Data Communications and Networking

Textbook
William Stallings, Data and Computer Communications, 6e

Chapter 18 Network Security

Yang Xianchun
Department of Computer Science and Technology
Nanjing University

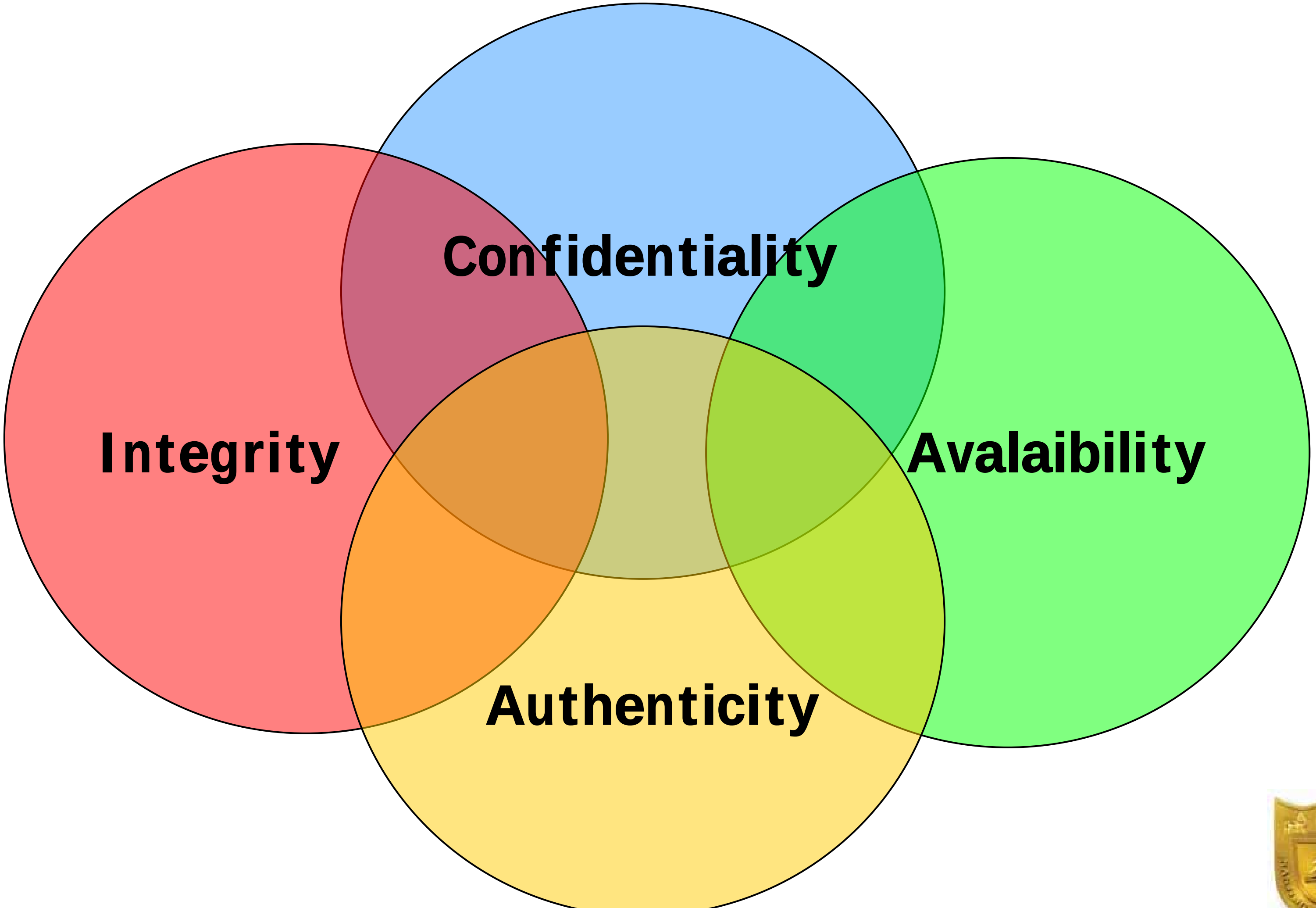
18.1 Security requirements and attacks

- Attacks, services and mechanisms
- Security requirements and goals
- Security threads
- Types of security attacks
- Categorization of security attacks
- security services
- Methods of defense
- A model for network security
- Network access security model

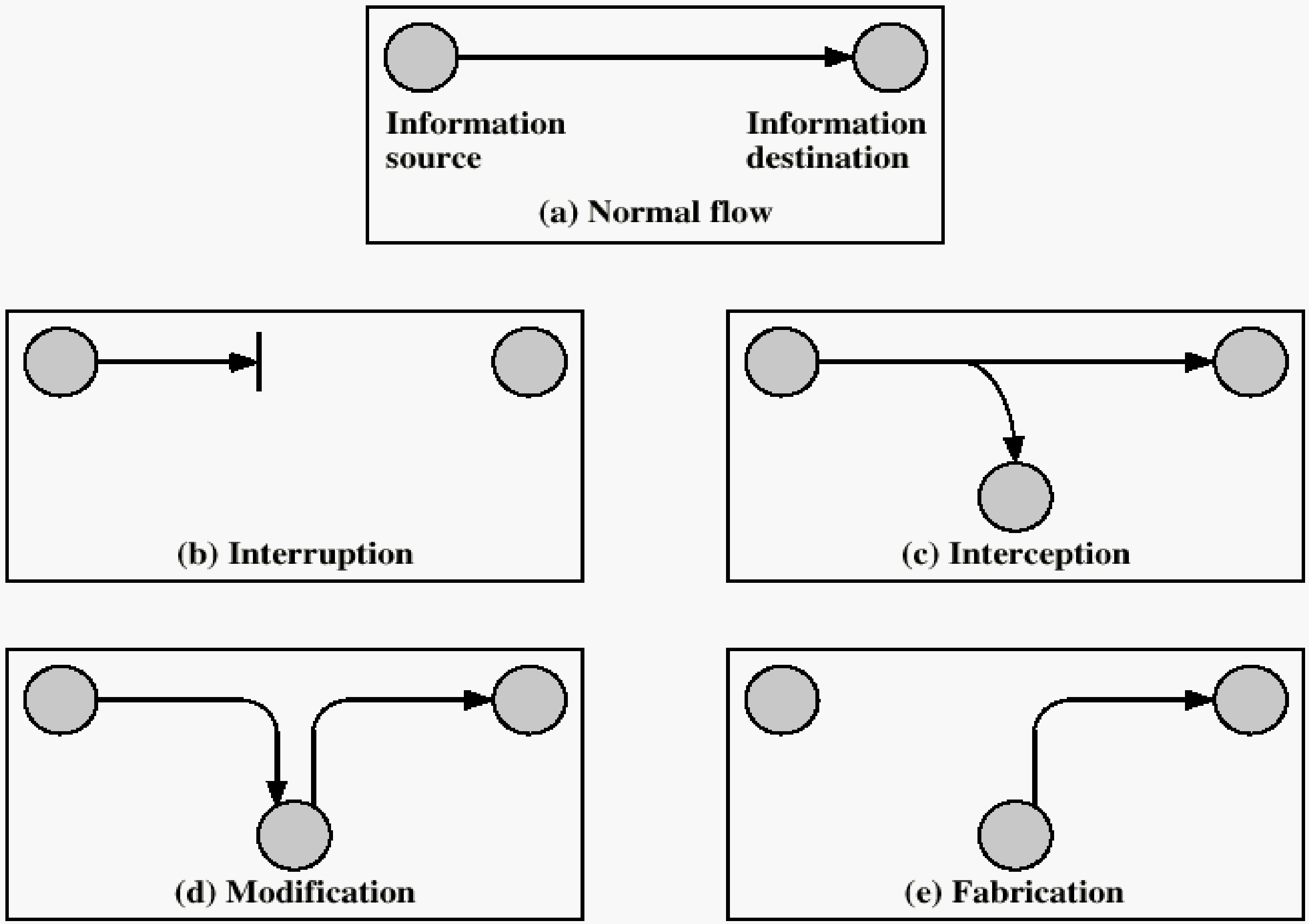
Attacks, Services and Mechanisms

- **Security Attack:** Any action that compromises the security of information.
- **Security Mechanism:** A mechanism that is designed to detect, prevent, or recover from a security attack.
- **Security Service:** A service that enhances the security of data processing systems and information transfers. A security service makes use of one or more security mechanisms.

Security Requirements and Goals



Security Threats

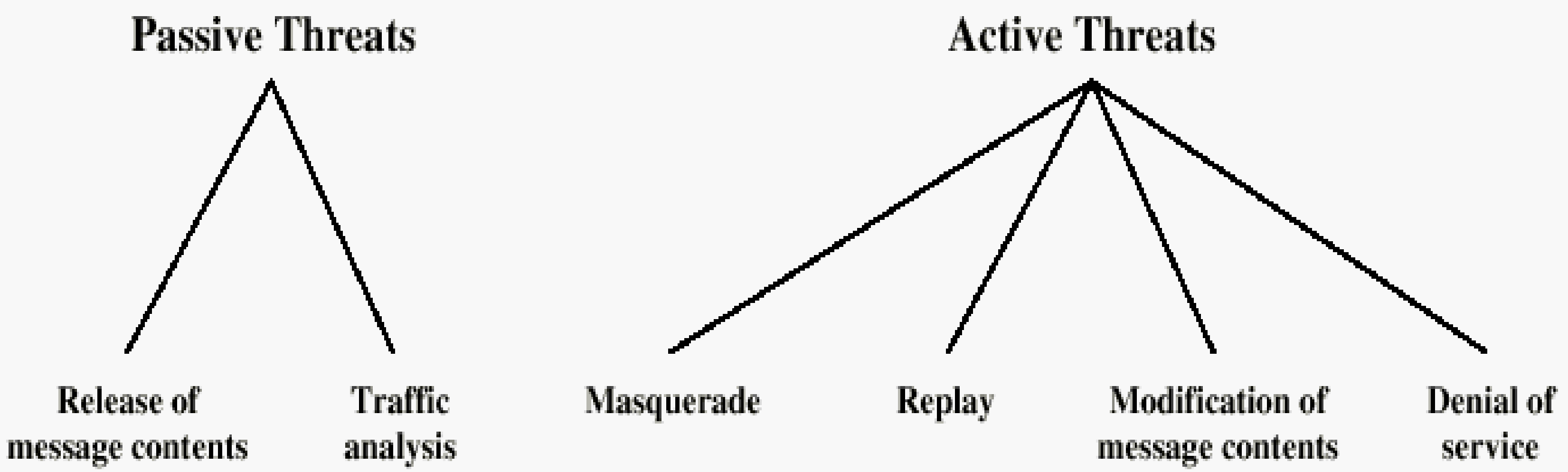


Security Threats

Types of Security Attacks

- **Interruption:** This is an attack on availability
  - Denial of service
  - Virus that deletes files
- **Interception:** an attack on confidentiality
  - Release of message contents
  - Traffic analysis
- **Modification:** This is an attack on integrity
  - Modification of message contents
- **Fabrication:** This is an attack on authenticity
  - Masquerade
  - Replay

Categorization of Security Attacks



Active and Passive Security Threats

Passive Attacks

- Eavesdropping on transmissions
- To obtain information
- Release of message contents
  - Outsider learns content of transmission
- Traffic analysis
  - By monitoring frequency and length of messages, even encrypted, nature of communication may be guessed
- Difficult to detect
- Can be prevented

Passive Attackers

- Sniffer
- Wiretap
- Tempest
- Dumpster diving

Active Attacks

- Masquerade
  - Pretending to be a different entity
- Replay
  - Intercept and capture, then retransmit it
- Modification of messages
- Denial of service
- Easy to detect
  - Detection may lead to deterrent
- Hard to prevent

Active Attackers

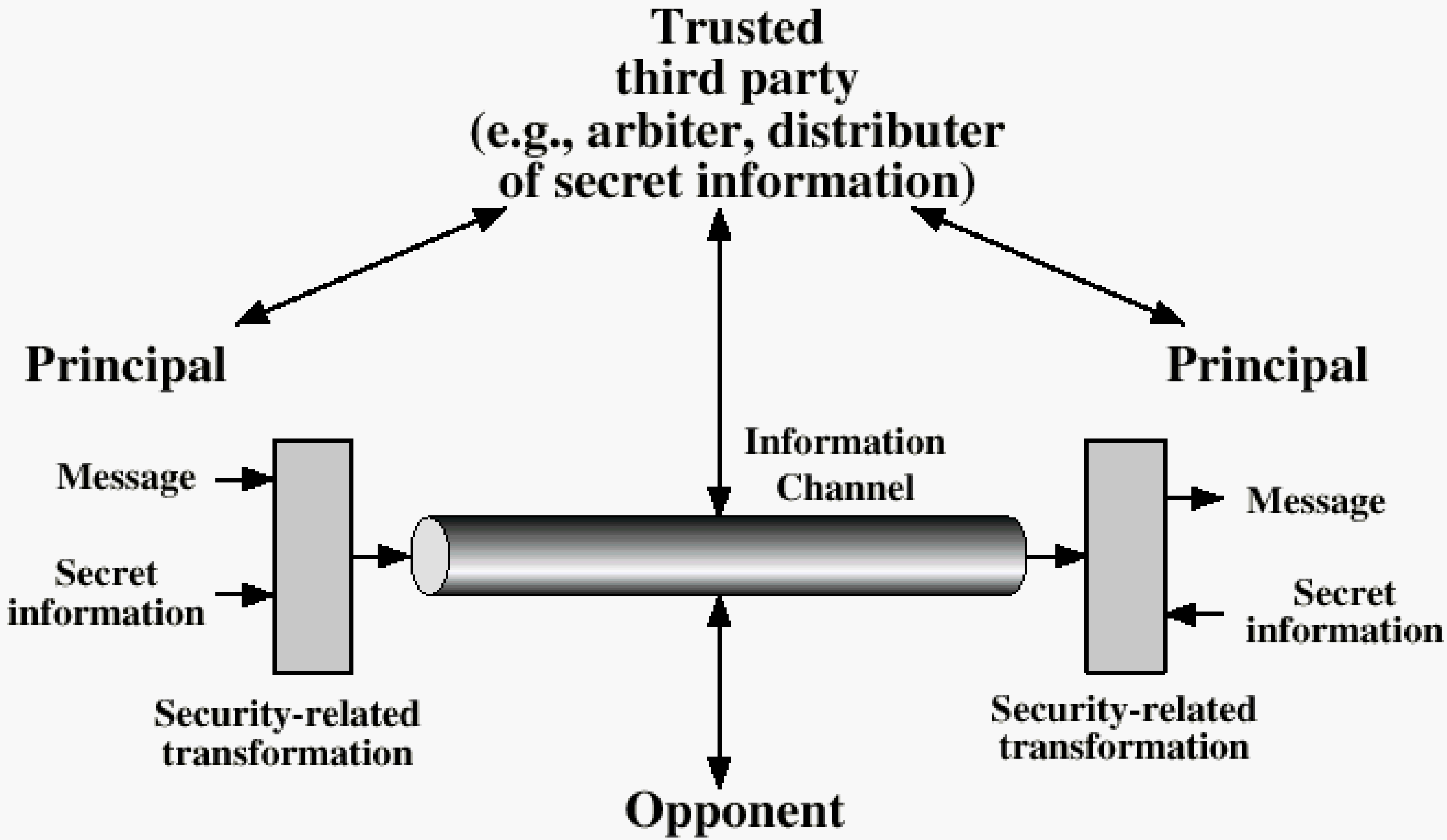
- Intruders
  - Hackers
  - Crackers
  - Cyberpunker
- Rogue Programs
  - Computer virus
  - Computer worm
  - Trojan horse
  - Trapdoor
  - Logic bomb

Security Services

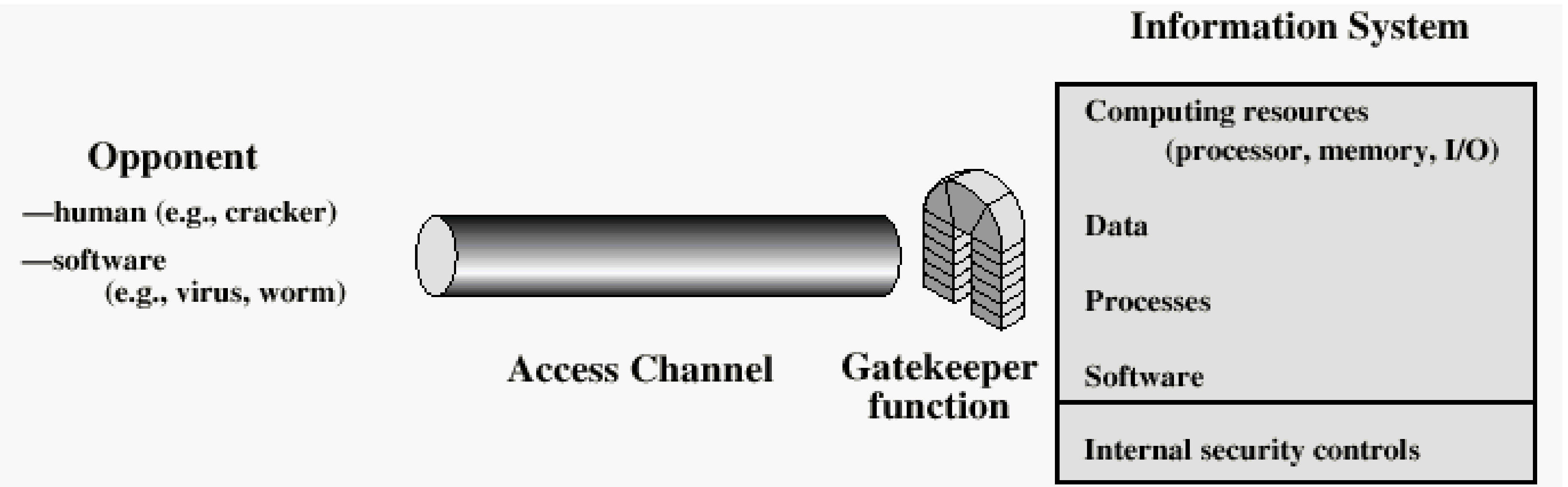
- ⌘ Confidentiality (privacy)
- ⌘ Authentication (who created or sent the data)
- ⌘ Integrity (has not been altered)
- ⌘ Non-repudiation (the order is final)
- ⌘ Access control (prevent misuse of resources)
- ⌘ Availability (permanence, non-erasure)

Methods of Defense

- **Encryption**
- Software Controls (access limitations in a data base, in operating system protect each user from other users)
- Hardware Controls (smartcard)
- Policies (frequent changes of passwords)
- Physical Controls



Model for Network Security



Network Access Security Model

Recommended Reading

- Pfleeger, C. *Security in Computing.* Prentice Hall, 1997.

- Mel, H.X. Baker, D. *Cryptography Decrypted.* Addison Wesley, 2001.

