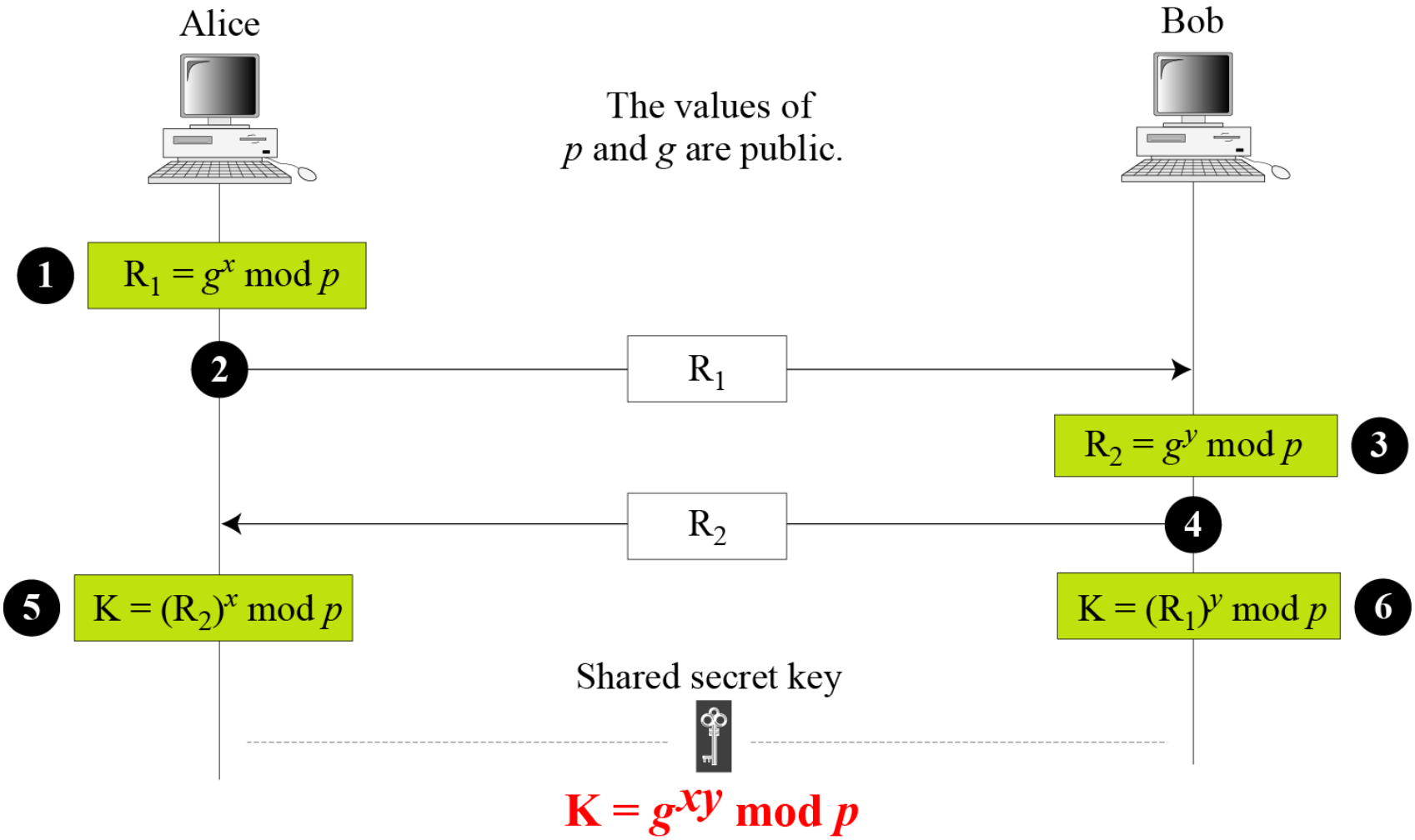
15-3 SYMMETRIC-KEY AGREEMENT

Alice and Bob can create a session key between themselves without using a KDC. This method of session-key creation is referred to as the symmetric-key agreement.

Topics discussed in this section:

15.3.1 Diffie-Hellman Key Agreement

Figure 15.9 *Diffie-Hellman method*



Note

The symmetric (shared) key in the Diffie-Hellman method is $K = g^{xy} \bmod p$.

Example 15.1

Let us give a trivial example to make the procedure clear. Our example uses small numbers, but note that in a real situation, the numbers are very large. Assume that $g = 7$ and $p = 23$. The steps are as follows:

1. **Alice chooses $x = 3$ and calculates $R_1 = 73 \bmod 23 = 21$.**
2. **Bob chooses $y = 6$ and calculates $R_2 = 76 \bmod 23 = 4$.**
3. **Alice sends the number 21 to Bob.**
4. **Bob sends the number 4 to Alice.**
5. **Alice calculates the symmetric key $K = 43 \bmod 23 = 18$.**
6. **Bob calculates the symmetric key $K = 216 \bmod 23 = 18$.**
7. **The value of K is the same for both Alice and Bob; $g^{xy} \bmod p = 718 \bmod 35 = 18$.**

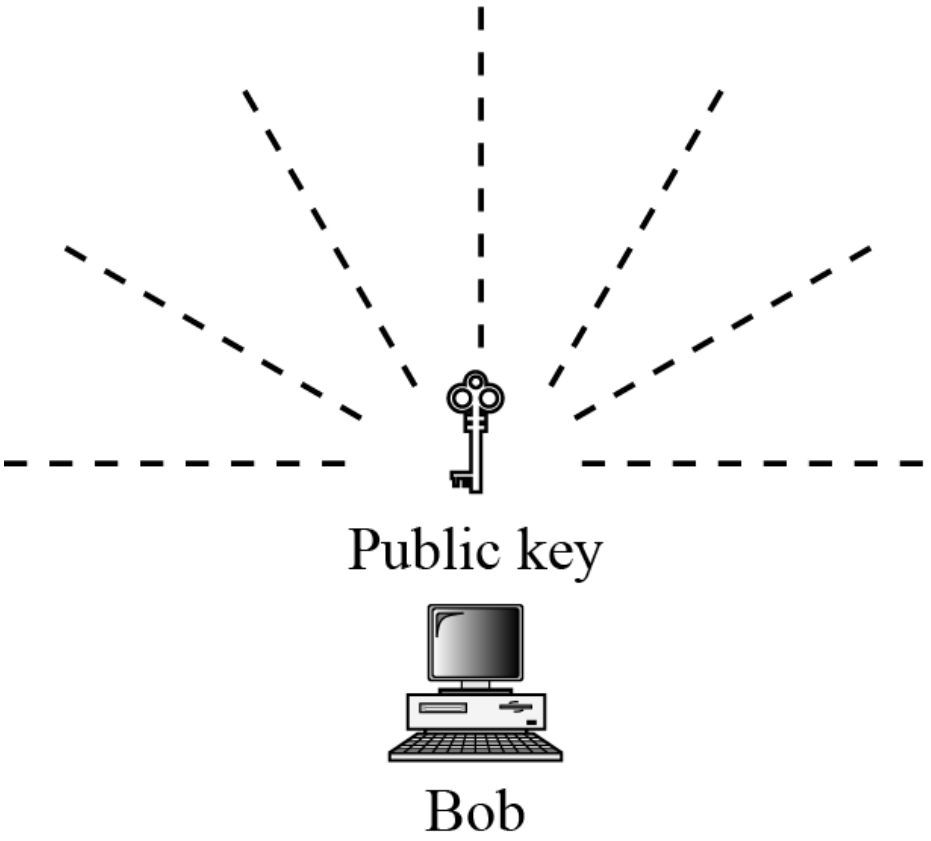
15-4 PUBLIC-KEY DISTRIBUTION

In asymmetric-key cryptography, people do not need to know a symmetric shared key; everyone shields a private key and advertises a public key.

Topics discussed in this section:

