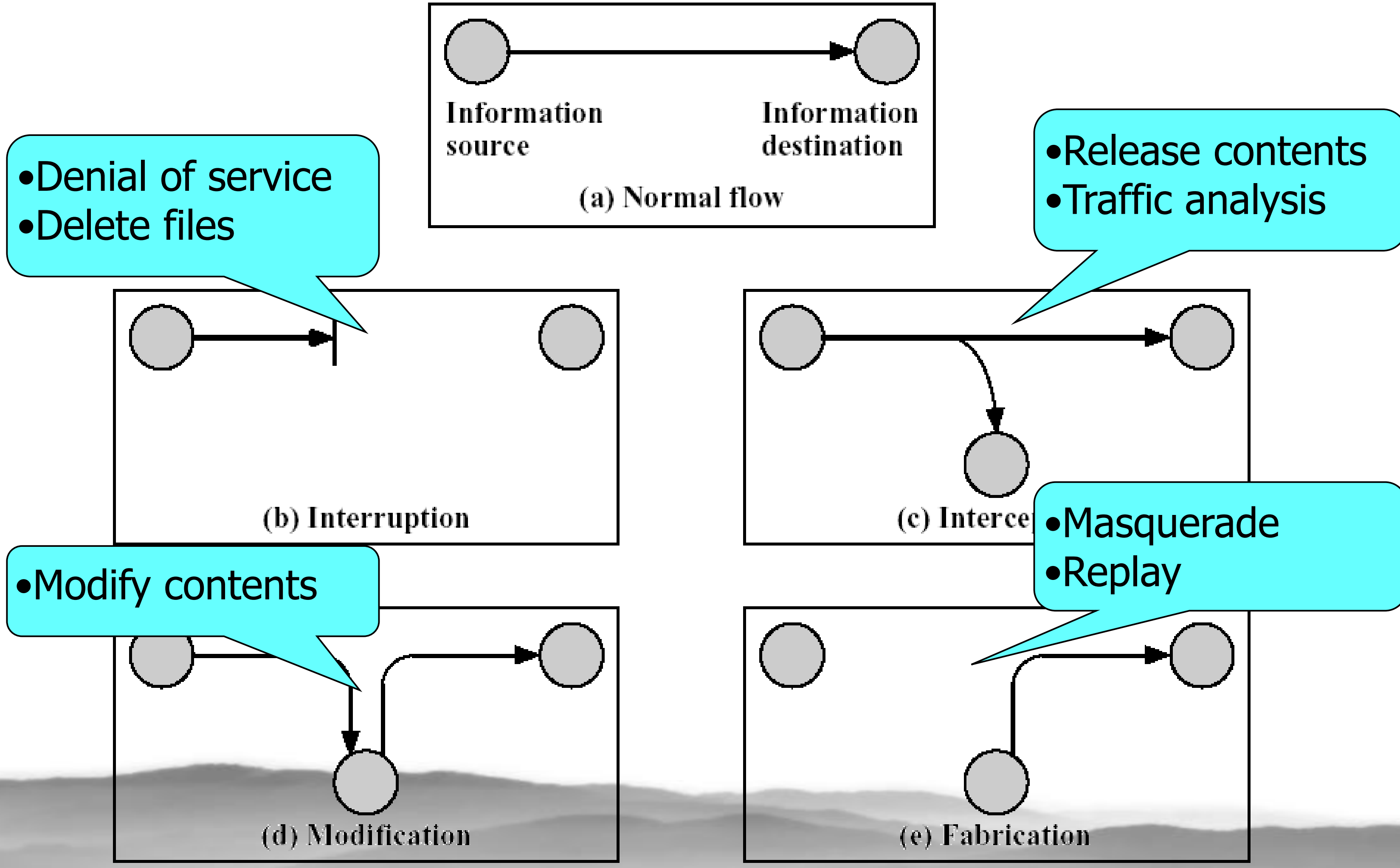
Computer Networks

Wenzhong Li
Nanjing University

Chapter 7. Network Security

- Network Attacks
- Cryptographic Technologies
- Message Integrity and Authentication
- Key Distribution
- Securing Wireless LANs
- Transport Layer Security
- IP Security
- Firewalls

Network Attacks



Passive Attacks

- Eavesdropping on transmissions
  - Release of message contents
- Traffic analysis
  - By monitoring frequency and length of msgs between pair of hosts
  - Nature of communication may be guessed
- Difficult to detect, but can be prevented

Active Attacks

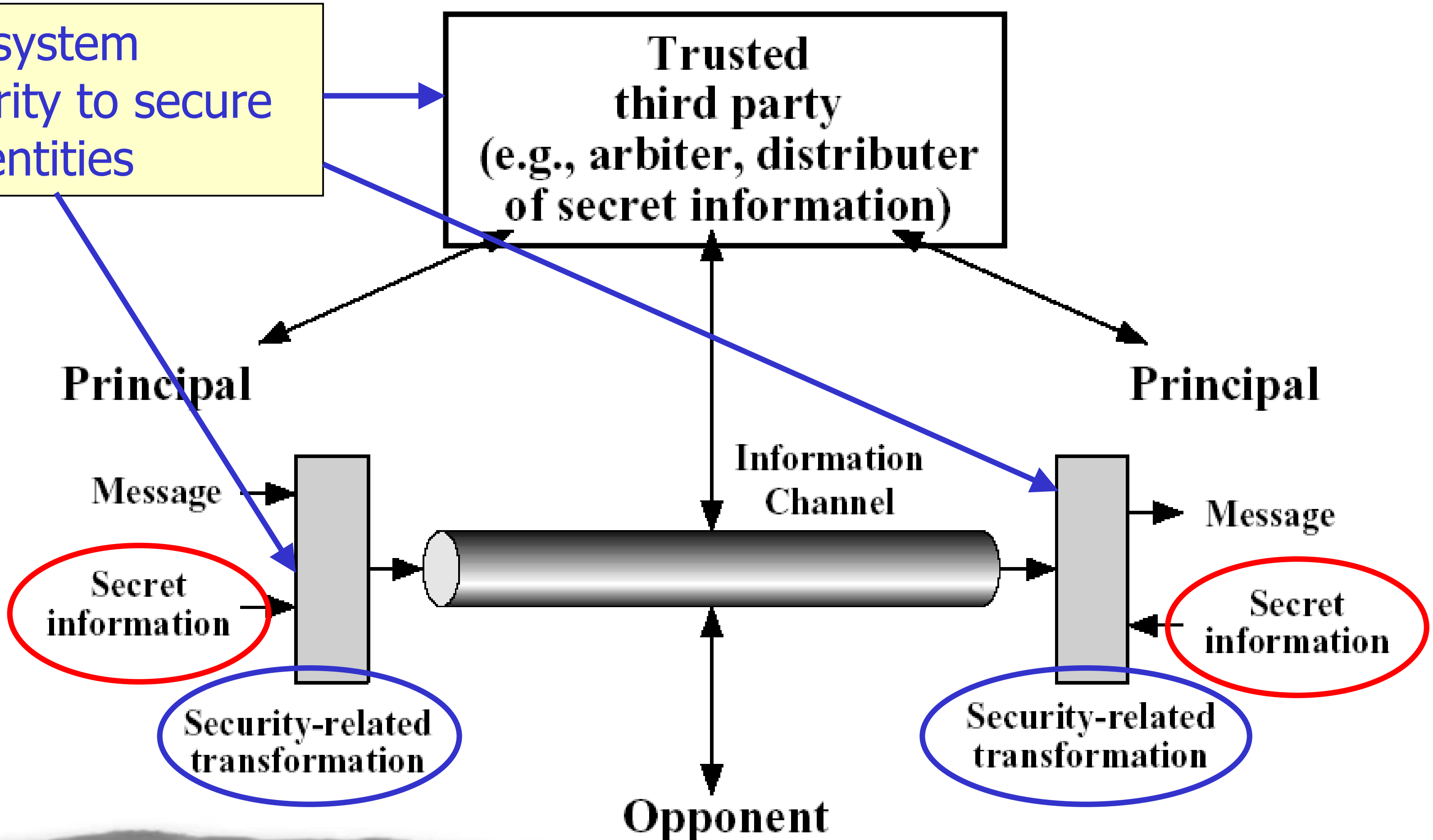
- Masquerade
  - Pretending to be a different entity
- Replay
  - Intercept and capture, then retransmit
- Modification of message contents
- Denial of service
- Hard to prevent, but can be detected

Security Requirements

- Availability, for Interruption
  - Ensure resource is available
- Confidentiality, for Interception
  - Only sender, intended receiver can understand the msgs
- Integrity, for Modification
  - Ensure msgs not altered (e.g. in transit) without detection
- Authenticity, for Fabrication
  - Sender, receiver confirm identity of each other and origin of data