18.2 Conventional Encryption

- Cryptography
- Conventional encryption principles
- Classical encryption algorithms
- Feistel Cipher Structure
- DES(Data Encryption Standard) algorithm
- Strength of DES
- Triple DEA algorithm
- Cipher Block Modes of Operation
- Location of encryption devices
- Key distribution
- Traffic padding

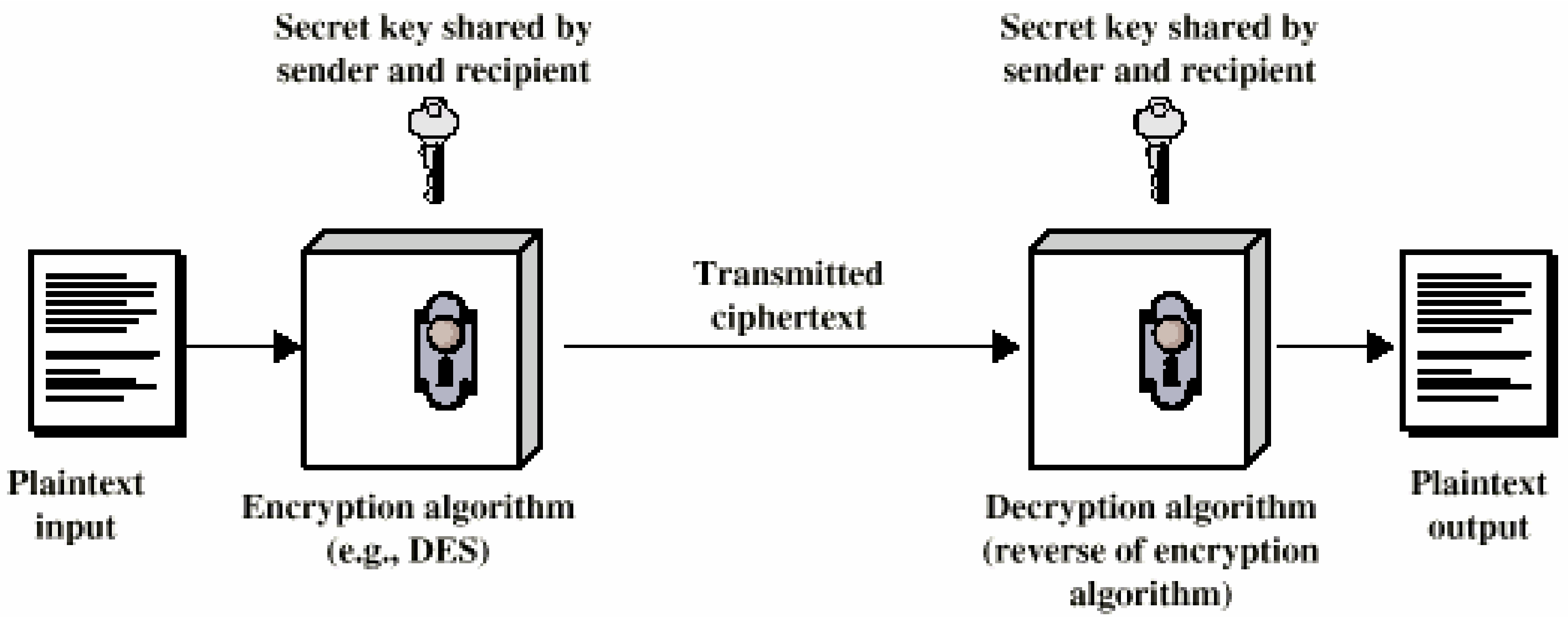
Cryptography

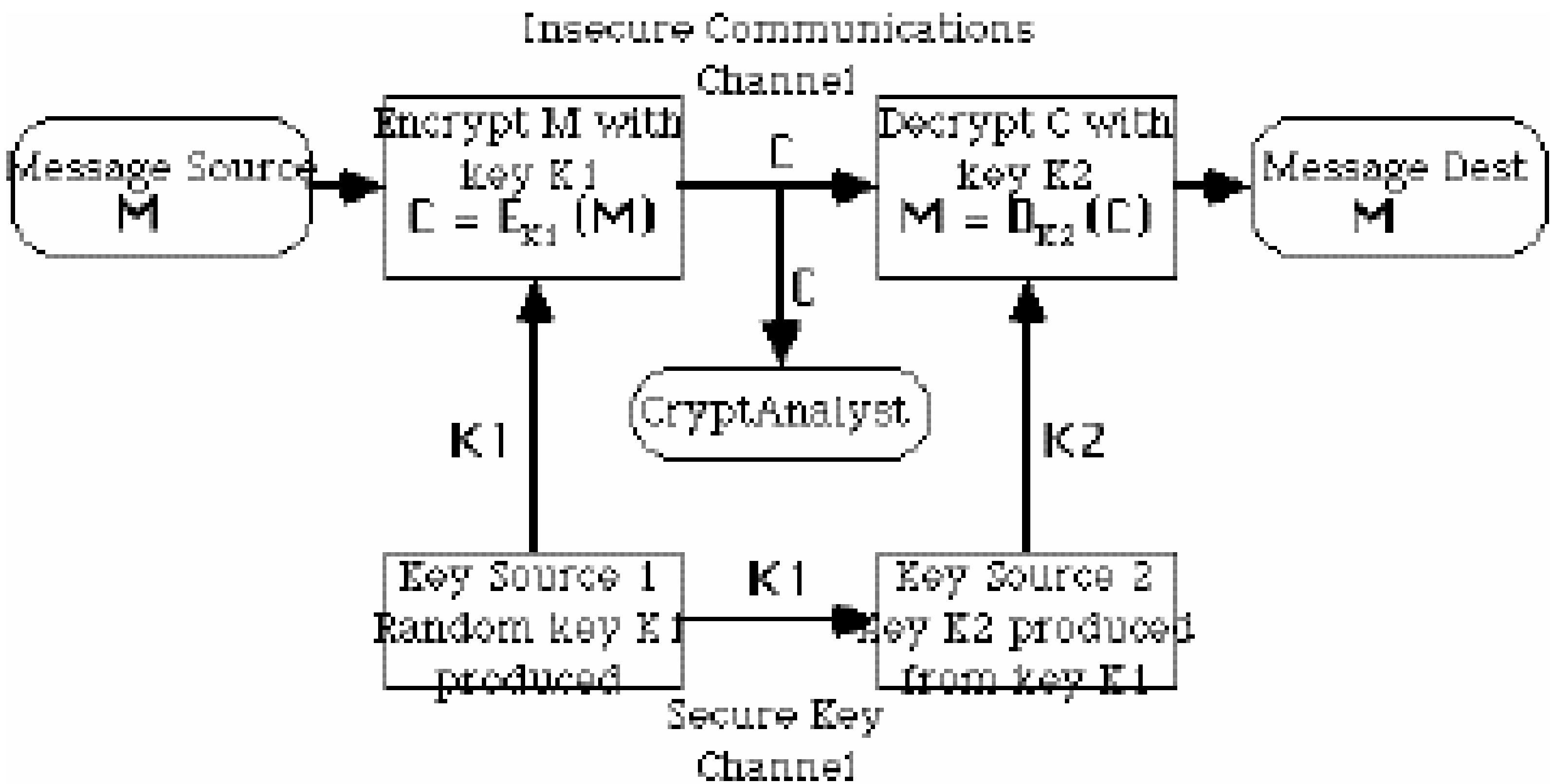
- Classified along three independent dimensions:
  - The type of operations used for transforming plaintext to ciphertext
    - substitution
    - transposition
    - product (multiple substitution and transposition)
  - The number of keys used
    - symmetric (single key)
    - asymmetric (two-keys, or public-key encryption)
  - The way in which the plaintext is processed
    - stream
    - block

Conventional Encryption Principles

- An encryption scheme has five ingredients:
  - Plaintext
  - Encryption algorithm
  - Secret Key
  - Ciphertext
  - Decryption algorithm
- Security depends on the secrecy of the key, not the secrecy of the algorithm

A Simplified Model of Conventional Encryption





Symmetric (Private-Key) Encryption System

Requirements for Security

- Strong encryption algorithm
  - Even if known, should not be able to decrypt or work out key
  - Even if a number of cipher texts are available together with plain texts of them
- Sender and receiver must obtain secret key securely
  - Once key is known, all communication using this key is readable

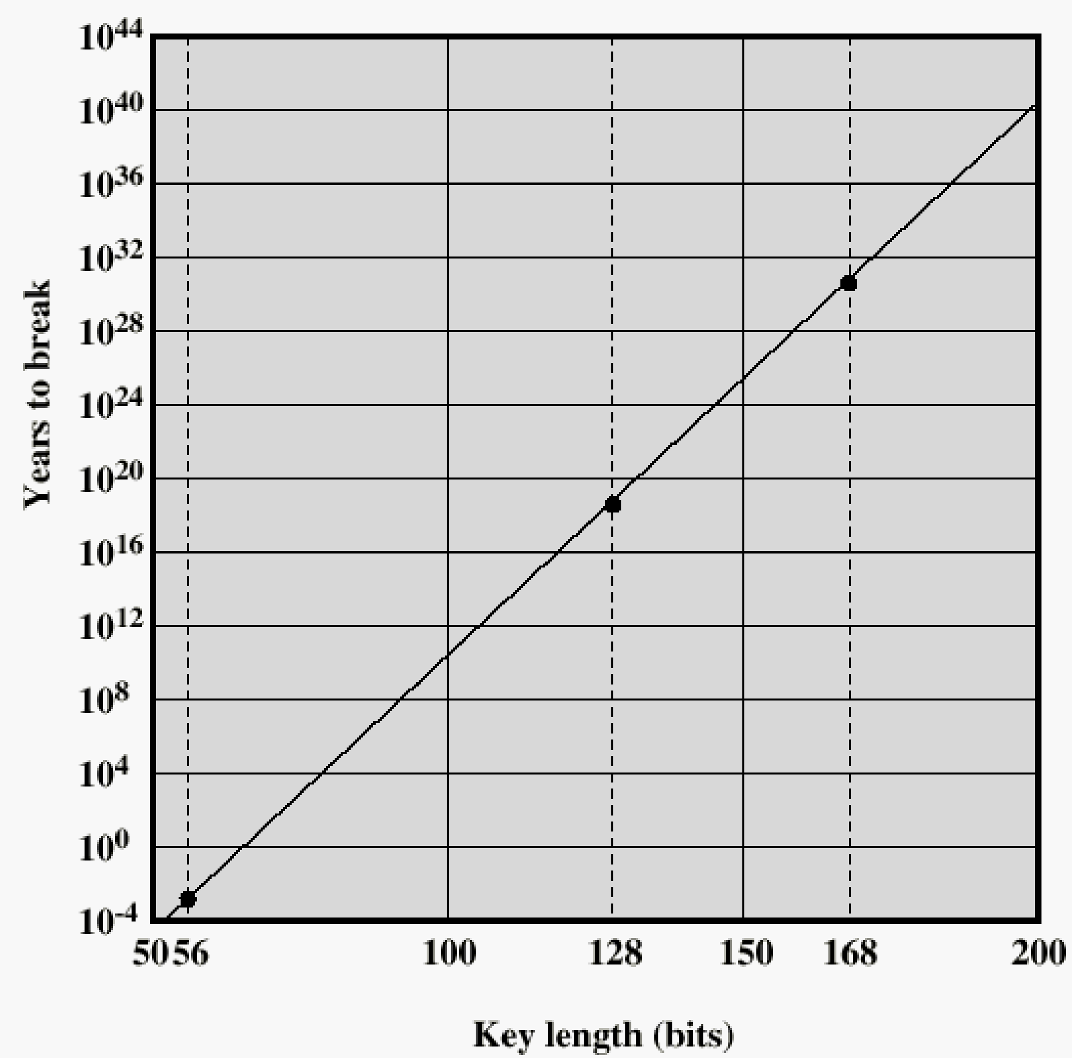
Attacking Encryption

- Crypt analysis
  - Relay on nature of algorithm plus some knowledge of general characteristics of plain text
  - Attempt to deduce plain text or key
- Brute force
  - Try every possible key until plain text is achieved

Average time required for exhaustive key search

Key Size (bits)	**Number of Alternative Keys**	**Time required at 10^6 Decryption/µs**
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Time to break a code (10^6 decryptions/µs)



Classical Encryption Algorithms (1)

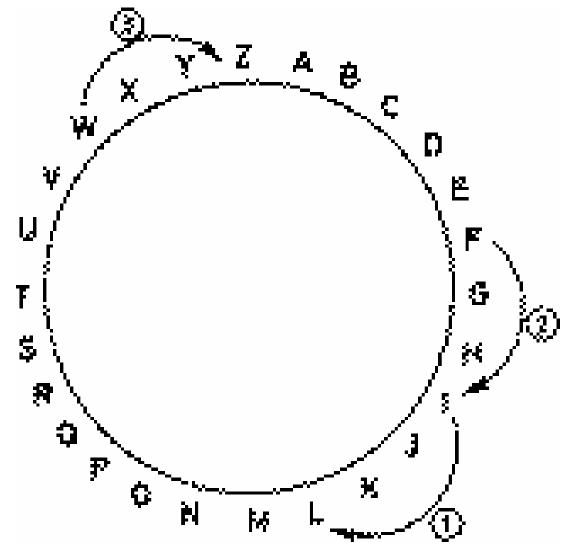
- Substitution ciphers
  - Caesar cipher
  - Playfair cipher Charles Wheatstone,1854
    - http://www.pbs.org/wgbh/nova/decoding/playfair.html
    - http://www.pbs.org/wgbh/nova/decoding/playfair2.html
  - Hill cipher Lester Hill, 1929
    - http://home.ecn.ab.ca/~jsavard/crypto/ro020103.htm
  - Vigenère cipher Blaise de Vigenère, 1586
    - http://www.metaweb.com/wiki/wiki.phtml?title=The_Vigen%E8re_Cipher_(Talith)

Caesar Cipher

- a monoalphabetic cipher
  - replace each letter of message by a letter a fixed distance away eg use the 3rd letter on
  - reputedly used by Julius Caesar
- eg.

  Cipher text: L FDPH L VDZ L FRQTXHUHG

  Plain text: I CAME I SAW I CONQUERED



- ie. mapping is

  ABCDEFGHIJKLMNOPQRSTUVWXYZ

  DEFGHIJKLMNOPQRSTUVWXYZABC
- can describe this cipher as:
  - Encryption E_(k) : i -> i + k mod 26
  - Decryption D_(k) : i -> i - k mod 26

Classical Encryption Algorithms (2)

- Transposition ciphers
  - Scytale cipher
  - Reverse cipher
  - Rail Fence cipher
  - Geometric Figure
  - Row Transposition ciphers
  - Block (Columnar) Transposition ciphers
  - Nihilist ciphers
- Combination of substitution and transposition
  - Product ciphers
  - ADFGVX Product Cipher

Row Transposition ciphers

- can use a word, with letter order giving sequence: to write in the plain text; or read off the cipher
- eg.

Plain: ACONVENIENTWAYTOEXPRESSTHEPERMUTATION

Key (W): **C O M P U T E R**

Order: 1 4 3 5 8 7 2 6

A	**N**	**O**	**V**	**I**	**N**	**C**	**E**
E	**W**	**T**	**A**	**O**	**T**	**N**	**Y**
E	**R**	**P**	**E**	**T**	**S**	**X**	**S**
H	**E**	**P**	**R**	**T**	**U**	**E**	**M**
A	**O**	**I**	**N**	**Z**	**Z**	**T**	**Z**

Cipher: ANOVI NCEEW TAOTN YERPE TSXSH EPRTU EMAOI NZZTZ

Modern Conventional Encryption Algorithms (1)

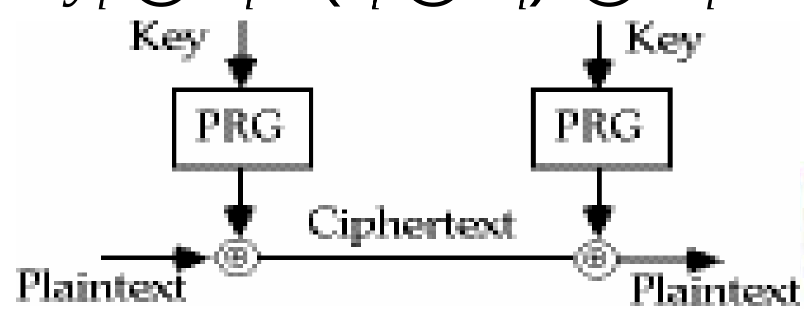
⌘ Stream ciphers

- Process plaintext in sequential bit strame
- Assume plaintext X=$x_1x_2\ldots$, Key sequence K= $k_1k_2\ldots$, Encrypt x_i using k_i, then ciphertext is

  $E_k(X)=E_{k1}(x_1)E_{k2}(x_2) \wedge$
- Vernam cipher (one-time pad) Gilbert Vernam, 1918

  Encryption: $y_i = E_{ki}(x_i) = x_i \oplus k_i$

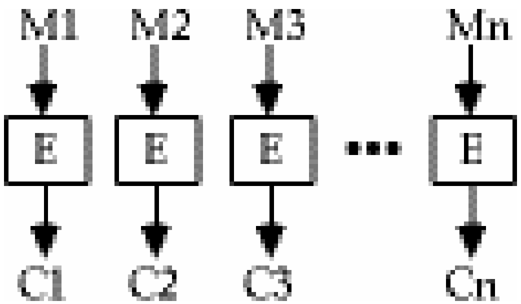
  Decryption: $D_{ki}(y_i) = y_i \oplus k_i = (x_i \oplus k_i) \oplus k_i = x_i$



Modern Conventional Encryption Algorithms (2)

⌘Block ciphers

- Process plain text in fixed block sizes producing block of cipher text of equal size
- Data Encryption Standard (DES)
- International Data Encryption Algorithm (IDEA)
- Triple DES (TDEA)



Feistel Cipher Structure

- Virtually all conventional block encryption algorithms, including DES have a structure first described by Horst Feistel of IBM in 1973
- The realization of a Fesitel Network depends on the choice of the following parameters and design features (see next slide):

Feistel Cipher Structure

- **Block size:** larger block sizes mean greater security
- **Key Size:** larger key size means greater security
- **Number of rounds:** multiple rounds offer increasing security
- **Subkey generation algorithm:** greater complexity will lead to greater difficulty of cryptanalysis.
- **Fast software encryption/decryption:** the speed of execution of the algorithm becomes a concern

