計算機ネットワーク

開講クォーター: 1-2Q

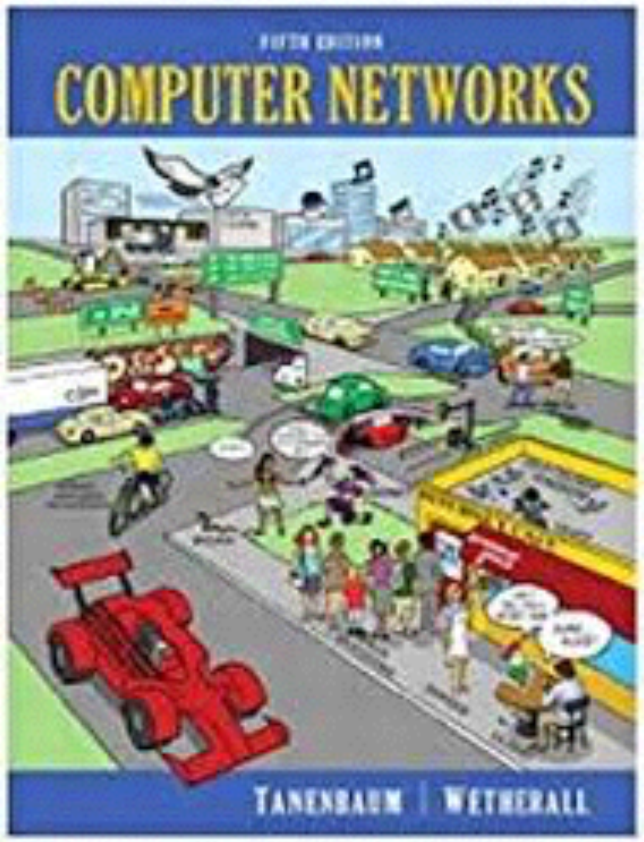
曜日・時限: 火7-8限

講義室: 1Q @ W834, 2Q @ W931

横田理央

rioyokota@gsic.titech.ac.jp

参考書



教科書

講義日程（2Q）

		授業計画		課題
06/14	第９回	ネットワーク層１ ルーティング・輻輳制御	５章	ルーティングの種類を理解し 輻輳制御手法を説明できる
06/21	第１０回	ネットワーク層２ インターネットとサービス品質	５章	インターネットの制御プロトコルを理解し ネットワーク間の接続について説明できる
06/28	第１１回	トランスポート層１ トランスポート・プロトコルの要素	６章	誤り制御とフロー制御を理解し 輻輳制御について説明できる
07/05	第１２回	トランスポート層２ UDP と TCP	６章	TCP の信頼性を理解し TCP のコネクション管理を説明できる
07/12	第１３回	アプリケーション層 DNS, 電子メール, www	７章	DNS, 電子メール, www のしくみを理解し ストリーミング，P2P について説明できる
07/26	第１４回	ネットワークセキュリティ１ 対称鍵暗号，公開鍵暗号	８章	暗号アルゴリズムを理解し SHA-1,2 と RSA について説明できる
08/02	第１５回	ネットワークセキュリティ２ デジタル署名，認証プロトコル	８章	電子メール,Web のセキュリティ の脅威について把握できる

Network Security

Categories of network security

1. Secrecy (機密)
2. Authentication (認証)
3. Nonrepudiation (否認防止)
4. Integrity control (一貫性)

People who may cause security problems

Adversary	Goal
Student	To have fun snooping on people's email
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Corporation	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Identity thief	To steal credit card numbers for sale
Government	To learn an enemy's military or industrial secrets
Terrorist	To steal biological warfare secrets

Kerckhoff'sprinciple:

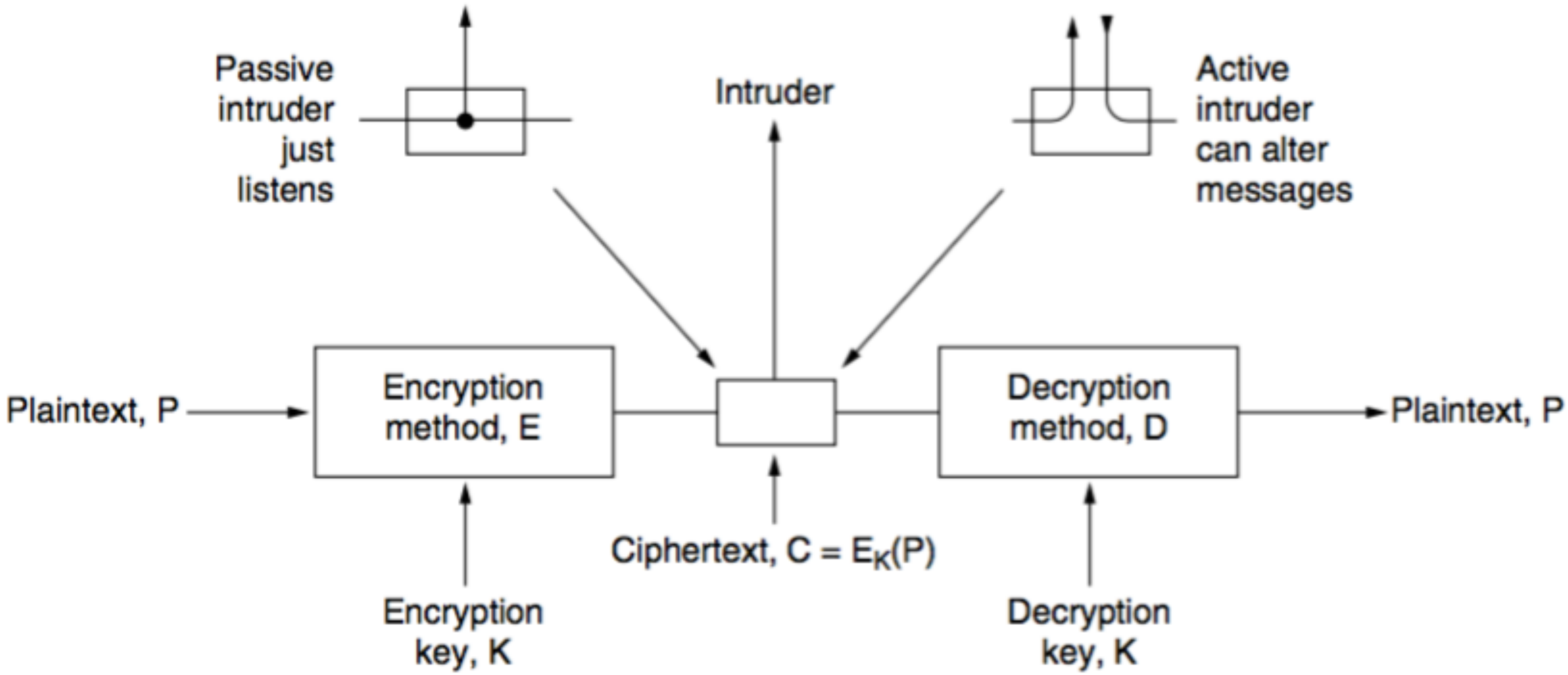
All algorithms must be public; only the keys are secret