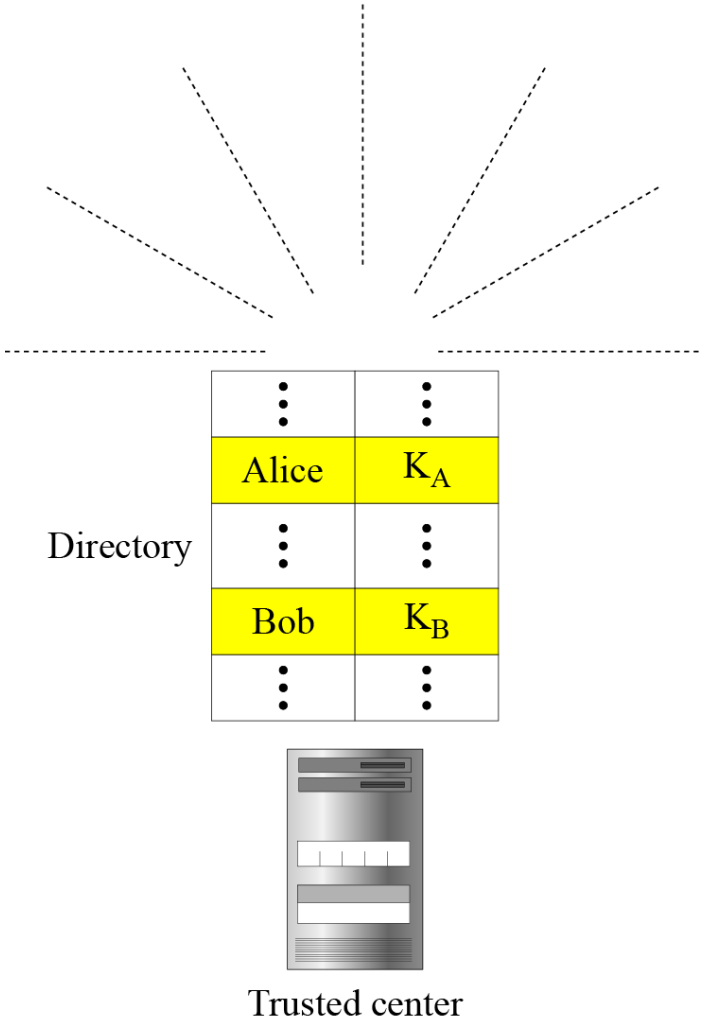
15.4.1 Public Announcement

Figure 15.13 *Announcing a public key*



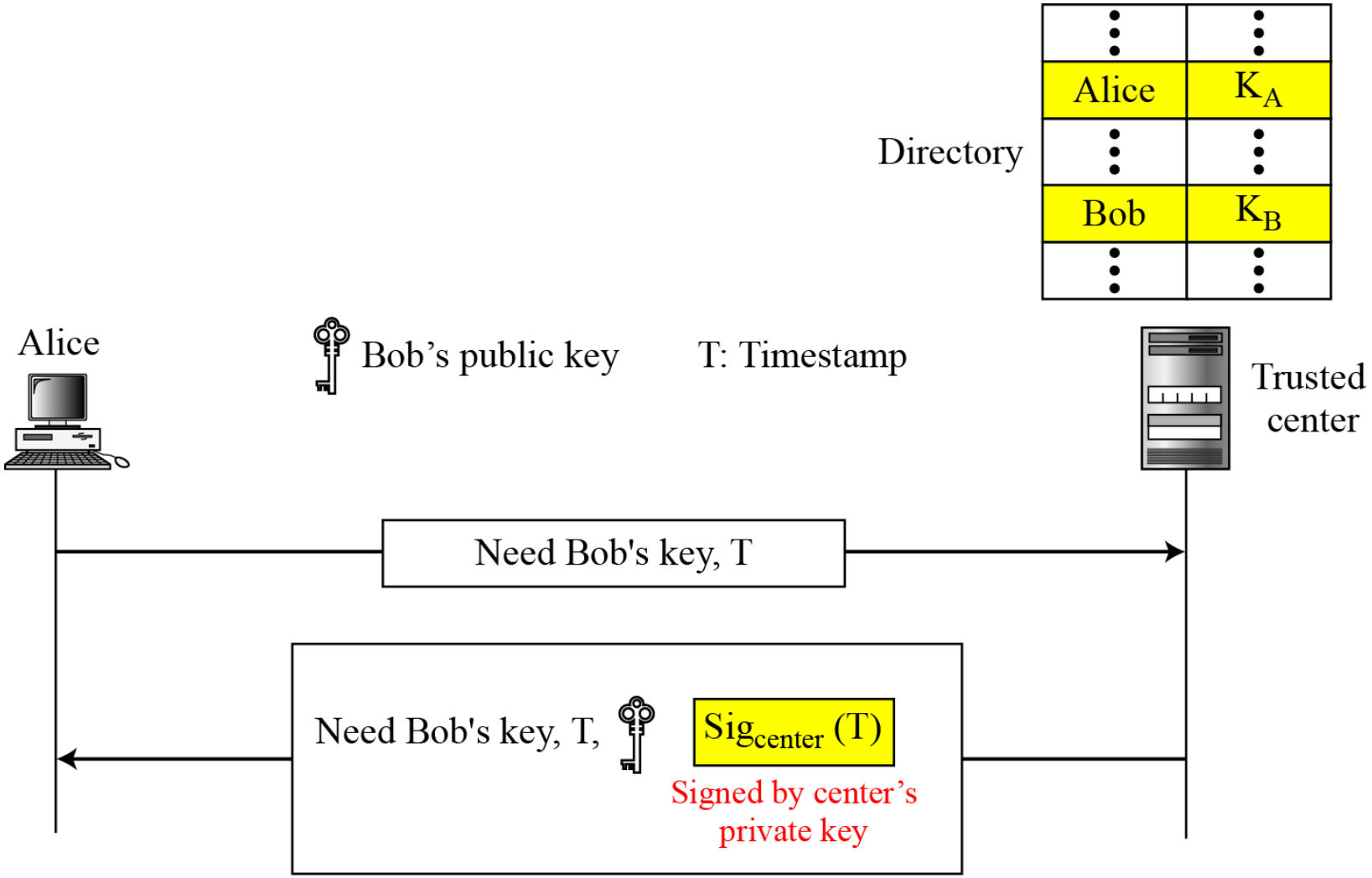
15.4.2 Trusted Center

Figure 15.14 *Trusted center*



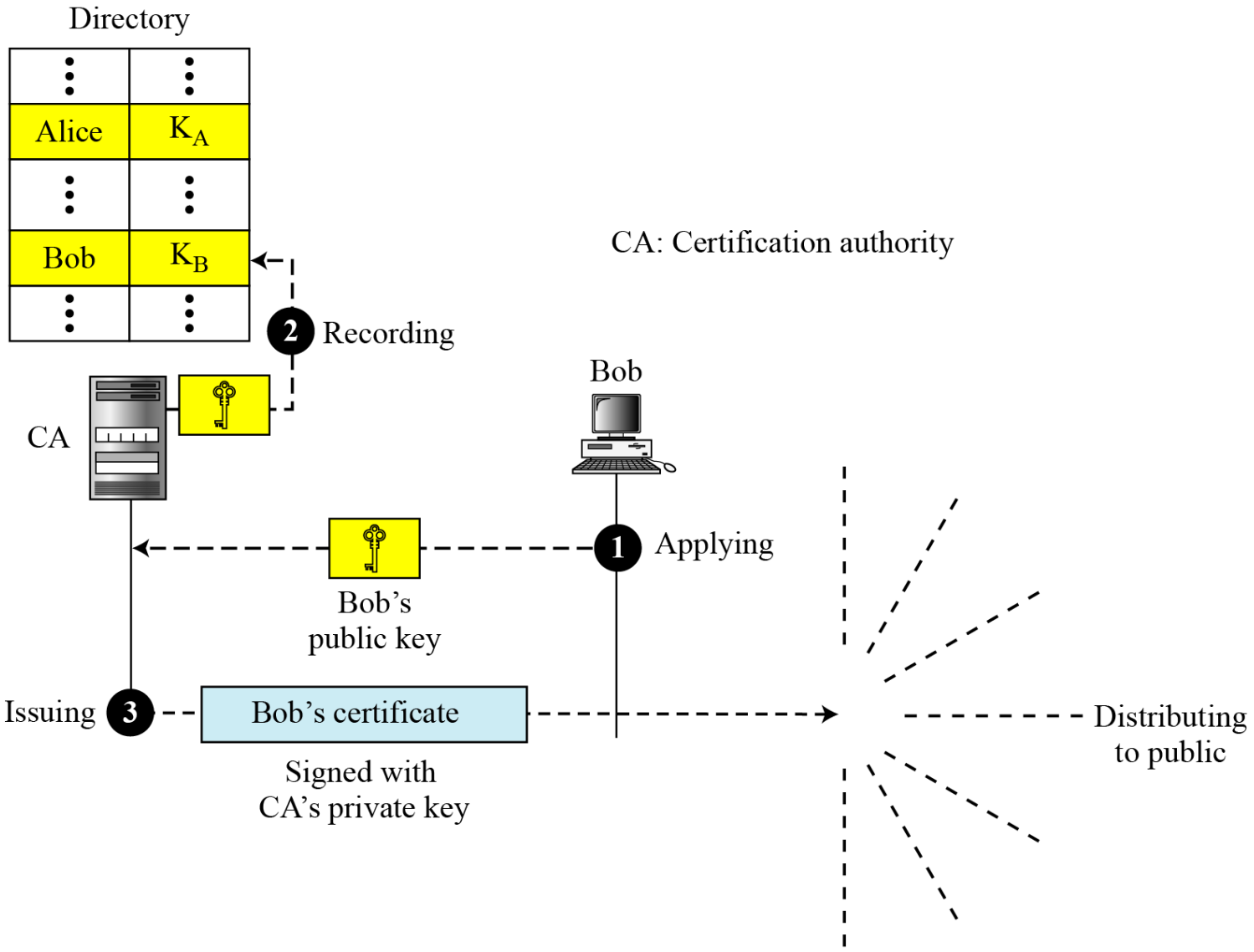
15.4.3 Controlled Trusted Center

Figure 15.15 *Controlled trusted center*



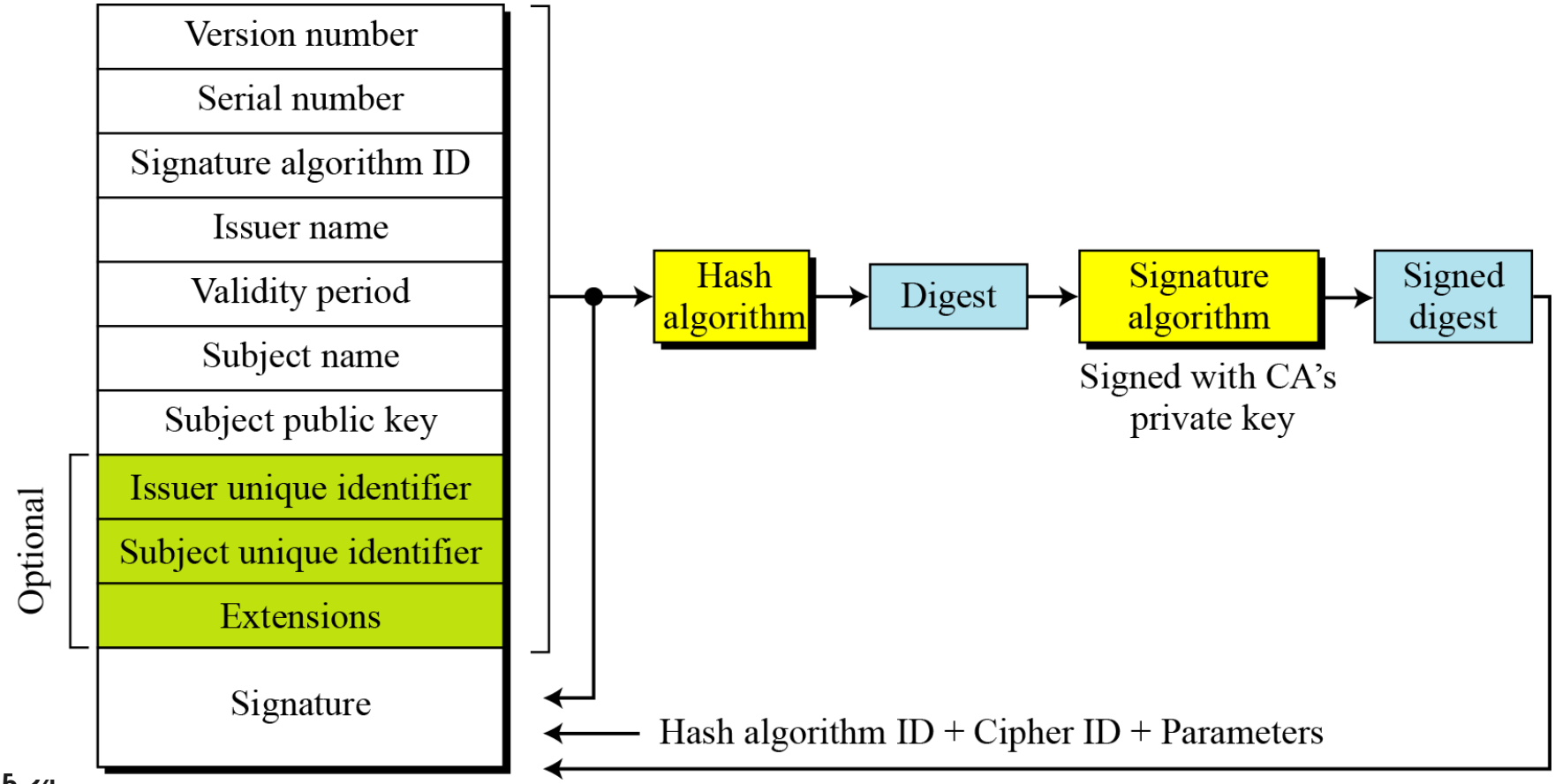
15.4.4 Certification Authority

Figure 15.16 *Certification authority*



Certificate

Figure 15.17 shows the format of a certificate.



15.4.5 Continued

Certificate Renewal

Each certificate has a period of validity. If there is no problem with the certificate, the CA issues a new certificate before the old one expires.

Certificate Revocation

In some cases a certificate must be revoked before its expiration.

15.4.5 Continued

Various reasons for certification revocation:

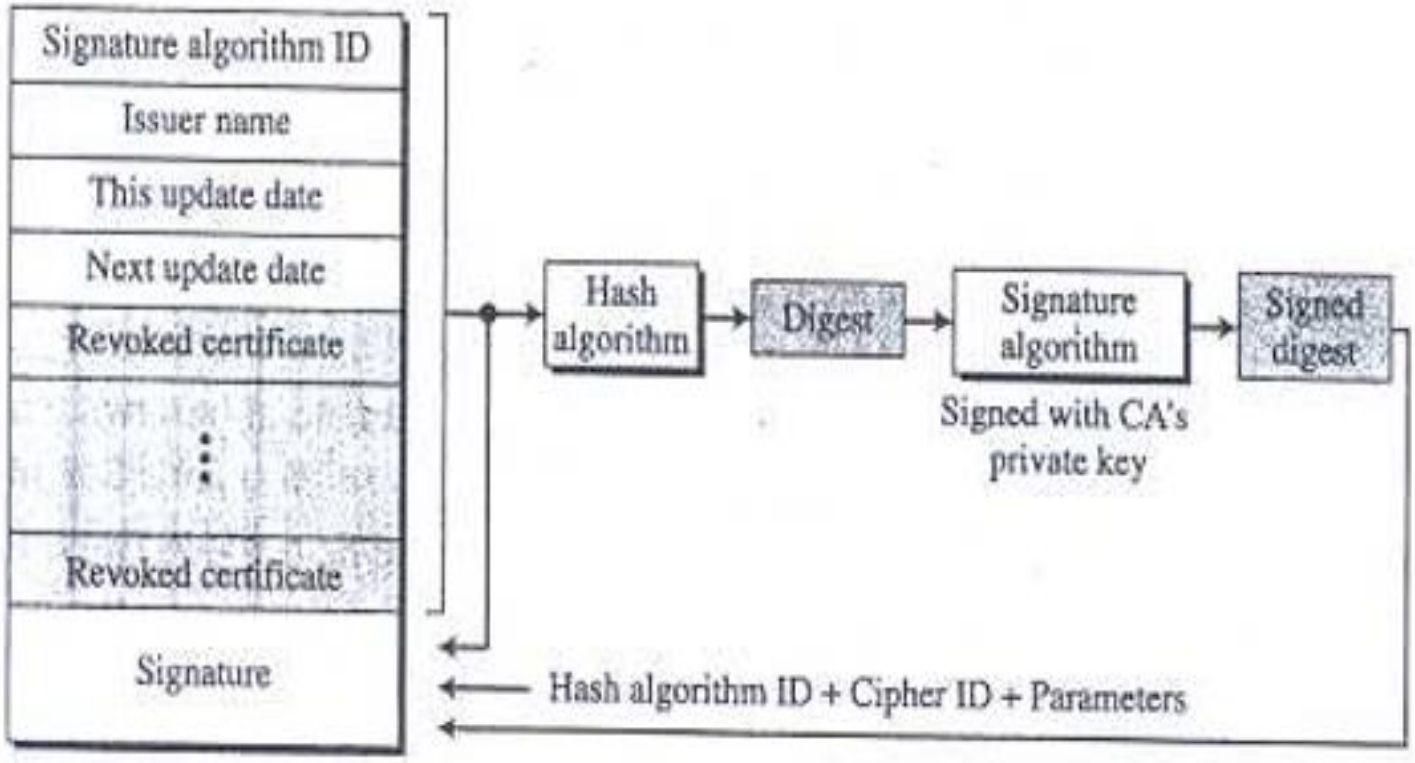
1. *The user's private key is compromised.*
2. *The CA is no longer willing to certify the user.*
3. *The CA's private key, which can verify certificates have been compromised.*

Delta Revocation

To make revocation more efficient, the delta certificate revocation list (delta CRL) has been introduced.
Delta CRL contains only changes made after last CRL.

15.4.5 Continued

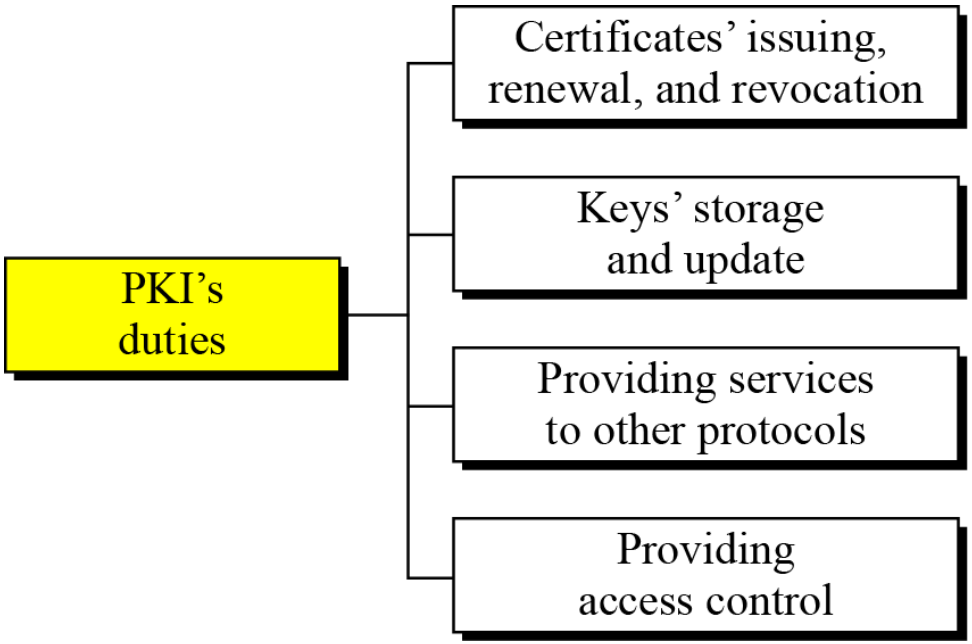
Figure 15.17 *Certificate revocation format*



Public Key Infrastructure

PKI is a model for creating, distributing and revoking certificates based on X.509.