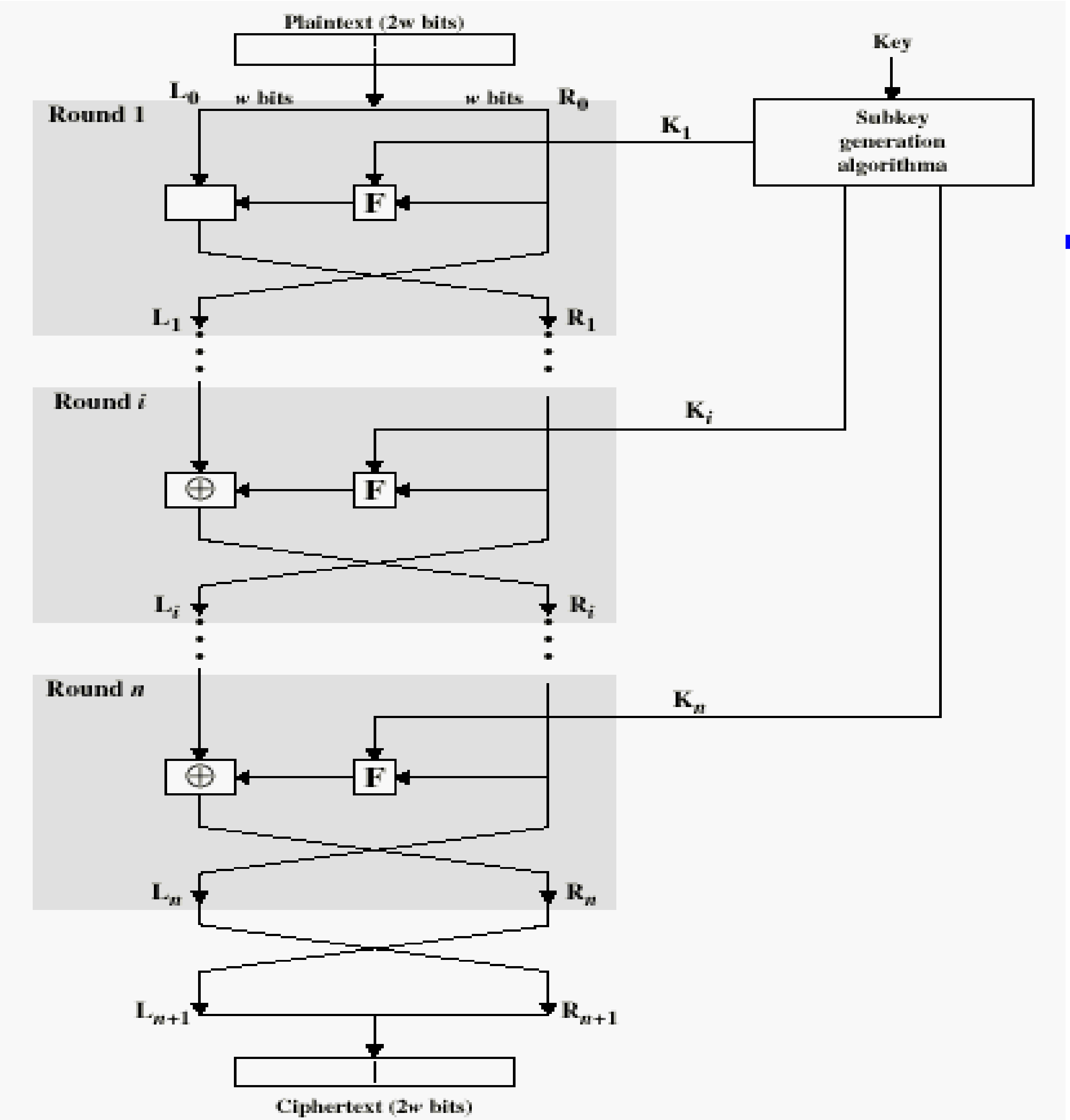
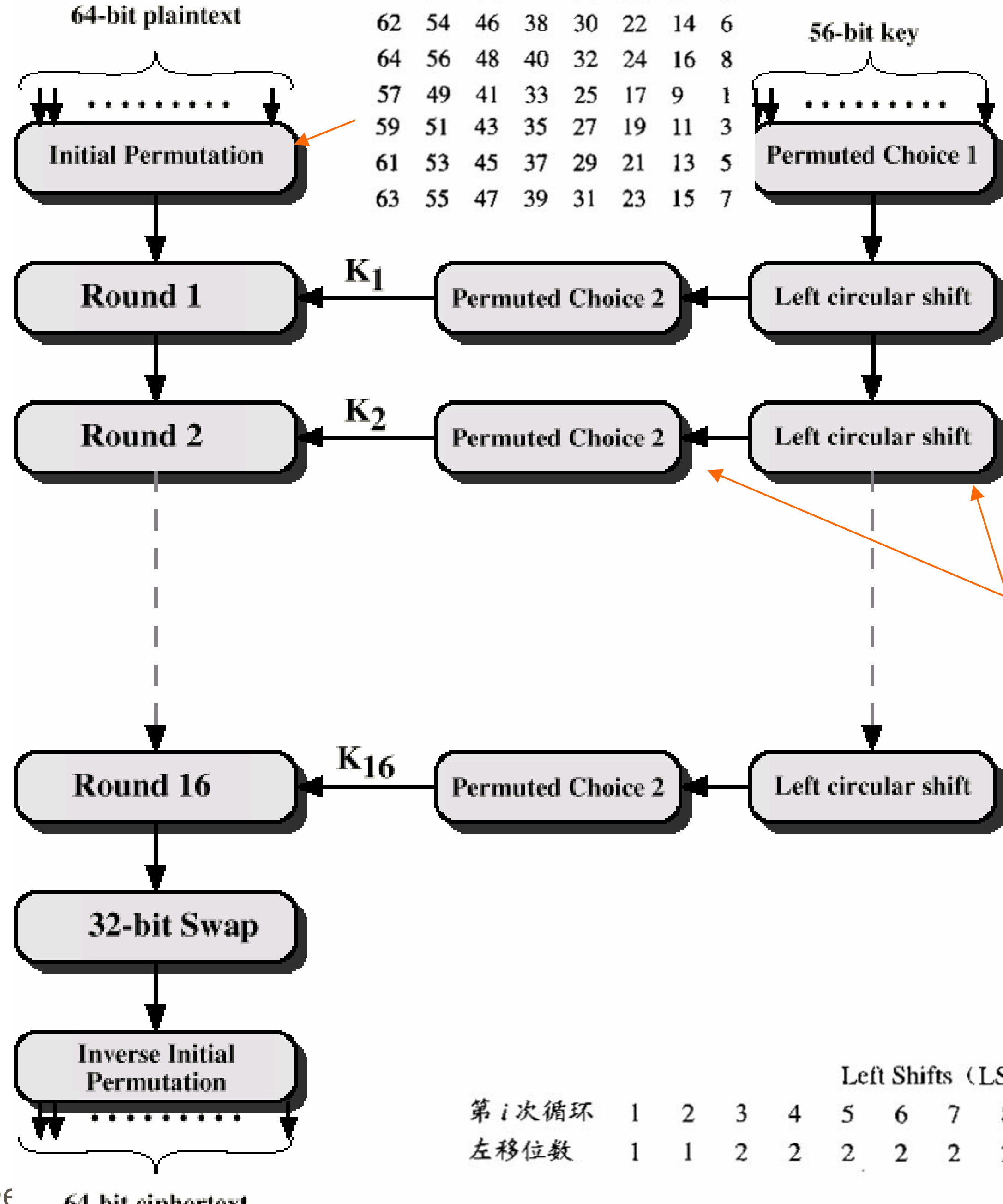


Figure 2.2 Classical Feistel Network

Data Encryption Standard (DES)

- The most widely used encryption scheme
- The algorithm is reffered to the Data Encryption Algorithm (DEA)
- DES is a block cipher
- The plaintext is processed in 64-bit blocks
- The key is 56-bits in length

DES Encryption Algorithm



Initial Permutation （IP）

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Permuted Choice 1 （PC-1）

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Permuted Choice 2 （PC-2）

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

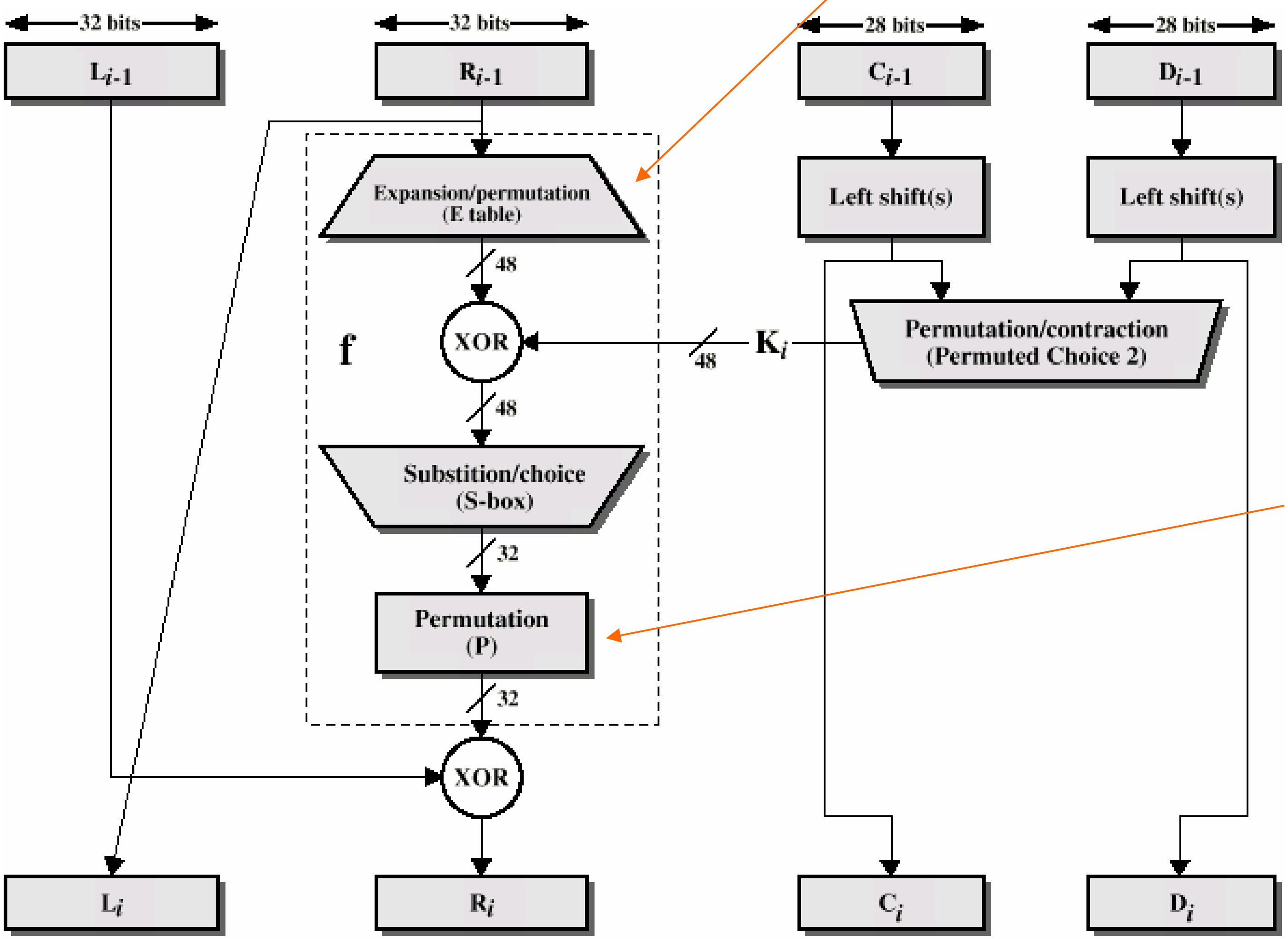
Left Shifts（LS）

第 i 次循环	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
左移位数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Single Iteration

Expansion （E）

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Permutation P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

S-BOXES1（S[1]）

行/列	0	1	2	3
0	14	0	4	15
1	4	15	1	12
2	13	7	14	8
3	1	4	8	2
4	2	14	13	4
5	15	2	6	9
6	11	13	2	1
7	8	1	11	7
8	3	10	15	5
9	10	6	12	11
10	6	12	9	3
11	12	11	7	14
12	5	9	3	10
13	9	5	10	0
14	0	3	5	6
15	7	8	0	13

S-BOXES2（S[2]）

行/列	0	1	2	3
0	15	3	0	13
1	1	13	14	8
2	8	4	7	10
3	14	7	11	1
4	6	15	10	3
5	11	2	4	15
6	3	8	13	4
7	4	14	1	2
8	9	12	5	11
9	7	0	8	6
10	2	1	12	7
11	13	10	6	12
12	12	6	9	0
13	0	9	3	5
14	5	11	2	14
15	10	5	15	9

S-BOXES3（S[3]）

行/列	0	1	2	3
0	10	13	13	1
1	0	7	6	10
2	9	0	4	13
3	14	9	9	0
4	6	3	8	6
5	3	4	15	9
6	15	6	3	8
7	5	10	0	7
8	1	2	11	4
9	13	8	1	15
10	12	5	2	14
11	7	14	12	3
12	11	12	5	11
13	4	11	10	5
14	2	15	14	2
15	8	1	7	12

S-BOXES4（S[4]）

行/列	0	1	2	3
0	7	13	10	3
1	13	8	6	15
2	14	11	9	0
3	3	5	0	6
4	0	6	12	10
5	6	15	11	1
6	9	0	7	13
7	10	3	13	8
8	1	4	15	9
9	2	7	1	4
10	8	2	3	5
11	5	12	14	11
12	11	1	5	12
13	12	10	2	7
14	4	14	8	2
15	15	9	4	14

S-BOXES5（S[5]）

行/列	0	1	2	3
0	2	14	4	11
1	12	11	2	8
2	4	2	1	12
3	1	12	11	7
4	7	4	10	1
5	10	7	13	14
6	11	13	7	2
7	6	1	8	13
8	8	5	15	6
9	5	0	9	15
10	3	15	12	0
11	15	10	5	9
12	13	3	6	10
13	0	9	3	4
14	14	8	0	5
15	9	6	14	3

S-BOXES6（S[6]）

行/列	0	1	2	3
0	12	10	9	4
1	1	15	14	3
2	10	4	15	2
3	15	2	5	12
4	9	7	2	9
5	2	12	8	5
6	6	9	12	15
7	8	5	3	10
8	0	6	7	11
9	13	1	0	14
10	3	13	4	1
11	4	14	10	7
12	14	0	1	6
13	7	11	13	0
14	5	3	11	8
15	11	8	6	13

S-BOXES7（S[7]）

行/列	0	1	2	3
0	4	13	1	6
1	11	0	4	11
2	2	11	11	13
3	14	7	13	8
4	15	4	12	1
5	0	9	3	4
6	8	1	7	10
7	13	10	14	7
8	3	14	10	9
9	12	3	15	5
10	9	5	6	0
11	7	12	8	15
12	5	2	0	14
13	10	15	5	2
14	6	8	9	3
15	1	6	2	12

S-BOXES8（S[8]）

行/列	0	1	2	3
0	13	1	7	2
1	2	15	11	1
2	8	13	4	14
3	4	8	1	7
4	6	10	9	4
5	15	3	12	10
6	11	7	14	8
7	1	4	2	13
8	10	12	0	15
9	9	5	6	12
10	3	6	10	9
11	14	11	13	0
12	5	0	15	3
13	0	14	3	5
14	12	9	5	6
15	7	2	8	11

DES

- **The overall processing at each iteration:**
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- Concerns about:
  - The algorithm and the key length (56-bits)

Strength of DES

- Declared insecure in 1998
- Electronic Frontier Foundation
- DES Cracker machine
- DES now worthless
- Alternatives include TDEA

Triple DEA

- Use three keys and three executions of the DES algorithm (encrypt-decrypt-encrypt)

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

  - C = ciphertext
  - P = Plaintext
  - EK[X] = encryption of X using key K
  - DK[Y] = decryption of Y using key K
- Effective key length of 168 bits
- ANSI X9.17 (1985)
- Incorporated in DEA standard 1999

Triple DEA

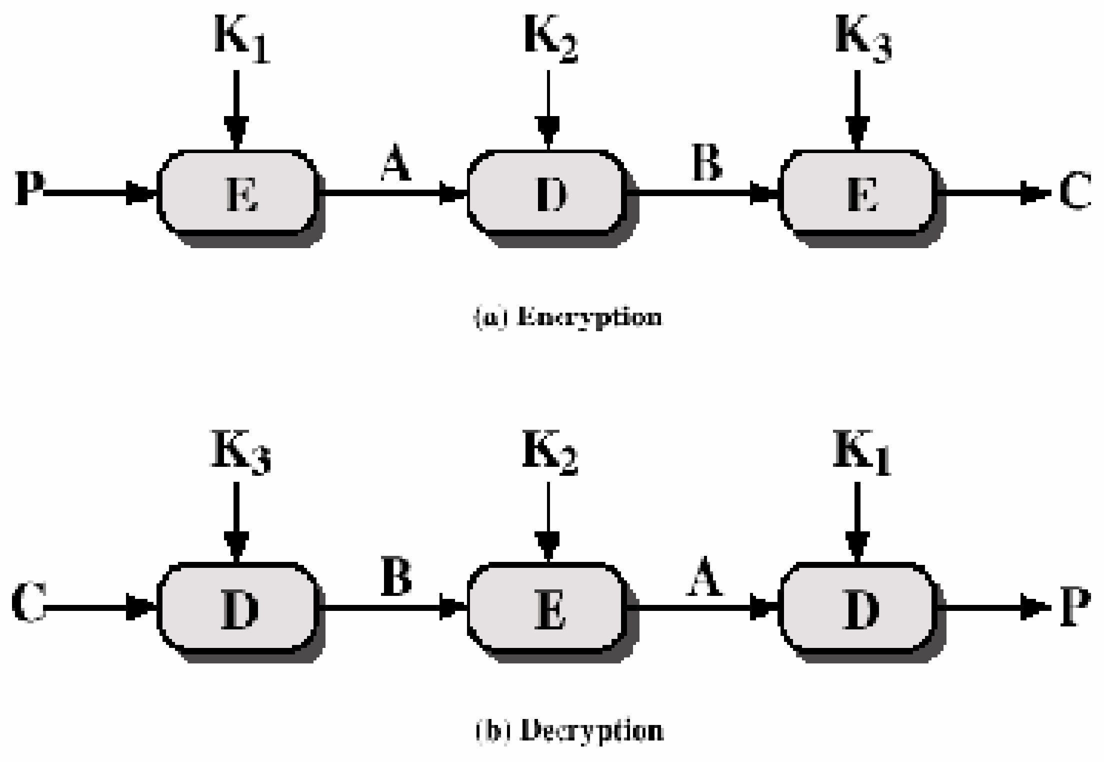


Figure 2.6 Triple DEA

Other Symmetric Block Ciphers

- **Advanced Encryption Standard (AES)**
  - 128-bit key
  - Easier to implement than TDEA
- **International Data Encryption Algorithm (IDEA)**
  - 128-bit key
  - Used in PGP (Pretty Good Privacy)—a software packet
- **Blowfish**
  - Easy to implement
  - High execution speed
  - Run in less than 5K of memory

Other Symmetric Block Ciphers

- **RC5**
  - Suitable for hardware and software
  - Fast, simple
  - Adaptable to processors of different word lengths
  - Variable number of rounds
  - Variable-length key
  - Low memory requirement
  - High security
  - Data-dependent rotations
- **Cast-128**
  - Key size from 40 to 128 bits
  - The round function differs from round to round

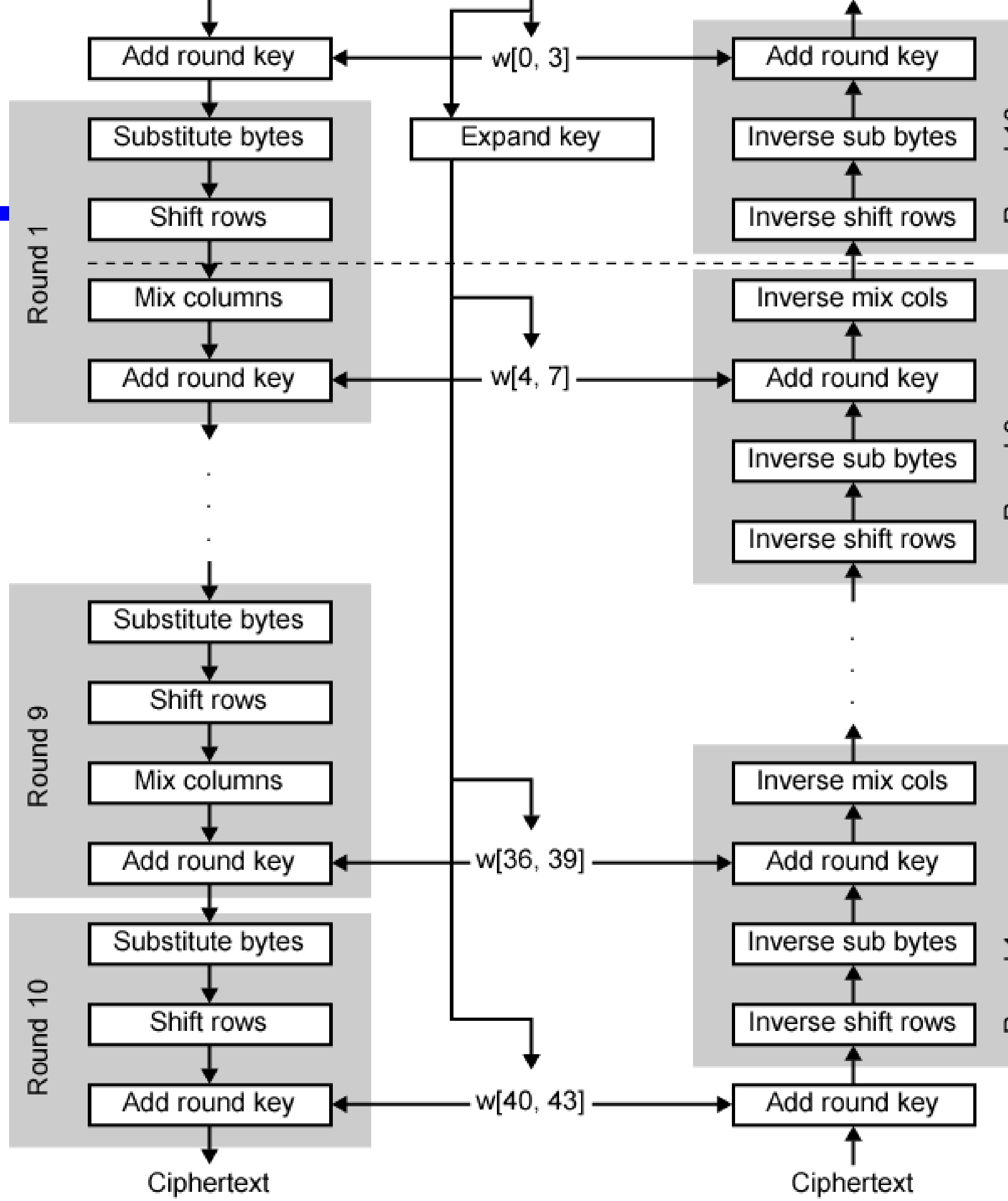
Advanced Encryption Standard

- National Institute of Standards and Technology (NIST) in 1997 issued call for Advanced Encryption Standard (AES)
  - Security strength equal to or better than 3DES
  - Improved efficiency
  - Symmetric block cipher
  - Block length 128 bits
  - Key lengths 128, 192, and 256 bits
  - Evaluation include security, computational efficiency, memory requirements, hardware and software suitability, and flexibility
  - 2001, AES issued as federal information processing standard (FIPS 197)

