Security and Privacy

Cryptography Concepts

Cryptography is everywhere

- **Secure communication**:
  - web traffic: HTTPS
  - wireless traffic: 802.11i WPA2 (and WEP), GSM, Bluetooth
- **Encrypting files on disk**: EFS, TrueCrypt
- **User authentication**
- Digital Signature
- More and more …

Cryptographic algorithms and protocols can be grouped into four main areas:

Symmetric encryption

- Used to conceal the contents of blocks or streams of data of any size, including messages, files, encryption keys, and passwords

Asymmetric encryption

- Used to conceal small blocks of data, such as encryption keys and hash function values, which are used in digital signatures

Data integrity algorithms

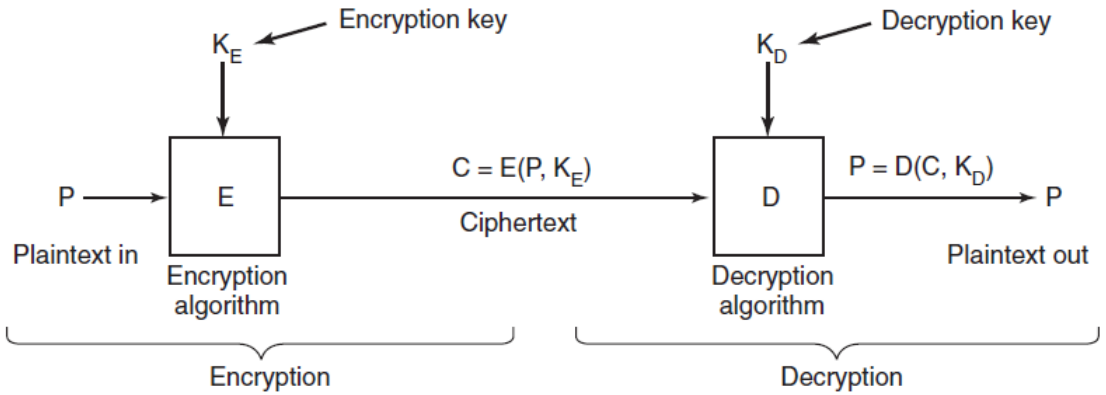
- Used to protect blocks of data, such as messages, from alteration

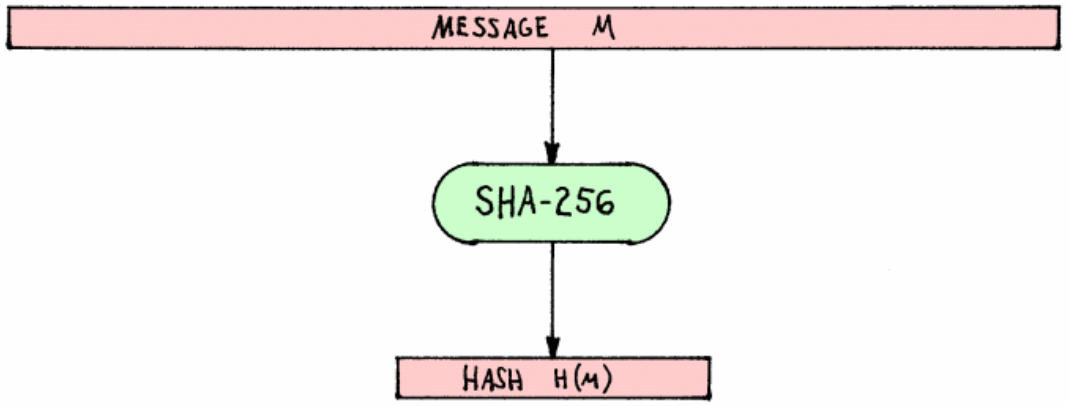
Authentication protocols

- Schemes based on the use of cryptographic algorithms designed to authenticate the identity of entities