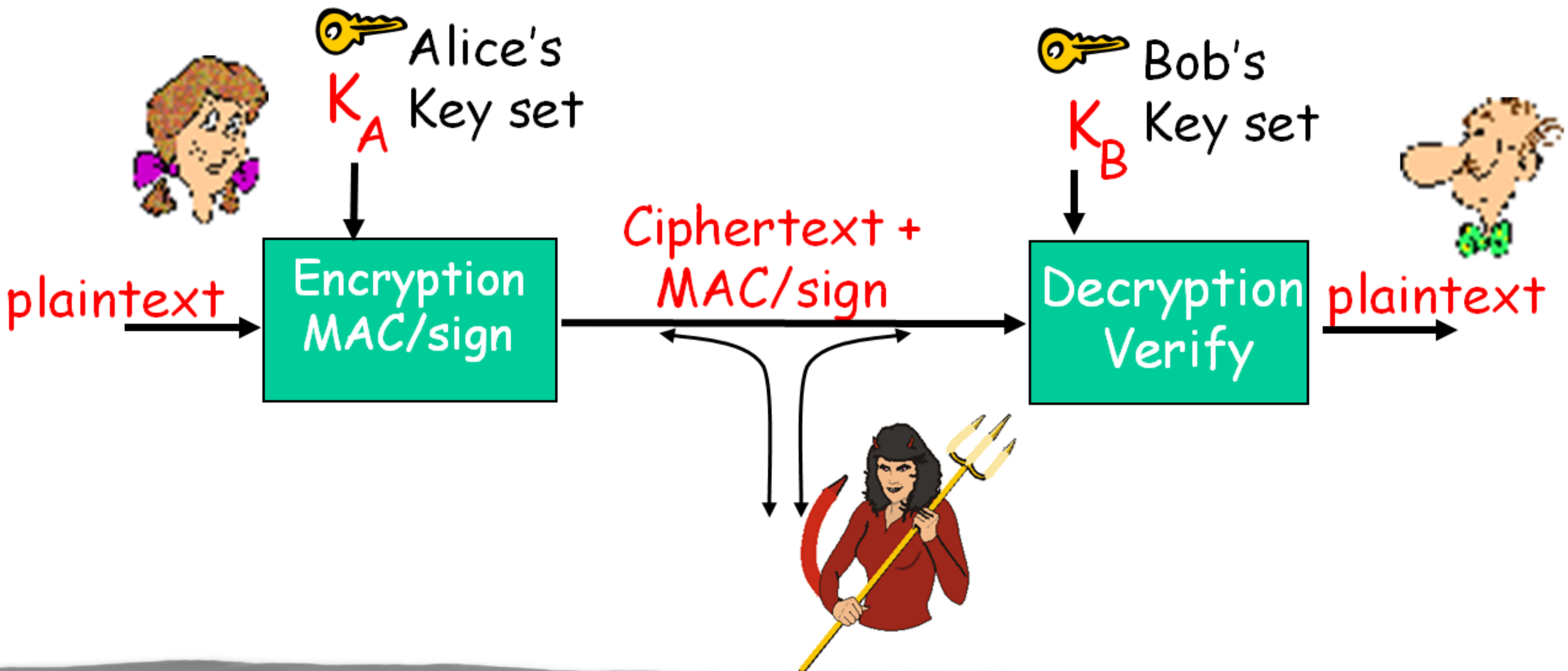
Model for Network Security

Use system security to secure the entities

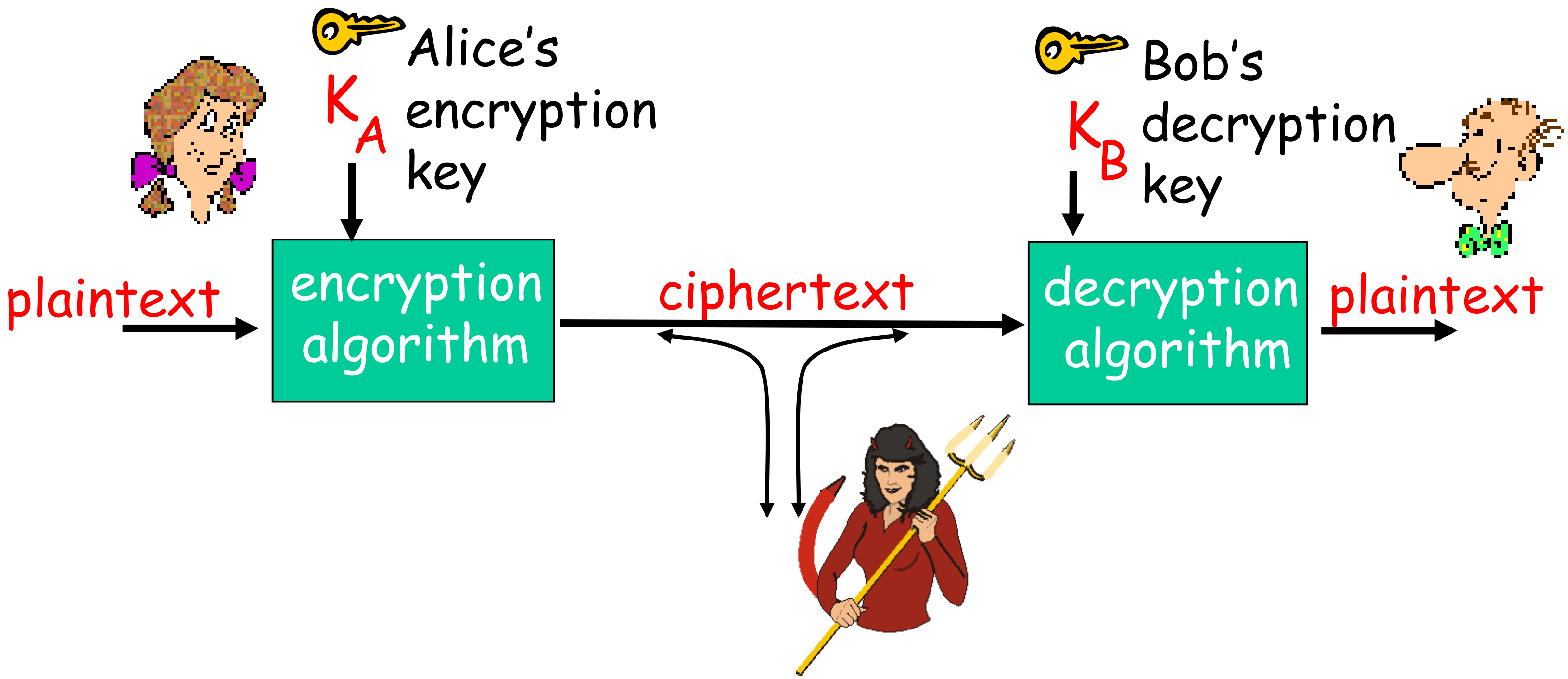


Handling Network Security

- Encryption: the message cannot be understood
- MAC (Message Authentication Code): the message cannot be altered
- Sign: the source cannot be forged/fabricated



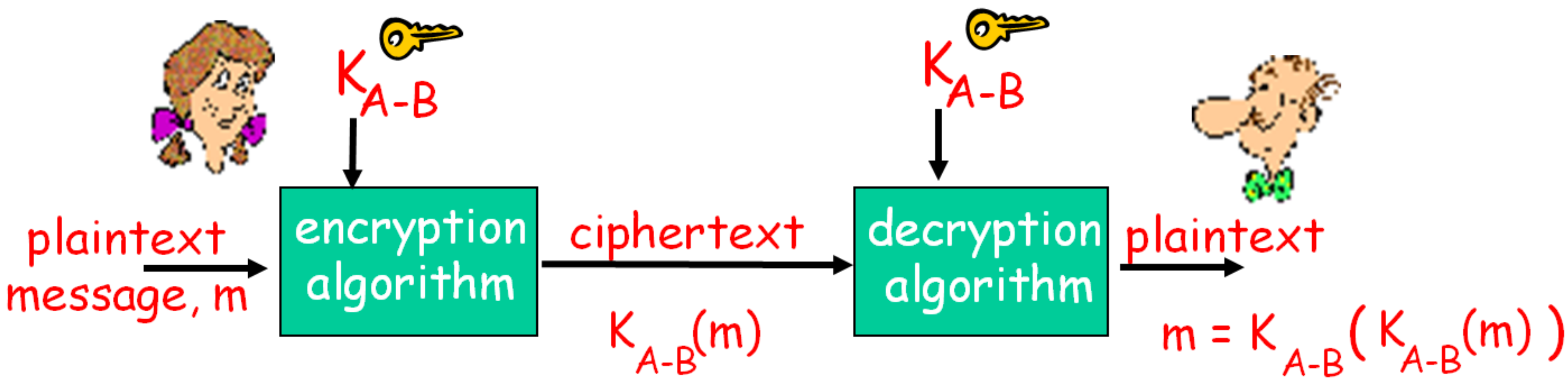
Cryptographic Technologies



- Symmetric key crypto: sender, receiver keys identical
- Public-key crypto: encryption key public, decryption key secret