AES Description

- Assume key length 128 bits
- Input is single 128-bit block
  - Depicted as square matrix of bytes
  - Block copied into State array
    - Modified at each stage
  - After final stage, State copied to output matrix
- 128-bit key depicted as square matrix of bytes
  - Expanded into array of key schedule words
  - Each four bytes
  - Total key schedule 44 words for 128-bit key
- Byte ordering by column
  - First four bytes of 128-bit plaintext input occupy first column of in matrix
  - First four bytes of expanded key occupy first column of w matrix

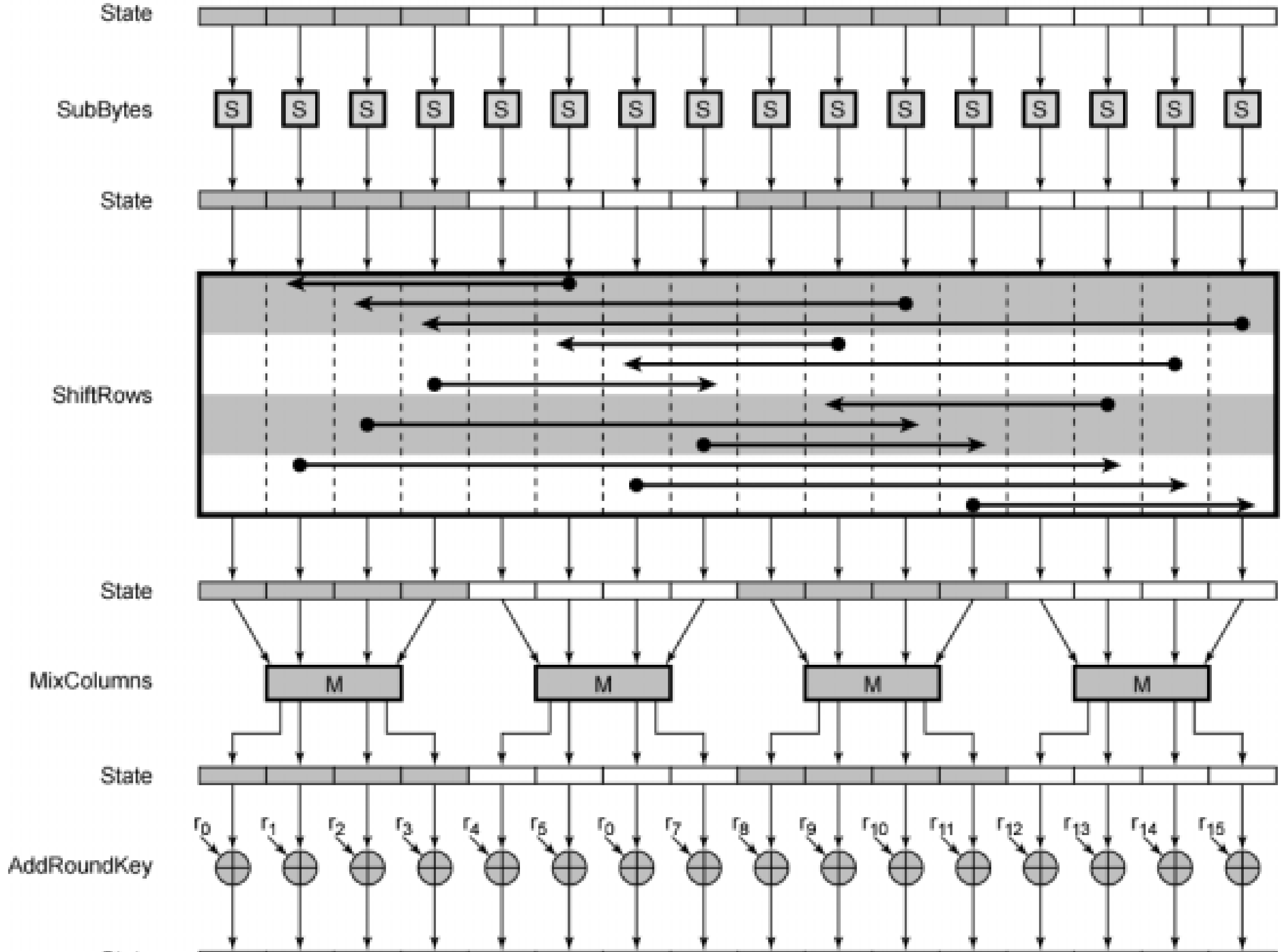
AES Encryption and Decryption



AES Comments (1)

- Key expanded into array of forty-four 32-bit words, w[i]
  - Four distinct words (128 bits) serve as round key for each round
- Four different stages
  - One permutation and three substitution
    - Substitute bytes uses S-box table to perform byte-by-byte substitution of block
    - Shift rows is permutation that performed row by row
    - Mix columns is substitution that alters each byte in column as function of all of bytes in column
    - Add round key is bitwise XOR of current block with portion of expanded key
- Simple structure
  - For both encryption and decryption, cipher begins with Add Round Key stage
  - Followed by nine rounds,

AES Encryption Round



AES Comments (2)

- Only Add Round Key stage uses key
  - Begin and ends with Add Round Key stage
  - Any other stage at beginning or end, reversible without key
    - Adds no security
- Add Round Key stage by itself not formidable
  - Other three stages scramble bits
  - By themselves provide no security because no key
- Each stage easily reversible
- Decryption uses expanded key in reverse order
  - Not identical to encryption algorithm
- Easy to verify that decryption does recover plaintext
- Final round of encryption and decryption consists of only three stages
  - To make the cipher reversible

Cipher Block Modes of Operation

- Cipher Block Chaining Mode (CBC)
  - The input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block.
  - Repeating pattern of 64-bits are not exposed

$$C_i = E_k[C_{i-1} \oplus P_i]$$

$$D_K[C_i] = D_K[E_K(C_{i-1} \oplus P_i)]$$

$$D_K[C_i] = (C_{i-1} \oplus P_i)$$

$$C_{i-1} \oplus D_K[C_i] = C_{i-1} \oplus C_{i-1} \oplus P_i = P_i$$

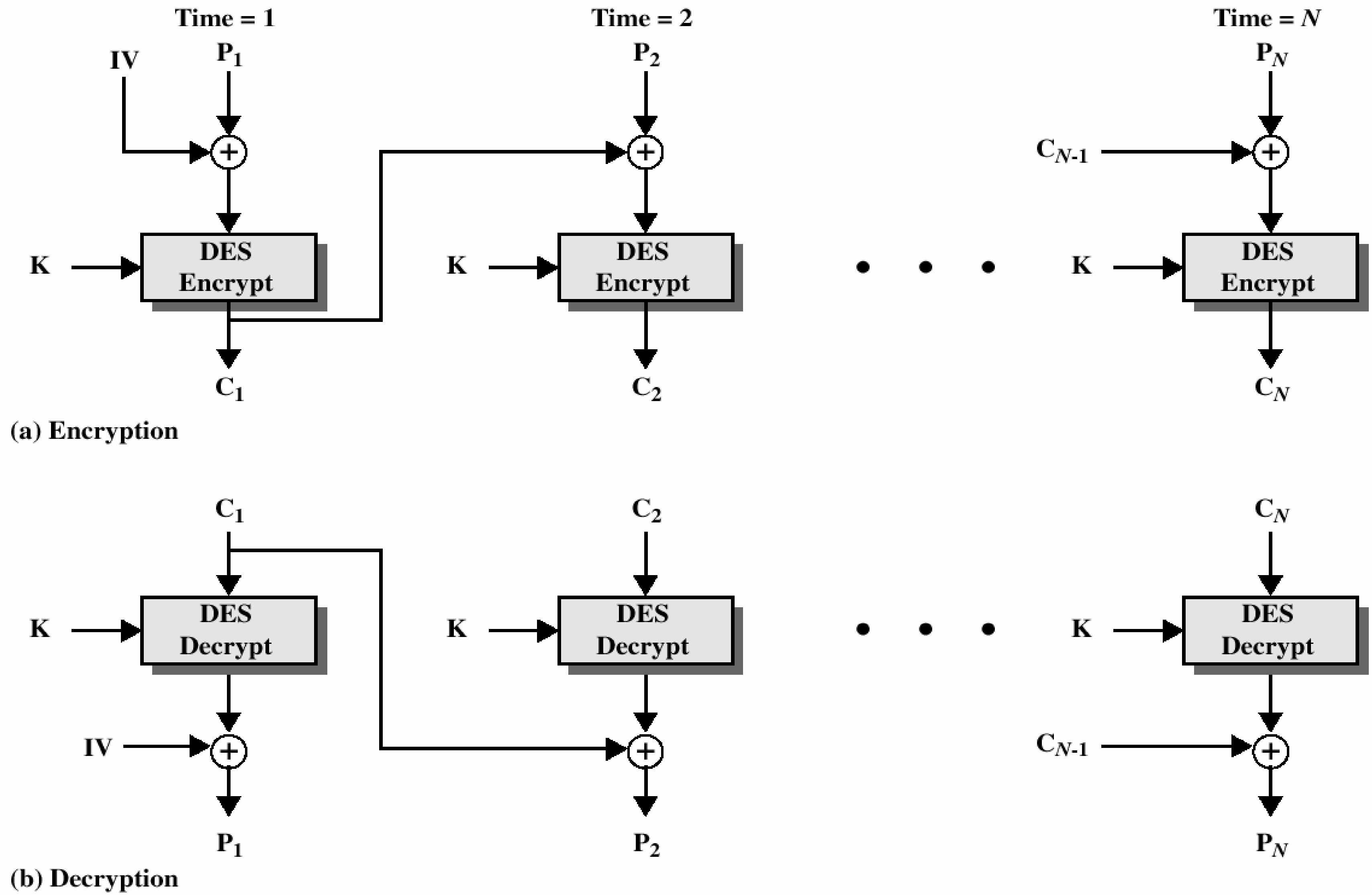
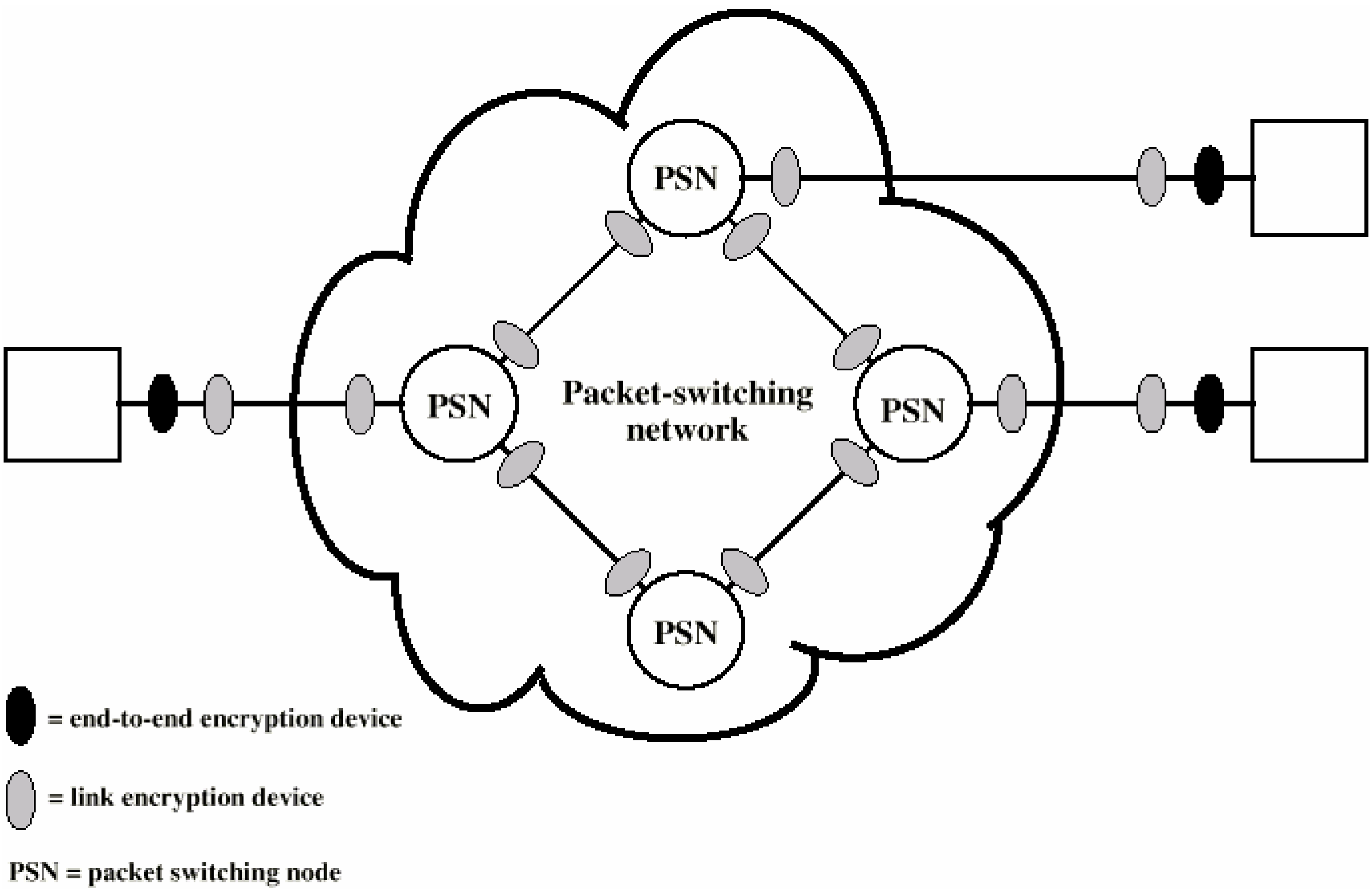


Figure 2.7 Cipher Block Chaining (CBC) Mode

Location of Encryption Device

- **Link encryption:**
  - A lot of encryption devices
  - High level of security
  - Decrypt each packet at every switch
- **End-to-end encryption**
  - The source encrypt and the receiver decrypts
  - Payload encrypted
  - Header in the clear
- **High Security:** Both link and end-to-end encryption are needed (see the figure on next slide)

Encryption across a packet-switching network



Link Encryption

- Each communication link equipped at both ends
- All traffic secure
- High level of security
- Requires lots of encryption devices
- Message must be decrypted at each switch to read address (virtual circuit number)
- Security vulnerable at switches
  - Particularly on public switched network

End to End Encryption

- Encryption done at ends of system
- Data in encrypted form crosses network unaltered
- Destination shares key with source to decrypt
- Host can only encrypt user data
  - Otherwise switching nodes could not read header or route packet
- Traffic pattern not secure
- Use both link and end to end

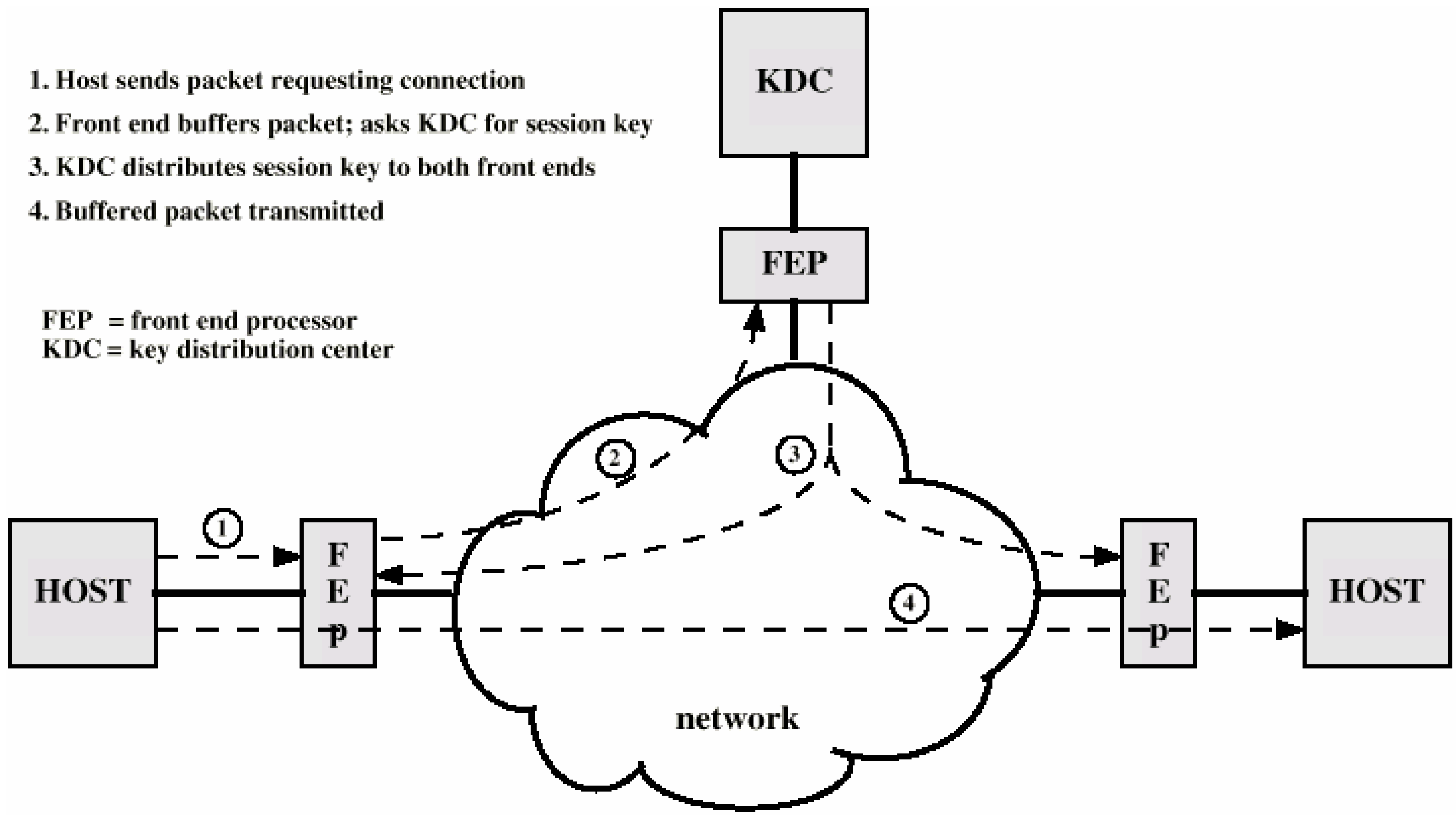
Key Distribution

- ⌘ A key could be selected by A and physically delivered to B.
- ⌘ A third party could select the key and physically deliver it to A and B.
- ⌘ If A and B have previously used a key, one party could transmit the new key to the other, encrypted using the old key.
- ⌘ If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Automatic Key Distribution

- **Session key:**
  - Data encrypted with a one-time session key.At the conclusion of the session the key is destroyed
- **Permanent key:**
  - Used between entities for the purpose of distributing session keys
- **Key distribution center**
  - Determines which systems may communicate
  - Provides one session key for that connection
- **Front end processor**
  - Performs end to end encryption
  - Obtains keys for host

Automatic Key Distribution for Connection-Oriented Protocol



Traffic Padding

- Produce cipher text continuously
- If no plain text to encode, send random data
- Make traffic analysis impossible

Recommended Reading

- Stallings, W. *Cryptography and Network Security: Principles and Practice, 2nd edition*. Prentice Hall, 1999
- Scneier, B. *Applied Cryptography*, New York: Wiley, 1996
- Mel, H.X. Baker, D. *Cryptography Decrypted*. Addison Wesley, 2001

18.3 Authentication and Hash

- Message authentication
- Authentication using encryption
- Authentication without encryption
  - Message authentication code
  - One way hash function
- Secure hash functions
- Secure hash function SHA-1

Message Authentication

- **Encryption** protects against passive attacks
  - Eavesdropping
- **Message Authentication** Protects against active attacks
  - Falsification of data and transactions
- Message is authentic if it is genuine and comes from the alleged source
- Authentication allows receiver to verify that message is authentic
  - **Message has not been altered**
  - **Message is from authentic source**
  - Message timeline (not be artificially delayed and replayed)
  - Message sequence is correct (not altered)

Approach 1
Authentication Using Encryption

- Assumes sender and receiver are only entities that know key
- Message includes:
  - error detection code
  - sequence number
  - time stamp