Cryptography

Categories of cryptography

1. Cipher: Character-for-character or bit-for-bit transformation
2. Code: Replaces one word with another word or symbol

   Navaho Indians code during world war II



Encryption model

plaintext: Message to be encrypted

key: Encryption key

ciphertext: Output of the encryption process

intruder: Third party with malicious intent

$$D_K(E_K(P)) = P$$

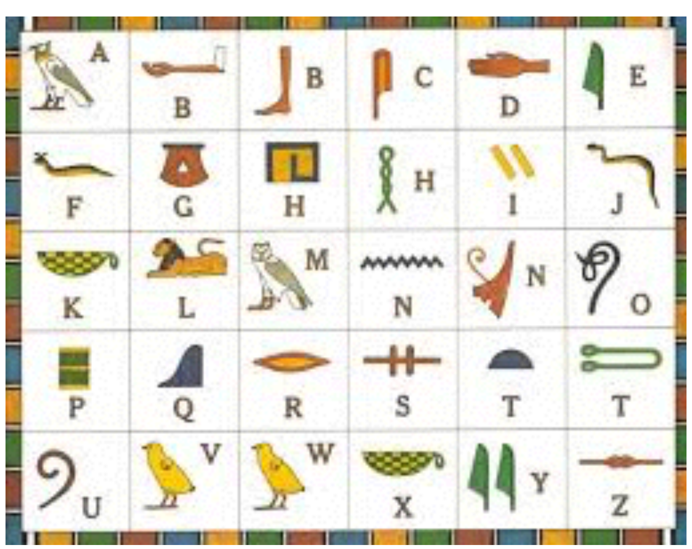
Substitution Ciphers

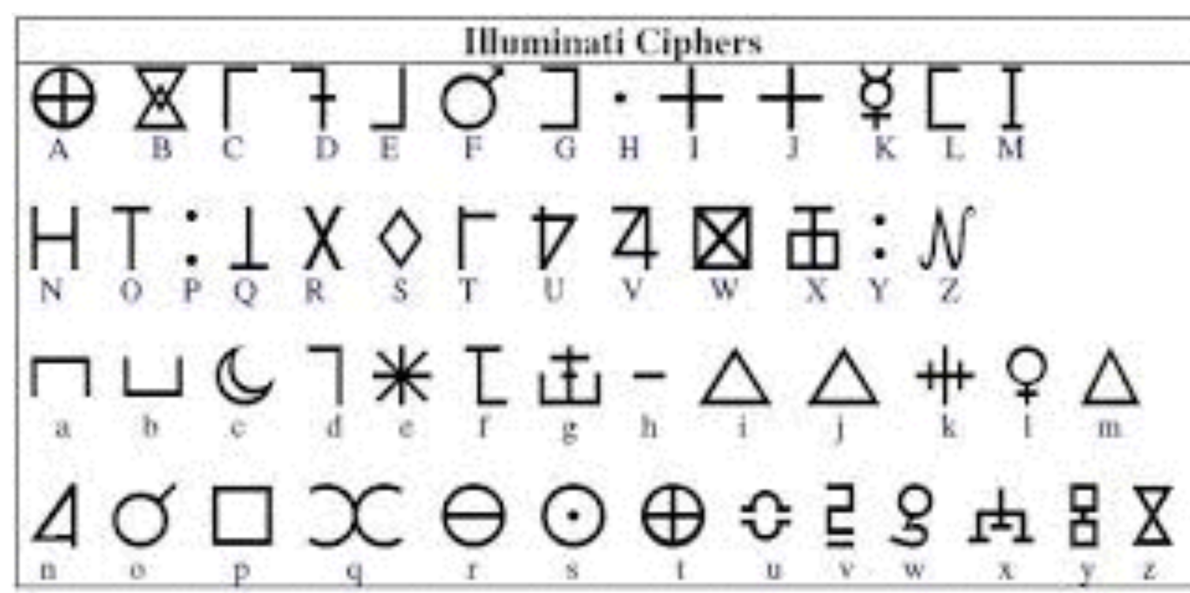
plaintext: a b c d e f g h i j k l m n o p q r s t u v w x y z
ciphertext: Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

Monoalphabetic substitution cipher: e.g. Caesar's cipher

Attack (攻略法): Take advantage of the statistical properties of natural languages

1. Count the frequencies of all letters in the text
2. Look at diagrams and trigrams





Transposition Ciphers

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Attack (攻略法)

1. First be aware that he is dealing with a transposition cipher
2. Make a guess at the number of columns
3. Reorder the columns

One-Time-Pads

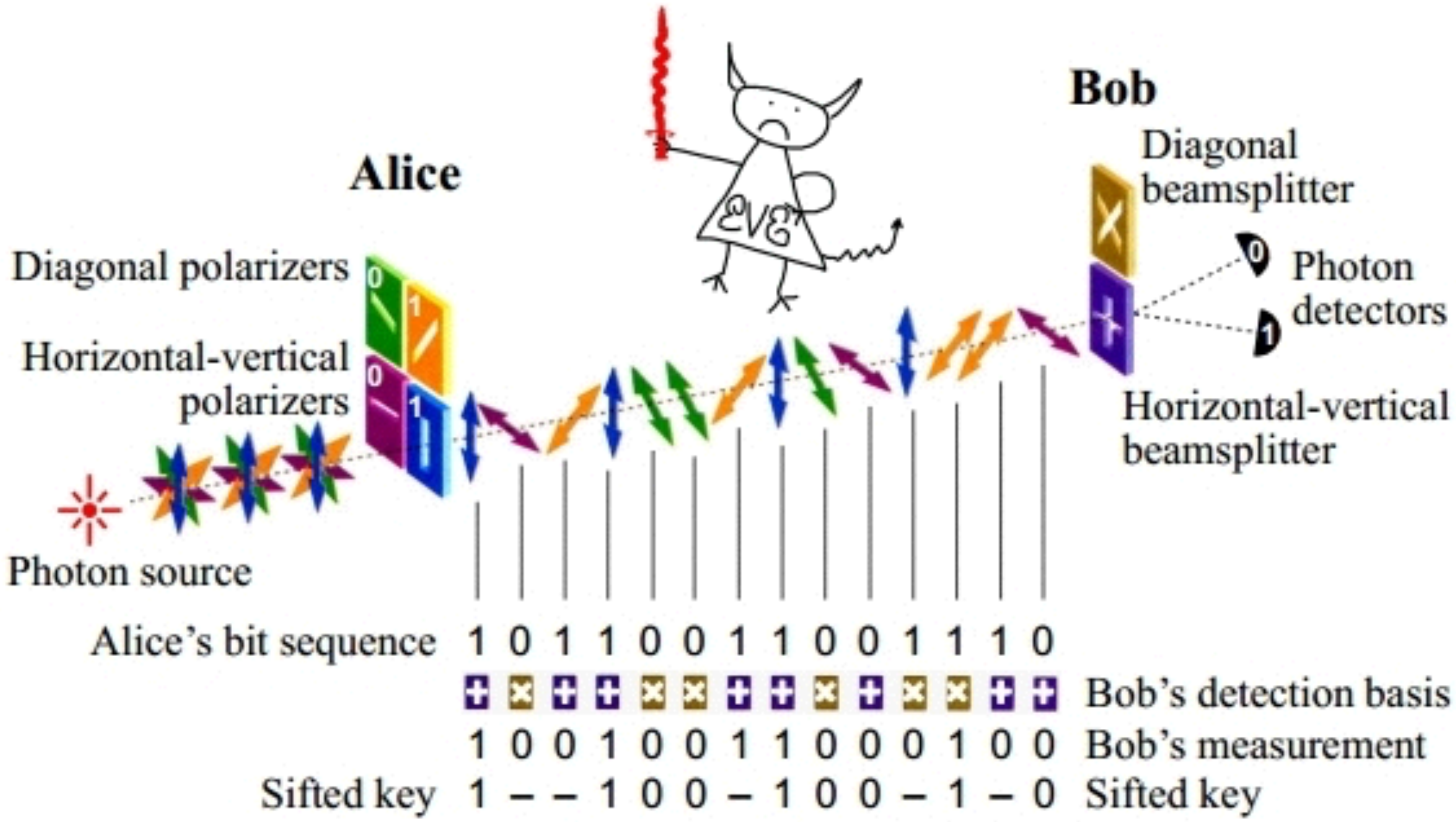
Message 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Pad 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Ciphertext:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101