Symmetric Key Cryptography

- Bob and Alice share the same (symmetric) key: K_{A-B}



Ingredients

- Plain text
- Encryption algorithm
- Secret key
- Cipher text
- Decryption algorithm

Requirements for Encryption

- Strong encryption algorithm
  - Even if known, a number of “cipher texts, plain texts” pairs available
  - Should not be able to decrypt or work out key
- Sender and receiver must obtain secret key securely – Key distribution
  - Once key is known, all communication using this key is readable

Attacking Encryption

- Cryptoanalysis
  - Rely on nature of algorithm plus knowledge of general characteristics of plain / cipher text
  - Attempt to deduce plain text or key
- Brute force
  - Try every possible key until plain text is achieved

Traditional Encryption Techniques

- Substitution methods
  - Letters of the alphabet are replaced with other letters / numbers / symbols
  - Caesar Cipher
  - Mono-alphabetic Cipher
  - Vigenere Cipher
- Transposition (Permutation) methods
  - Rearrange (shuffle) the input without altering the actual letters used
  - Rail Fence Cipher
  - Row-Column Cipher

Caesar Cipher (1)

凯撒密码

- Invented by Julius Caesar, first attested use in military affairs

- Cyclic shift of the 26 letters of the alphabet by 3

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
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Plain text

attack from east at dawn

Cipher text

dwwdfn iurp hdvw dw gdzq

Caesar Cipher (2)

- In mathematical terms
  - $C = encrypt(P) = P+3 \pmod{26}$
  - $P = decrypt(C) = C-3 \pmod{26}$
- Generalizd
  - $C = encrypt(P) = P+k \pmod{26}$
  - $P = decrypt(C) = C-k \pmod{26}$, where $0<k<26$
- How to Attack?
  - There is only one key
  - Easy to break, only 26 different keys, use Brute force

Mono-alphabetic Cipher (1)

单表密码

- Map the N letters of the alphabet with one of N! permutations

- The key is one of the N! options

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
x	f	g	i	k	u	m	o	y	a	c	e	s	v	b	h	d	j	l	p	n	q	z	t	w	r

Plain text

attack from east at dawn

Cipher text

xppxgc ujbs kxlp xp ixzv

Mono-alphabetic Cipher (2)

- For N = 26 letters of the English alphabet
  - There are $N! = 26! \approx 2^{88}$ possible keys

- How to Attack？
  - Appearance frequency of letters (in long enough texts) in the language is well determined
  - Fixed substitution
  - Using the appearance frequencies of letters, words, and pairs-of-letters can easily get the key

Vigenere Cipher (1)

维吉尼亚密码

- Collection of Mono-Alphabetic Ciphers consists of the 26 options for Caesar Cipher (k = 0, 1, 2, …, 25)

- Each of the 26 Caesar Ciphers is denoted by a letter, which is the ciphertext letter that replaces the letter 'a'

- A keyword is used (in cycle) to select which of the Caesar Ciphers to use
  - Denoted by the current letter in the keyword

Vigenere Cipher (2)

- Ciphers Table

	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
a	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
b	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	a
c	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
d	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
	… … …																									

Key

bad

Plain text

attack from east at dawn

Cipher text

btwbcn grrn edtt du ddxn

Vigenere Cipher (3)

- There are multiple ciphertext letters to which each plaintext letter can be mapped
  - Number of possible ciphertext letters a plaintext letter can map equals numbers of different letters in key
- How to Attack?
- Attack
  - Figure out the length of the key
  - Break each of the suspected Mono-Alphabetic Ciphers independently

Rail Fence Cipher

篱笆密码

- Plaintext is written down as a sequence of diagonals
- Ciphertext is then read of row by row

Plain text

a t c f o e s a d w

t a k r m a t t a n

Cipher text

atcfoesadwtakrmattan

- Key is number of rows used
- How to Attack?

- Attack is easy, just play the rows

Row-Column Cipher

行列密码

- Plaintext is written in a rectangle, row by row
  - Length of each row equal the key length
- Ciphertext is read from the rectangle, column by column
  - Column order corresponds to letter order of the key

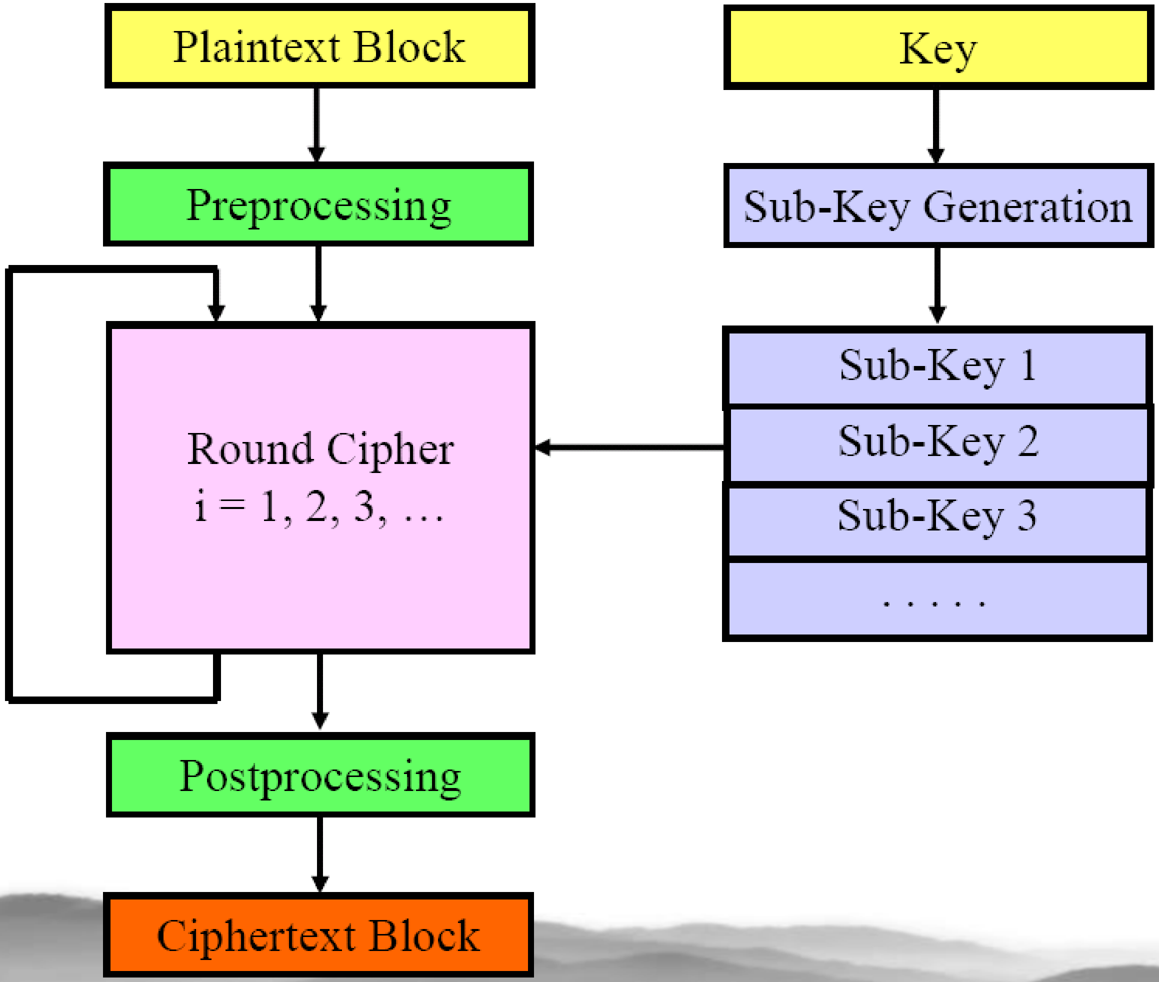
Key	Plaintext	Ciphertext
noise	3 4 2 5 1	cmantrsoakettfadaotw
34251	a t t a c	
	k f r o m	
	e a s t a	
	t d o w n	

- Key can be determined by placing the ciphertext in a rectangle and playing with the rows and the columns
- If with plaintext, break pure transposition ciphers is very easy

Modern Encryption Algorithms

- Block cipher
  - Process plain text in fixed block sizes
  - Produce block of cipher text of equal size
- Commonly used Algorithms
  - Data encryption standard (DES)
  - Triple DES (TDES)
  - Advanced Encryption Standard (AES)