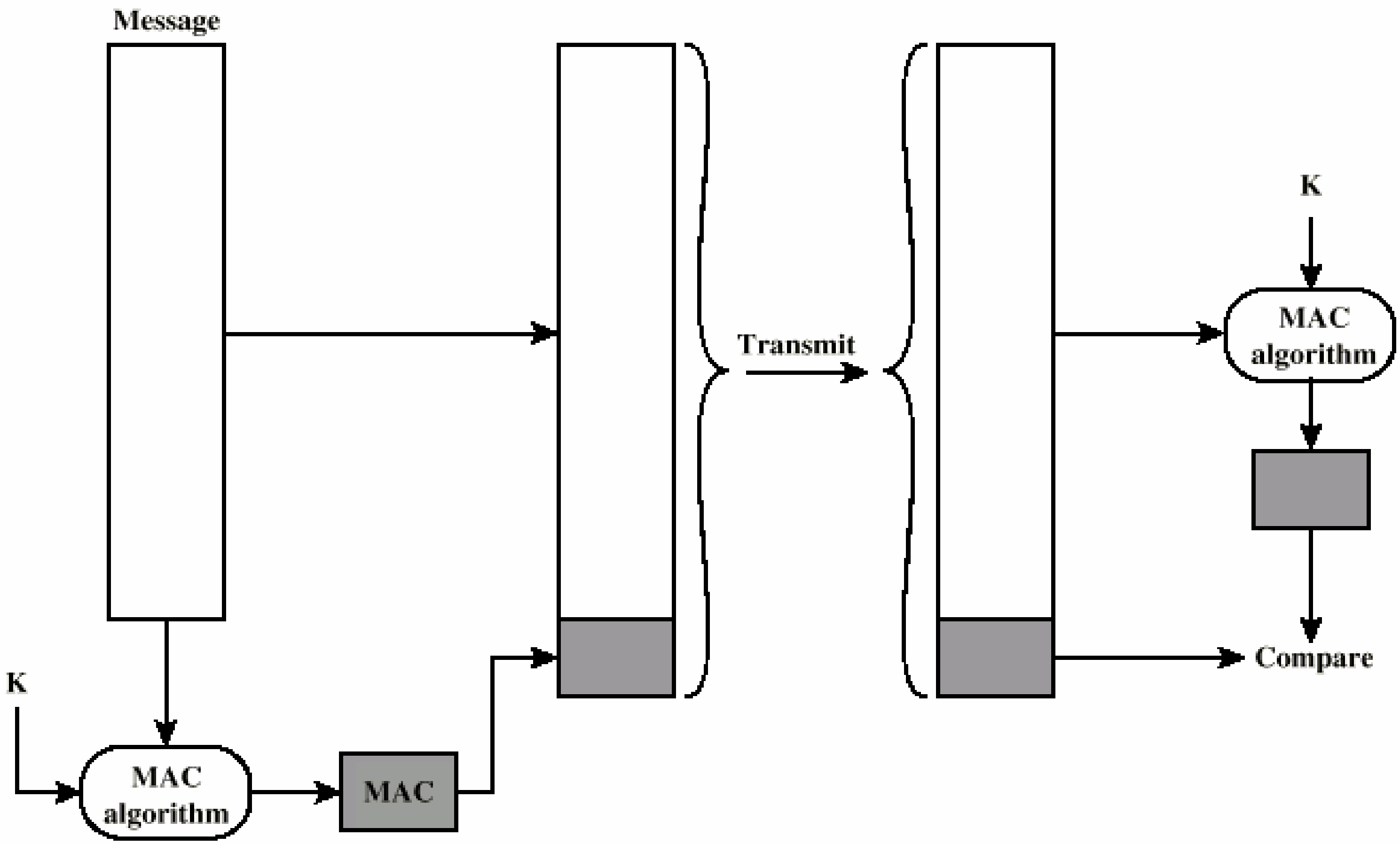
Approach 2
Authentication Without Encryption

- Authentication tag generated and appended to each message
- Message not encrypted
- Useful for:
  - Messages broadcast to multiple destinations
    - Have one destination responsible for authentication
  - One side heavily loaded
    - Encryption adds to workload
    - Can authenticate random messages
  - Programs authenticated without encryption can be executed without decoding

Approach 2-1 Message Authentication Code

- Generate authentication code based on shared key and message
- Common key shared between sender A and receiver B
- A and B calculate individually: $MAC_M=F(K_{AB},M)$
- If only sender and receiver know key and MAC code matches:
  - Receiver assured message has not altered
  - Receiver assured message is from alleged sender
  - If message has sequence number, receiver assured of proper sequence
- A number of algorithms can be used to generate MAC
  - NBS recommends use of DES
  - Last 16 or 32 bits of ciphertext used as MAC

Authentication Using Message Authentication Code



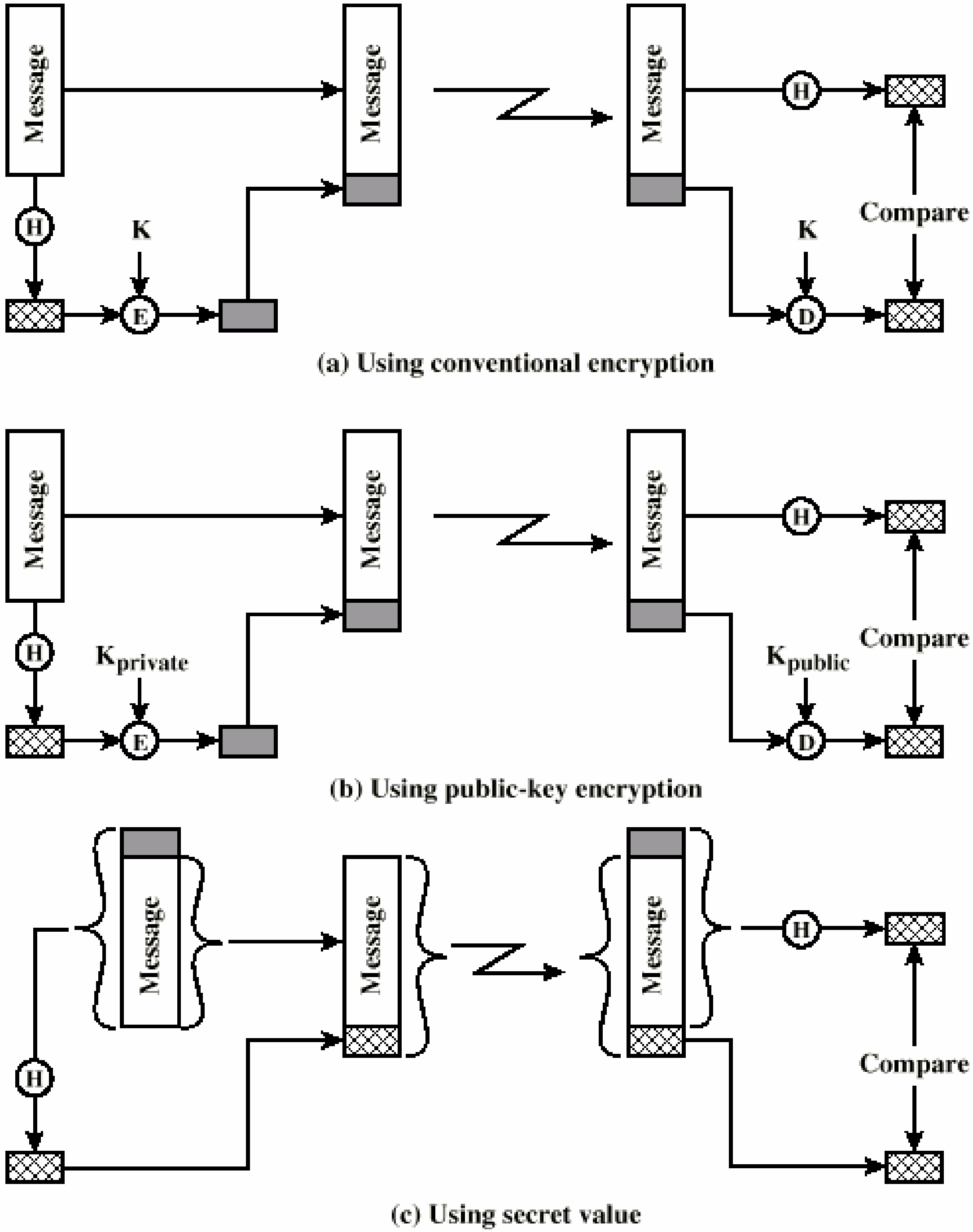
Question

- What is difference between encryption process and authentication algorithm ?
  - Authentication algorithm need not be reversible
  - Encryption must be reversible for decryption
  - Message authentication is less vulnerable to being broken than encryption because of mathematical properties of the authentication function

Approach 2-2
One Way Hash Function

- Accepts variable size message and produces fixed size tag (message digest)
  - Only function of message: H(M)
  - Difference from MAC: not tack secret key as input
  - Digest is sent with the massage
  - Digest is authentic
- Three ways in which message authenticated
  - Using conventional encryption
  - Using public-key encryption
  - Using secret value

Using One Way Hash



Advantages of authentication without encryption

- Encryption software is slow
- Encryption hardware expensive
- Encryption hardware optimized to large data
- Algorithms covered by patents
- Algorithms subject to export controls (from USA)

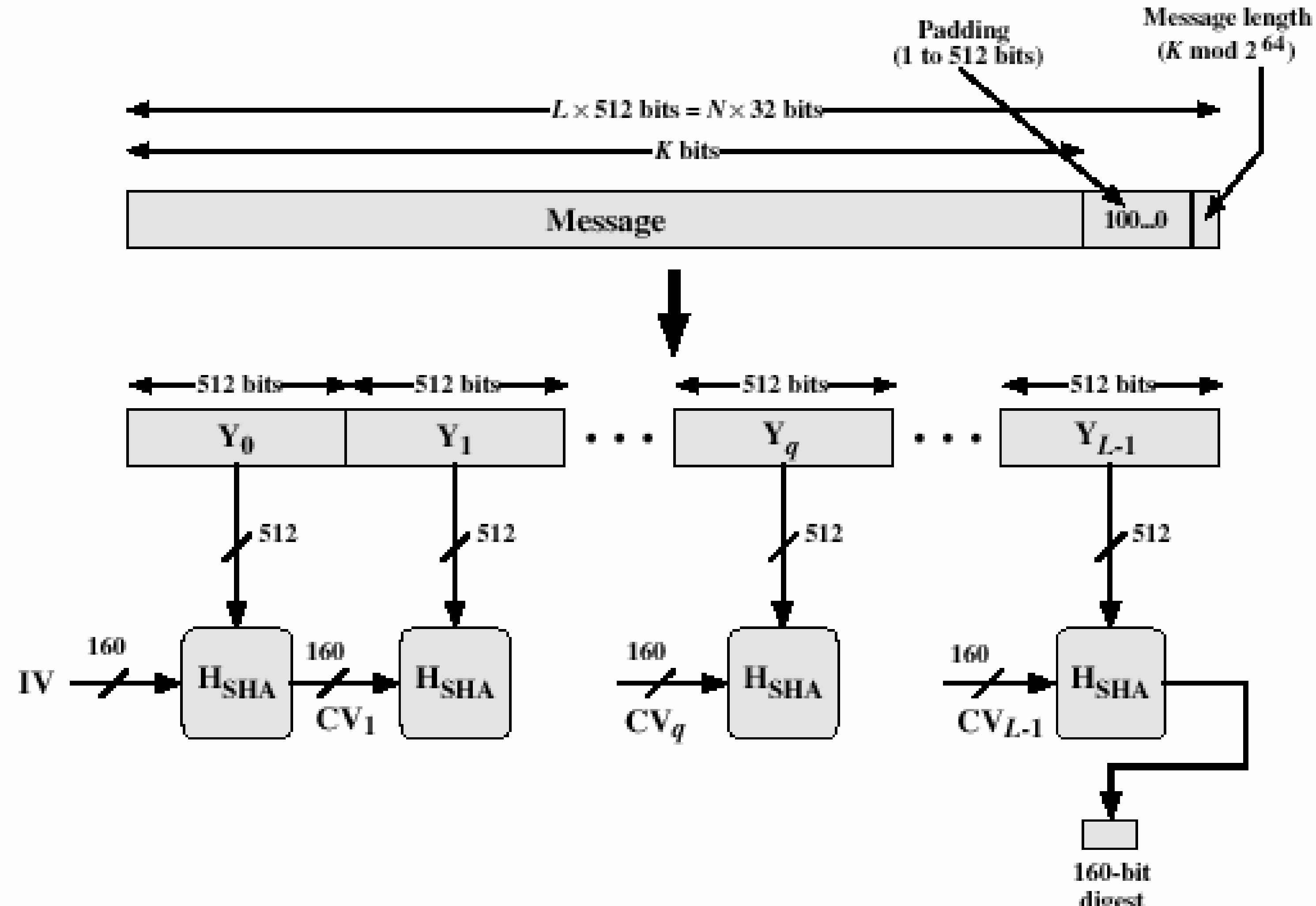
Secure Hash Functions

- Hash function must have following properties:
  - Can be applied to any size data block
  - Produce fixed length output
  - Easy to compute
  - Not feasible to reverse
  - Not feasible to find two message that give the same hash

SHA-1

- Secure Hash Algorithm 1
- Input message less than 2^{64} bits
  - Processed in 512 bit blocks
- Output 160 bit digest

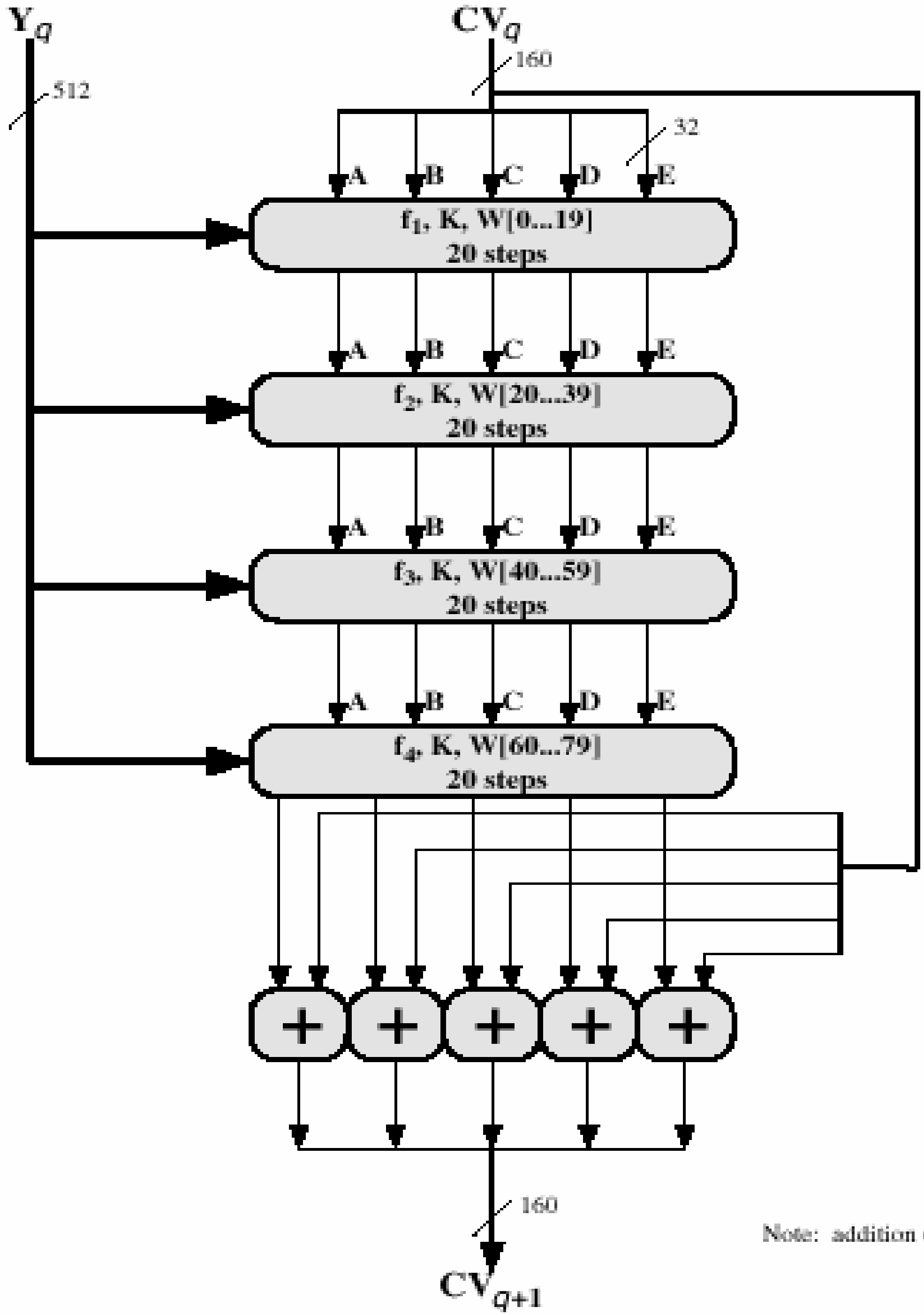
Message Digest Generation Using SHA-1



SHA-1 Processing

- Step 1: Append padding bits
  - message Length congruent to 448 modulo 512
- Step 2: Append length
  - a block of 64 bits is appended to the message
- Step 3: Initialize MD buffer
  - five 32-bit registers (A,B,C,D,E)
- Step 4: Process message
  - 512-bit (16-word) blocks
- Step 5: Output
  - 160-bit message digest

SHA-1 Processing of a Single 512-bit Block



Note: addition (+) is mod 2^{32}

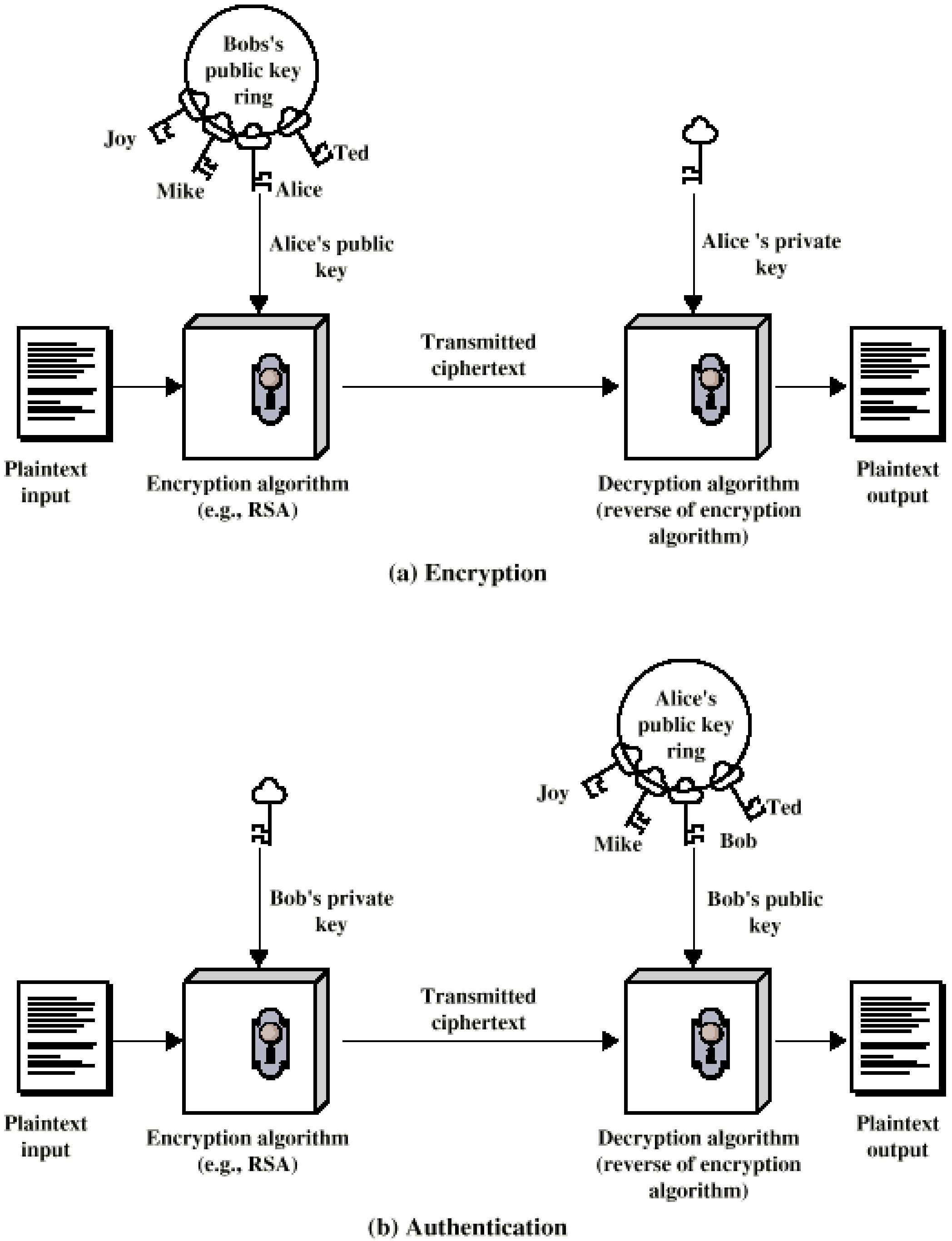
18.4 Public-Key Encryption and Digital Signature