Encryption vs Hashing

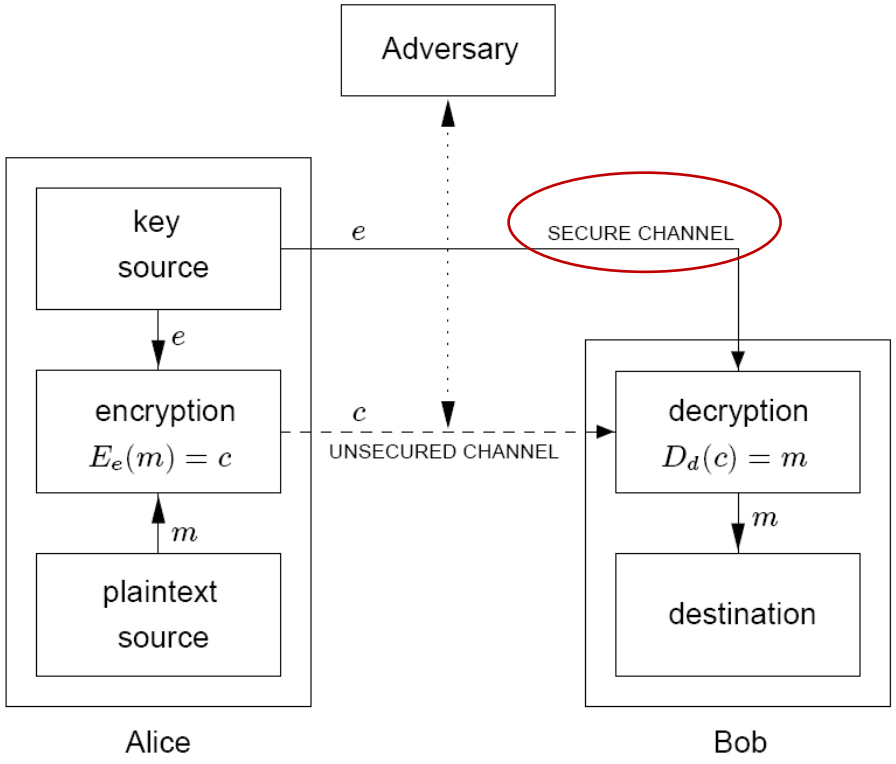
- **Hashing**
  - Mathematical algorithms that create a **fixed size** message summary or digest
  - Unidirectional
  - Used in signatures, password storage, etc.





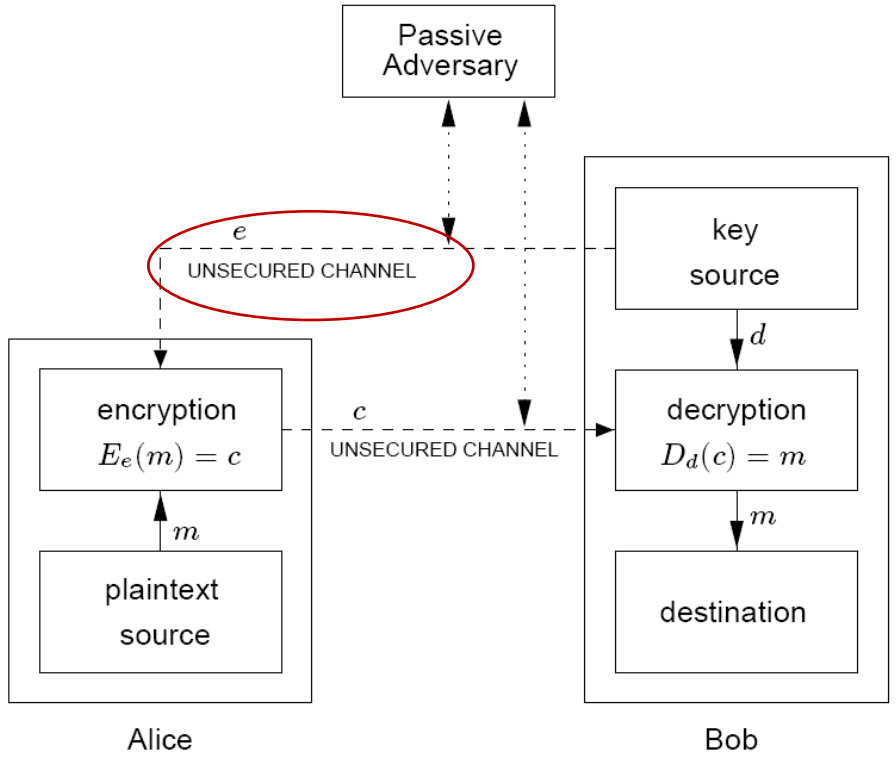
SYMMETRIC VS ASYMMETRIC

- Shared key e is used for both encryption and decryption
- Possible to derive ***d*** from ***e***



A public key e is used to encrypt and a private key d is used to decrypt

Not possible to derive ***d*** from ***e***



Cryptography

Introduction to Number Theory

Divisibility

- We say that a nonzero *b* **divides** *a* if *a = mb* for some *m,* where *a, b,* and *m* are integers
- *b* divides *a* if there is no remainder on division
- The notation ***b | a*** is commonly used to mean *b* divides *a*
- If *b | a* we say that *b* is a **divisor** of *a*

The positive divisors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24
13 | 182; - 5 | 30; 17 | 289; - 3 | 33; 17 | 0

Properties of Divisibility

- If $a \mid 1$, then $a = \pm 1$
- If $a \mid b$ and $b \mid a$, then $a = \pm b$
- Any $b \neq 0$ divides 0
- If $a \mid b$ and $b \mid c$, then $a \mid c$

- If $b \mid g$ and $b \mid h$, then $b \mid (mg + nh)$ for arbitrary integers m and n

$11 \mid 66$ and $66 \mid 198 = 11 \mid 198$

Properties of Divisibility

- To see this last point, note that:
  - If $b \mid g$, then g is of the form $g = b * g_1$ for some integer g_1
  - If $b \mid h$, then h is of the form $h = b * h_1$ for some integer h_1
- So:
  - $mg + nh = mbg_1 + nbh_1 = b * (mg_1 + nh_1)$

    and therefore b divides $mg + nh$

$b = 7$; $g = 14$; $h = 63$; $m = 3$; $n = 2$
7 | 14 and 7 | 63.
To show 7 (3 * 14 + 2 * 63),
we have (3 * 14 + 2 * 63) = 7(3 * 2 + 2 * 9),
and it is obvious that 7 | (7(3 * 2 + 2 * 9)).