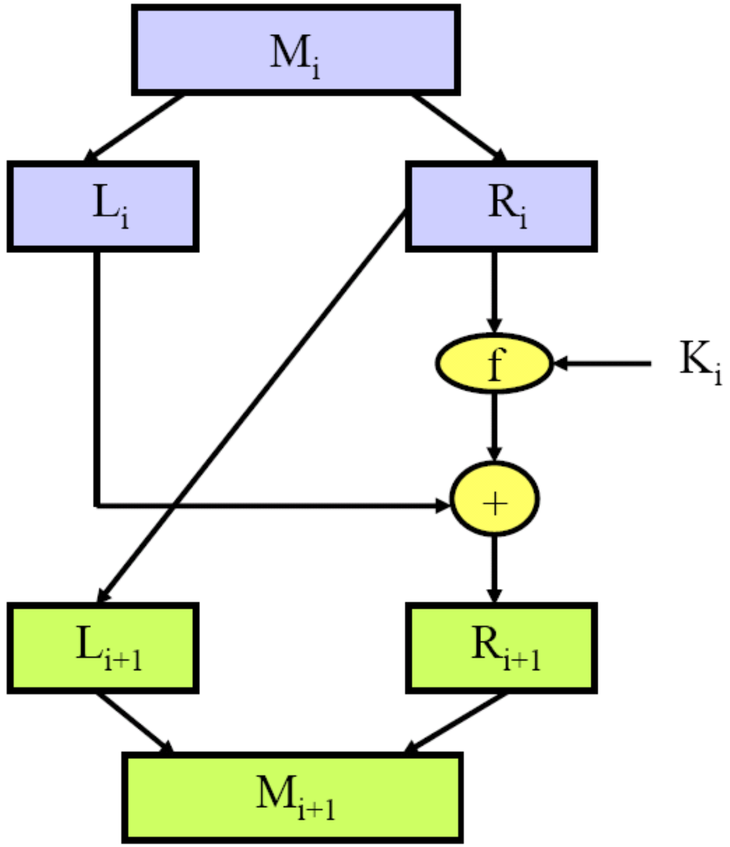
Pattern of Block Ciphers



Feistel Ciphers

- A scheme / template for specifying the algorithm of a block cipher
- Allows encryption and decryption with the same hardware circuit / piece of software

- Input block M_i broken into half-blocks L_i and R_i
- For next round
  - $L_{i+1} = R_i$
  - $R_{i+1} = f(R_i, K_i) \oplus L_i$

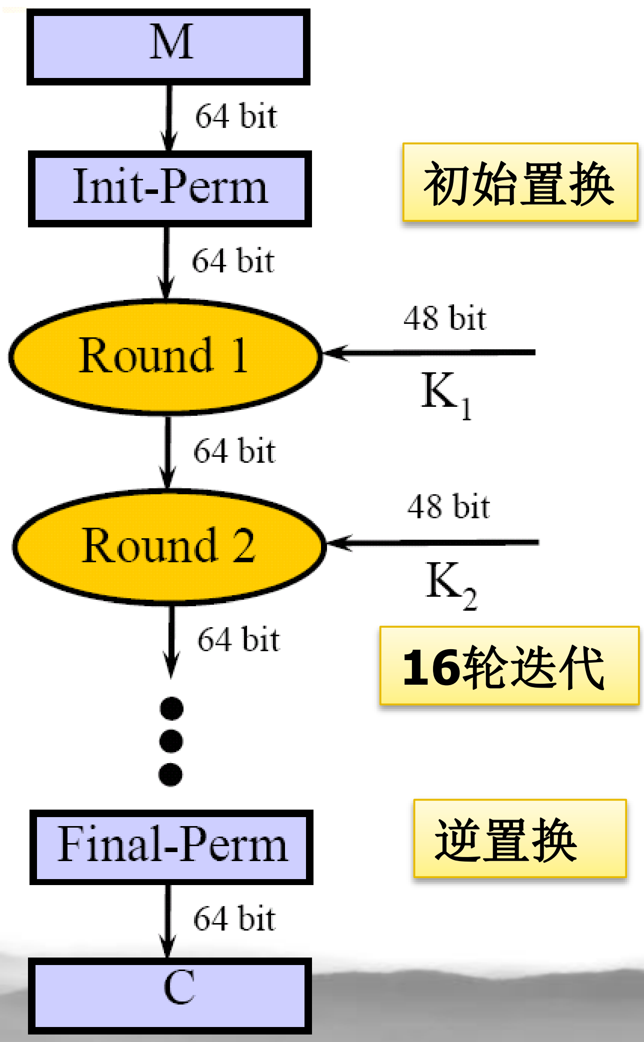


Data Encryption Standard

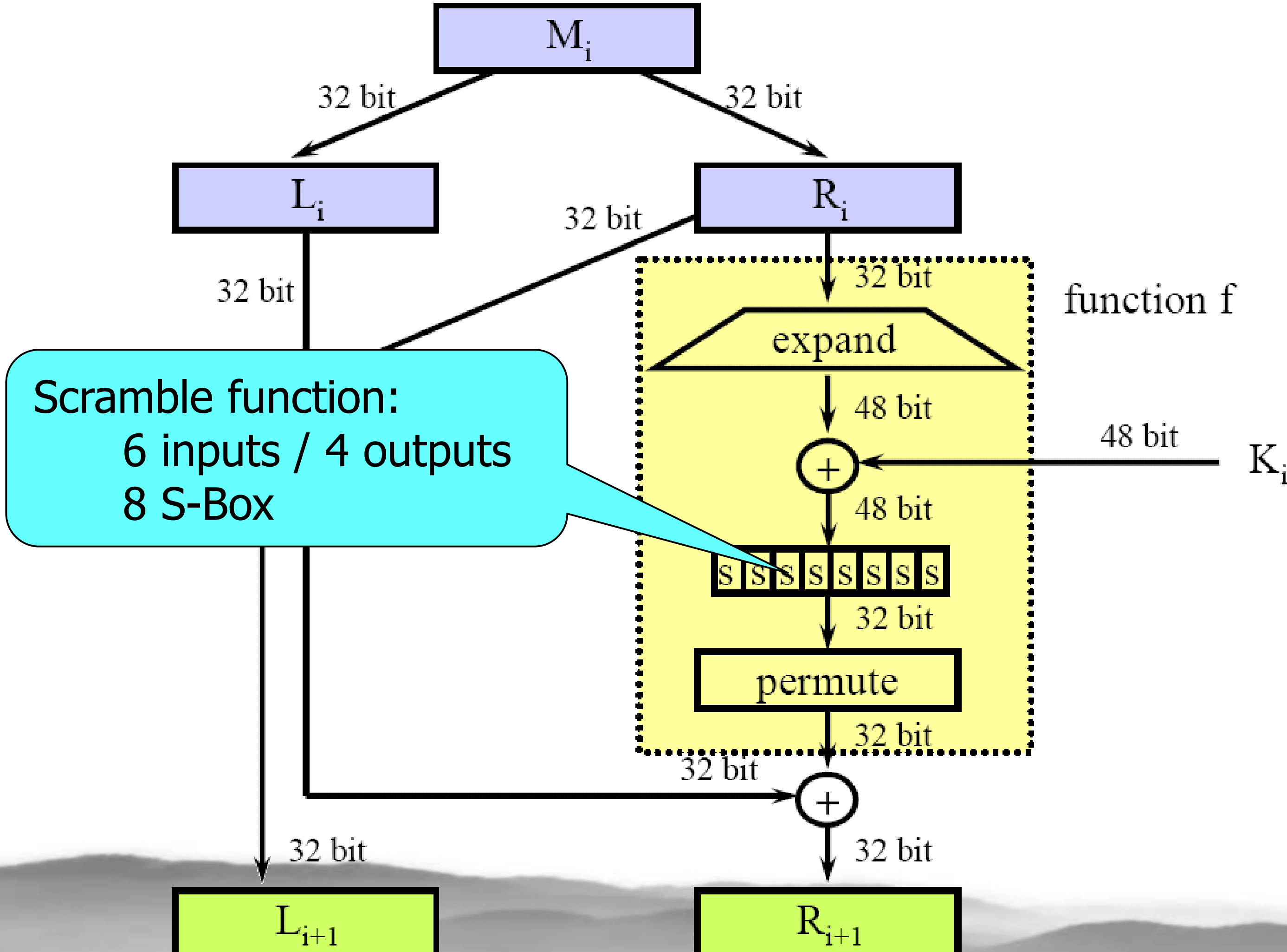
- 1973, NBS (National Bureau of Standards) came out with an RFP for a commercial encryption standard
- IBM proposed its strong Lucifer algorithm (developed by Feistel and others)
- NSA (National Security Agency) requested to weaken the strength of Lucifer (by shortening the key)
- NSA also made changes to IBM's Lucifer algorithm
- 1976, Data Encryption Standard (DES) accepted
- 1999, Triple DES (3-DES) defined by NIST (National Institute of Standards and Technology)