- ⌘ Public-key encryption
- ⌘ Digital signature
- ⌘ RSA public-key encryption algorithm
- ⌘ Key management

Public Key Encryption

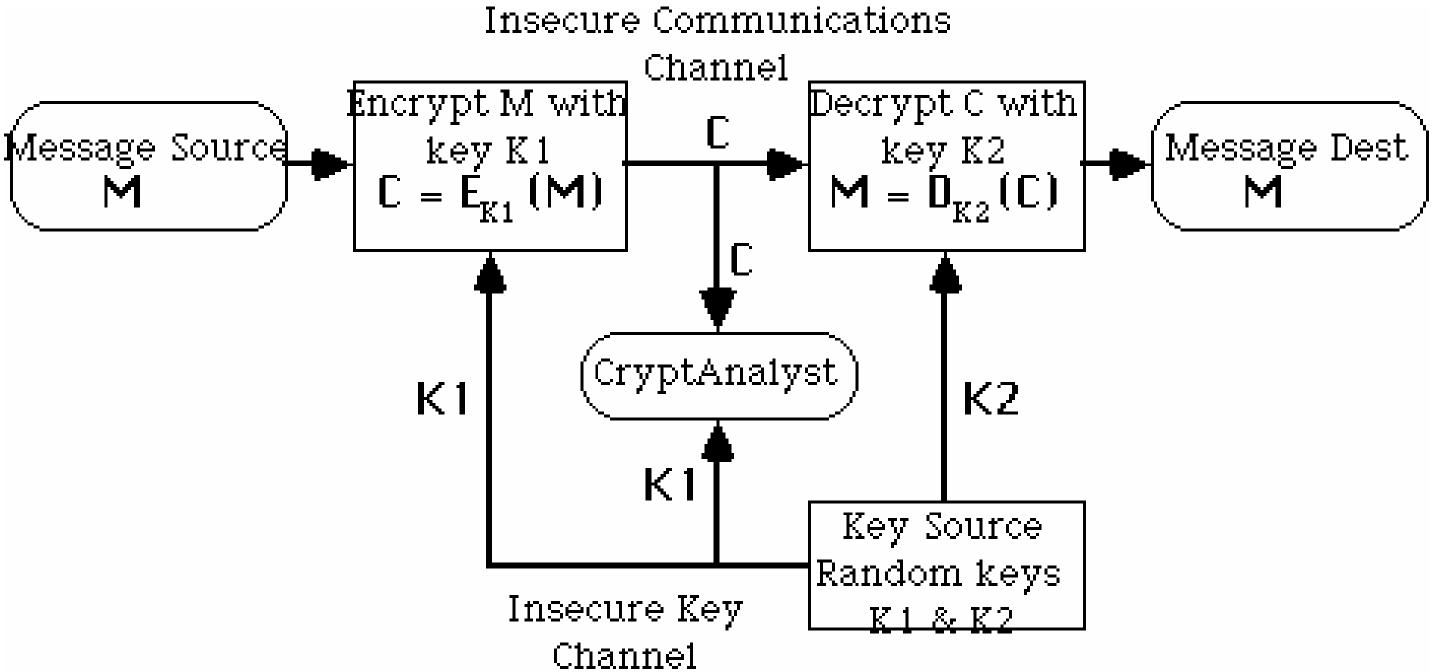
- Based on mathematical algorithms
- Asymmetric
  - Use two separate keys
- Ingredients
  - Plain text
  - Encryption algorithm
  - Public and private key
  - Cipher text
  - Decryption algorithm

Public Key Encryption



(a) Encryption

(b) Authentication



Asymmetric (Public-Key) Encryption System

Public Key Encryption - Operation

- One key made public
  - Used for encryption
- Other kept private
  - Used for decryption
- Infeasible to determine decryption key given encryption key and algorithm
- Either key can be used for encryption, the other for decryption

Steps

- User generates pair of keys
- User places one key in public domain
- To send a message to user, encrypt using public key
- User decrypts using private key

Digital Signature

- Sender encrypts message with their private key
- Receiver can decrypt using senders public key
- This authenticates sender, who is only person who has the matching key
- Does not give privacy of data
  - Decrypt key is public

RSA Algorithm

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

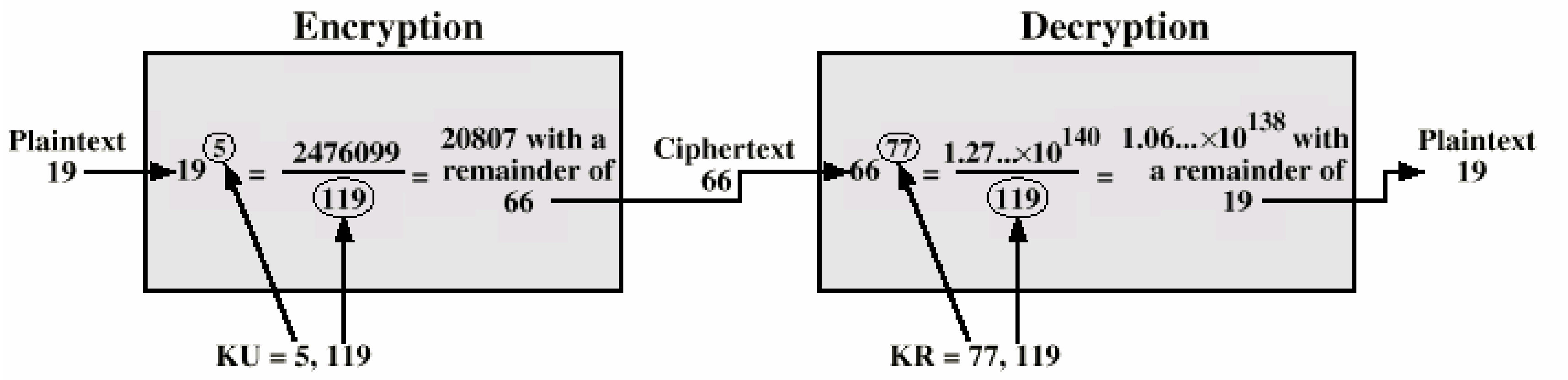
Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

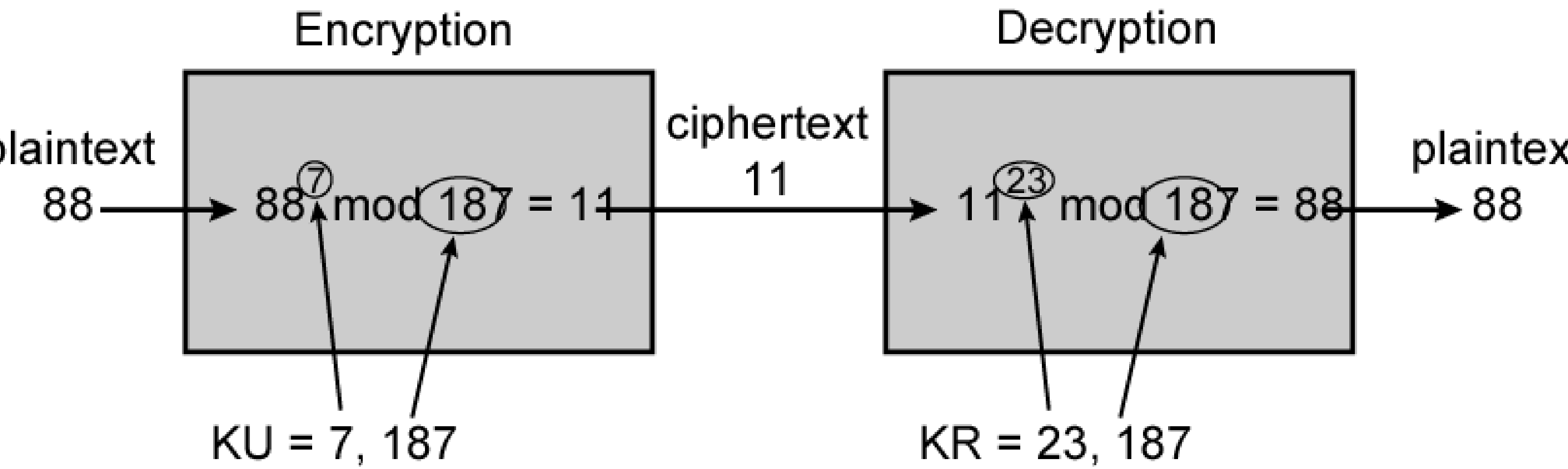
Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

RSA Example


Encryption
Decryption
Plaintext 19
19
5
2476099
119
20807 with a remainder of 66
Ciphertext 66
66
77
1.27...×10^140
119
1.06...×10^138 with a remainder of 19
Plaintext 19
KU = 5, 119
KR = 77, 119

RSA Example



Encryption
Decryption
plaintext
88
88 7 mod 187 = 11
ciphertext
11
11 23 mod 187 = 88
plaintex
88
KU = 7, 187
KR = 23, 187

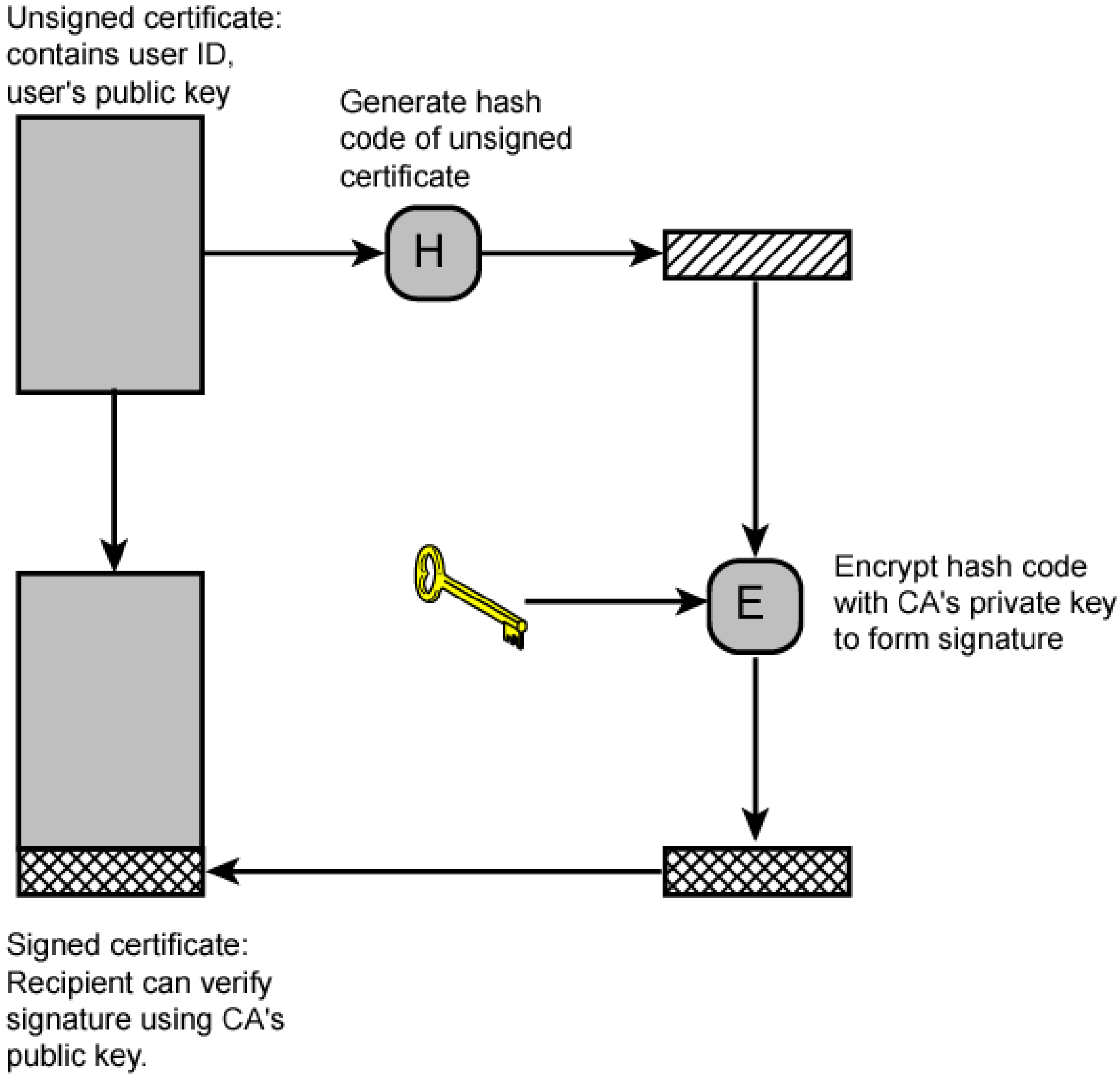
Key Distribution

- One Problem
  - How to distribute secret keys securely is the most difficult problem for symmetric encryption
- The problem is wiped away with public-key encryption
  - Private key is never distributed
- sender
  - Prepare a message
  - Encrypt the message using symmetric encryption with a one-time session key
  - Encrypt the session key using receiver's public key
  - Attach the encrypted session key to the message and send it to receiver

Key Distribution (cont.)

- **Another Problem**
  - Sender must be sure that the public key with the receiver's name written all over it is in fact the receiver's public key
- **Solution**
  - Public-key certificate
- Consist of
  - A public key plus a User ID of the key owner
  - Whole block signed by a trusted third part

Public Key Certificate Use



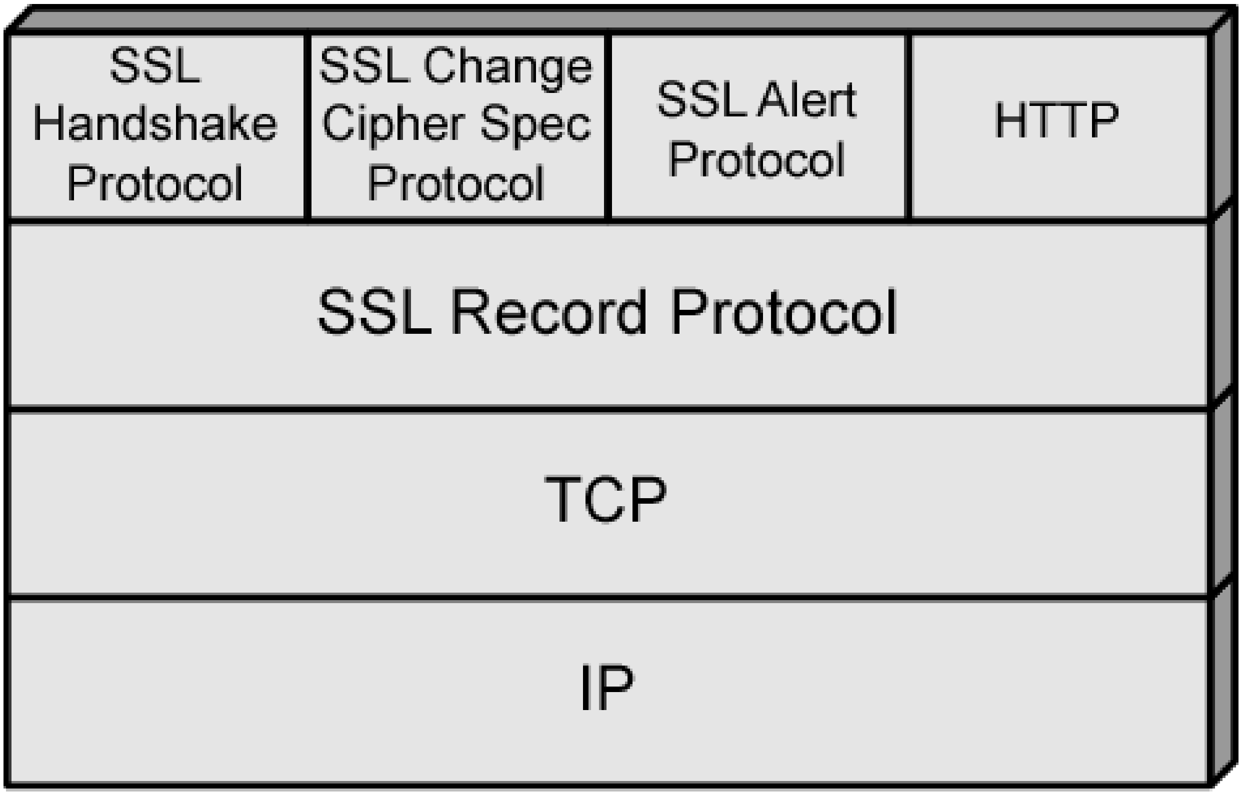
Secure Sockets Layer Transport Layer Security

- Security services
- Transport Layer Security defined in RFC 2246
- SSL general-purpose service
  - Set of protocols that rely on TCP
- Two implementation options
  - Part of underlying protocol suite
    - Transparent to applications
  - Embedded in specific packages
    - E.g. Netscape and Microsoft Explorer and most Web servers
- Minor differences between SSLv3 and TLS

SSL Architecture

- SSL uses TCP to provide reliable end-to-end secure service
- SSL two layers of protocols
- Record Protocol provides basic security services to various higher-layer protocols
  - In particular, HTTP can operate on top of SSL
- Three higher-layer protocols
  - Handshake Protocol
  - Change Cipher Spec Protocol
  - Alert Protocol
  - Used in management of SSL exchanges (see later)

SSL Protocol Stack



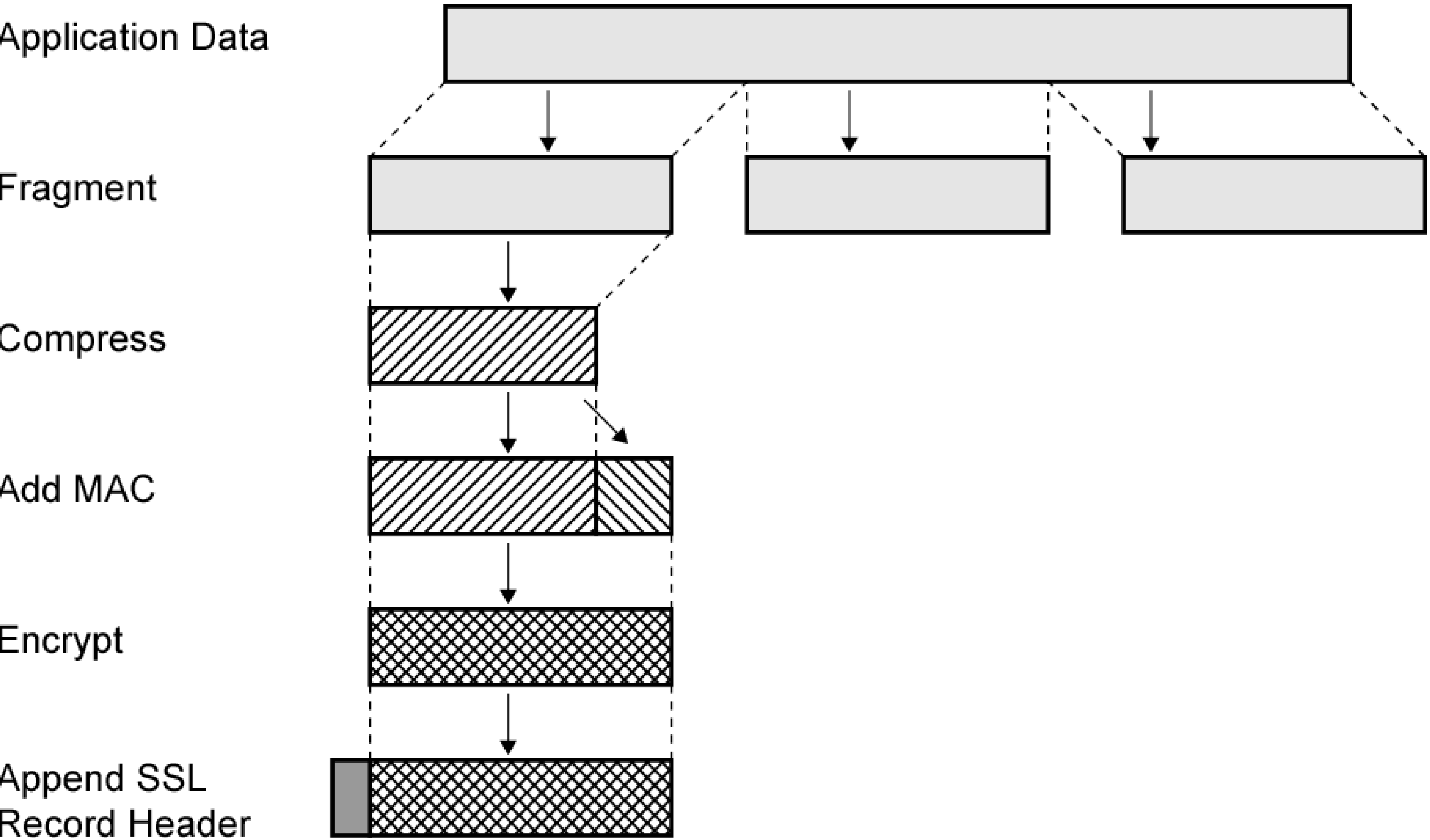
SSL Connection and Session

- Connection
  - Transport that provides suitable type of service
  - Peer-to-peer
  - Transient
  - Every connection associated with one session
- Session
  - Association between client and server
  - Created by Handshake Protocol
  - Define set of cryptographic security parameters
  - Used to avoid negotiation of new security parameters for each connection
- Maybe multiple secure connections between parties
- May be multiple simultaneous sessions between parties
  - Not used in practice