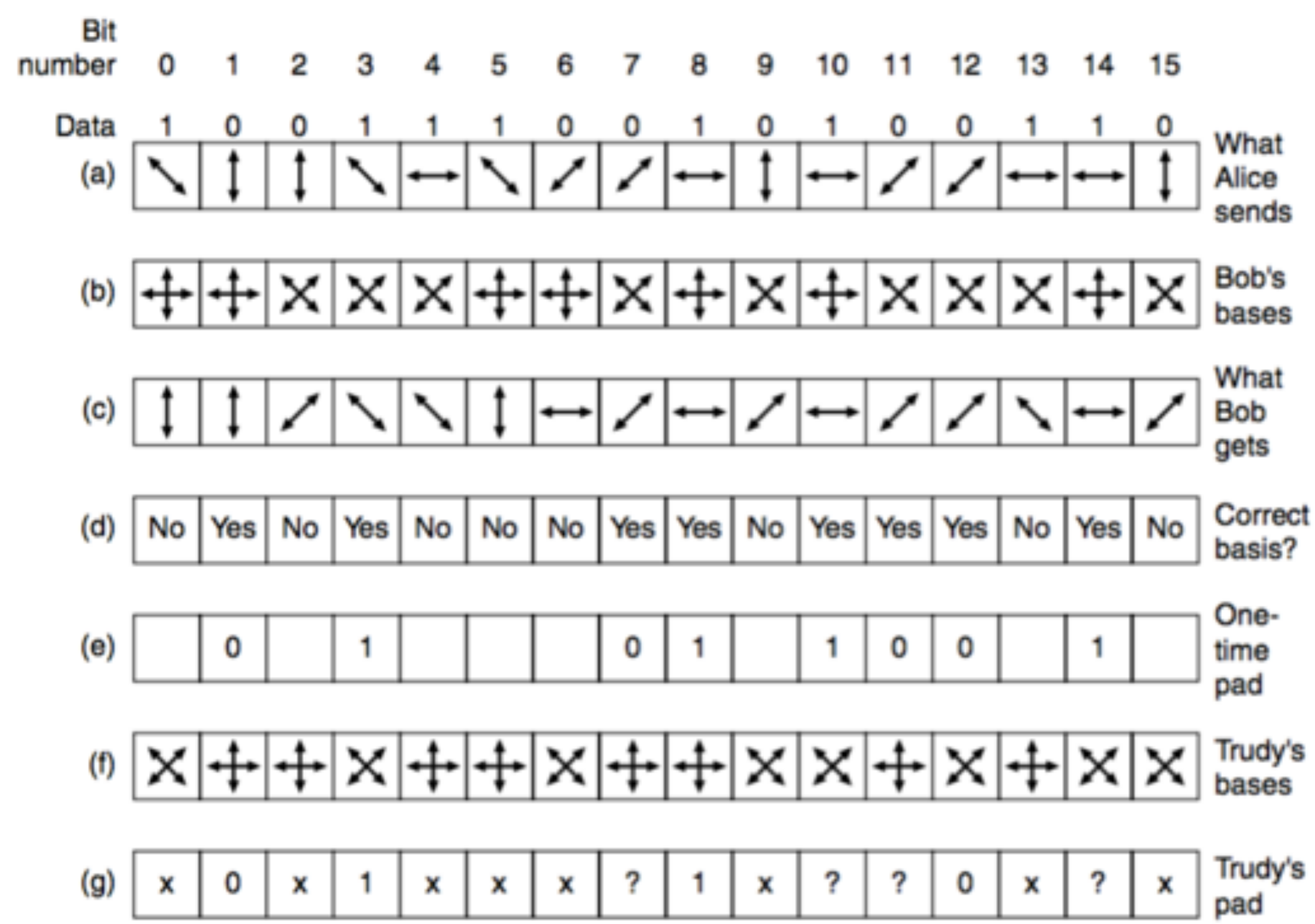
Pad 2:	1011110	0000111	1101000	1010011	1010111	0100110	1000111	0111010	1001110	1110110	1110110
Plaintext 2:	1000101	1101100	1110110	1101001	1110011	0100000	1101100	1101001	1110110	1100101	1110011

<u>Drawbacks</u>

1. Truly random (as opposed to pseudorandom) one-time pad values, which is a non-trivial requirement.
2. Secure generation and exchange of the one-time pad values, which must be at least as long as the message.
3. Careful treatment to make sure that it continues to remain secret, and is disposed of correctly preventing any reuse in whole or part —hence "one time".

Quantum Cryptography





Principles of Cryptography

<u>Redundancy</u>

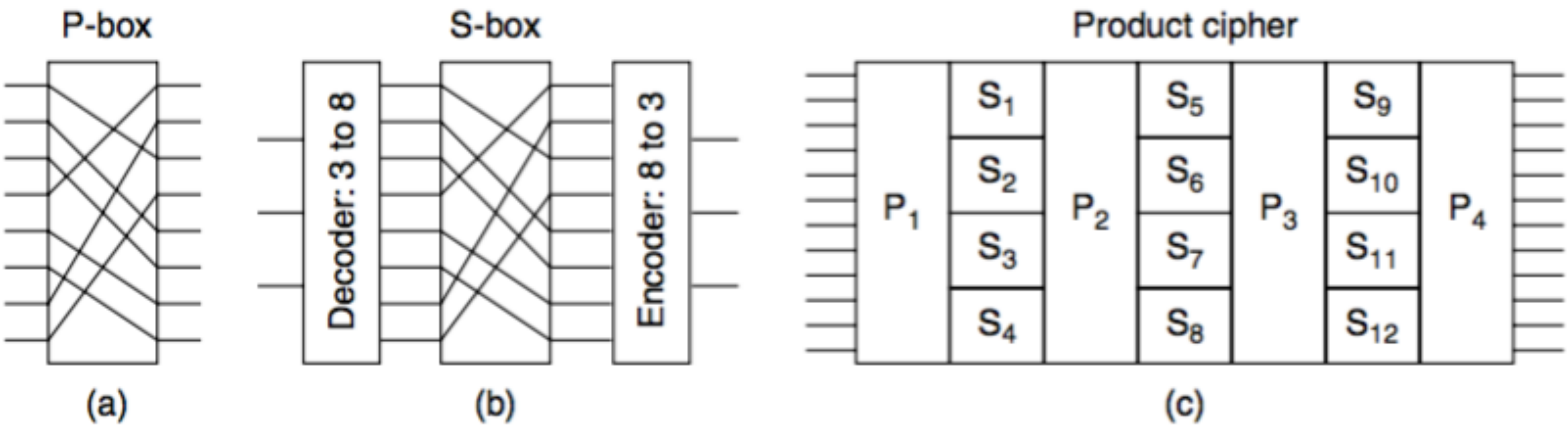
1. For TCP, almost every 3-byte message is valid
2. Intruders cannot read, but can manipulate the message
3. This problem can be solved by the addition of redundancy
4. However, adding redundancy makes it easier for cryptanalysts to break messages

<u>Freshness</u>

1. To prevent active intruders from playing back old messages
2. One such measure is including in every message a timestamp
3. Measures other than timestamps will be discussed later

Symmetric-key Algorithms

- Same key is used for encryption and decryption -

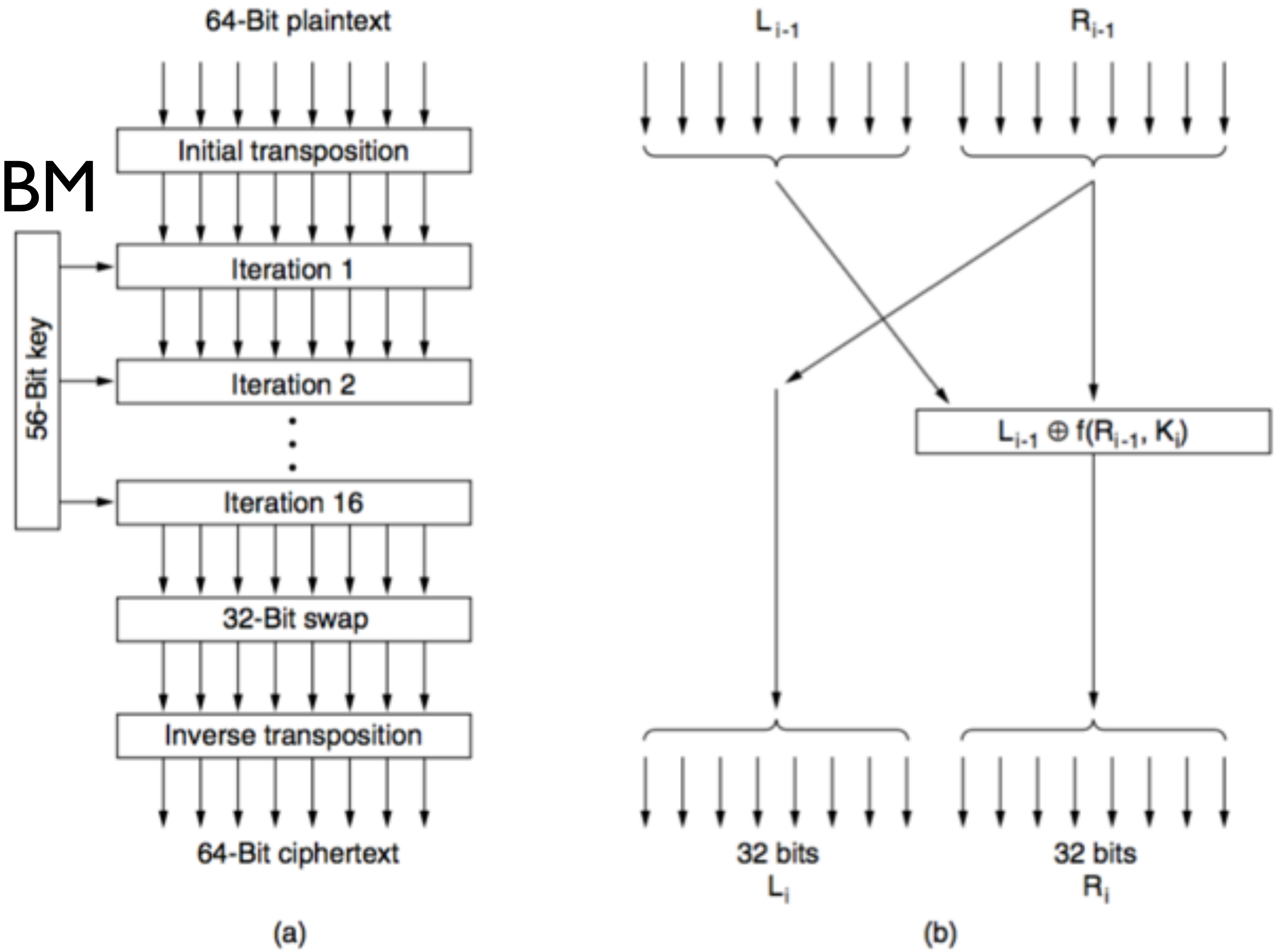


P-box: Can be made to perform any transposition and do it at practically the speed of light since no computation is involved