DES Structure

- Block size – 64 bits
- Key size – 56 bits (in a 64-bit buffer)
- Fixed initial permutation on input block (64 bits)
- 16 round keys (48 bits) derived from key (56 bits)
- Key scheduling scheme for 16 round keys
- 16 iterations each consisting of scrambling the round-block (64 bits) with the round-key (48 bits)
- Fixed inverse initial permutation on output block



Per Round DES



Attack DES

- Exhaustive Search Attack
  - Search space of $O(2^{56}) = O(10^{17})$ keys
  - No "backdoor" exists

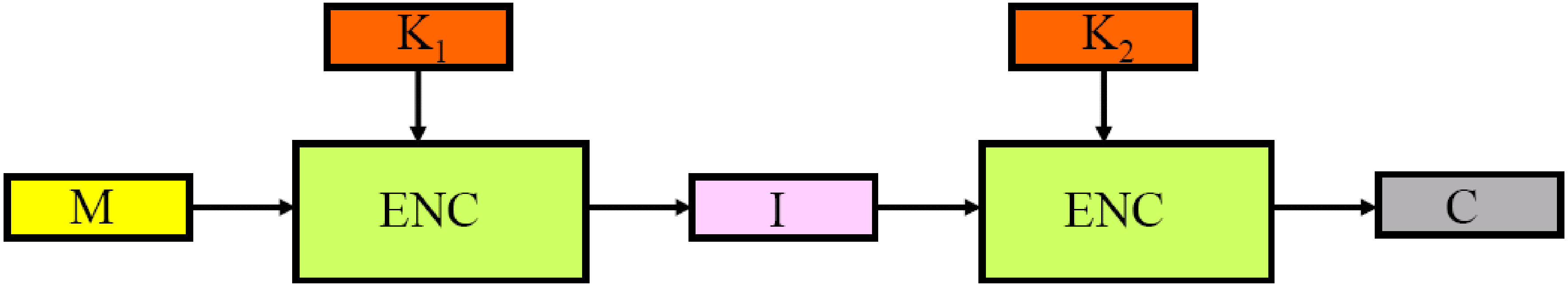
- DES now worthless
  - In the 1970's, Diffie and Hellman suggested a $20M machine that will crack DES in about one day
  - In the 1990's, Wiener suggested a $1M machine that will crack DES in 3.5 hours
  - In 1990's, DES challenges were broken in matter of days using distributed clusters of computers
  - Presumably, most national security agencies have the hardware and software to crack DES in hours

Triple DES

- 1985, ANSI X9.17
- 1999, Incorporated into DES standard
- Uses 3 keys and 3 executions of DES algorithm
- 2 Mode defined
  - EEE mode
  - EDE mode
- Problem
  - Slow, block size (64 bit) too small

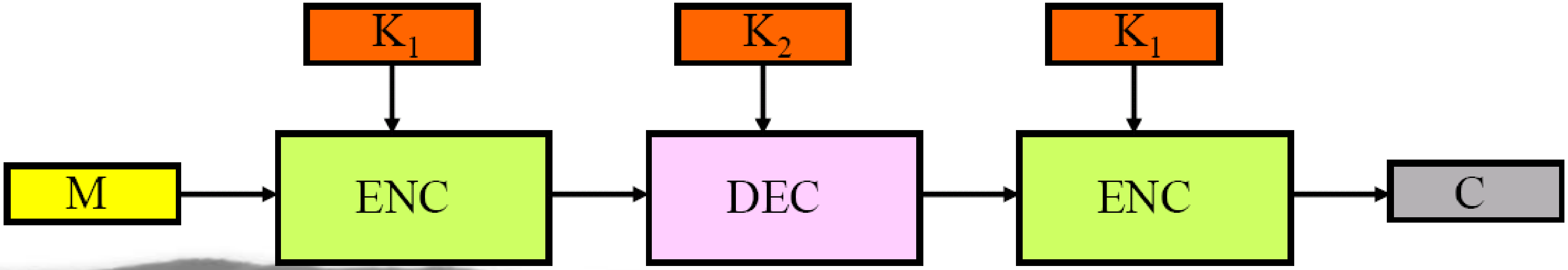
3-DES: EEE Mode

- DES Encrypt-Encrypt-Encrypt with 3 keys K_1, K_2, and K_3
- Properties
  - Three keys (168 bits)
  - Strength about $O(2^{110})$ against attack
  - Not compatible with regular DES



3-DES: EDE Mode

- DES Encrypt-Decrypt-Encrypt with 2 keys K_1 and K_2
- Properties:
  - Two keys (112 bits)
  - Strength about $O(2^{110})$ against attack
  - Compatible with regular DES when $K_1 = K_2$
  - Can Encrypt-Decrypt-Encrypt with keys K_1, K_2, and K_3, more resilient against attacks



Advanced Encryption Standard

- 1997, NIST published RFP for Advanced Encryption Algorithm
  - Symmetric block cipher
  - Security strength equal to or better than 3-DES with improved efficiency
  - Variable strength by key size (from 128 to 256 bits)
  - Efficient implementation on various SW & HW platforms
- About 20 algorithms were proposed
  - Open review process for about 3 years
  - Rijndael was selected in November 2001
- 2001, AES issued as federal information processing standard (FIPS 197)

AES Parameters

- B: Block size in 32-bit words, 4 means 128 bits
- K: Key size in 32-bit words, 4 / 6 / 8 (i.e. 128 / 192 / 256 bits)
- r: Number of rounds – 6 + max(B, K)
  - AES-128 – 10 rounds
  - AES-192 – 12 rounds
  - AES-256 – 14 rounds

AES Description

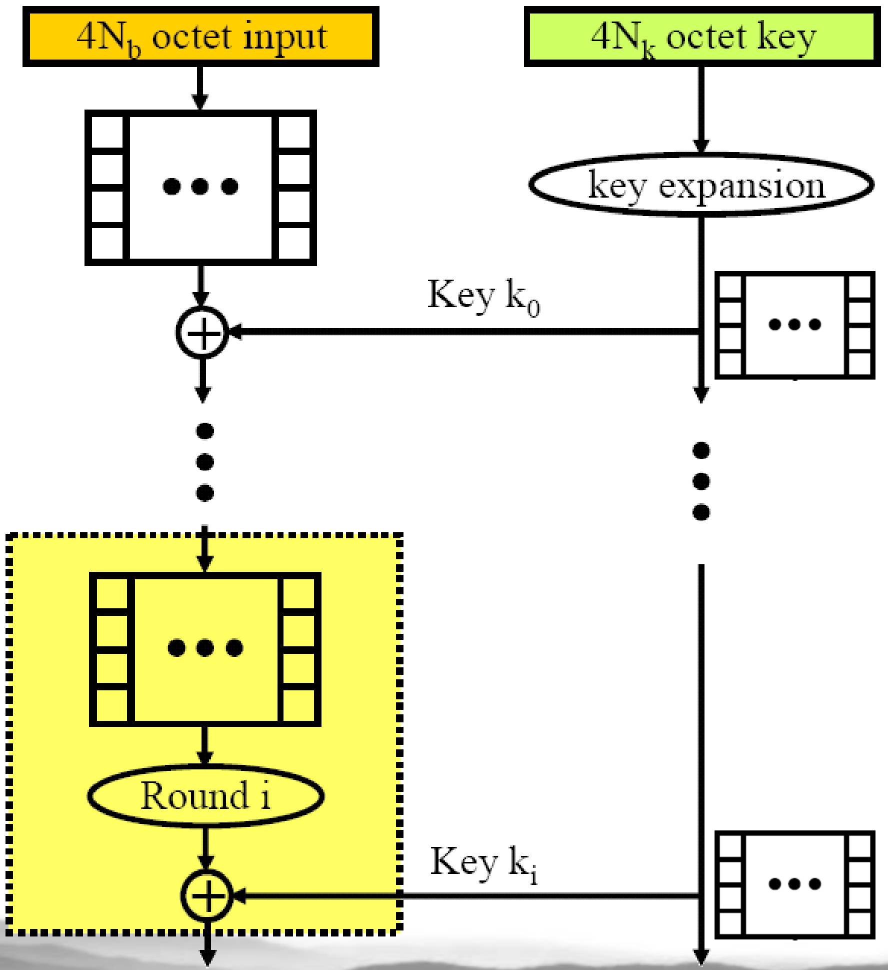
- State Array
  - A rectangular array of 4 rows by B (=4) columns
  - Each of the array entries holds an octet (8 bits)
  - Initial value of state array is the plaintext block entered column by column
  - State is transformed during r rounds
  - Final value of state is the ciphertext block read column by column
- key-expansion scheme
  - Key is organized as a sequence of key-sets
  - Initial key-set consists of K columns of 4 octets (32 bits) each
  - Key-expansion generates $(r+1)\times B$ 4-octet columns