SSL Record Protocol

- Confidentiality
  - Handshake Protocol defines shared secret key
  - Used for symmetric encryption
- Message Integrity
  - Handshake Protocol defines shared secret key
  - Used to form message authentication code (MAC)
- Each upper-layer message fragmented
  - 2^{14} bytes (16384 bytes) or less
- Compression optionally applied
- Compute message authentication code
- Compressed message plus MAC encrypted using symmetric encryption
- Prepend header

SSL Record Protocol Operation



Record Protocol Header

- Content Type (8 bits)
  - change_cipher_spec, alert, handshake, and application_data
  - No distinction between applications (e.g., HTTP)
    - Content of application data opaque to SSL
- Major Version (8 bits) – SSL v3 is 3
- Minor Version (8 bits) - SSLv3 value is 0
- Compressed Length (16 bits)
  - Maximum $2^{14} + 2048$
- Record Protocol then transmits unit in TCP segment
- Received data are decrypted, verified, decompressed, and reassembled and then delivered

Change Cipher Spec Protocol

- Uses Record Protocol
- Single message
  - Single byte value 1
- Cause pending state to be copied into current state
  - Updates cipher suite to be used on this connection

Alert Protocol

- Convey SSL-related alerts to peer entity
- Alert messages compressed and encrypted
- Two bytes
  - First byte warning(1) or fatal(2)
    - If fatal, SSL immediately terminates connection
    - Other connections on session may continue
    - No new connections on session
  - Second byte indicates specific alert
  - E.g. fatal alert is an incorrect MAC
  - E.g. nonfatal alert is close_notify message

Handshake Protocol

- Authenticate
- Negotiate encryption and MAC algorithm and cryptographic keys
- Used before any application data sent

Handshake Protocol – Phase 1 Initiate Connection

- Version
  - Highest SSL version understood by client
- Random
  - Client-generated random structure
  - 32-bit timestamp and 28 bytes from secure random number generator
  - Used during key exchange to prevent replay attacks
- Session ID
  - Variable-length
  - Nonzero indicates client wishes to update existing connection or create new connection on session
  - Zero indicates client wishes to establish new connection on new session
- CipherSuite
  - List of cryptographic algorithms supported by client
  - Each element defines key exchange algorithm and CipherSpec
- Compression Method
  - Compression methods client supports

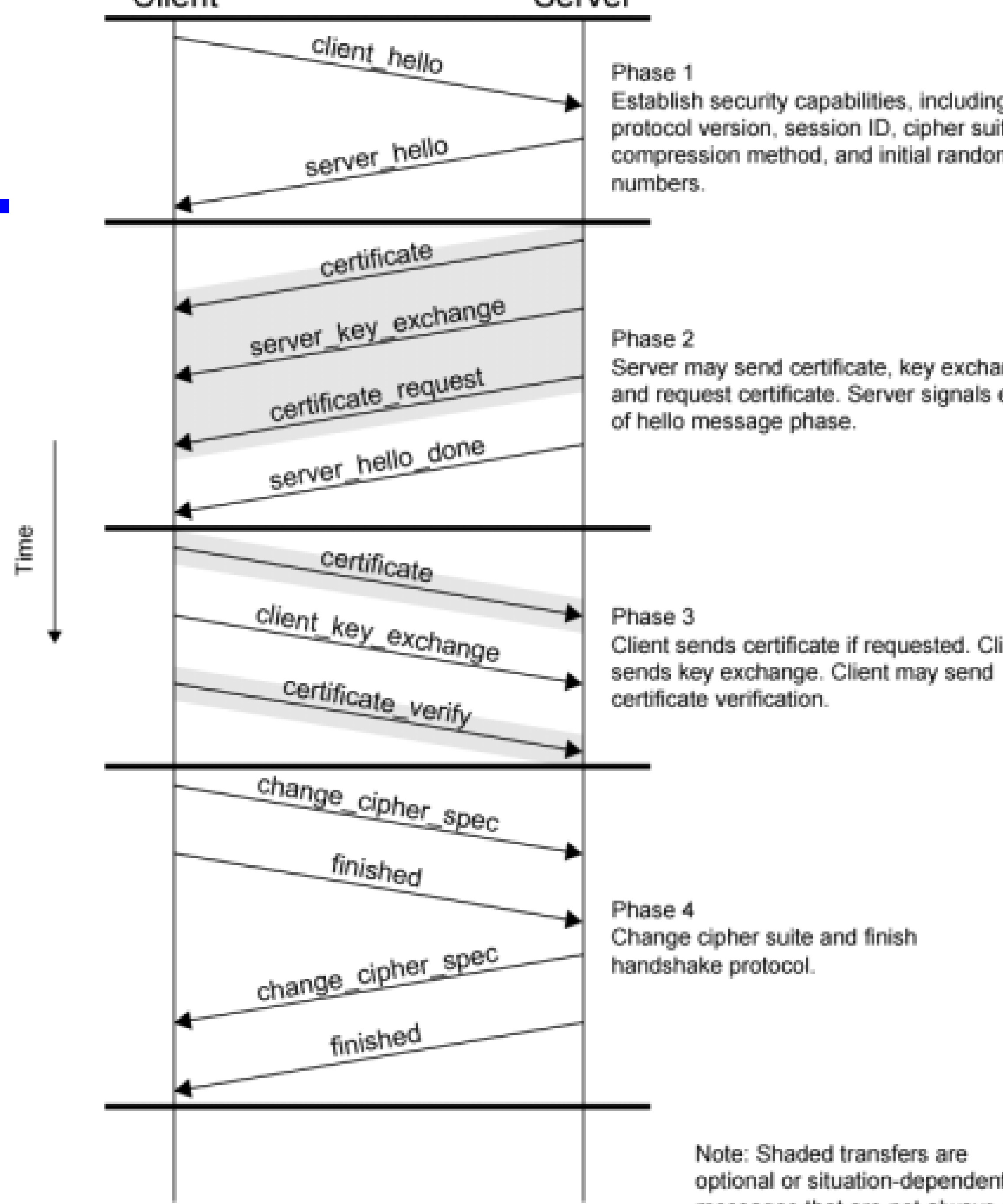
Handshake Protocol – Phase 2, 3

- Client waits for server_hello message
  - Same parameters as client_hello
- Phase 2 depends on underlying encryption scheme
- Final message in Phase 2 is server_done
  - Required
- Phase 3
  - Upon receipt of server_done, client verifies certificate if required and check server_hello parameters
  - Client sends messages to server, depending on underlying public-key scheme

Handshake Protocol – Phase 4

- Completes setting up
- Client sends change_cipher_spec
- Copies pending CipherSpec into current CipherSpec
  - Not considered part of Handshake Protocol
  - Sent using Change Cipher Spec Protocol
- Client sends finished message under new algorithms, keys, and secrets
- Finished message verifies key exchange and authentication successful
- Server sends own change_cipher_spec message
- Transfers pending to current CipherSpec
- Sends its finished message
- Handshake complete

Handshake Protocol Action



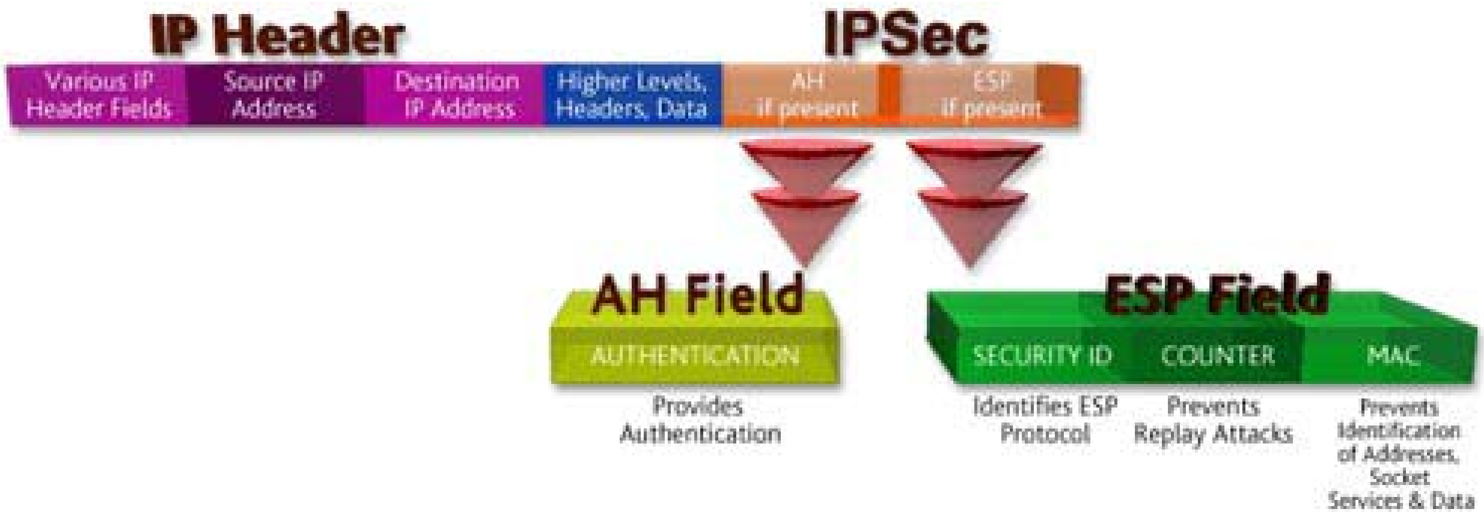
IPv4 and IPv6 Security

- IPSec
- Secure branch office connectivity over Internet
- Secure remote access over Internet
- Extranet and intranet connectivity
- Enhanced electronic commerce security

IPSec Scope

- Authentication header
- Encapsulated security payload
- Key exchange
- RFC 2401,2402,2406,2408

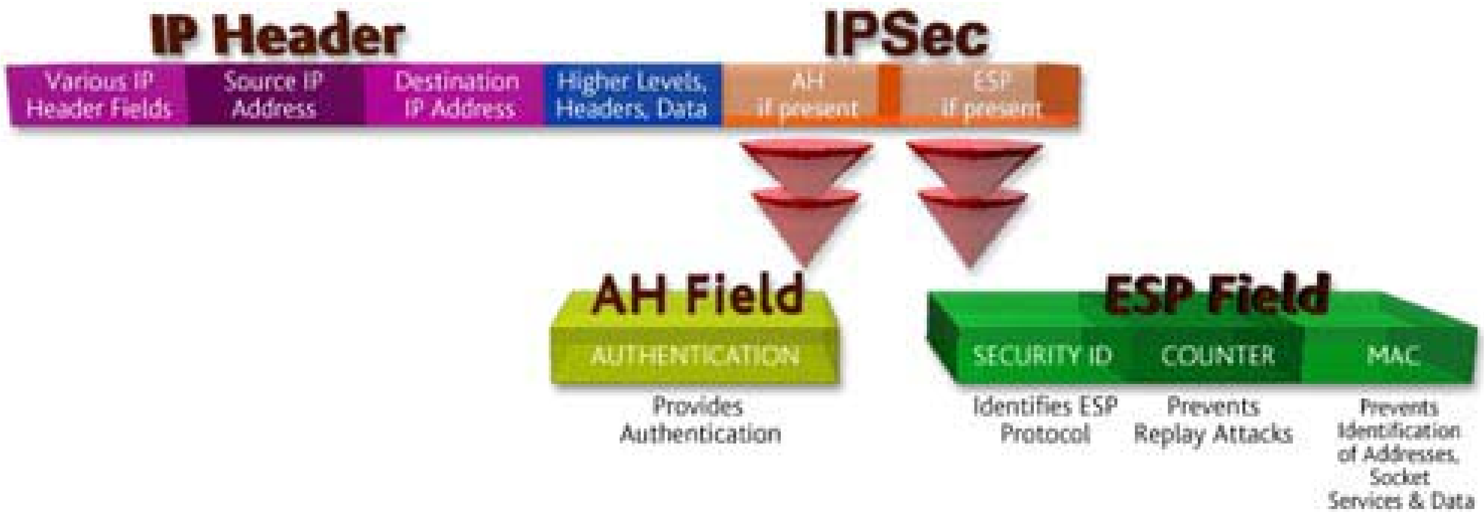
IPSec Components



Three Main Components that Comprise IPSec

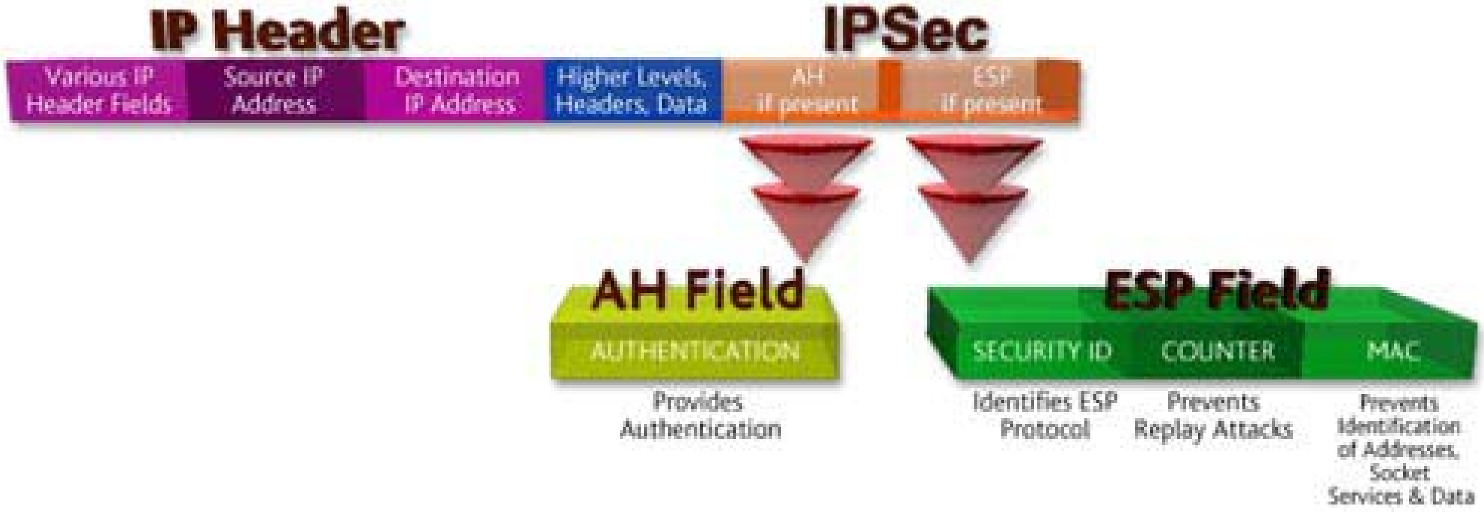
- **Authentication (AH)**
- **Encapsulation Security Protocol (ESP)**
- **Internet Key Exchange (IKE)**

IPSec Components



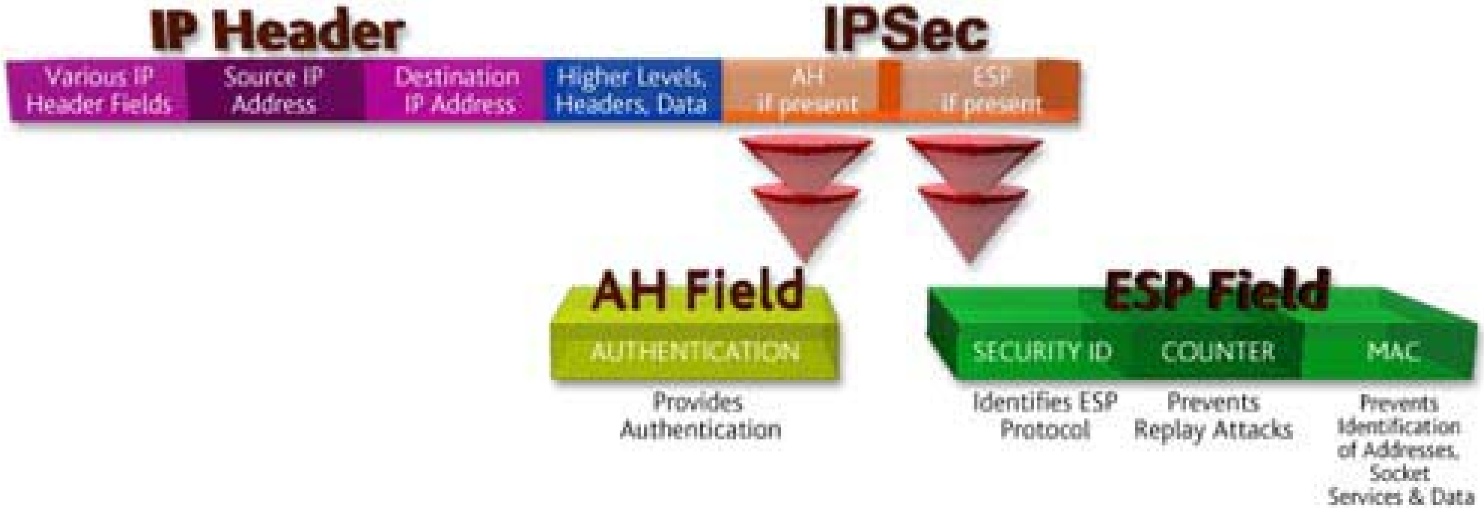
- **Authentication (AH)**
  - Ensures the integrity and authenticity of the data
  - Most common Authentication algorithms: MD5 and SHA-1
- **Encapsulation Security Protocol (ESP)**
- **Internet Key Exchange (IKE)**

IPSec Components



- Authentication (AH)
- **Encapsulation Security Protocol (ESP)**
  - Protects the confidentiality, integrity and authenticity
  - Most common Encryption algorithms: DES, 3DES and AES
- Internet Key Exchange (IKE)

IPSec Components



- Authentication (AH)
- Encapsulation Security Protocol (ESP)
- **Internet Key Exchange (IKE)**
  - Negotiates security association and exchanges key material
  - Uses pre-shared keys or digital certificates (X.509)

Security Association

- One way relationship between sender and receiver
- For two way, two associations are required
- Three SA identification parameters
  - Security parameter index
  - IP destination address
  - Security protocol identifier