Duties of PKI are



15.4.5 Continued

Trust Model

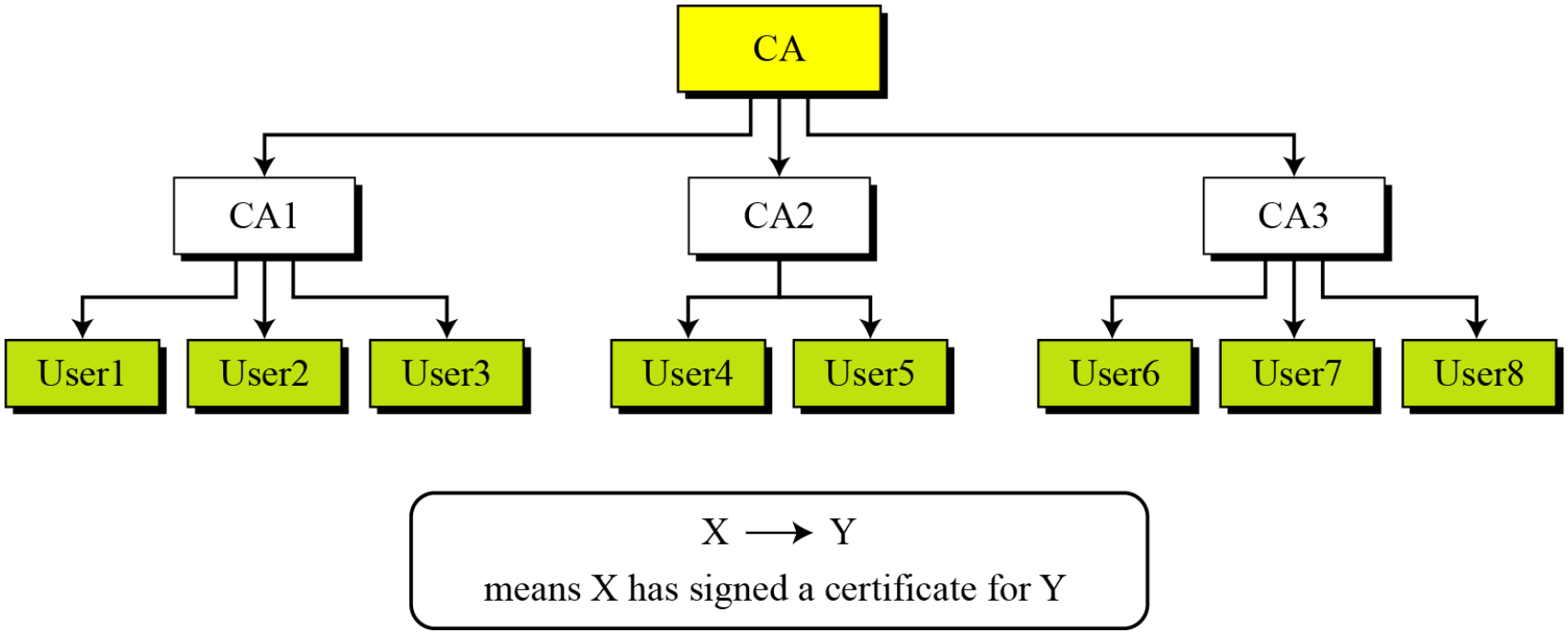
It is not possible to have just one CA issuing all certificates for all users.

There should me many CA's each responsible for creating, storing, issuing and revoking a limited number of certificates.

The trust model defines rules that specify how a user can verify a certificate received from CA.

Trust Model

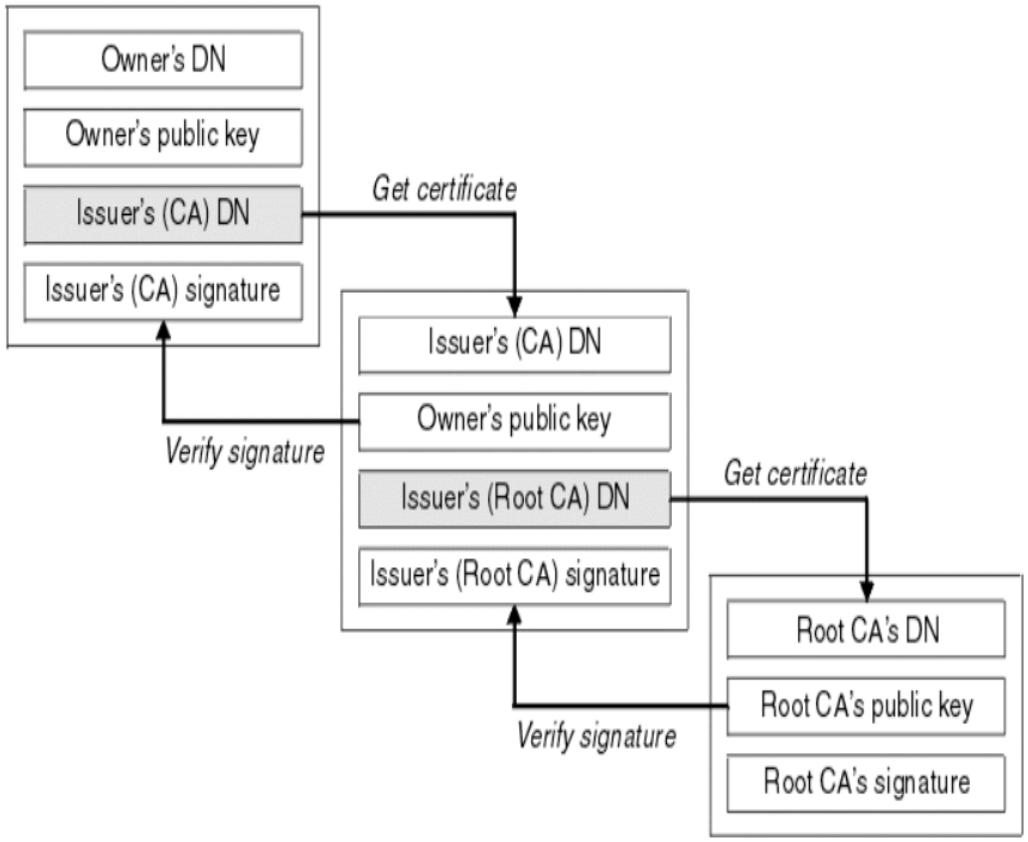
Figure 15.20 *PKI hierarchical model*



15.4.6 Continued

Certificate Chain or Certificate Path

When you get a certificate for your public key from a commercial CA then your certificate is associated with a chain of certificates or sometimes called chain of trust. The number of certificates in the chain depends on the CA's hierarchical structure. The following image shows a certificate chain for a two tier CA. The owners/users certificate is signed by a Issuing CA and issuing CA's certificate is signed by the Root CA. Root CA's certificate is self signed.



15.4.6 Continued

Overview of Services offered by licensed CAs

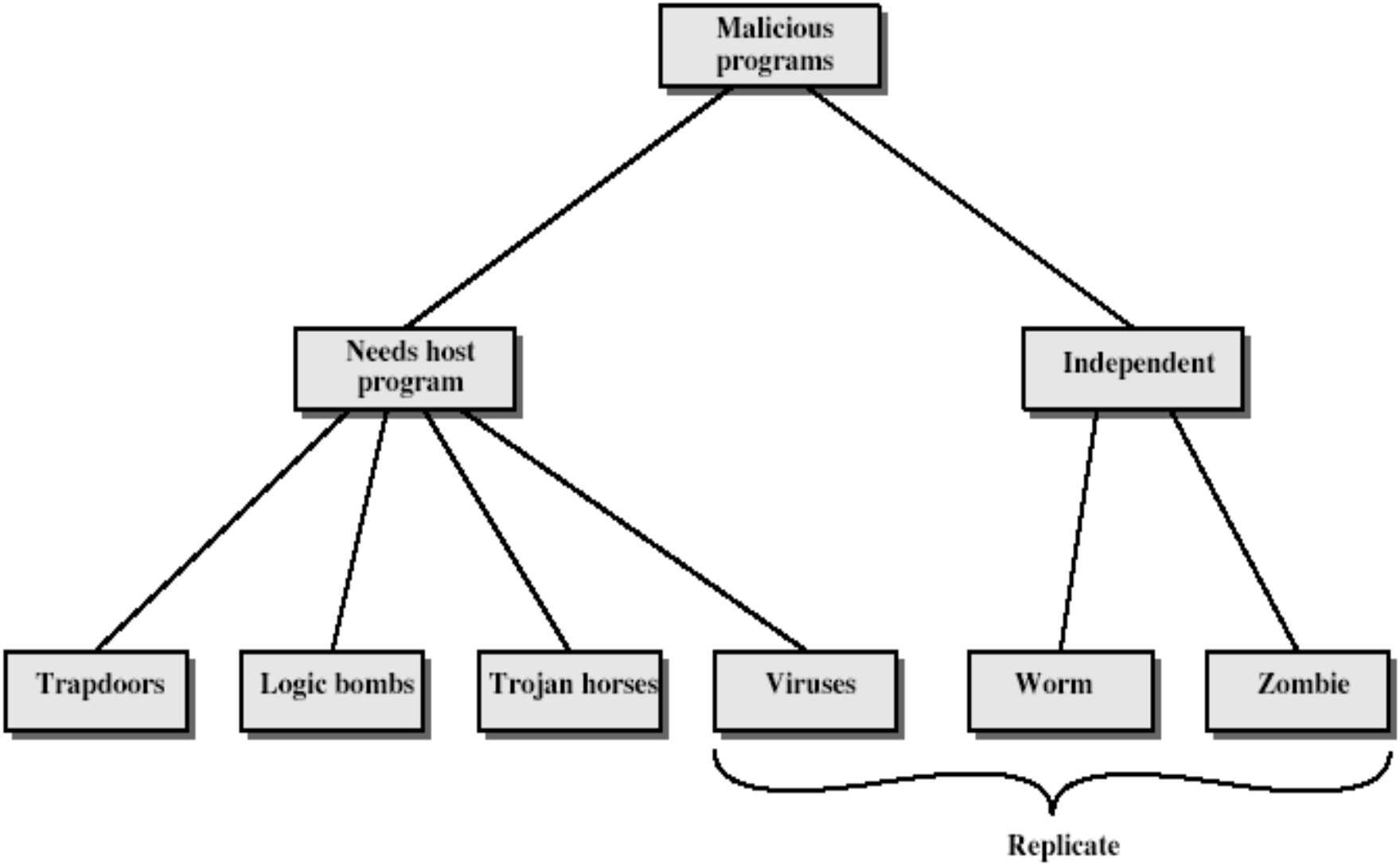
Licensed CAs	CLASS 1-3 DSCs	eSign	SSL & Code Signing certificates	TIMESTAMPING
Safescrypt	✓			✓
IDRBT	✓ ONLY TO BANKS		✓ ONLY TO BANKS*	✓ ONLY TO BANKS
(n)Code Solutions	✓	✓	✓ *	✓
e Mudhra	✓	✓	✓ *	✓
CDAC		✓		
Capricorn	✓	✓		✓
NSDL e-Gov		✓		
Indian Air Force	✓ ONLY TO IAF			✓ ONLY TO IAF
Verasys	✓			

* The Root CA certificate of India is listed only in Microsoft products (including IE).

CLOSED CAS

Licensed CAs	CLASS 1-3 DSCs	SSL & CODE SIGINING
MTNL	NA	NA
iCert	NA	NA
TCS	NA	NA
NIC	NA	NA

Malicious Programs



Trapdoors

- Secret entry point into a program.
- Allows those who know access bypassing usual security procedures.
- Have been commonly used by developers.
- Threat when left in production programs, allowing exploited by attackers
- Requires good s/w development & update.

Logic Bomb

- One of oldest types of malicious software.
- Code embedded in legitimate program.
- Activated when specified conditions met.
  - presence/absence of some file
  - particular date/time
  - particular user
- When triggered typically damage system.
  - modify/delete files/disks

Trojan Horse

- Program with hidden side-effects.
- Which is usually superficially attractive
  - eg game, ZIP software etc.
- When run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- Often used to propagate a virus/worm or install a backdoor or to destroy data.

Trojan Horse

This article is about the mythological Trojan Horse. For the type of malware, see Trojan horse (computing). For other uses, see Trojan horse (disambiguation).

The **Trojan Horse** is a story from the Trojan War about the subterfuge that the Greeks used to enter the independent city of Troy and win the war. In the canonical version, after a fruitless 10-year siege, the Greeks constructed a huge wooden horse, and hid a select force of men inside including Odysseus. The Greeks pretended to sail away, and the Trojans pulled the horse into their city as a victory trophy. That night the Greek force crept out of the horse and opened the gates for the rest of the Greek army, which had sailed back under cover of night. The Greeks entered and destroyed the city of Troy, ending the war.

Metaphorically, a "Trojan Horse" has come to mean any trick or stratagem that causes a target to invite a foe into a securely protected bastion or place. A malicious computer program which tricks users into willingly running it is also called a "Trojan horse" or simply a "Trojan".

The main ancient source for the story is the *Aeneid* of Virgil, a Latin epic poem from the time of Augustus. The event is also referred to in Homer's *Odyssey*.[1] In the Greek tradition, the horse is called the "wooden horse" (δουράτεος ἵππος *dourateos hippos* in Homeric/Ionic Greek (*Odyssey* 8.512); δούρειος ἵππος, *doureios hippos* in Attic Greek).

Trojan Horse - Canakkale Waterfront - Dardanelles - Turkey

Zombie

- Program which secretly takes over another networked computer then uses it to indirectly launch attacks.
- Often used to launch distributed denial of service (DDoS) attacks.

Viruses

- A virus is a piece of software that can "infect" other programs by modifying them.
- The modification includes a copy of the virus program, which can then go on to infect other programs.
- A virus can be pre-pended or post-pended to an executable program.
- The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.
- The virus may perform some action, usually detrimental to the system. This action could be a logic bomb that triggers only under certain conditions.
- If the infection phase of the program is reasonably rapid, a user is unlikely to notice any difference between the execution of an infected and uninfected program.

Types of Viruses

- Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.

- Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.

- Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

Types of Viruses

- Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software. For ex: Using compression to hide increased file size due to infection.

- Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible. For ex: To insert superfluous instructions or change the order of independent instructions.

E-mail Virus

- The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment.
- If the recipient opens the e-mail attachment, the Word macro is activated.
- Then e-mail virus sends itself to everyone on the mailing list in the user's e-mail package.
- Later, more powerful version of the e-mail virus appeared, which can be activated merely by opening an e-mail.
- Ultimately, a greater degree of security must be built into Internet utility and application software on PCs to counter the growing threat.

Worms

- A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate. Also, the worm usually performs some unwanted function.

- An e-mail virus has some of the characteristics of a worm, however, we can still classify it as a virus because it requires a human to move it forward. To replicate itself, a network worm uses some sort of network vehicle. For ex. : Electronic mail facility, Remote login capability etc.

- The new copy of the worm program is then run on the remote system continues to spread in the same fashion.

- Ex: **Stuxnet** is a malicious computer worm, first uncovered in 2010. Thought to have been in development since at least 2005, Stuxnet targets SCADA systems and is believed to be responsible for causing substantial damage to Iran's nuclear program. Although neither country has openly admitted responsibility, the worm is believed to be a jointly built American/Israeli cyberweapon.[1][2]

Worms

- **How to tell if your computer has a worm**

- If you suspect your devices are infected with a computer worm, run a virus scan immediately. Even if the scan comes up negative, continue to be proactive by following these steps.