Per Round AES (1)

- 4×B octet input
- Placement of input in the state array

- 4×K octet key
- Key expansion

- XOR input with K_0
- r rounds of state array transformation and XOR with K_i

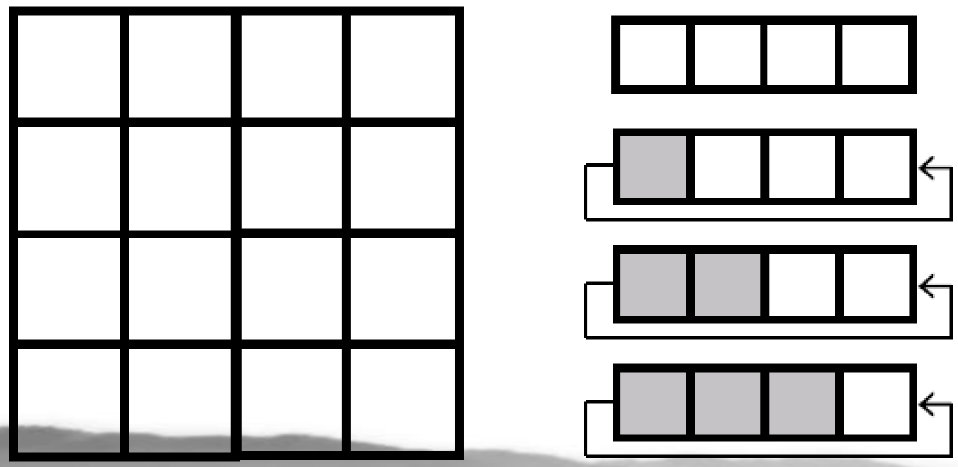


Per Round AES (2)

- 4 primitive operations inside each round
  - S-box that substitutes octet for octet, S-box substitution is implemented as a table lookup
  - Rearrangement of octets that consists of rotating rows by some number of cells
  - A Mix-Column operation that replaces a 4-octet column with another 4-octet column, uses table lookup
  - Bit-wise XOR with current K_i

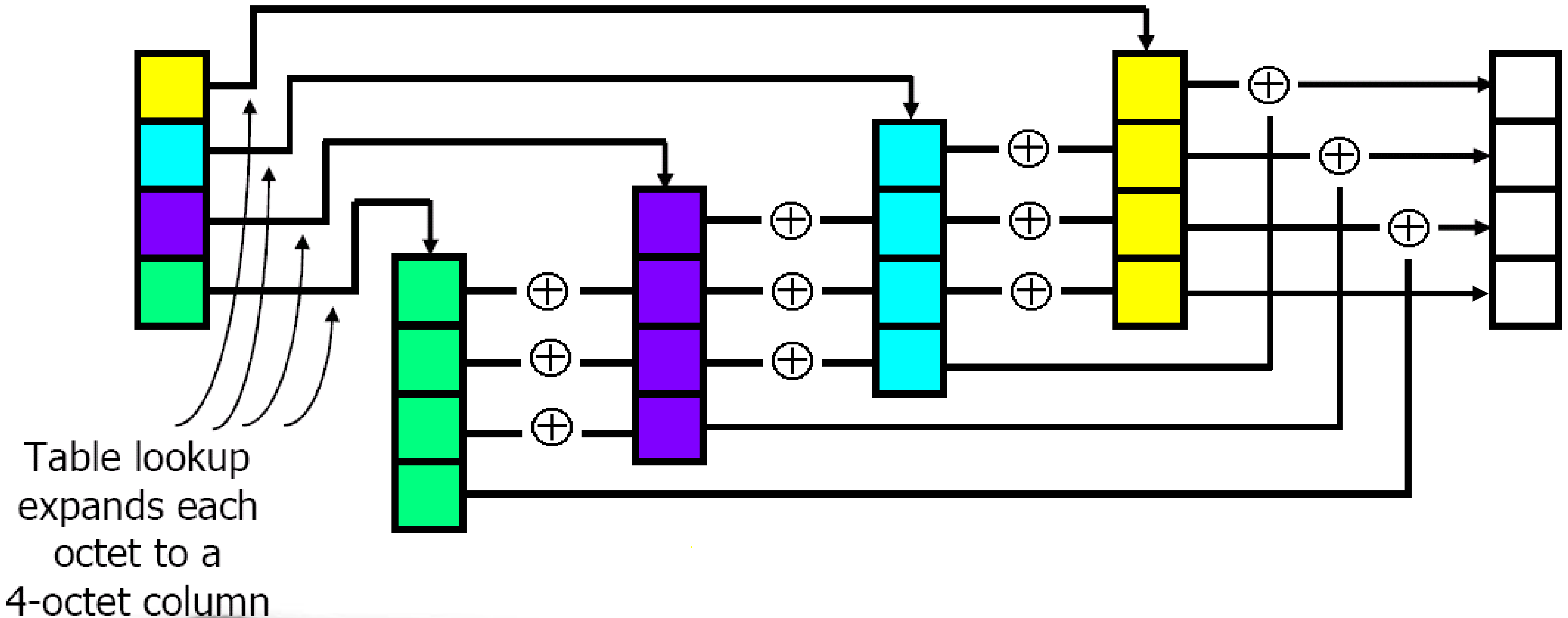
Row Rotations

- 4×B cells of 8 bits each
  - Row 0 of the state array is not rotated
  - Row 1 of the state array is rotated left 1 column
  - Row 2 of the state array is rotated left 2 columns
  - Row 3 of the state array is rotated left 3 columns