SA Parameters

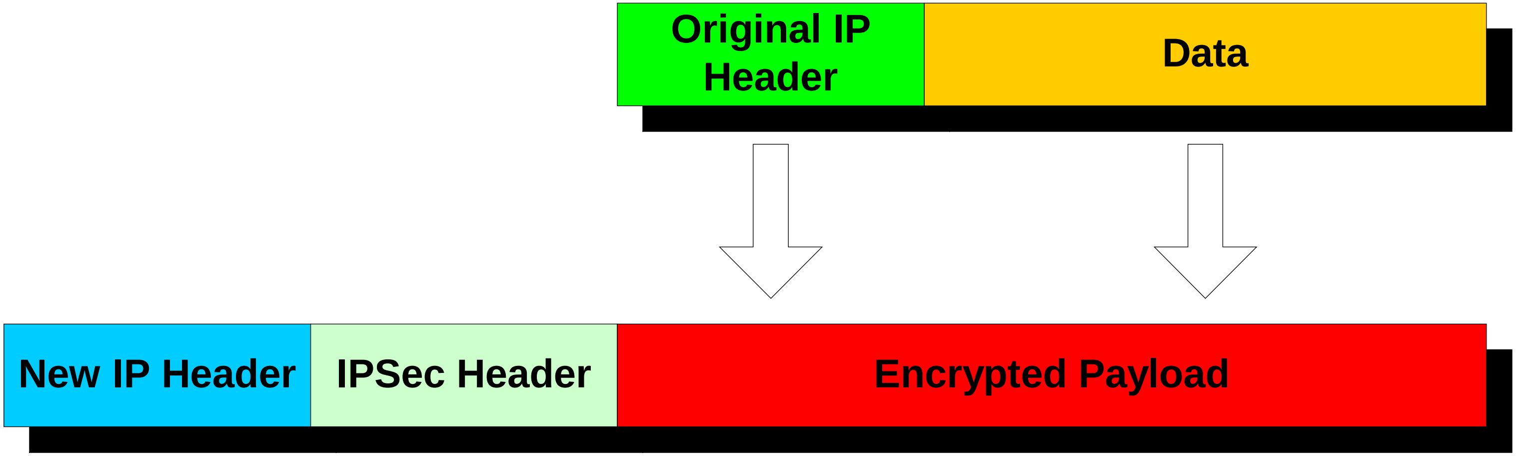
- Sequence number counter
- Sequence counter overflow
- Anti-reply windows
- AH information
- ESP information
- Lifetime of this association
- IPSec protocol mode
  - Tunnel, transport or wildcard
- Path MTU

Transport and Tunnel Modes

- Transport mode
  - Protection for upper layer protocols
  - Extends to payload of IP packet
  - End to end between hosts
- Tunnel mode
  - Protection for IP packet
  - Entire packet treated as payload for outer IP "packet"
  - No routers examine inner packet
  - May have different source and destination address
  - May be implemented at firewall

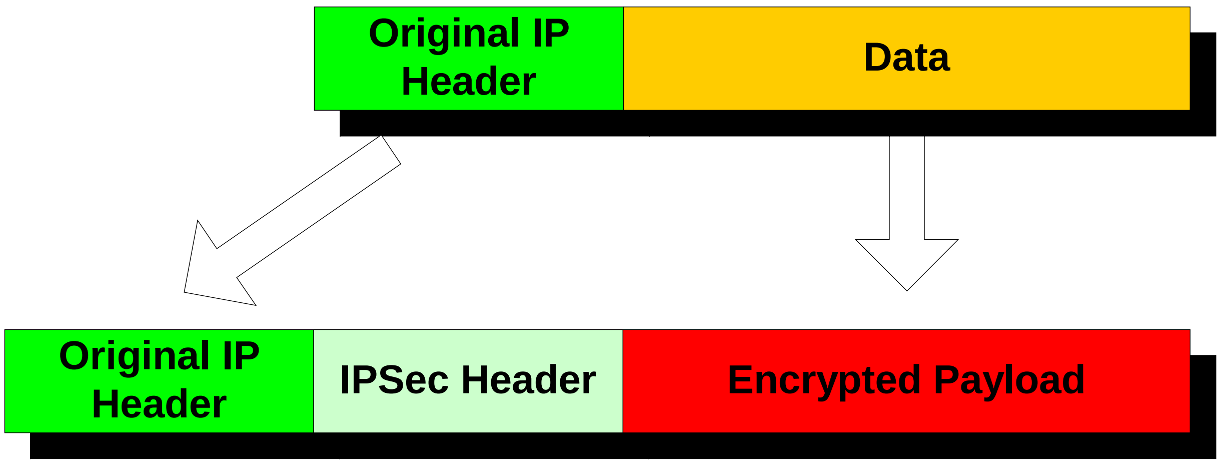
IPSec Modes – Two Types (Tunnel and Transport)

- **Tunnel**
  - The entire original IP packet is encrypted and it becomes the payload in a new IP packet
  - Protects the end networks and the data (Site to site)



IPSec Modes – Two Types (Tunnel and Transport)

- **Transport**
  - **Only the IP payload is encrypted and the original IP headers are left intact**
  - **Protects only the data**
  - **Peer to peer**



IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 1**

  Workstation wants information from the Server and sends a frame.

IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 2**

  Gateway A sees the destination and knows that the policy for talking to the Server's IP Address requires a secure tunnel with Gateway B.

IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 3**

  Gateway A then creates an Uni-directional IPSec SA for the Workstation

IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 4**

  The information from the Workstation is transformed by Gateway A and sent to Gateway B which in turns authenticates and/or decrypts it before forwarding to the Server.

IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 5**

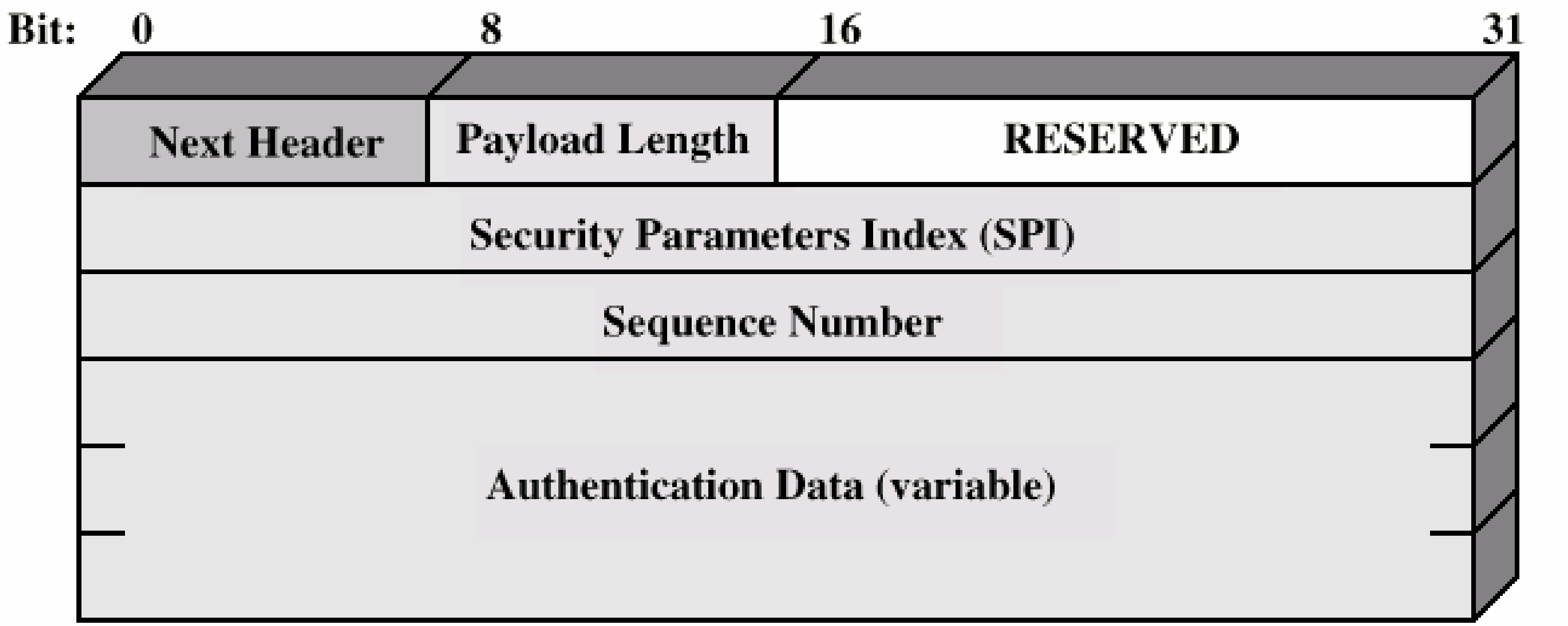
  The Server responds and Gateway B creates an Uni-directional SA to Gateway A and forwards the information.

IPSec In Action (Preshared Keys, Tunnel Mode)

- **Step 6**

  Now that IPSec SAs are created, information is freely shared between the Workstation and Server.

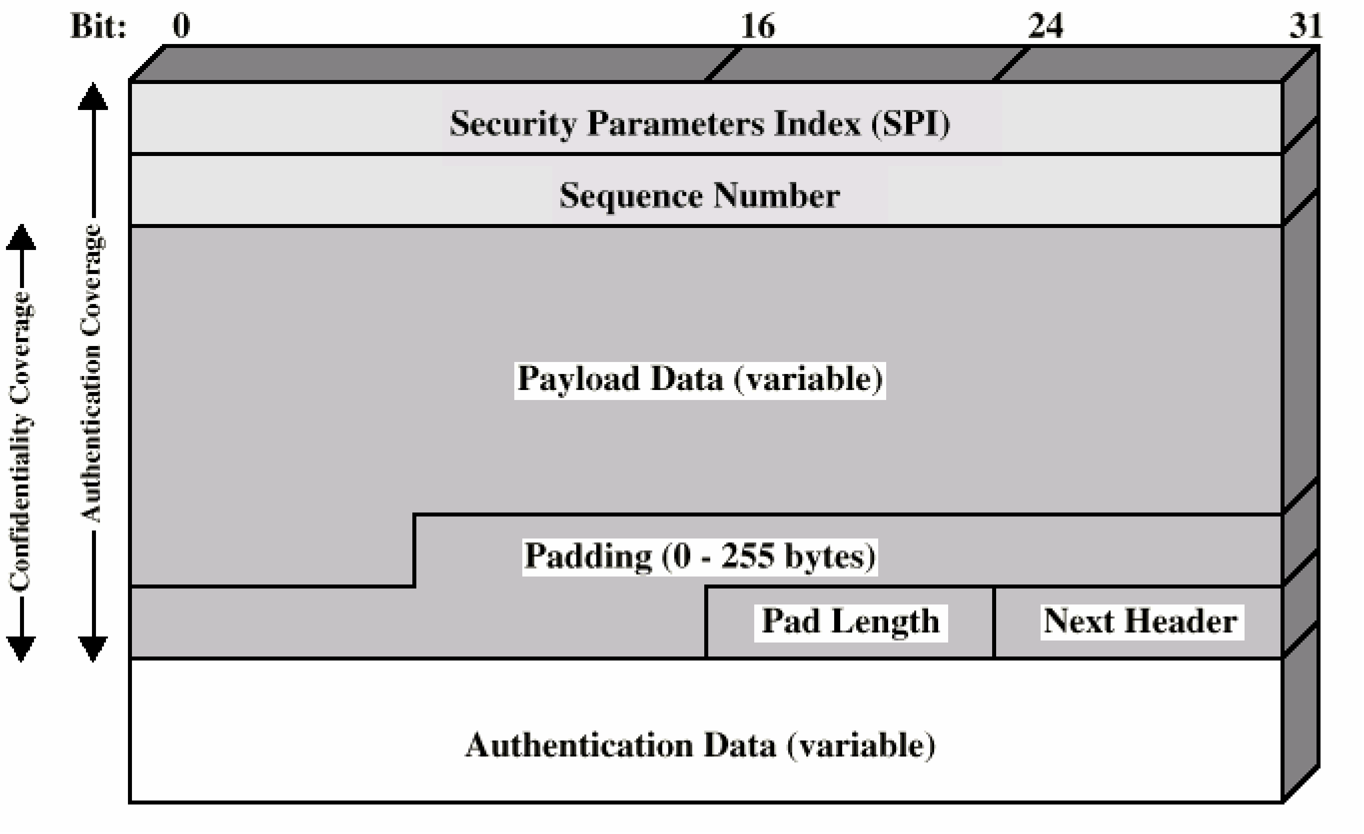
Authentication Header



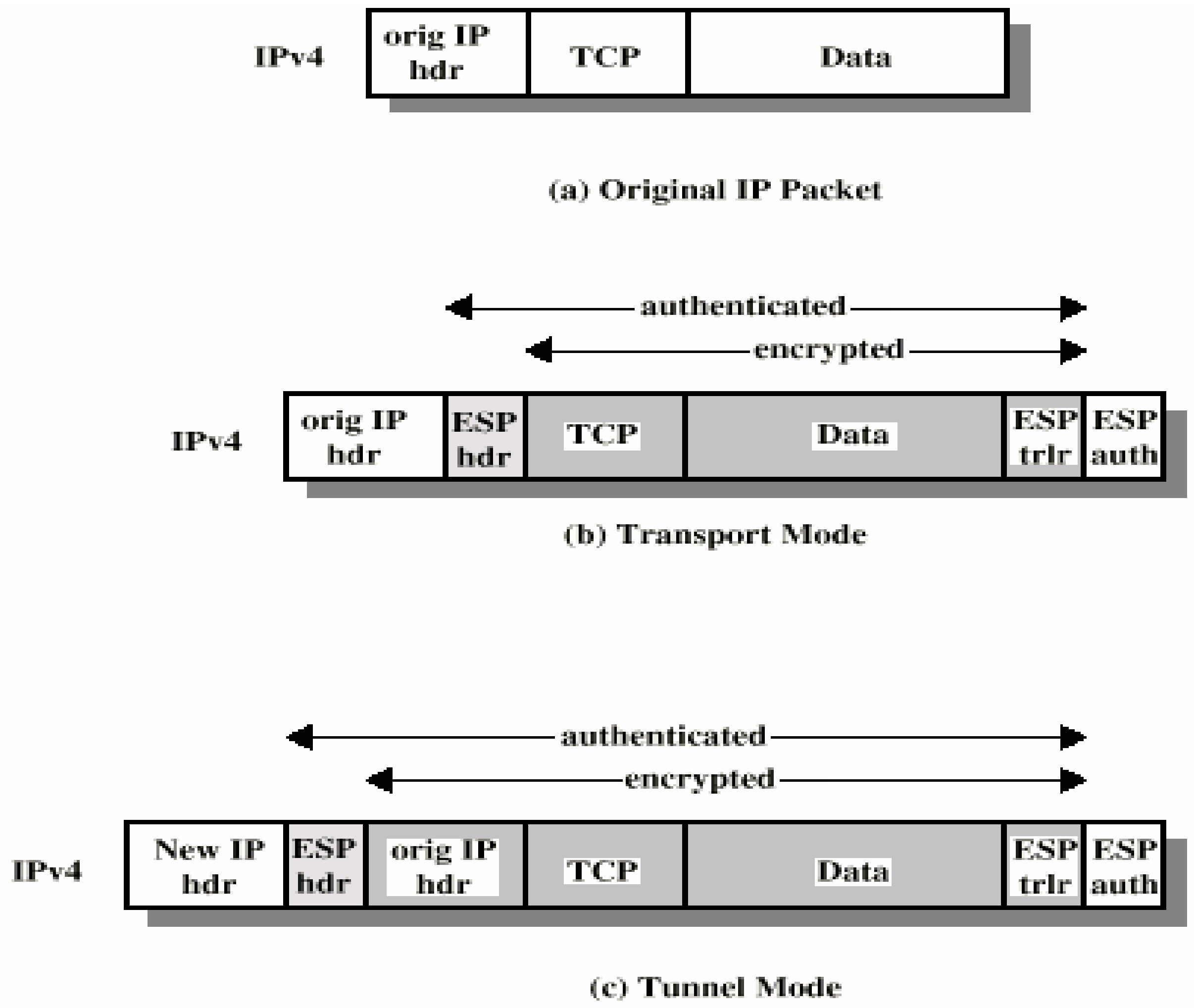
Encapsulating Security Payload

- ESP
- Confidentiality services

ESP Packet


Bit:
0
16
24
31
Security Parameters Index (SPI)
Sequence Number
Payload Data (variable)
Padding (0 - 255 bytes)
Pad Length
Next Header
Authentication Data (variable)
Confidentiality Coverage
Authentication Coverage

Scope of ESP


IPv4
orig IP hdr
TCP
Data
(a) Original IP Packet
authenticated
encrypted
IPv4
orig IP hdr
ESP hdr
TCP
Data
ESP trlr
ESP auth
(b) Transport Mode
authenticated
encrypted
IPv4
New IP hdr
ESP hdr
orig IP hdr
TCP
Data
ESP trlr
ESP auth
(c) Tunnel Mode

Key Management

- Manual
- Automatic
  - ISAKMP/Oakley
    - Oakley key determination protocol
    - Internet security association and key management protocol

What is Diffie-Hellman?

- Diffie-Hellman key agreement protocol
  - also called exponential key agreement
  - developed by Diffie and Hellman in 1976
  - published in the ground-breaking paper "New Directions in Cryptography “
- The protocol allows two users to exchange a secret key over an insecure medium
  - without any prior secrets.

What is Diffie-Hellman? (cont.)

- two system parameters *p* and *g*.
  - They are both public and may be used by all the users in a system.
  - *p* is a prime number
  - *g* (usually called a generator) is an integer less than *p*
- Property
  - for every number *n* between 1 and *p*-1 inclusive, there is a power *k* of *g* such that $n = g^k \bmod p$.

What is Diffie-Hellman? (cont.)

- Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol
- proceed as follows:
  - First, Alice generates a random private value *a* and Bob generates a random private value *b*, Both *a* and *b* are drawn from the set of integers
  - Then they derive their public values using parameters *p* and *g* and their private values. Alice's public value is g^a mod p and Bob's public value is g^b mod p
  - They then exchange their public values
  - Finally, Alice computes $g^{ab} = (g^b)^a$ mod p, and Bob computes $g^{ba} = (g^a)^b$ mod p. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k.

What is Diffie-Hellman? (cont.)

- The protocol depends on the discrete logarithm problem for its security
- It assumes that
  - it is computationally infeasible to calculate the shared secret key $k = g^{ab} \bmod p$
  - given the two public values $g^a \bmod p$ and $g^b \bmod p$ when the prime p is sufficiently large
  - Maurer has shown that breaking the Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions.

What is Diffie-Hellman? (cont.)

- The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack
  - In this attack, an opponent Carol intercepts Alice's public value and sends her own public value to Bob
  - When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice.
  - Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key
  - After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party
  - This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants
  - Possible solutions include the use of digital signatures and other protocol variants.

What is Diffie-Hellman? (cont.)

- The authenticated Diffie-Hellman key agreement protocol, or Station-to-Station (STS) protocol
  - was developed by Diffie, van Oorschot, and Wiener in 1992
  - defeat the man-in-the-middle attack on the Diffie-Hellman key agreement protocol
- The immunity is achieved by allowing the two parties to authenticate themselves to each other by the use of digital signatures and public-key certificates

IPSec Component - IKE (2 Phase Process)

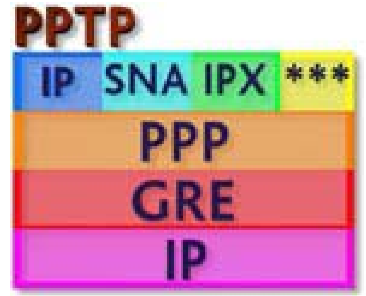
- Phase I: Creation of an IKE SA
  - **Pre-shared keys or a 3rd party digital certificate vendor for verification (X.509).**
  - **Describes the AH and ESP to be used.**
- Phase II: Creation of Uni-directional IPSec SA
  - **Uses the information from IKE to create Uni-directional SA.**

IPSec Component - IKE (2 Phase Process)

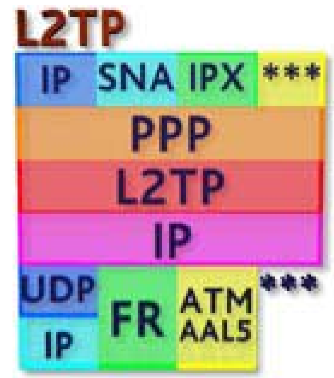
- Phase I: Creation of an IKE SA
  - Two types of tunnel creation modes:
    - **Main – 6 Step handshake**
    - **Aggressive – 3 Step handshake (faster but does not provide identity protection)**
- Phase II: Creation of Uni-directional IPSec SA
  - **One type of tunnel creation mode: Quick**

VPN Technologies

Layer 2



PPTP – Developed by Microsoft for dial-up and LAN-to-LAN PPP connections.



L2TP – Developed by IETF based on Cisco’s L2F to run over any technology. Uses IPSec for encryption.

Layer 3

IPSec – Developed by IETF for VPN over IP and is used by IPv6.

Required Reading

- ⌘Stallings chapter 18