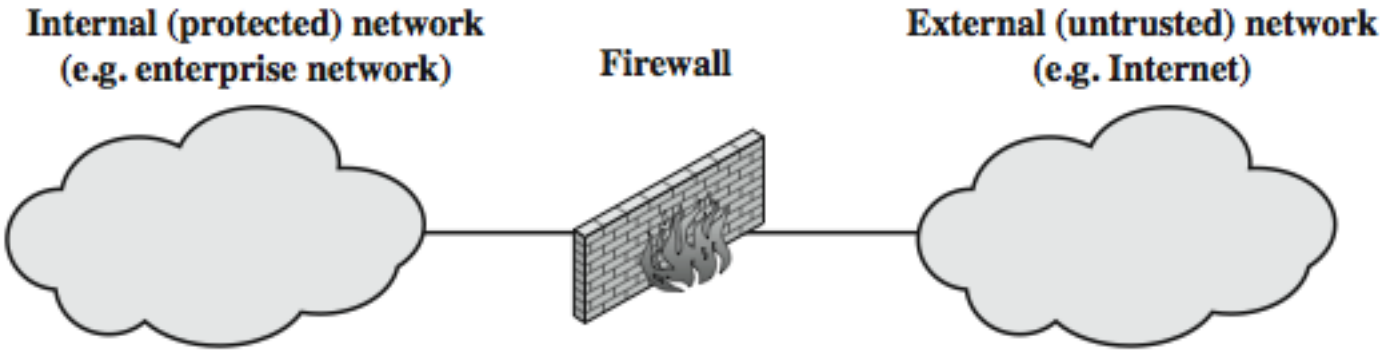
- **Keep an eye on your hard drive space**. When worms repeatedly replicate themselves, they start to use up the free space on your computer.

- **Monitor speed and performance**. Has your computer seemed a little sluggish lately? Are some of your programs crashing or not running properly? That could be a red flag that a worm is eating up your processing power.

- **Be on the lookout for missing or new files**. One function of a computer worm is to delete and replace files on a computer.

Firewall?

- A **choke point** of control and monitoring.
- Only authorized traffic is allowed.
- Auditing and controlling access.
- implement VPNs using IPSec.

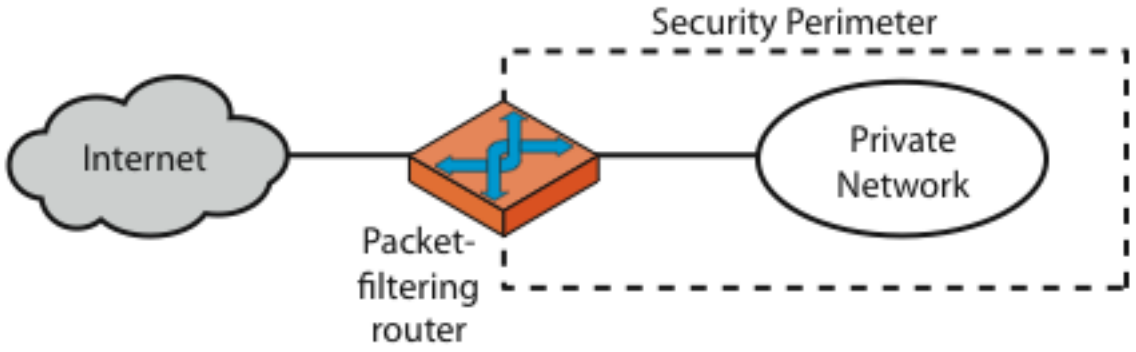


Firewall Limitations

- Cannot protect from attacks bypassing it.
- Cannot protect against internal threats.
- Cannot protect against access via WLAN.
- Cannot protect against malware imported via laptop, PDA, storage infected outside.

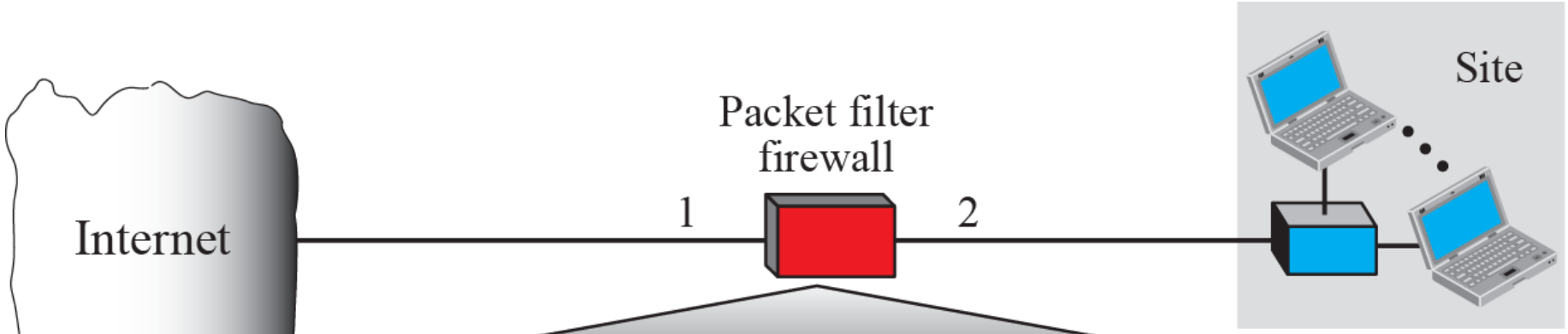
Packet Filter Firewall

- Types of firewalls: Packet filters, Application-level gateways, & Circuit-level gateways.
- A packet-filtering router applies a set of rules to each incoming and outgoing IP packet to forward or discard the packet. Filtering rules are based on information contained in a network packet.



(a) Packet-filtering router

Packet Filter Firewall

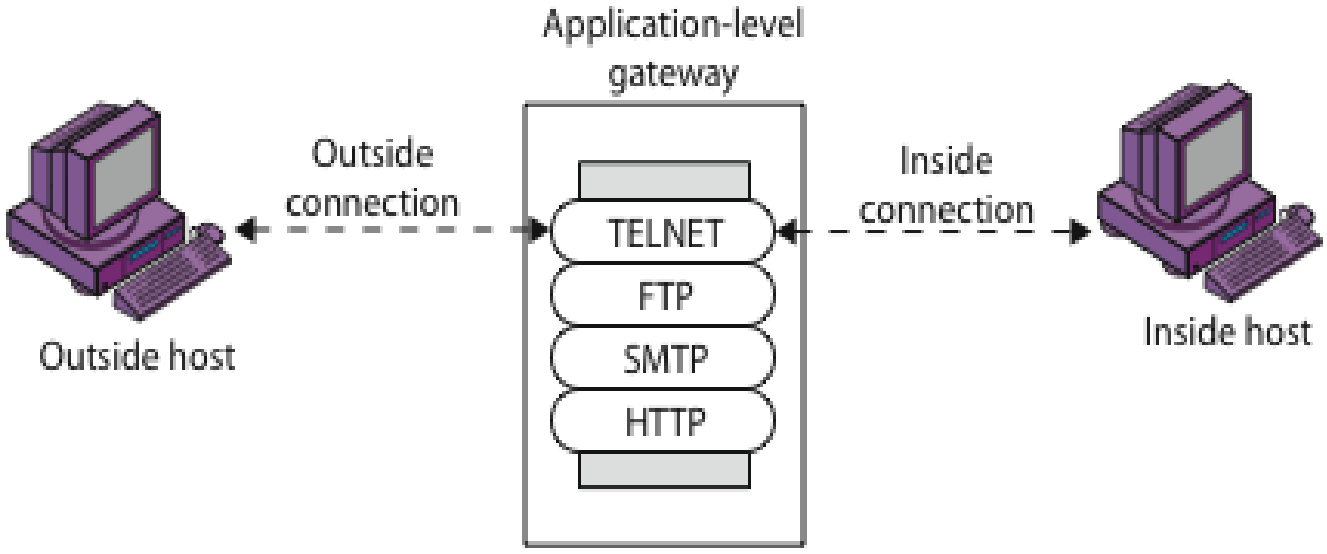


Interface	Source IP	Source port	Destination IP	Destination port
1	131.34.0.0	*	*	*
1	*	*	*	23
1	*	*	194.78.20.8	*
2	*	*	*	80

Application-level Gateway

- An Application level gateway, also called a proxy server.
- The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.
- When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints.
- A prime disadvantage is the additional processing overhead on each connection.

Application-level Gateway

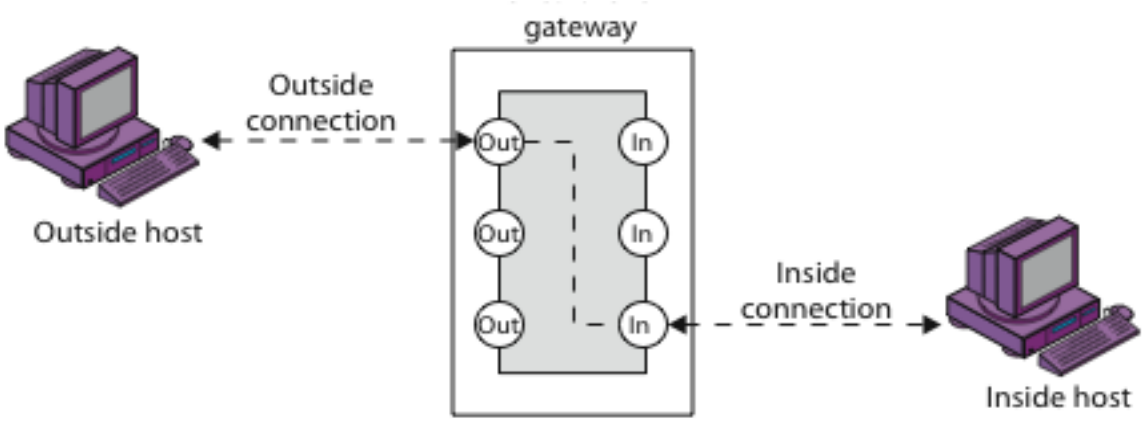


(b) Application-level gateway

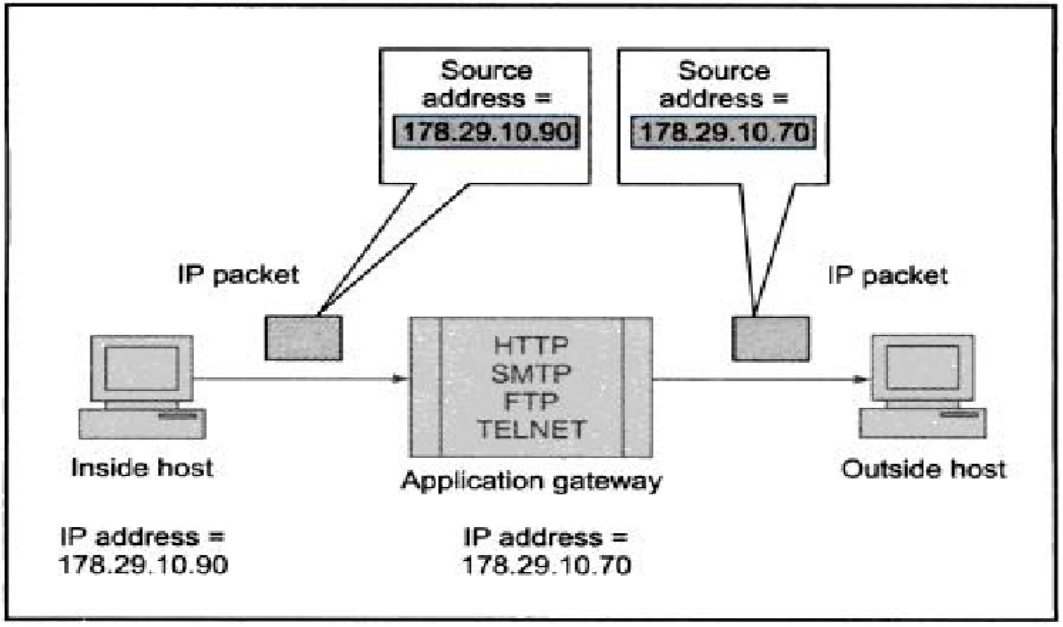
Circuit-Level Gateway

- Circuit level gateway can be a stand-alone system or it can be a specified function performed by an application level gateway for certain applications.
- A Circuit level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections.
- One between itself and a TCP user on an inner host and one between itself and a TCP user on an outer host.
- Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents.

Circuit-Level Gateway



(c) Circuit-level gateway



E-mail Security- Pretty Good Privacy (PGP)

- Provides a confidentiality and authentication service that can be used for electronic mail and file storage applications
- Developed by Phil Zimmermann
  - Selected the best available cryptographic algorithms as building blocks
  - Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands
  - Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks
  - Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP

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.

PGP Authentication

- Combination of SHA-1 and RSA provides an effective digital signature scheme
  - Because of the strength of RSA the recipient is assured that only the possessor of the matching private key can generate the signature
  - Because of the strength of SHA-1 the recipient is assured that no one else could generate a new message that matches the hash code
- As an alternative, signatures can be generated using DSS/SHA-1

PGP Confidentiality

- Provided by encrypting messages to be transmitted or to be stored locally as files
  - In both cases the symmetric encryption algorithm CAST-128 may be used
  - Alternatively IDEA or 3DES may be used
  - The 64-bit cipher feedback (CFB) mode is used

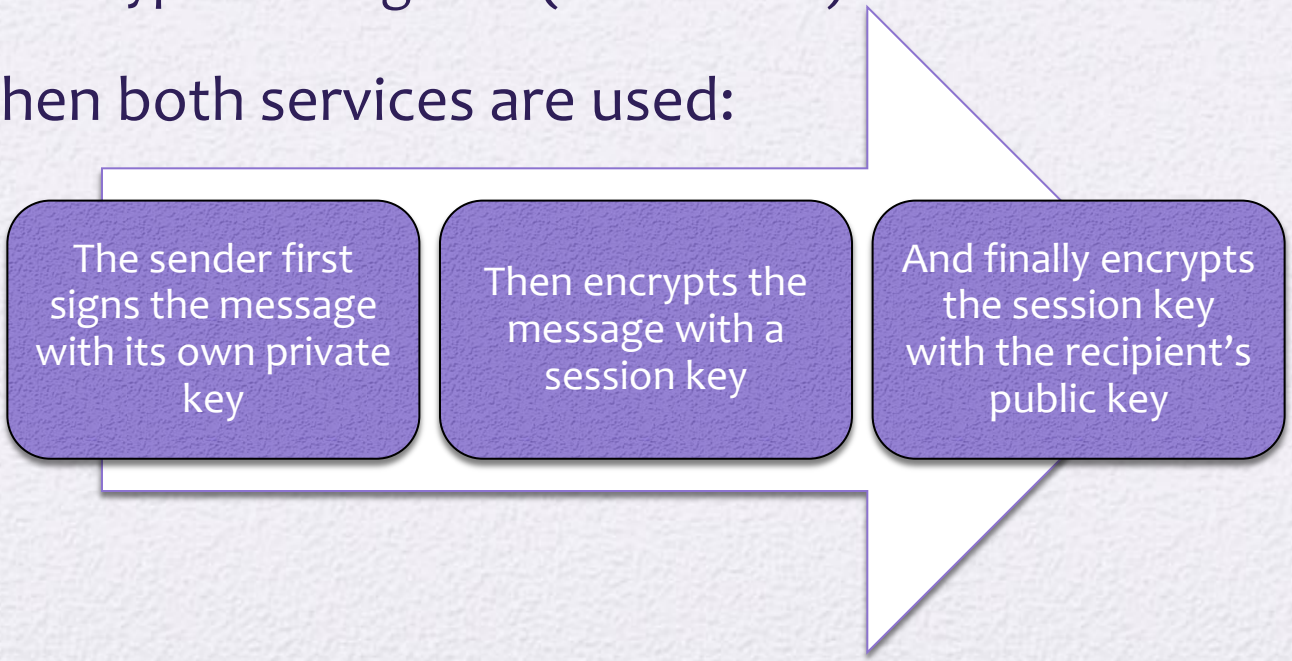
In PGP each symmetric key is used only once