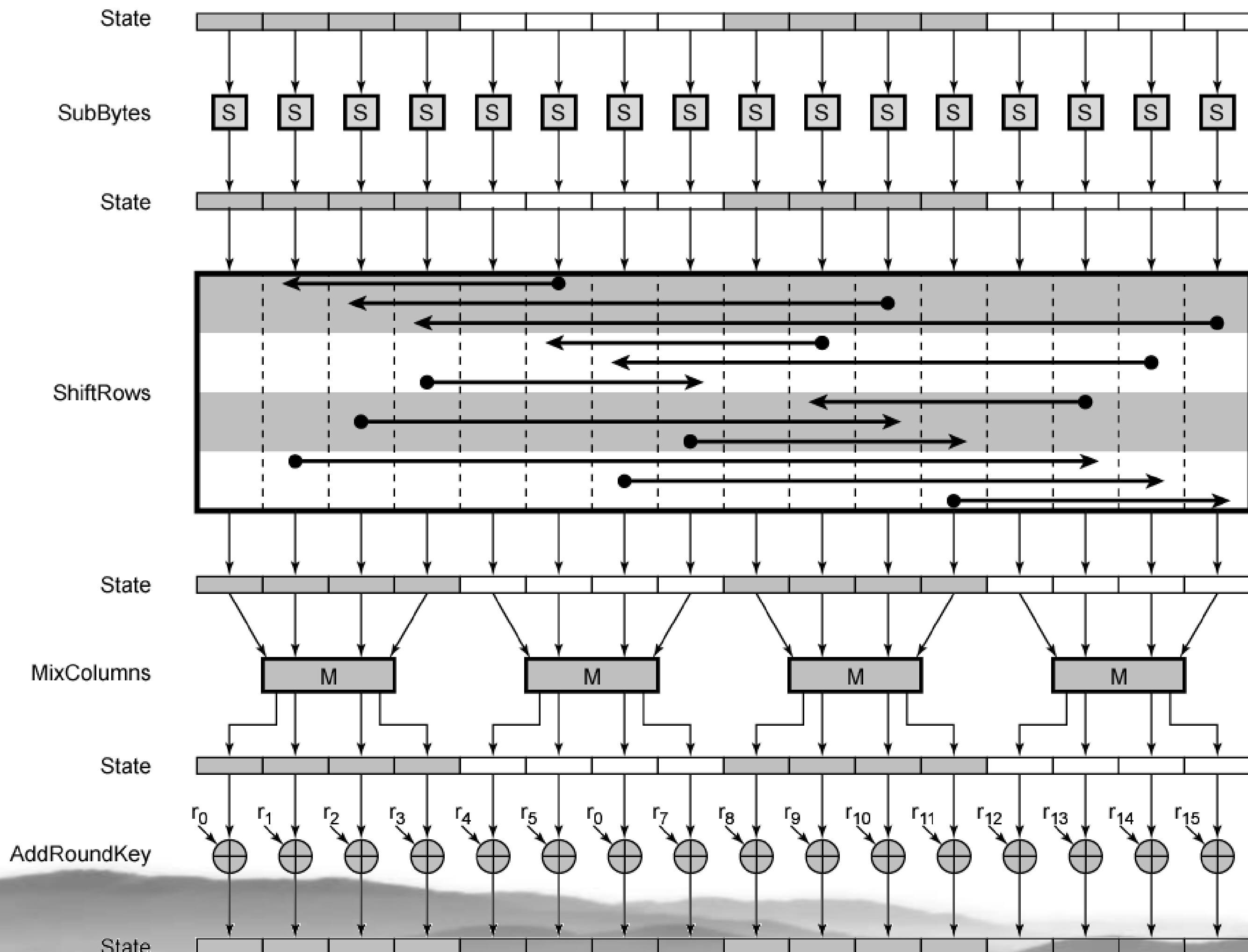


Mix-Column Operation

- On each 4-octet column
  - A new 4-octet column is computed



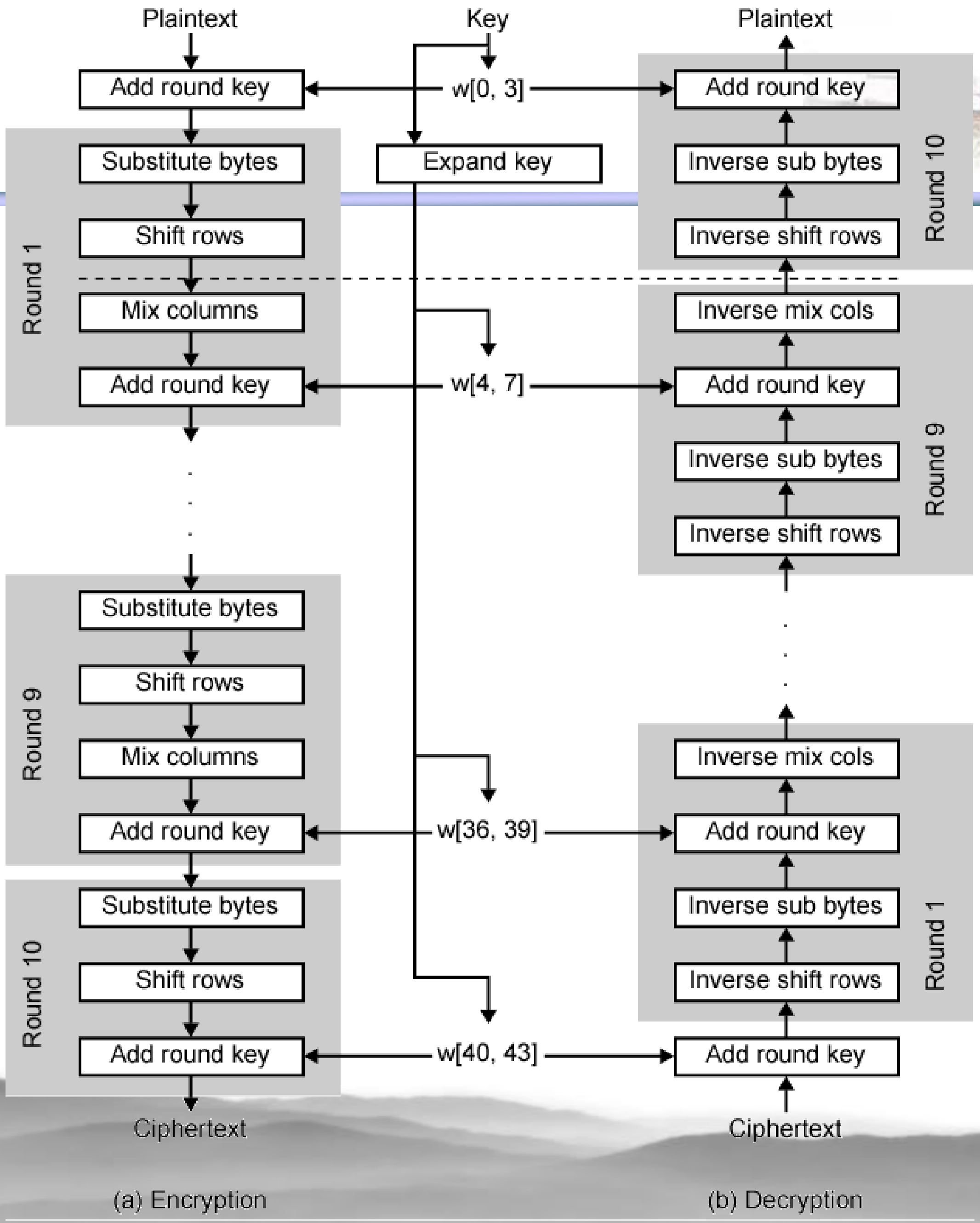
AES Encryption Round



AES Description

- Simple structure
  - For both encryption and decryption, cipher begins with Add Round Key stage
  - Followed by r-1 rounds, each includes all 4 operations
  - Followed by last round of 3 operations
- Add Round Key plus 3 operations of scrambling bits
- Decryption uses expanded key in reverse order
  - Not identical to encryption algorithm

AES Structure



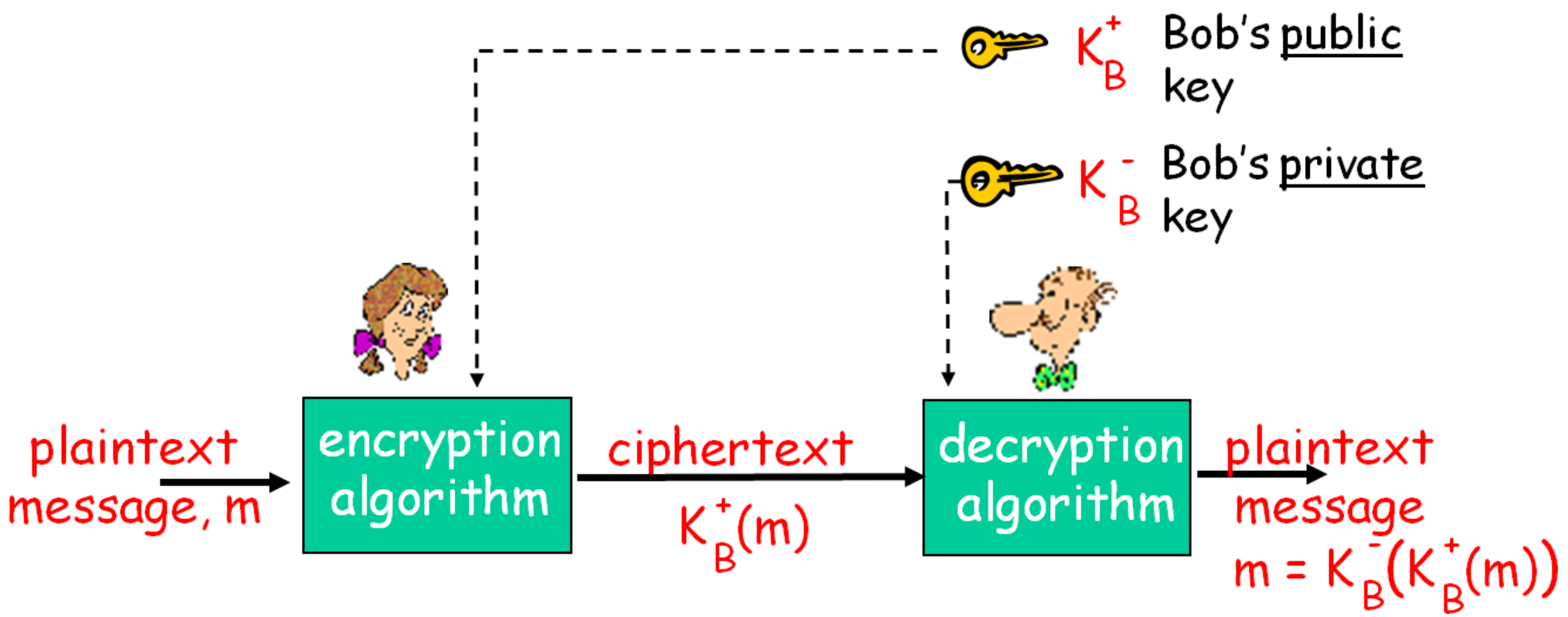
128 bit key
r = 10

Public Key Cryptography

- Radically different approach
  - Based on mathematical algorithms
- Asymmetric keys
  - Use two separate keys
  - Sender, receiver do not share symmetric secret key
- Each end system has
  - A public encryption key known to all
  - A private decryption key only the owner knows