S-box: The 3-bit input selects one of the eight lines exiting from the first stage and sets it to 1; all the other lines are 0. The second stage is a P-box. The third stage encodes the selected input line in binary again

DES — The Data Encryption Standard

1977 Developed by IBM
NSA had discussions with IBM
128-bit reduced to 56-bit



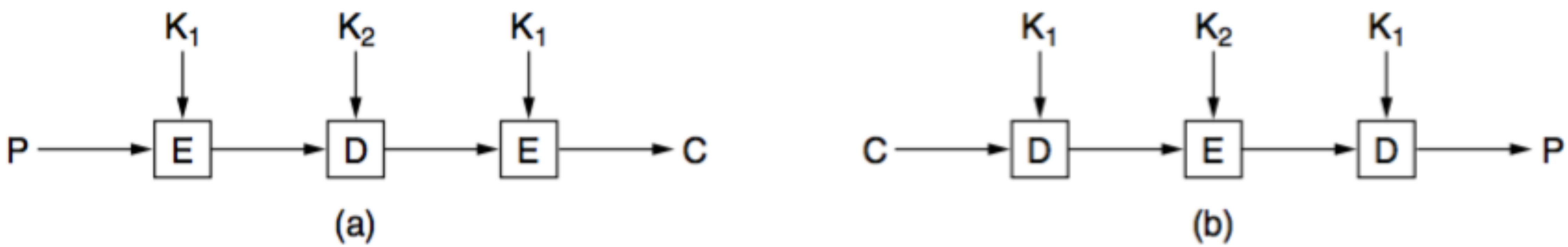
1. A 48-bit number, *E*, is constructed by expanding the 32-bit R_{i-1} according to a fixed transposition and duplication rule
2. E and Ki are XORed together
3. This output is then partitioned into eight groups of 6 bits each, each of which is fed into a different S-box
4. Each of the 64 possible inputs to an S-box is mapped onto a 4- bit output
5. These 8 × 4 bits are passed through a P-box

Triple DES

1979 IBM realized that the DES key length was too short

Two keys and three stages are used

1. The plaintext is encrypted using DES in the usual way with K_1
2. DES is run in decryption mode, using K_2 as the key
3. Another DES encryption is done with K_1



Encryption Decryption

AES—The Advanced Encryption Standard

1997 NIST (National Institute of Standards and Technology) sponsored a contest for a new standard

1. The algorithm must be a symmetric block cipher.
2. The full design must be public.
3. Key lengths of 128, 192, and 256 bits must be supported.
4. Both software and hardware implementations must be possible.
5. The algorithm must be public or licensed on nondiscriminatory terms.

2000, NIST announced that it had selected Rijndael, by Joan Daemen and Vincent Rijmen

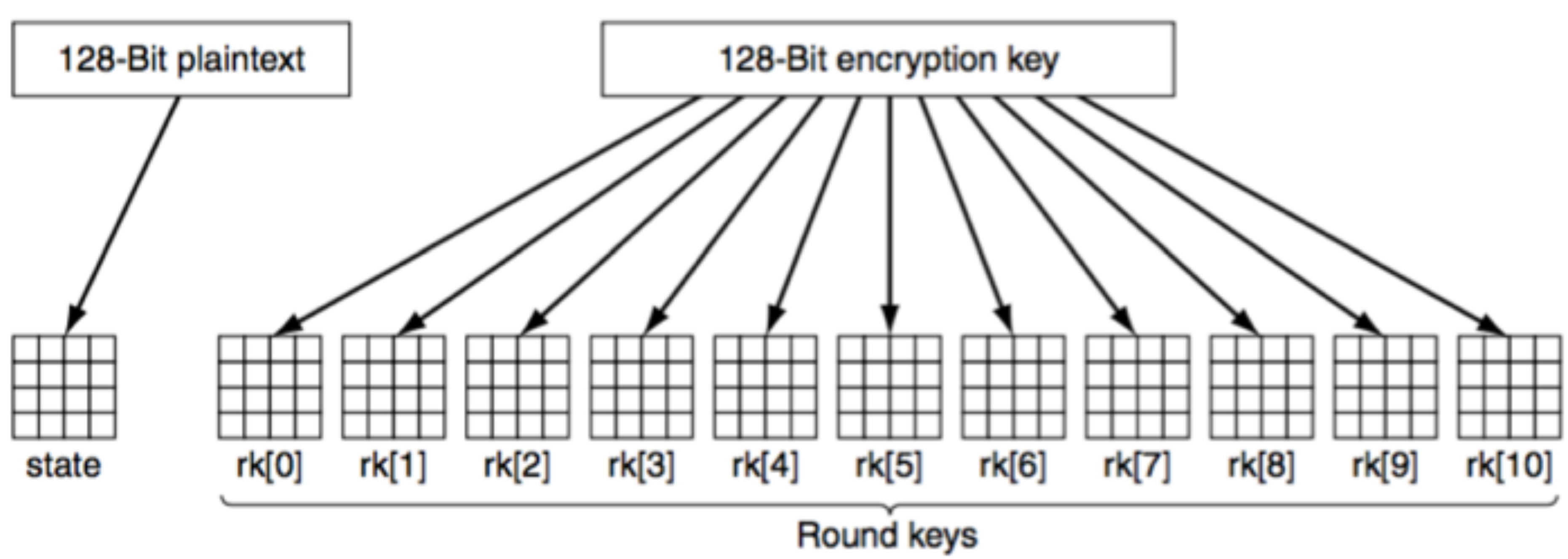
Rijndael

```
#define LENGTH 16                                    /* # bytes in data block or key */
#define NROWS 4                                      /* number of rows in state */
#define NCOLS 4                                      /* number of columns in state */
#define ROUNDS 10                                    /* number of iterations */
typedef unsigned char byte;                          /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
 int r;                                              /* loop index */
 byte state[NROWS][NCOLS];                           /* current state */
 struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];      /* round keys */

 expand_key(key, rk);                                /* construct the round keys */
 copy_plaintext_to_state(state, plaintext);          /* init current state */
 xor_roundkey_into_state(state, rk[0]);              /* XOR key into state */

 for (r = 1; r <= ROUNDS; r++) {
      substitute(state);                             /* apply S-box to each byte */
      rotate_rows(state);                            /* rotate row i by i bytes */
      if (r < ROUNDS) mix_columns(state);            /* mix function */
      xor_roundkey_into_state(state, rk[r]);         /* XOR key into state */
 }
 copy_state_to_ciphertext(ciphertext, state);        /* return result */
}
```

Rijndael

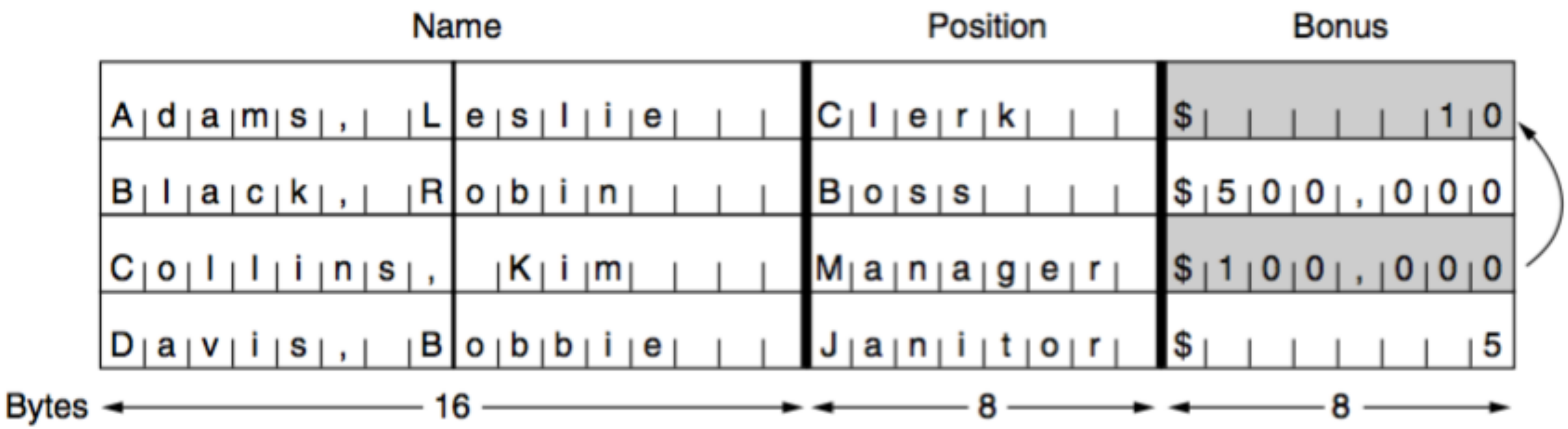
The loop executes 10 iterations, one per round, transforming state on each iteration. The contents of each round is produced in four steps.