- Although referred to as a session key, it is in reality a one-time key
- Session key is bound to the message and transmitted with it
- To protect the key, it is encrypted with the receiver's public key

- As an alternative to the use of RSA for key encryption, PGP uses ElGamal, a variant of Diffie-Hellman that provides encryption/decryption

PGP Confidentiality and Authentication

- Both services may be used for the same message
  - First a signature is generated for the plaintext message and prepended to the message
  - Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES) and the session key is encrypted using RSA (or ElGamal)
- When both services are used:



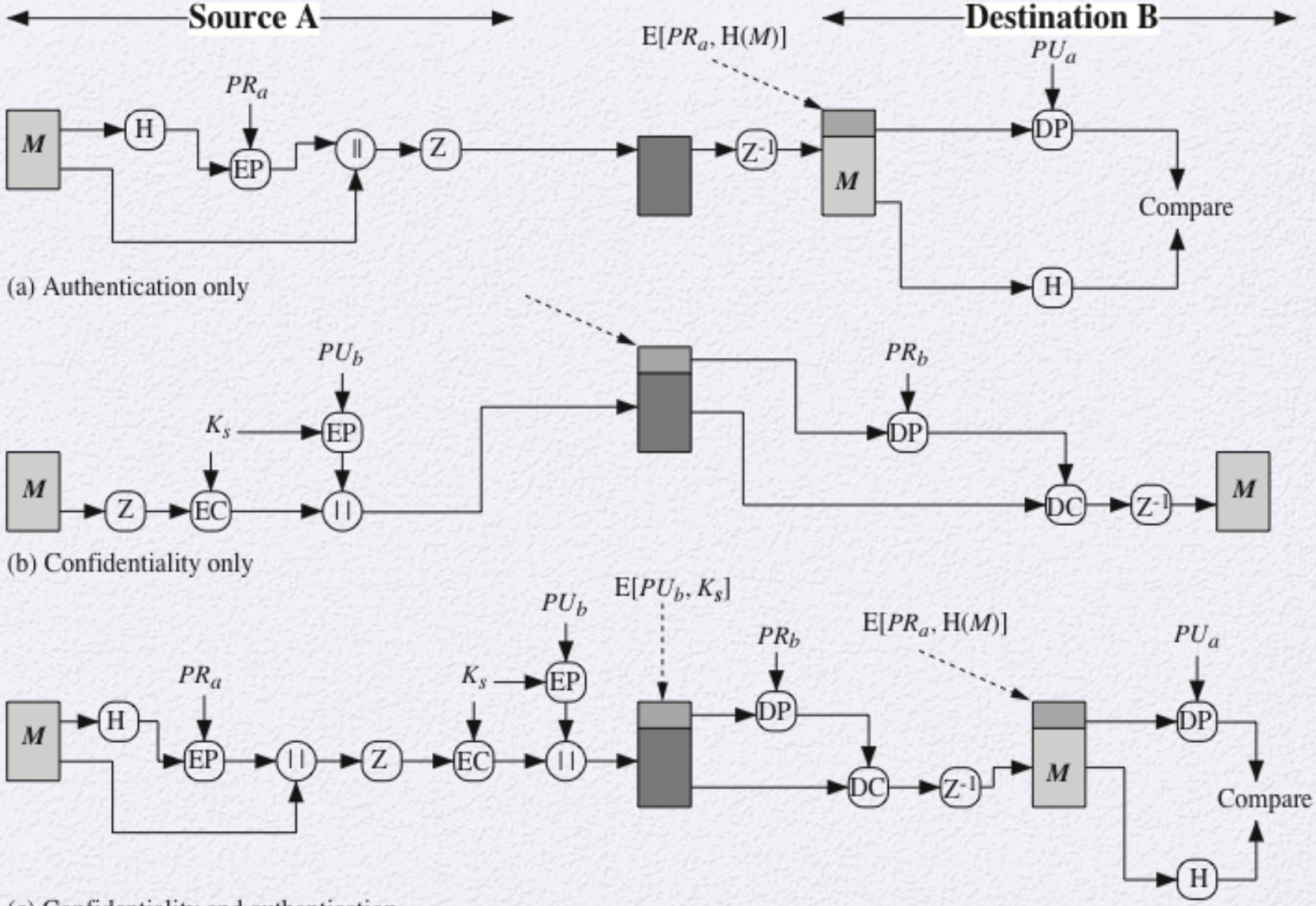


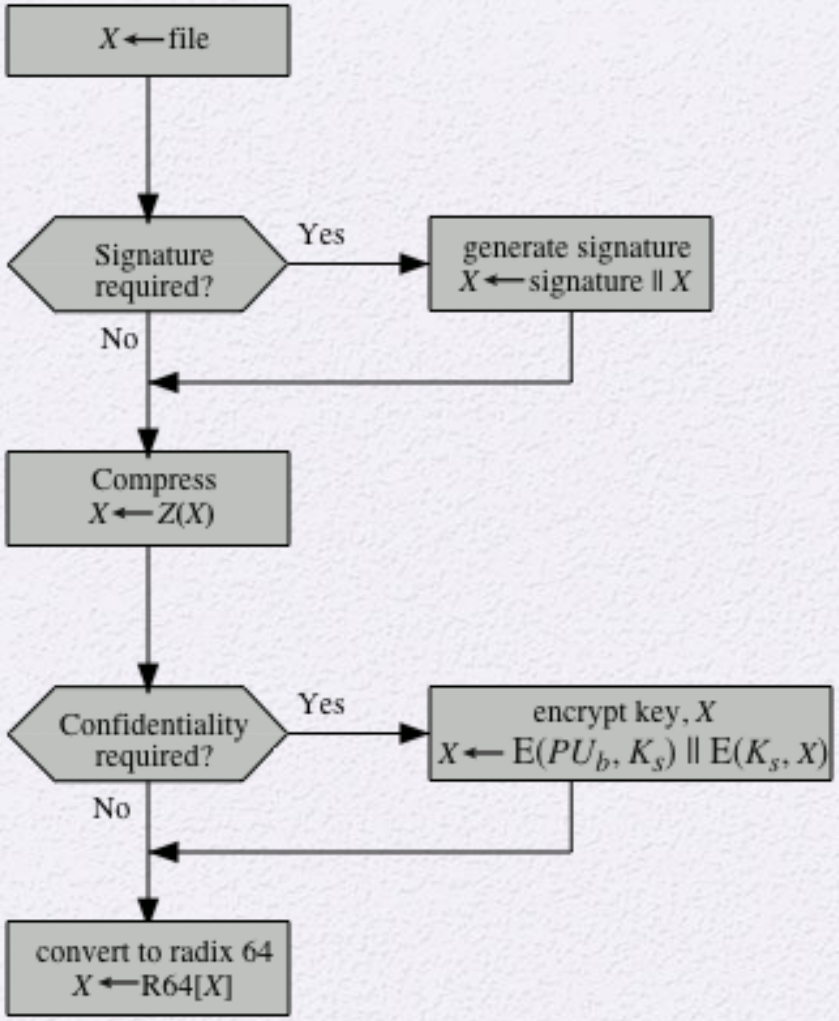
Figure 19.1 PGP Cryptographic Functions

PGP Compression

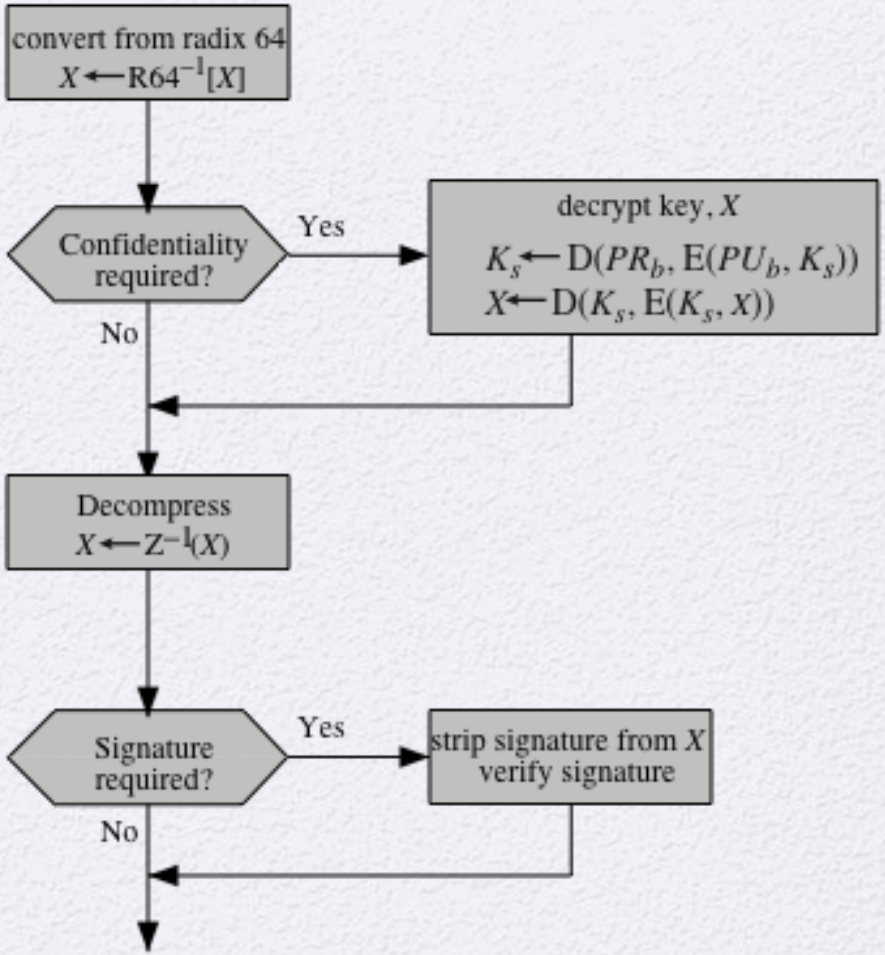
- As a default, PGP compresses the message after applying the signature but before encryption
  - This has the benefit of saving space both for e-mail transmission and for file storage
  - The placement of the compression algorithm is critical
    - It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.
    - Message encryption is applied after compression to strengthen cryptographic security
  - The compression algorithm used is ZIP

PGP E-mail Compatibility

- Many electronic mail systems only permit the use of blocks consisting of ASCII text
  - To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters
  - The scheme used for this purpose is radix-64 conversion
    - Each group of three octets of binary data is mapped into four ASCII characters
    - This format also appends a CRC to detect transmission errors



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

Figure 19.2 Transmission and Reception of PGP Messages

Radix 64 Conversion

Suppose the text to be encrypted has been converted into binary using ASCII coding and encrypted to give a ciphertext stream of binary.

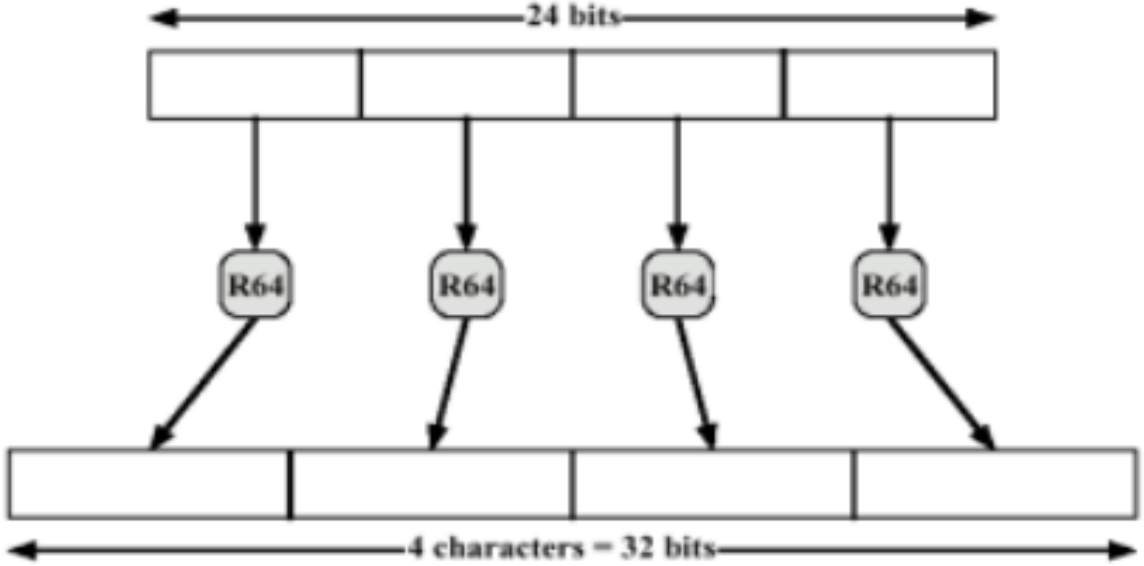
Radix-64 conversion maps arbitrary binary into printable characters as follows:

Radix 64 Conversion

1. The binary input is split into blocks of 24 bits (3 bytes).
2. Each 24 block is then split into four sets each of 6-bits.
3. Each 6-bit set will then have a value between 0 and 2^6-1 (=63).
4. This value is encoded into a printable character.