Public Key Cryptography

- Bob and Alice both have their own key sets



Public Key Ingredients

- Plain text
- Encryption algorithm
- Public and private key
- Cipher text
- Decryption algorithm

Public-Key Model

- Entity A has two keys
  - PRV_A – Private Key of A, kept secretly only at A
  - PUB_A – Public Key of A, made public to all

- There are two functions
  - Encrypt – uses one of the two keys
  - Decrypt – uses the other key

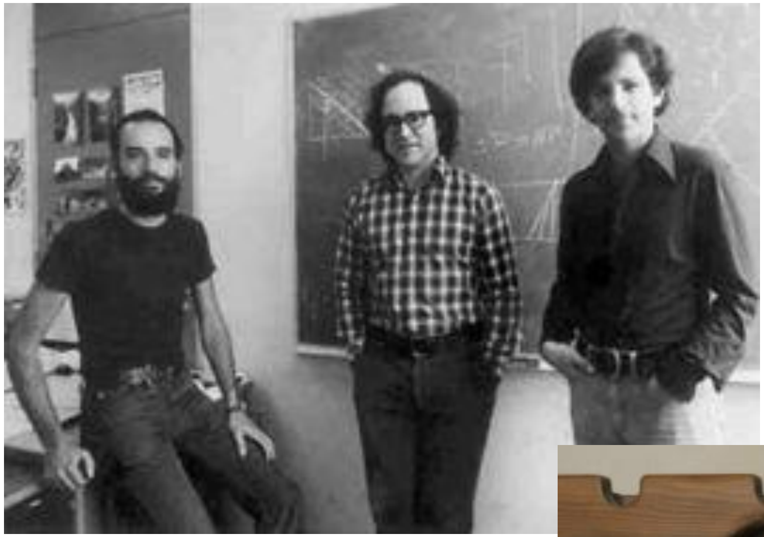
- Such that
  - $Decrypt(Encrypt(M, PUB_A), PRV_A) = M$
  - $Decrypt(Encrypt(M, PRV_A), PUB_A) = M$

Public Key Requirements

- The algorithms of encryption and decryption functions should be known
- Given the public key and the algorithms, it should be hard to find the private key
- Given a ciphertext using the public key, it should be hard to get the plaintext
- Given pairs of (plaintext, ciphertext) with the public key, it should be hard to find the private key

RSA Algorithm

- 1977, Rivest, Shamir, Adleman algorithm (Turing Award 2002)



RSA Algorithm

欧几里德，辗转相除法，前**330**年
欧拉定理，**1736**年

- 1977, Rivest, Shamir, Adleman algorithm (Turing Award 2002)
- Pick 2 large primes : p and q
  - About 500/1024 bits each

欧拉函数，计算小于**N**并与**N**互素的整数个数

- Compute : $N = p \times q$
- Compute : $\Phi = (p-1)\times(q-1)$

满足条件的**e**有多个，可随机选一个

- Pick e relatively prime to Φ, i.e. $GCD(e, \Phi) = 1$
- Calculate d inverse of e modulo Φ, i.e. $d\times e \bmod \Phi = 1$

d称为**e**的模反元素

- The key set is ready
  - Public Key: (e, N), and Private Key: (d, N)

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
3^i mod 5	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4
3^i mod 7	3	2	6	4	5	1	3	2	6	4	5	1	3	2	6	4	5	1	3	2	6	4
3^i mod 35	3	9	27	11	33	29	17	16	13	4	12	1	3	9	27	11	33	29	17	16	13	4
i	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
3^i mod 5	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1
3^i mod 7	5	1	3	2	6	4	5	1	3	2	6	4	5	1	3	2	6	4	5	1	3	2
3^i mod 35	12	1	3	9	27	11	33	29	17	16	13	4	12	1	3	9	27	11	33	29	17	16

欧拉数： Φ (N)=1到N范围内和N互质的整数的个数(包括1)

例：对于素数**5，7，Φ (5×7)=（5-1）×（7-1）=24**

欧拉定理：任意数a的i次方除以N的余数的周期

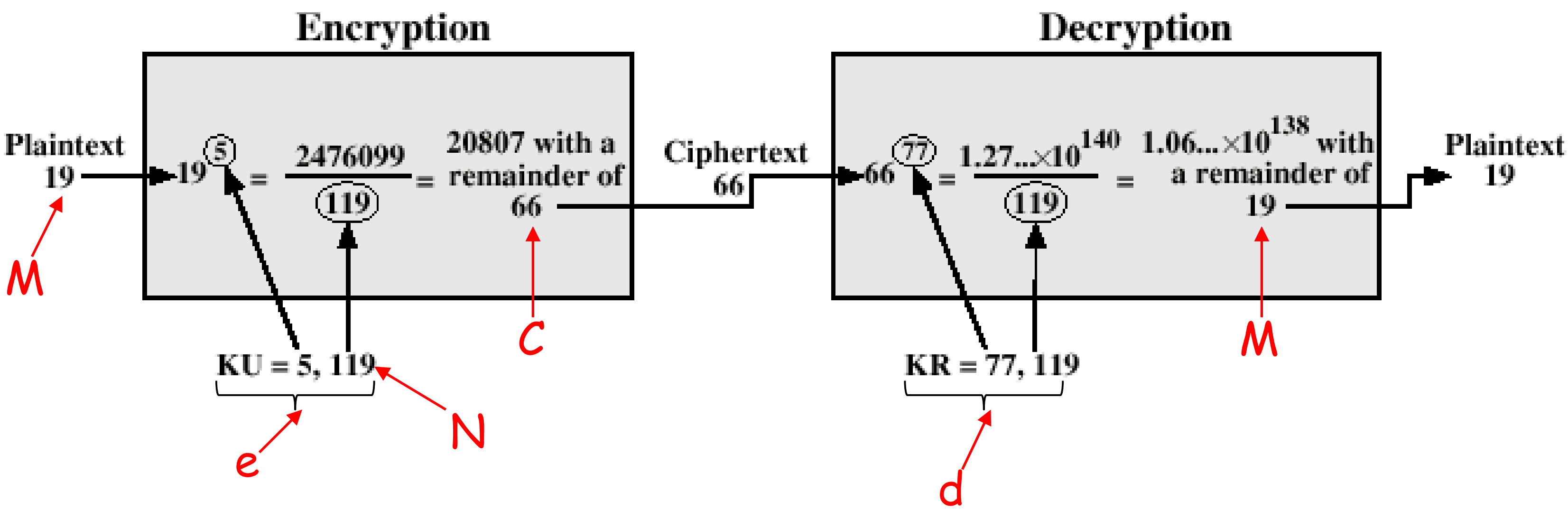
例：因为模反函数满足：**d×e mod Φ = 1**

故对任意数**a，a**的**d*e**次方除以**N**的余数等于**a**除以**N**的余数

RSA Encryption & Decryption

- Euler's Theorem
  - $(M^e \bmod N)^d \bmod N = M^{ed} \bmod N = M (\bmod N)$
- Encryption
  - Use public-key (e, N) to encrypt message block $M<N$
  - $C = M^e (\bmod N)$
  - Send only C
- Decryption
  - Use private-key (d, N) to decrypt cipher $C<N$
  - $M = C^d (\bmod N)$

RSA Example



N=7×17=119 （已知）
Ø=6×16=96 （欧拉函数，保密）
GCD（5，96）=1
存在模反元素**x**，使得**5x mod 96=1**，此处令**x=77**

RSA Considerations

- Primes p and q should be of about the same length
  - About half the length of N
- Primes p and q should be unrelated
- Public exponent e can be small
  - No need to add much burden on encryption
- Private exponent d must be large
  - Disallow searching on small values (brute force)
- Strength of RSA depends on the fact that
  - Large N with large prime factors, factoring is a hard problem
  - N's length can be enlarged to make it stronger

构造大素数非常容易
素数分解则非常困难

More about RSA

- The other way around
  - $(M^d \bmod N)^e \bmod N = M^{de} \bmod N = M (\bmod N)$
- Signature
  - A use private-key (d, N) to sign message block $M<N$
  - $S = M^d (\bmod N)$
  - Send M and S
- Verification
  - B (and others) use public-key (e, N) to check signature S on message M
  - $S^e = M (\bmod N)$
  - This authenticate A, since only A has the private key

Summary

- Symmetric Key Cryptography
- Traditional
  - Substitution methods
    - Caesar Cipher
    - Mono-alphabetic Cipher
    - Vigenere Cipher
  - Transposition (Permutation) methods
    - Rail Fence Cipher
    - Row-Column Cipher
- Modern: Block cipher
  - Data encryption standard (DES)
  - Triple DES (TDES)
  - Advanced Encryption Standard (AES)
- Asymmetric Key Cryptography: RSA

Homework

- 第8章：R5, R6, P3, P7, P8