1. Do a byte-for-byte substitution on state. Each byte in turn is used as an index into an S-box to replace its value by the contents of that S-box entry. This step is a straight monoalphabetic substitution cipher.
2. Rotate each of the four rows to the left. Row 0 is rotated 0 bytes (i.e., not changed), row 1 is rotated 1 byte, row 2 is rotated 2 bytes, and row 3 is rotated 3 bytes. This step diffuses the contents of the current data around the block.
3. Mix up each column independently of the other ones. The mixing is done using matrix multiplication in which the new column is the product of the old column and a constant matrix, with the multiplication done using the finite Galois field.
4. XOR the key for this round into the state array for use in the next round.

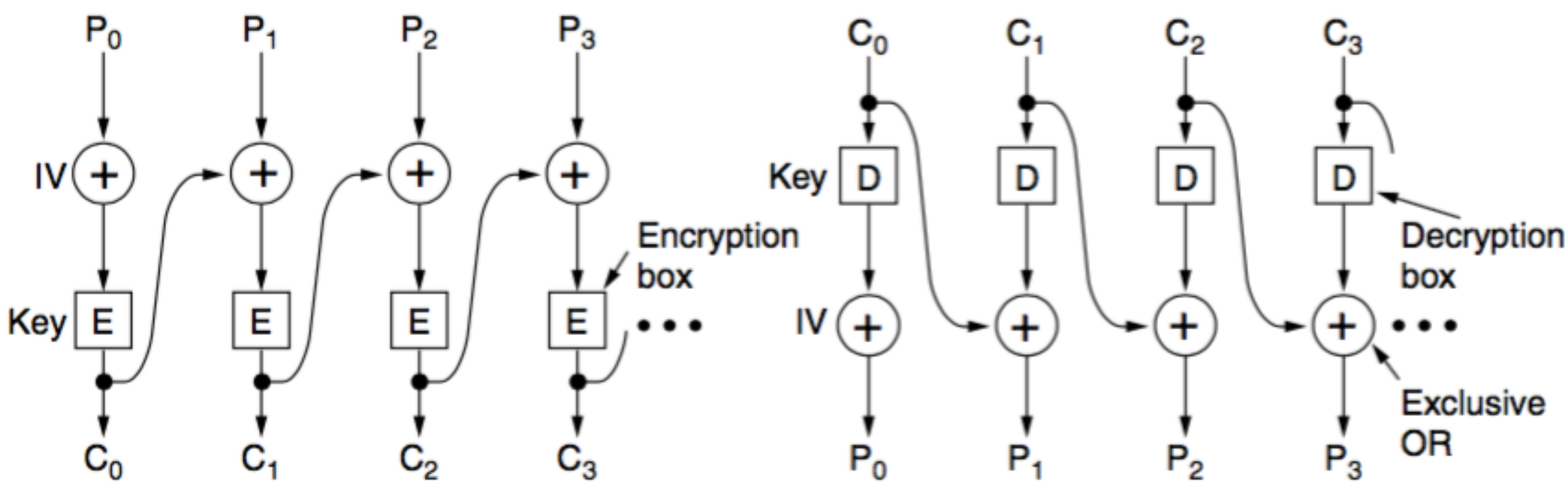


Cipher Modes

Electronic Codebook Mode



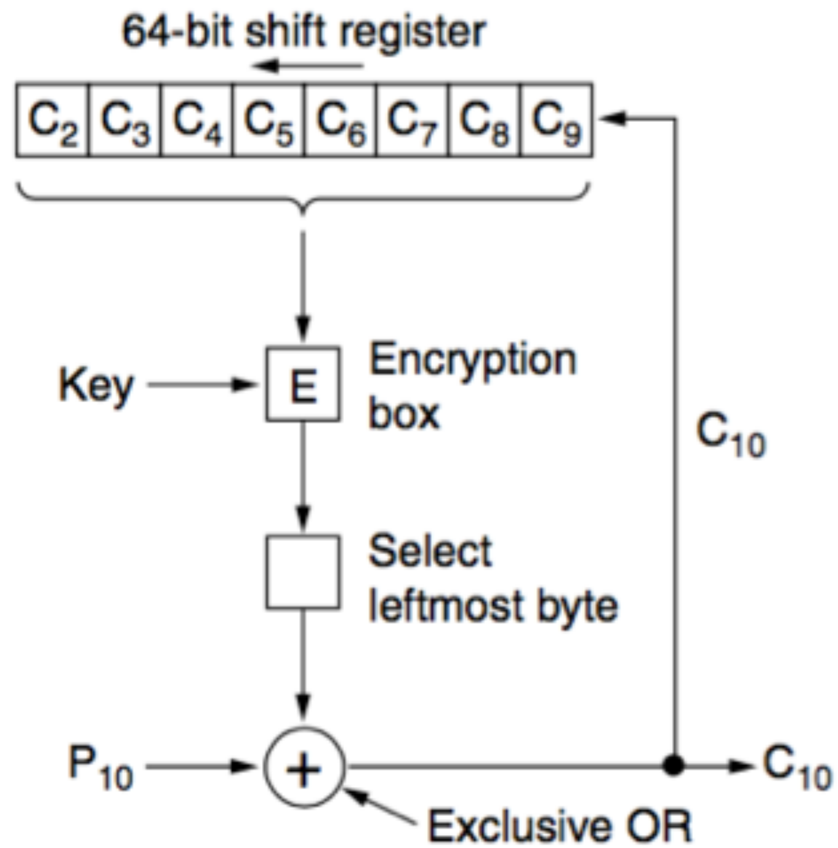
Cipher Block Chaining Mode



(a) (b)

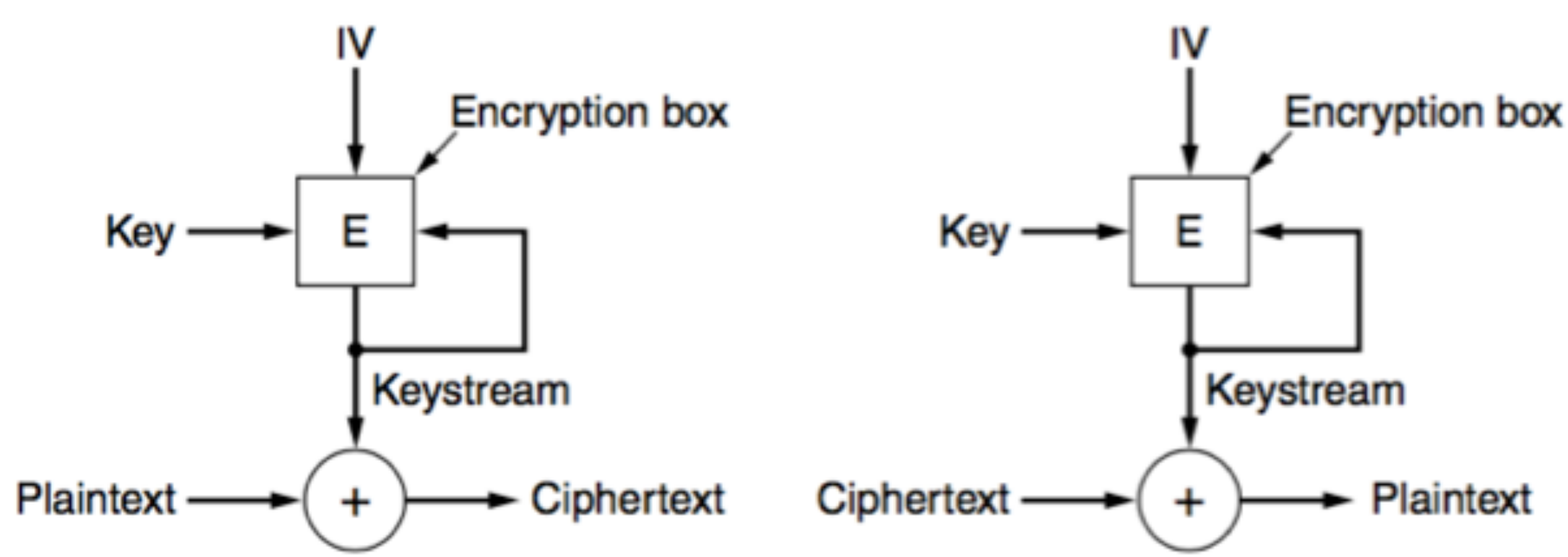
Cipher Modes

Cipher Feedback Mode



(a) (b)