Radix 64 Conversion

- To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion
- Radix-64 expands a message by 33%



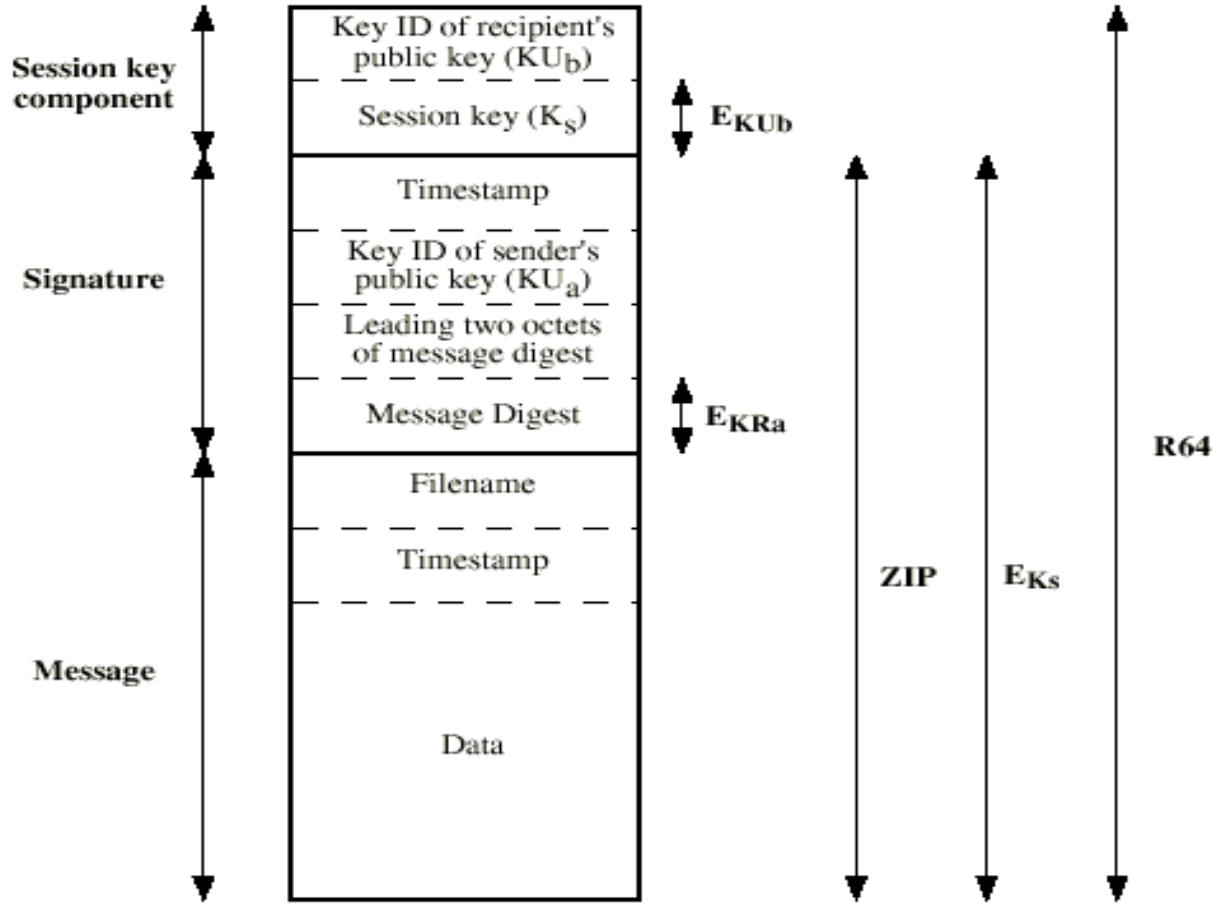
Radix 64 Conversion

6 bit value	Character encoding	6 bit value	Character encoding	6 bit value	Character encoding	6 bit value	Character encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

Example

- Suppose the email message is: new
- ASCII format: 01101110 01100101 01110111
- After encryption: 10010001 10011010 10001000
- The Radix-64 conversion:
  - The 24-bit block: 10010001 10011010 10001000
  - Four 6-bit blocks: 100100 011001 101010 001000
  - Integer version: 36 25 38 8
  - Printable version: k Z m I

Format of PGP Message



Key Rings

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

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

* = field used to index table

Figure 5.4 General Structure of Private and Public Key Rings

16.2.3 PGP Certificates

X.509 Certificates

Protocols that use X.509 certificates depend on the hierarchical structure of the trust.

Note

In X.509, there is a single path from the fully trusted authority to any certificate.

PGP Certificates

In PGP, there is no need for CAs; anyone in the ring can sign a certificate for anyone else in the ring.

Note

In PGP, there can be multiple paths from fully or partially trusted authorities to any subject.

Trusts and Legitimacy

The entire operation of PGP is based on introducer trust, the certificate trust, and the legitimacy of the public keys.

16.2.3 Continued

Trust & Legitimacy

Introducer trust: It specifies the trust levels issued by introducer for other people in the ring. There are three levels for introducer trust- fully, partial & none.

Certificate trust: Certificate trust is normally the same as the introducer trust.

Key legitimacy: The purpose of using introducer and certificate trusts is to determine the legitimacy of a public key.

Figure 16.7 *Format of private key ring table*

User ID	Key ID	Public key	Encrypted private key	Timestamp
⋮	⋮	⋮	⋮	⋮

16.2.3 Continued

Example 16.1

Let us show a private key ring table for Alice. We assume that Alice has only two user IDs, alice@some.com and alice@anet.net. We also assume that Alice has two sets of private/public keys, one for each user ID.

Table 16.5 *Private key ring table for Example 1*

User ID	*Key ID*	*Public Key*	*Encrypted Private Key*	*Timestamp*
alice@anet.net	AB13...45	AB13...45...59	**32452398...23**	031505-16:23
alice@some.com	FA23...12	FA23...12...22	**564A4923...23**	031504-08:11

16.2.3 Continued

Figure 16.8 *Format of a public key ring table*

Public ring

User ID	Key ID	Public key	Producer trust	Certificate(s)	Certificate trust(s)	Key Legitimacy	Timestamp
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

16.2.3 Continued

Example 16.2

A series of steps will show how a public key ring table is formed for Alice.

Table 16.6 *Example 2, starting table*

User ID	*Key ID*	*Public key*	*Prod. trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....

Table 16.7 *Example 2, after Bob is added to the table*

User ID	*Key ID*	*Public key*	*Prod. trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....
Bob...	12...	12.....	F			F	.....

16.2.3 Continued

Example 16.2 Continued

Table 16.8 *Example 2, after Ted is added to the table*

User ID	*Key ID*	*Public key*	*Prod. trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....
Bob...	12...	12.....	F			F	.....
Ted...	48...	48.....	F	Bob's	F	F	.....

Table 16.9 *Example 2, after Anne is added to the table*

User ID	*Key ID*	*Public key*	*Prod. trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....
Bob...	12...	12.....	F			F	.....
Ted...	48...	48.....	F	Bob's	F	F	.....
Anne...	71...	71.....	P	Bob's	F	F	.....

16.2.3 Continued

Example 16.2 Continued

Table 16.10 *Example 2, after John is added to the table*

User ID	*Key ID*	*Public key*	*Prod. Trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....
Bob...	12...	12.....	F			F	.....
Ted...	48...	48.....	F	Bob's	F	F	.....
Anne...	71...	71.....	P	Bob's	F	F	.....
John...	31...	31.....	N	Anne's	P	P	.....

16.2.3 Continued

Example 16.2 Continued

Table 16.11 *Example 2, after one more certificate received for John*

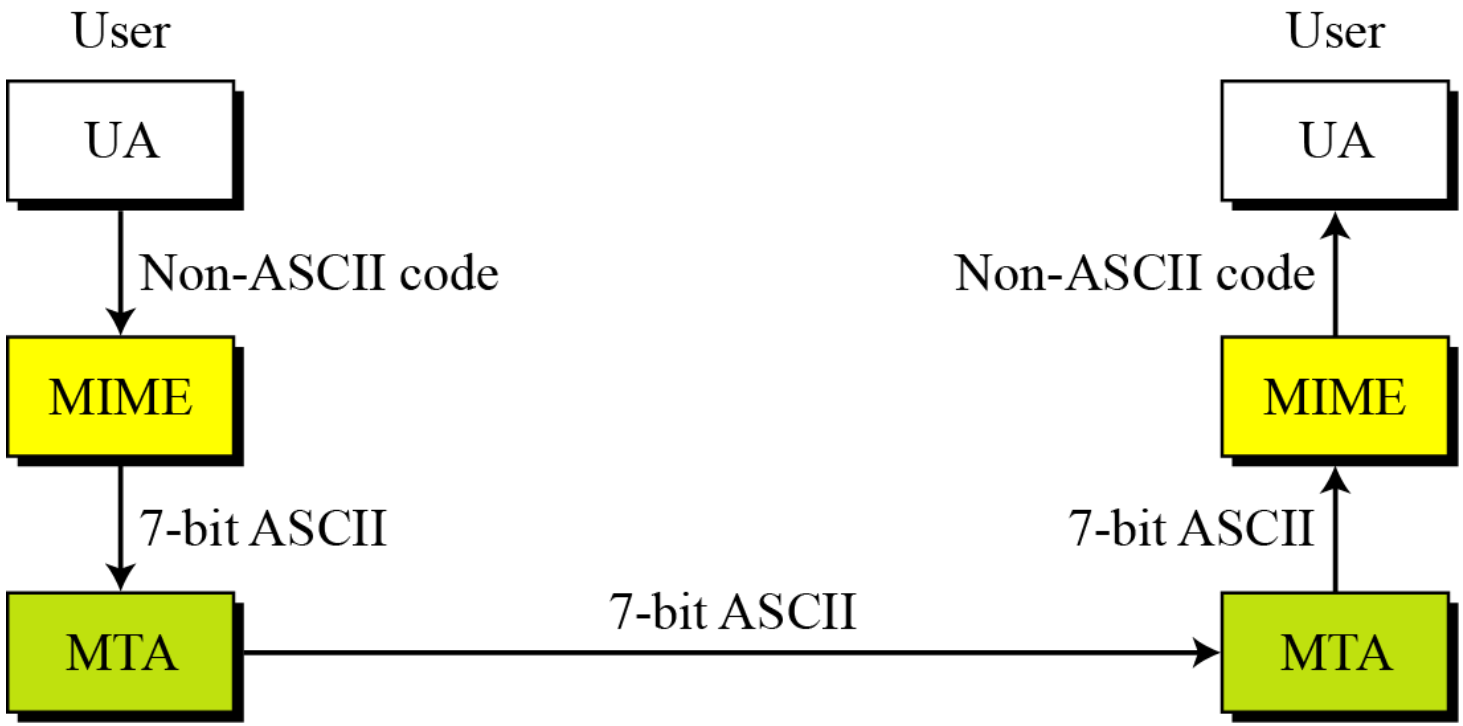
User ID	*Key ID*	*Public key*	*Prod. trust*	*Certificate*	*Cert. trust*	*Key legit.*	*Time-stamp*
Alice...	AB...	AB.....	F			F	.....
Bob...	12...	12.....	F			F	.....
Ted...	48...	48.....	F	Bob's	F	F	.....
Anne...	71...	71.....	P	Bob's	F	F	.....
John...	31...	31.....	N	Anne's Ted's	P F	F	.....

16-3 S/MIME

Another security service designed for electronic mail is Secure/Multipurpose Internet Mail Extension (S/MIME). The protocol is an enhancement of the Multipurpose Internet Mail Extension (MIME) protocol.

Topics discussed in this section:

Figure 16.23 *MIME*



Multipurpose Internet Mail Extensions (MIME)

- In 1992, a new standard was defined by an Internet Engineering Task Force Working Group - called MIME.
- MIME is a specification for enhancing the capabilities of standard Internet electronic mail.

When using the MIME standard, messages can contain the following types:

- Text messages in US-ASCII.
- Character sets other than US-ASCII.
- Multi-media: Image, Audio, and Video messages.
- Multiple objects in a single message.
- Multi-font messages.
- Messages of unlimited length.
- Binary files.

- MIME is defined to be completely backwards compatible, yet flexible and open to extensions. Therefore, it builds on the older standard by defining additional fields for the mail message header, that describes new content types, and a distinct organization of the message body.

Background

- SMPT (Simple Mail Transfer Protocol) is widely used around the world, it is the standard protocol for transferring mail between hosts in the TCP/IP suite.

- However, SMPT has been limited to the delivery of simple text messages which does not meet the rising demand for capability of delivery mail containing various types of data, including voice, images and video clips.
- To satisfy this requirement, a new electronic mail standard, which builds on SMPT, has been defined.

Limitations of the SMPT scheme

- The message may contain only US-ASCII characters
- The maximum line length allowed is 1000 characters
- The message must not be longer than a predefined maximum size
  - Cannot transmit executable files or other binary objects.

Limitations of the SMPT scheme ....contd

- It cannot transmit text data that includes national language characters (8-bit codes) because it is limited to 7-bit ASCII.
- SMPT servers may reject mail message over a certain size.
- SMPT gateways that translate between ASCII and the Character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.

MIME is compatible with existing implementations

- MIME is intended to resolve these problems in a manner that is compatible with exiting implementations.
  - A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.
  - Transfer encodings are defined that enable the conversion of any content format that is protected from alteration by the mail system.

Technical Specifications

- It explicitly describes the set of allowable *Content-types*.
  1. *Text* - Used to represent textual information.
  2. *Image* - this type is for transmitting still images.
  3. *Audio* - this content type is for transmitting audio or voice data..
  4. *Video* - The Video content type is for transmission of video data or moving image data.

Technical Specifications . .contd

- MIME *encapsulates* binary data in ASCII mail envelope.
- *Multipart* - Used to combine several body parts of possibly different types & subtypes.
- *Application* - Can be used to transmit application data (such as executables) or binary data.

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)
  - SMTP cannot transmit executable files or other binary objects.
  - SMTP cannot transmit text data that includes national language characters because there are represented by 8-bit codes and SMTP is restricted to 7-bit ASCII.
  - SMTP servers may reject mail messages over a certain size.
  - MIME is intended to resolve these problems that is compatible with RFC 5322 implementation.

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 19.3
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

- 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

S/MIME

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

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.

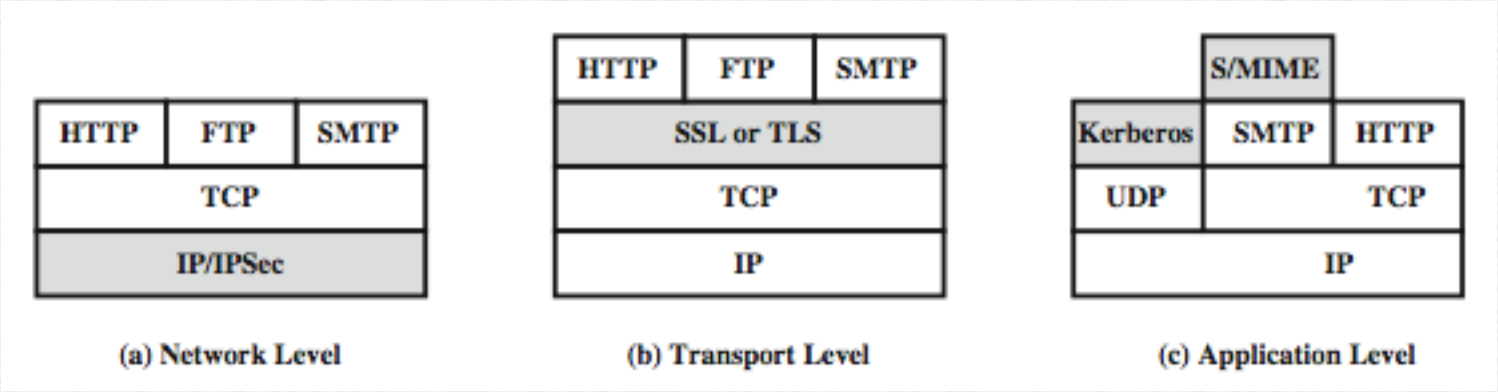
Table 19.5

Cryptographic Algorithms Used in S/MIME

Web Security

- Web now widely used by business, government, individuals
- but Internet & Web are vulnerable
- have a variety of threats
  - integrity
  - confidentiality
  - denial of service
  - authentication
- need added security mechanisms

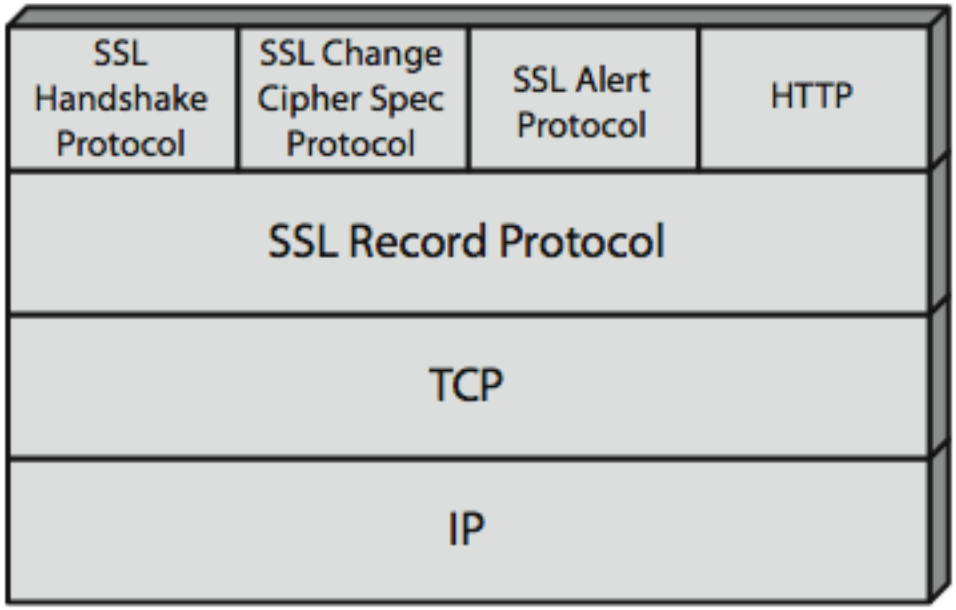
Web Traffic Security Approaches



SSL (Secure Socket Layer)

- transport layer security service
- originally developed by Netscape
- version 3 designed with public input
- subsequently became Internet standard known as TLS (Transport Layer Security)
- uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



SSL Architecture

- **SSL connection**
  - a transient, peer-to-peer, communications link
  - associated with 1 SSL session
- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections

SSL Architecture

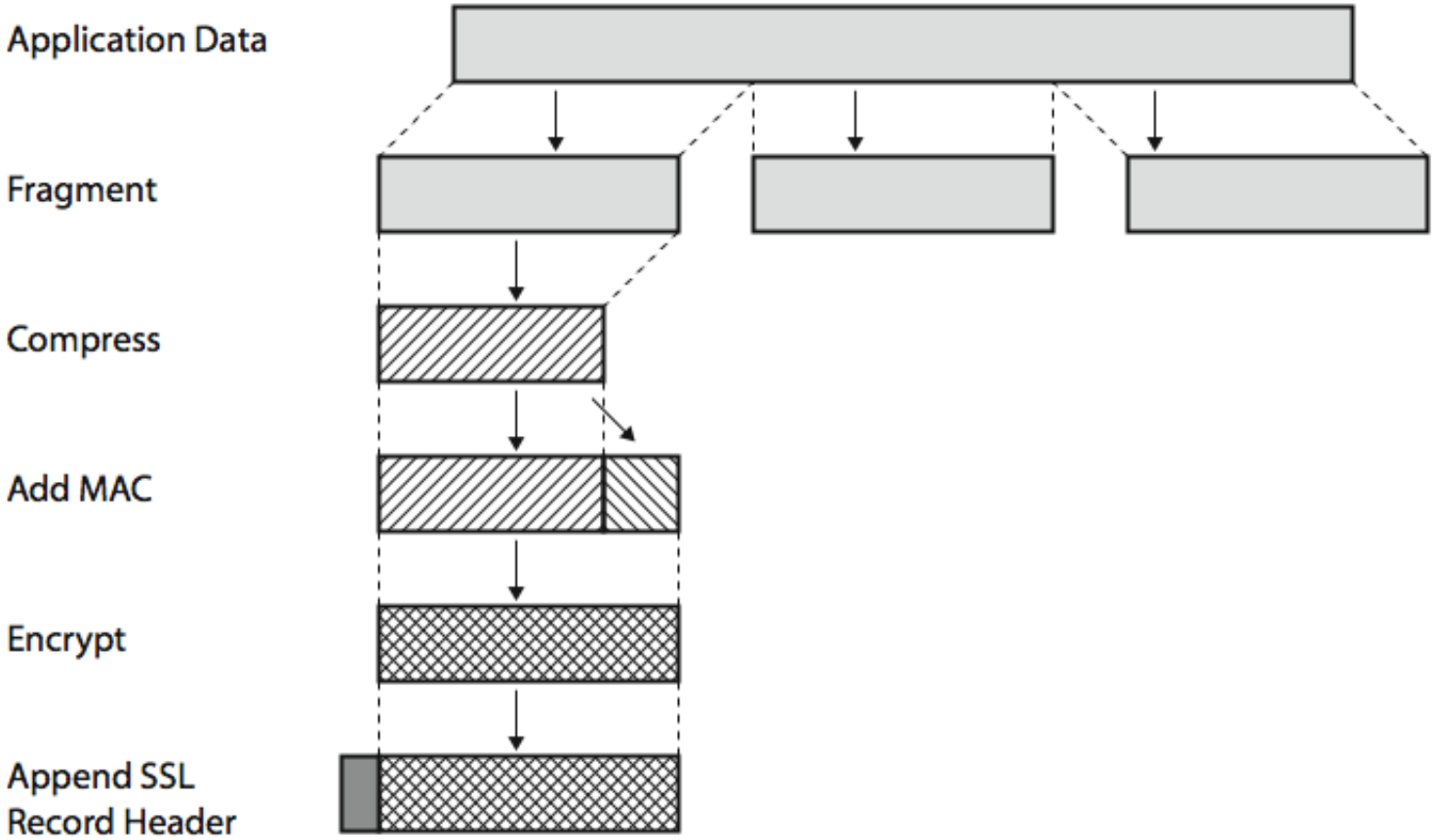
Session state is defined by

- Session Identifier
- Peer Certificate
- Compression methods
- Cipher Spec
- Master Secret
- Is Resumable

Connection state is defined by

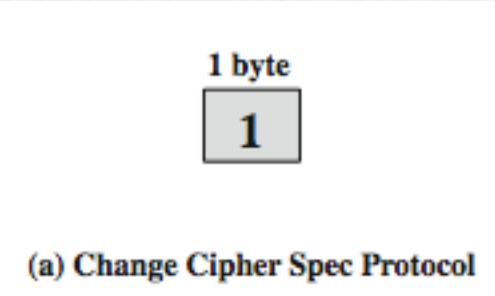
- Server and Client Random
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key
- Initialization vectors

SSL Record Protocol Operation



SSL Change Cipher Spec Protocol

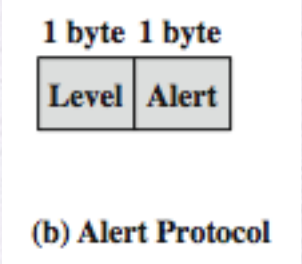
- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use



(a) Change Cipher Spec Protocol

SSL Alert Protocol

- conveys SSL-related alerts to peer entity
- severity
  - warning or fatal
- specific alert
  - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- compressed & encrypted like all SSL data

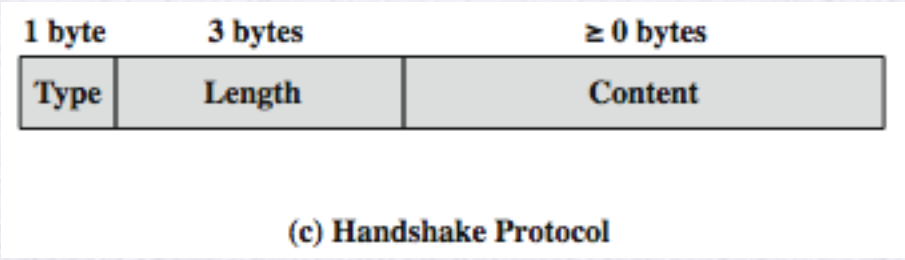


(b) Alert Protocol

SSL Handshake Protocol

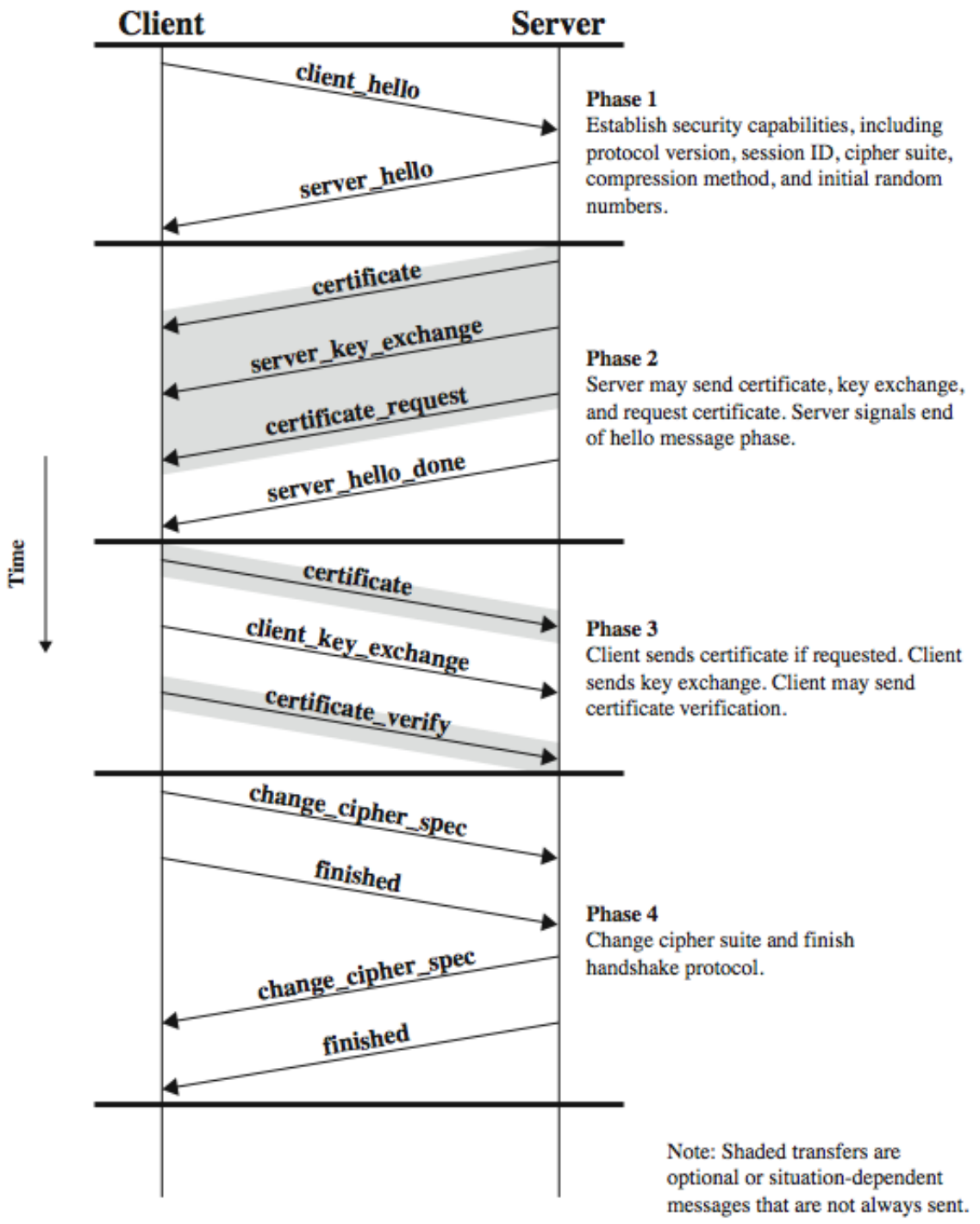
- allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used

- comprises a series of messages in phases
  1. Establish Security Capabilities
  2. Server Authentication and Key Exchange
  3. Client Authentication and Key Exchange
  4. Finish



(c) Handshake Protocol

SSL Handshake Protocol



Cryptographic Computations

- master secret creation
  - a one-time 48-byte value
  - generated using secure key exchange (RSA / Diffie-Hellman) and then hashing info
- generation of cryptographic parameters
  - client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV
  - generated by hashing master secret