Stream Cipher Mode

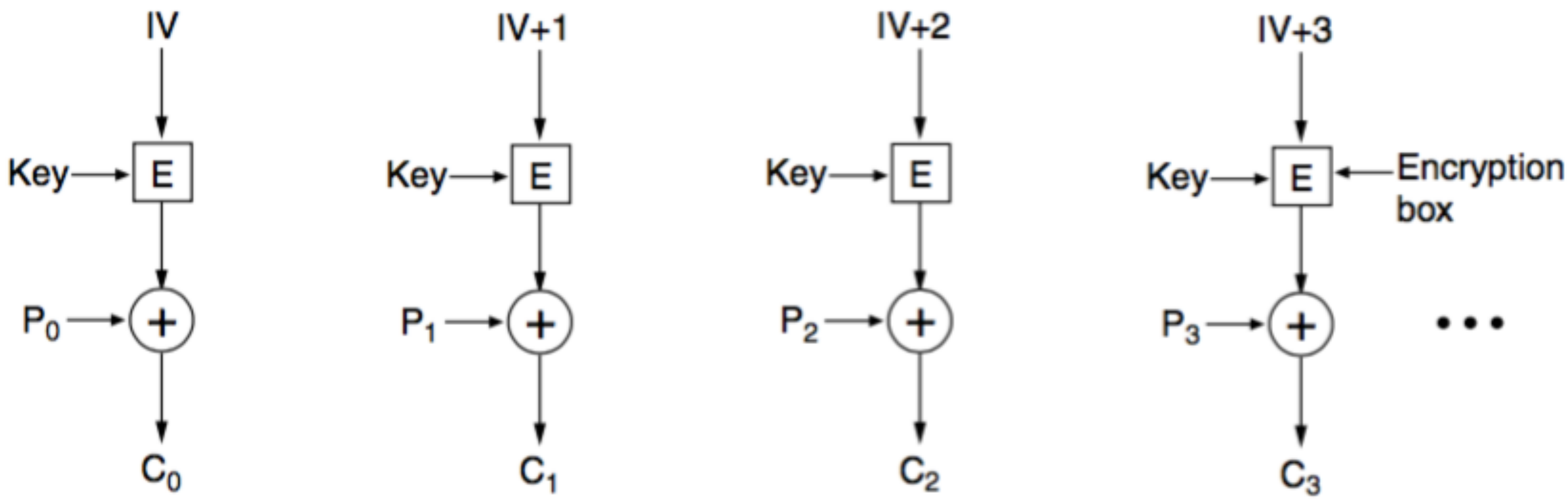


(a) (b)

Cipher Modes

Counter Mode

Random access



Other Ciphers

Cipher	Author	Key length	Comments
DES	IBM	56 bits	Too weak to use now
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
AES (Rijndael)	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Good, but getting old
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

Cryptanalysis

Differential cryptanalysis: Can be used to attack any block cipher

Linear cryptanalysis: XOR bits in the plaintext and ciphertext together and use the bias in the result to decipher faster

Power measurements: Processing a 1 takes more electrical energy than processing a 0

Timing analysis: If the then and else parts take different amounts of time

Public-Key Algorithms


1976 Proposed by Diffie and Hellman at Stanford

1. D(E(P)) = P.
2. It is exceedingly difficult to deduce D from E.
3. E cannot be broken by a chosen plaintext attack.

RSA

1. Choose two large primes, p and q (typically 1024 bits).
2. Compute $n = p \times q$ and $z = (p-1)\times(q-1)$.
3. Choose a number relatively prime to z and call it d.
4. Find e such that $e \times d = 1 \mod z$.

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation

Receiver's computation

Public-Key Algorithms

<u>Fastest Integer Factorization Algorithm</u>

GNFS (General Number Field Sieve)

$$O\left(\exp\sqrt[3]{\frac{64}{9}b(\log b)^2}\right)$$

<u>RSA (Rivest-Shamir-Adleman)</u>

Based on Galois fields (ガロア体)

<u>DSA (Digital Signature Algorithm)</u>

Discrete logarithm problem (離散対数)

<u>ECDSA (Elliptic Curve Digital Signature Algorithm)</u>

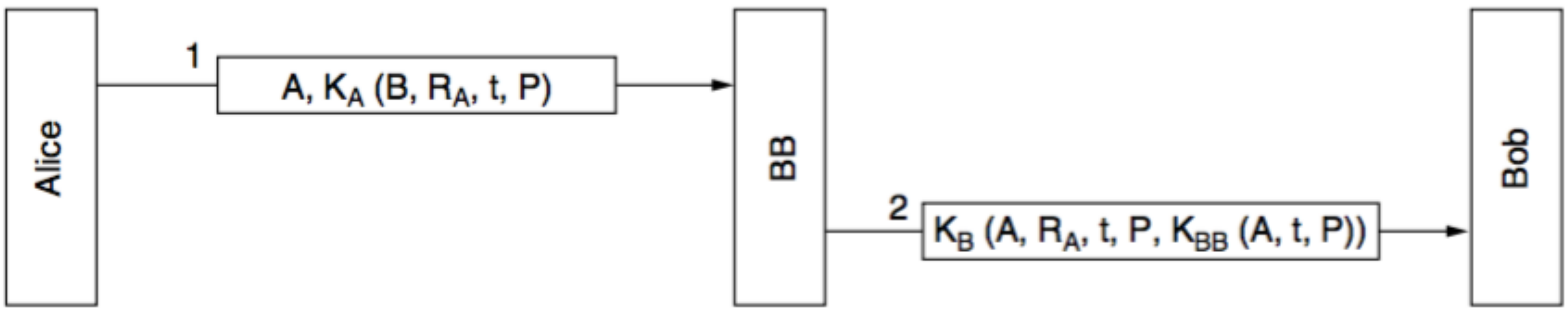
Based on elliptic curves (楕円曲線)

DSA is generally faster in decryption but slower for encryption

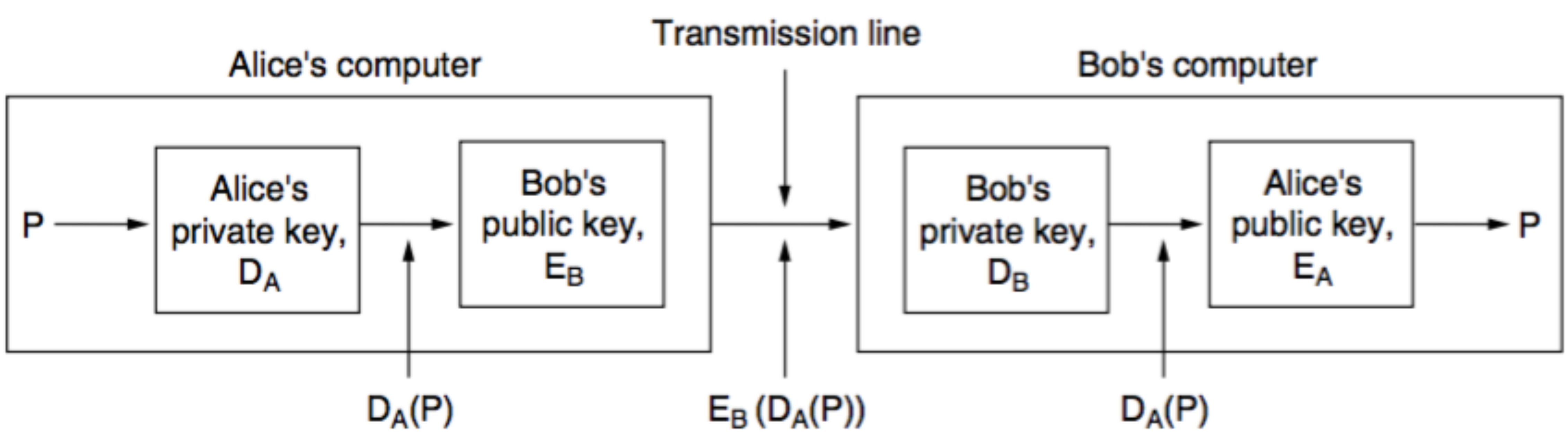
Digital Signatures

1. The receiver can verify the claimed identity of the sender.
2. The sender cannot later repudiate the contents of the message.
3. The receiver cannot possibly have concocted the message himself.