Calculation of key material from Master Secret

Figure 17.9 *Calculation of key material from master secret*

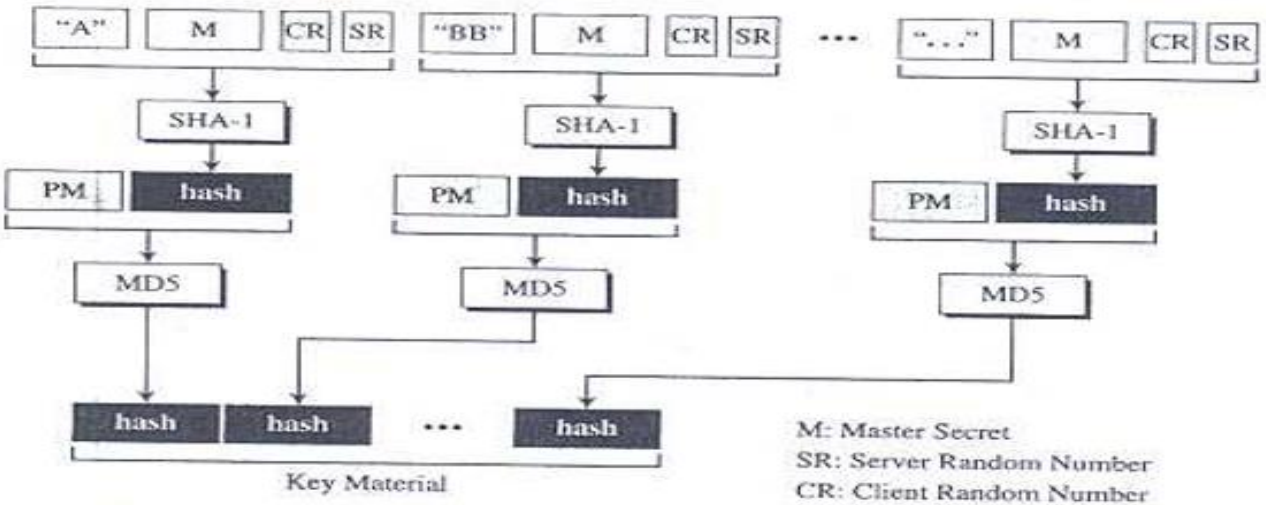
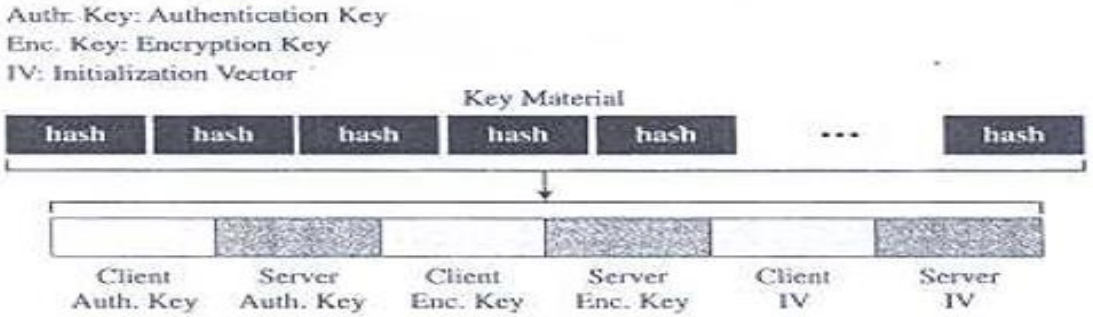


Figure 17.10 *Extractions of cryptographic secrets from key material*



IPSec

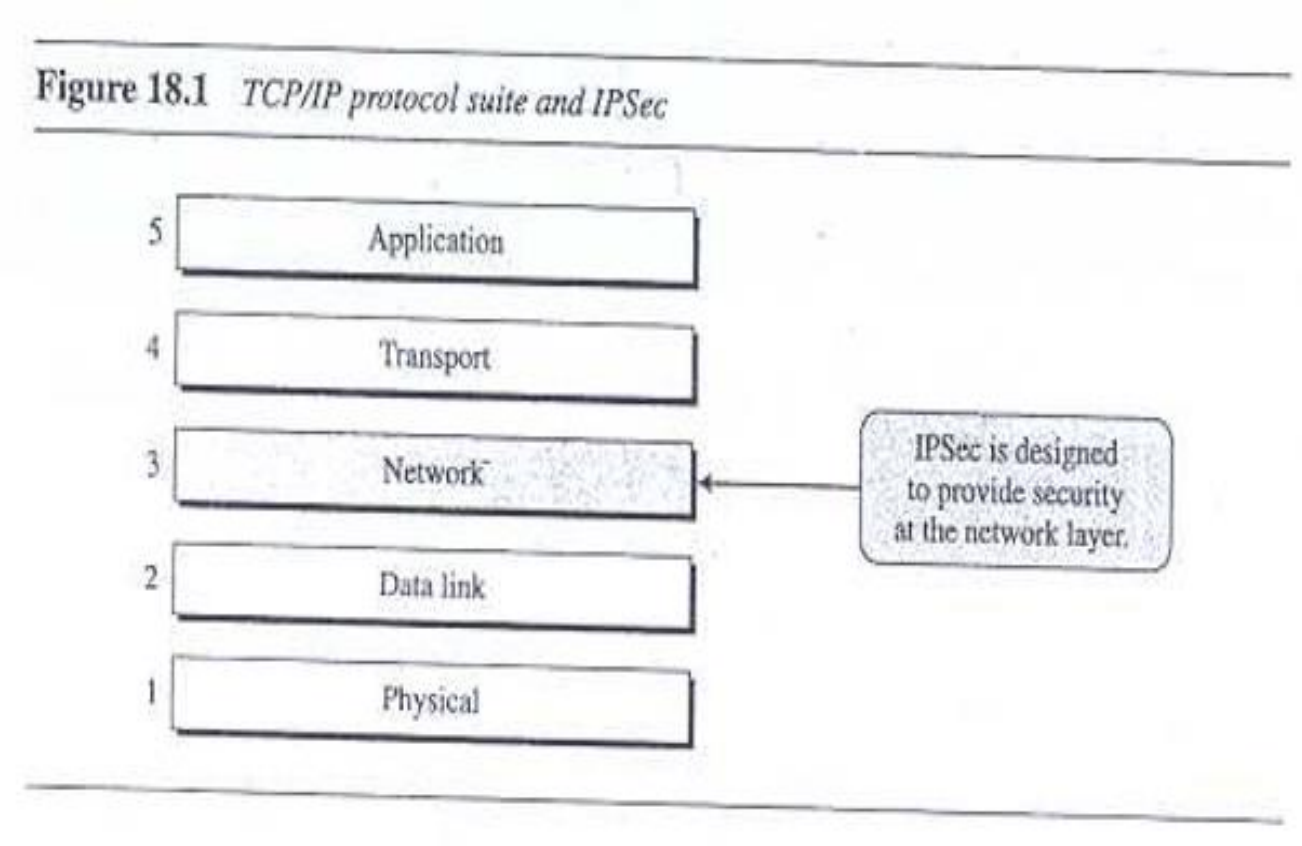


Figure 18.1 TCP/IP protocol suite and IPSec
5
Application
4
Transport
3
Network
2
Data link
1
Physical
IPSec is designed to provide security at the network layer.

Transport Mode

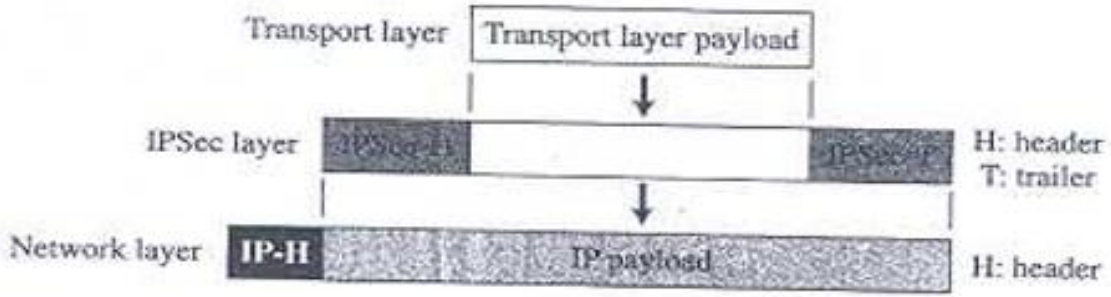
18.1 TWO MODES

IPSec operates in one of two different modes: transport mode or tunnel mode.

Transport Mode

In **transport mode,** IPSec protects what is delivered from the transport layer to the network layer. In other words, transport mode protects the network layer payload, the payload to be encapsulated in the network layer, as shown in Figure 18.2.

Figure 18.2 *IPSec in transport mode*

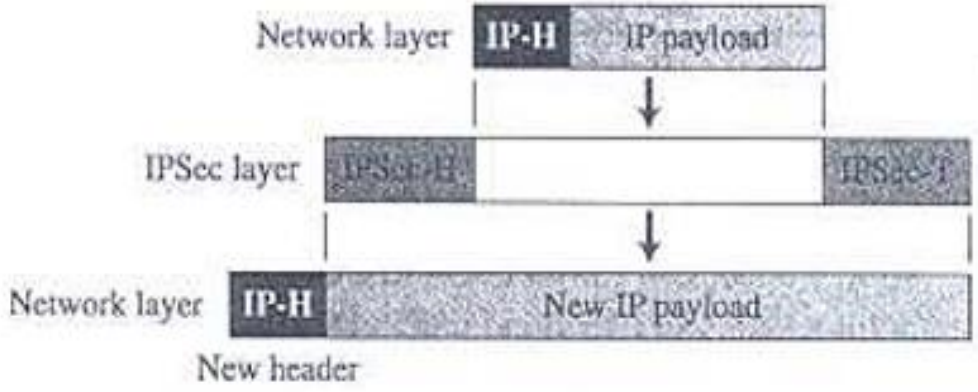


Tunnel Mode

Tunnel Mode

In **tunnel mode,** IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header, as shown in Figure 18.4.

Figure 18.4 *IPSec in tunnel mode*

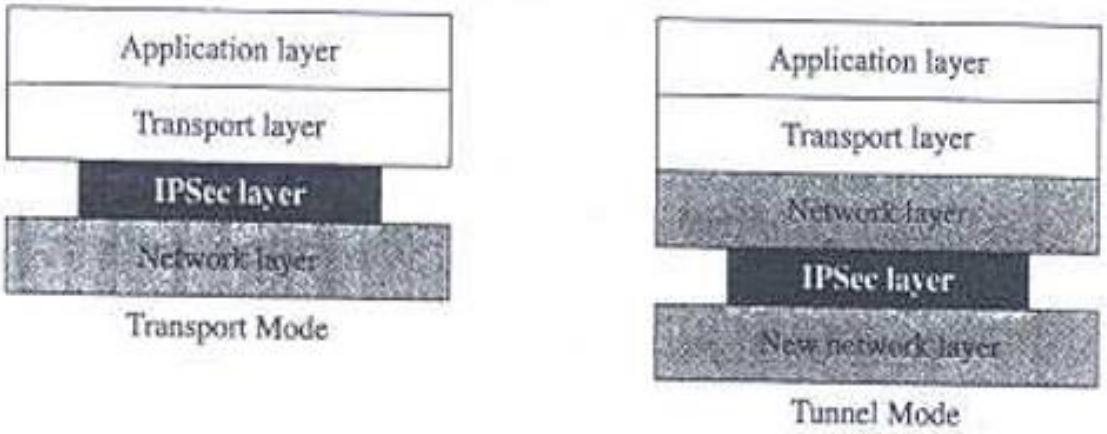


Comparison

Comparison

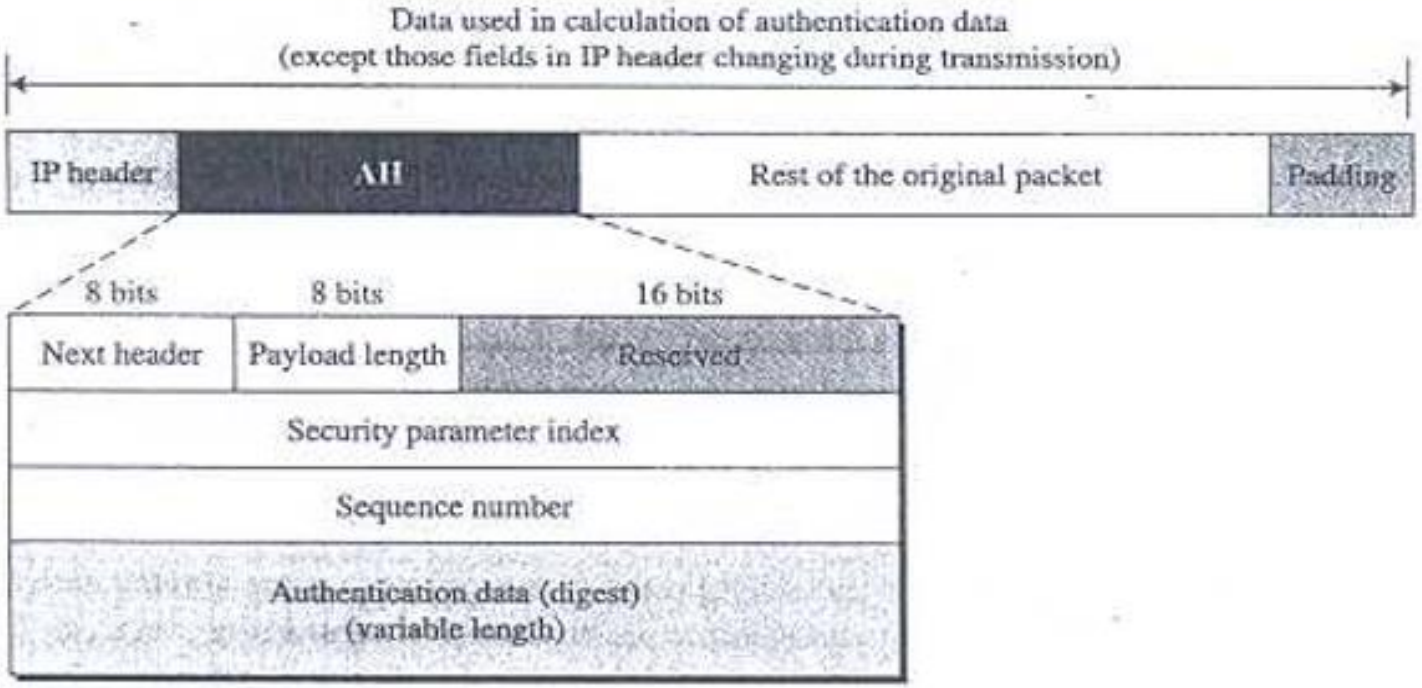
In transport mode, the IPSec layer comes between the transport layer and the network layer. In tunnel mode, the flow is from the network layer to the IPSec layer and then back to the network layer again. Figure 18.6 compares the two modes.

Figure 18.6 *Transport mode versus tunnel mode*



Two Security Protocols

Figure 18.7 *Authentication Header (AH) protocol*



ESP

Encapsulating Security Payload (ESP)

The AH protocol does not provide privacy, only source authentication and data integrity. IPSec later defined an alternative protocol, **Encapsulating Security Payload (ESP),** that provides source authentication, integrity, and privacy. ESP adds a header and trailer. Note that ESP's authentication data are added at the end of the packet, which makes its calculation easier. Figure 18.8 shows the location of the ESP header and trailer.

Figure 18.8 *ESP*