Symmetric-Key Signatures



Public-Key Signatures



Digital Signature Standard (DSS)

1991, NIST proposed using a variant of the El Gamal public-key algorithm for its new DSS

DSS was criticized for being

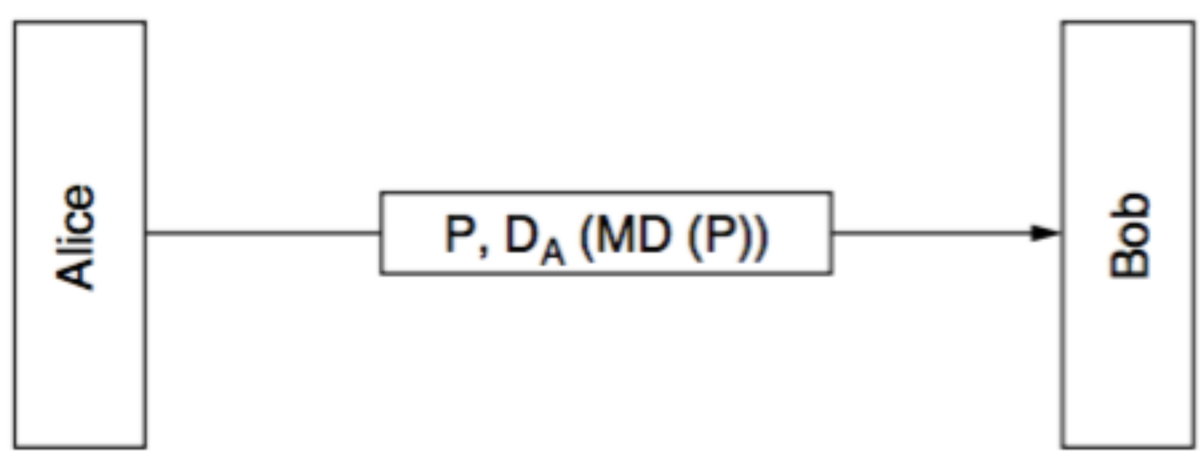
1. Too secret (NSA designed the protocol for using El Gamal).
2. Too slow (10 to 40 times slower than RSA for checking signatures).
3. Too new (El Gamal had not yet been thoroughly analyzed).
4. Too insecure (fixed 512-bit key).

Message Digests

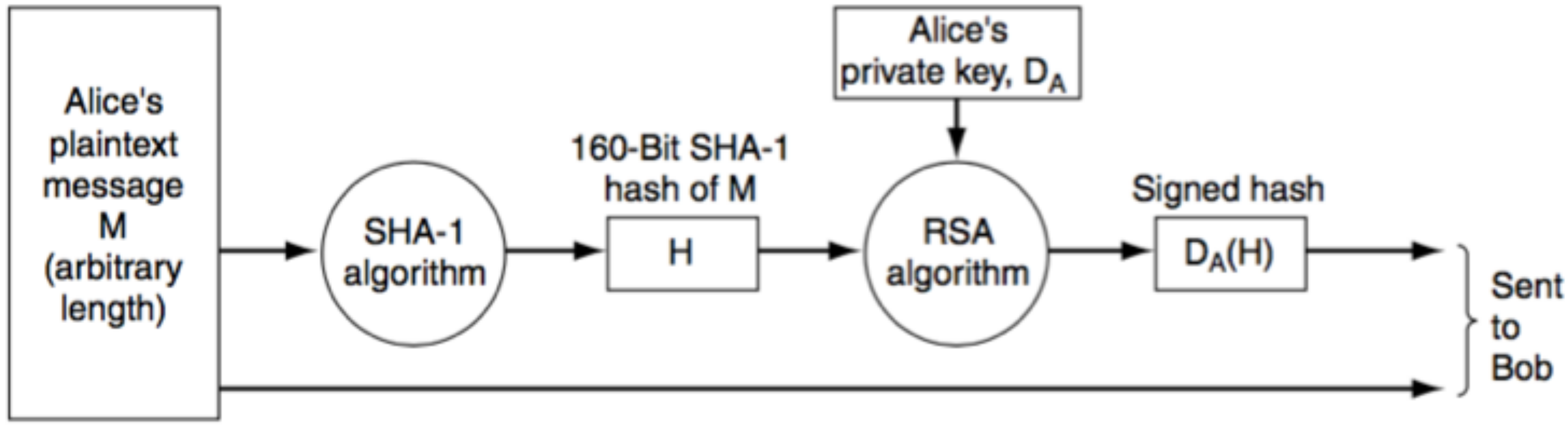
This scheme is based on the idea of a one-way hash function that takes an arbitrarily long piece of plaintext and from it computes a fixed-length bit string. This hash function, MD, often called a message digest, has four important properties:

1. Given P, it is easy to compute MD (P).
2. Given MD (P), it is effectively impossible to find P.
3. Given P, no one can find P′ such that MD (P′) = MD(P).
4. A change to the input of even 1 bit produces a very different output.

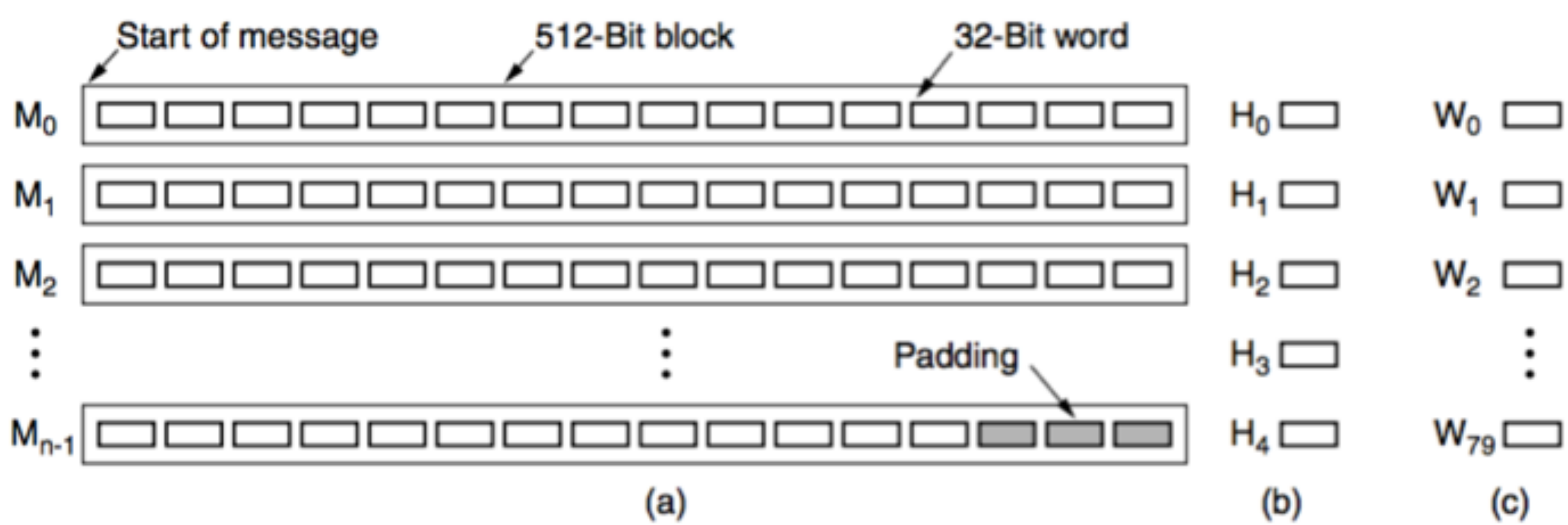
Computing a message digest from a piece of plaintext is much faster than encrypting that plaintext with a public-key algorithm, so message digests can be used to speed up digital signature algorithms.



SHA-1 and SHA-2



1. Pad the message by adding a 1 bit to the end, followed by as many 0 bits as are necessary
2. A 64-bit number containing the message length before padding is ORed into the low-order 64 bits
3. SHA-1 maintains five 32-bit variables, H0 through H4 , where the hash accumulates.
4. Each of the blocks M_0 through M_{n-1} is now processed in turn.



The Birthday Attack

Q. How many students do you need in a class before the probability of having two people with the same birthday exceeds 1/2?
A. 23

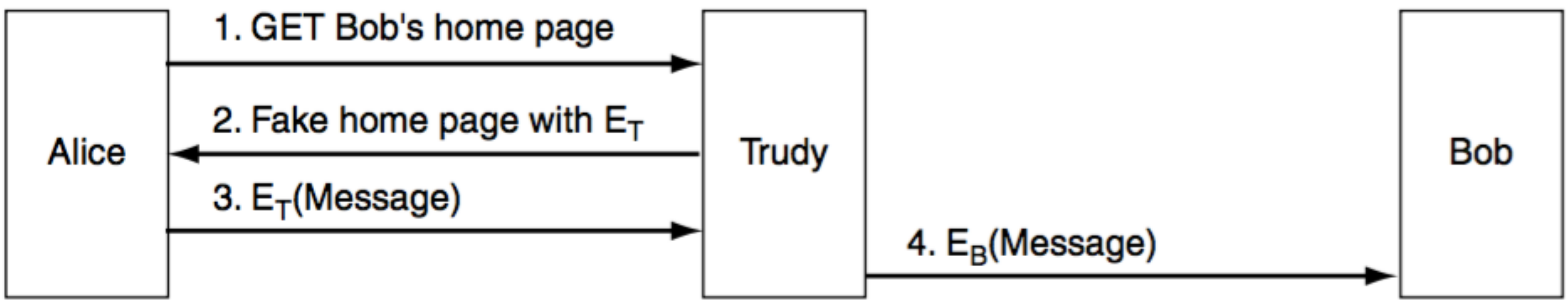
Q. How many operations to subvert an m-bit message digest?
A. $2^{m/2}$

Dear Dean Smith,
This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a candidate* | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Prof. Wilson for [*about* | *almost*] six years. He is an [*outstanding* | *excellent*] researcher of great [*talent* | *ability*] known [*worldwide* | *internationally*] for his [*brilliant* | *creative*] insights into [*many* | *a wide variety of*] [*difficult* | *challenging*] problems.

Dear Dean Smith,
This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a candidate* | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Tom for [*about* | *almost*] six years. He is a [*poor* | *weak*] researcher not well known in his [*field* | *area*]. His research [*hardly ever* | *rarely*] shows [*insight in* | *understanding of*] the [*key* | *major*] problems of [*the* | *our*] day.

Management of Public Keys

How to get each other's public keys to start the communication process?



<u>Certificates</u>

KDC key distribution center: Not scalable

CA (Certification Authority):

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

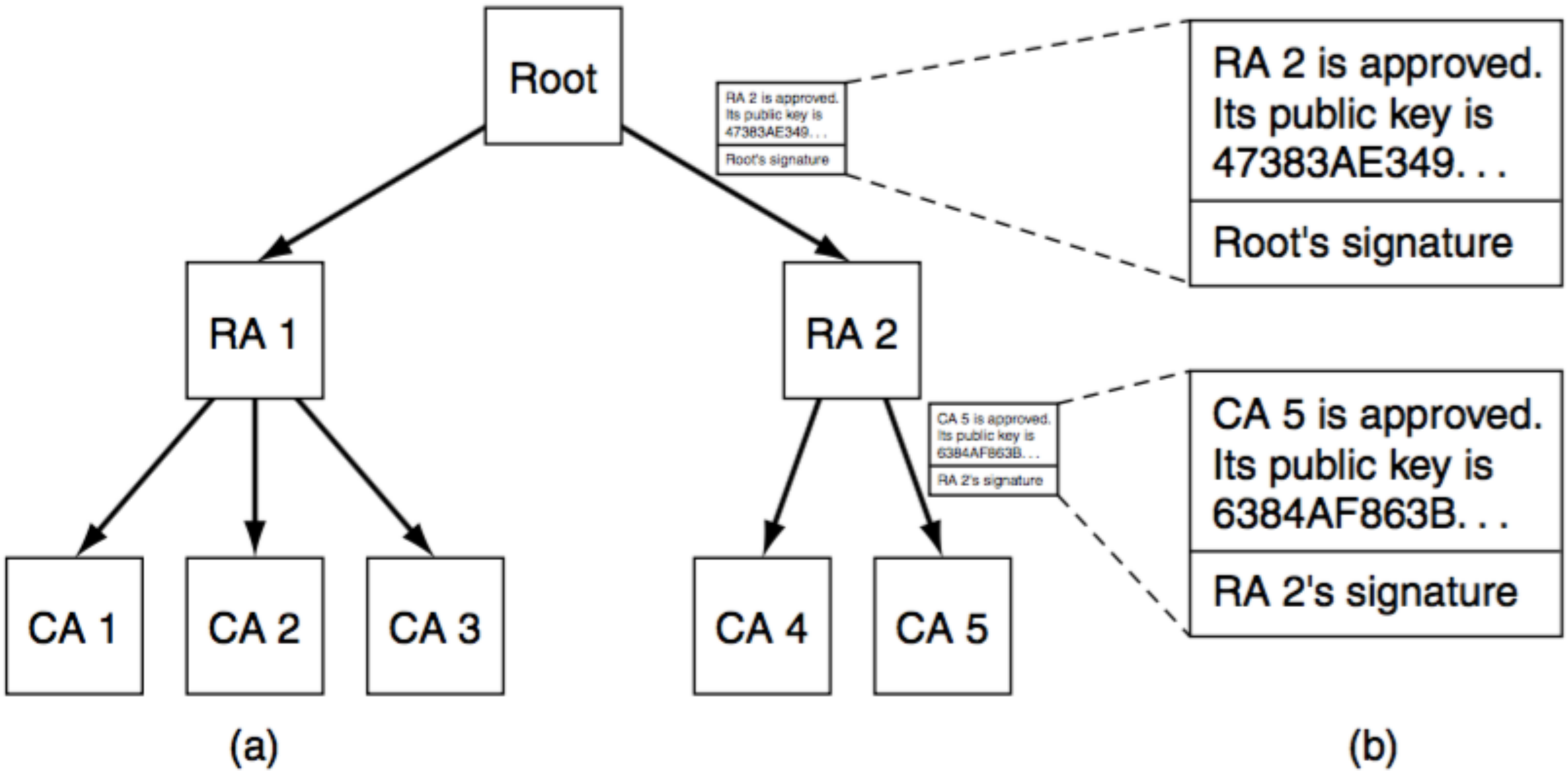
SHA-1 hash of the above certificate signed with the CA's private key

X.509 (Standard for Certificates)

Field	Meaning
Version	Which version of X.509
Serial number	This number plus the CA's name uniquely identifies the certificate
Signature algorithm	The algorithm used to sign the certificate
Issuer	X.500 name of the CA
Validity period	The starting and ending times of the validity period
Subject name	The entity whose key is being certified
Public key	The subject's public key and the ID of the algorithm using it
Issuer ID	An optional ID uniquely identifying the certificate's issuer
Subject ID	An optional ID uniquely identifying the certificate's subject
Extensions	Many extensions have been defined
Signature	The certificate's signature (signed by the CA's private key)

Certificates are encoded using OSI ASN.1
(Abstract Syntax Notation 1)

Public Key Infrastructures



Q. Where to store the certificates?

A. DNS servers, dedicated servers

Q. How to revoke a certificate?

A. Periodically issue a CRL (Certificate Revocation List)

講義日程（2Q）

		授業計画		課題
06/14	第９回	ネットワーク層１ ルーティング・輻輳制御	５章	ルーティングの種類を理解し 輻輳制御手法を説明できる
06/21	第１０回	ネットワーク層２ インターネットとサービス品質	５章	インターネットの制御プロトコルを理解し ネットワーク間の接続について説明できる
06/28	第１１回	トランスポート層１ トランスポート・プロトコルの要素	６章	誤り制御とフロー制御を理解し 輻輳制御について説明できる
07/05	第１２回	トランスポート層２ UDP と TCP	６章	TCP の信頼性を理解し TCP のコネクション管理を説明できる
07/12	第１３回	アプリケーション層 DNS, 電子メール, www	７章	DNS, 電子メール, www のしくみを理解し ストリーミング，P2P について説明できる
07/26	第１４回	ネットワークセキュリティ１ 対称鍵暗号，公開鍵暗号	８章	暗号アルゴリズムを理解し SHA-1,2 と RSA について説明できる
08/02	第１５回	ネットワークセキュリティ２ デジタル署名，認証プロトコル	８章	電子メール,Web のセキュリティ の脅威について把握できる