Division Algorithm

- Given any positive integer n and any nonnegative integer a, if we divide a by n we get an integer quotient q and an integer remainder r that obey the following relationship:

$$a = qn + r \qquad 0 \leq r < n;\ q = [a/n]$$

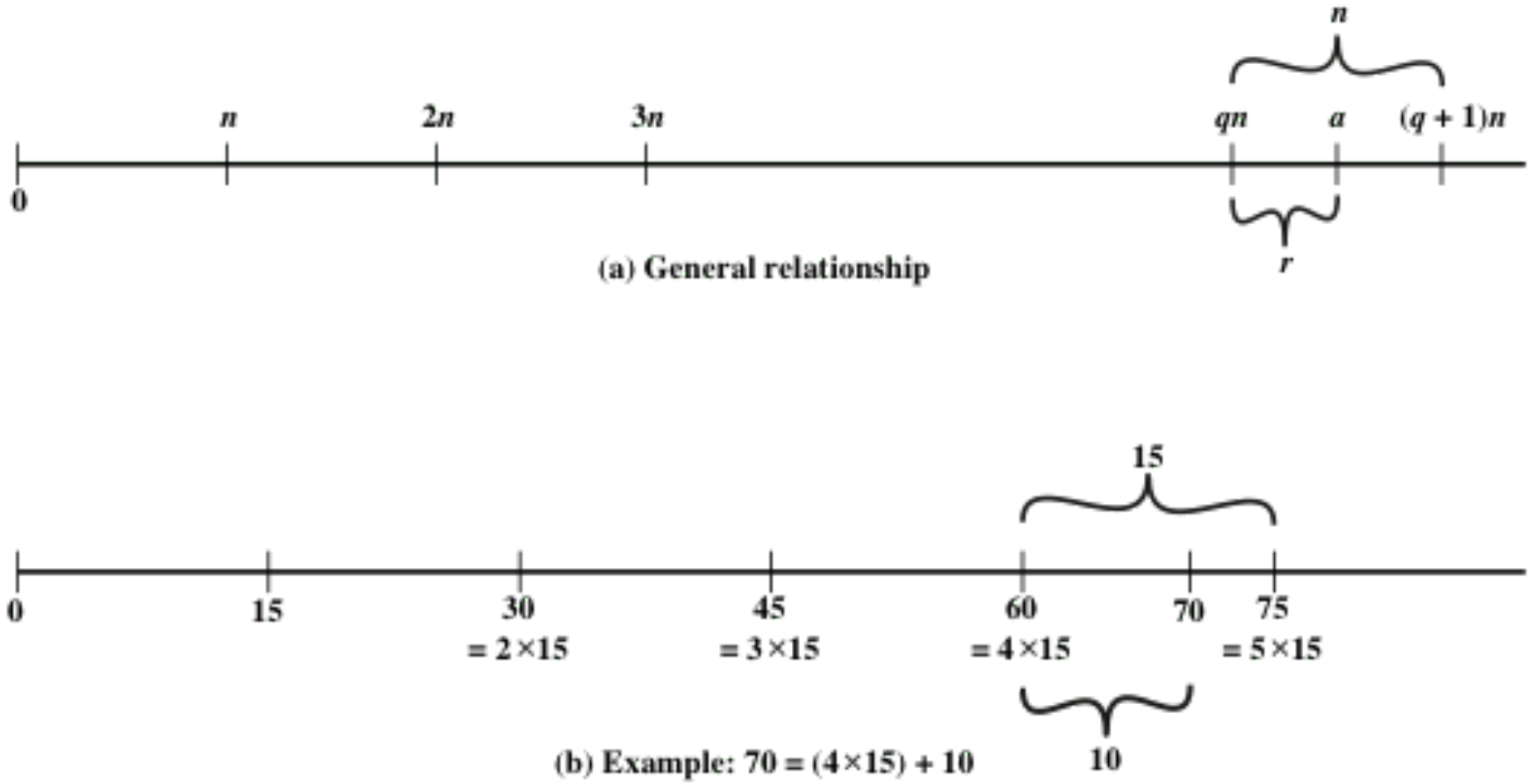


Figure 2.1 The Relationship $a = qn + r;\ 0 \leq r < n$

Euclidean Algorithm

- One of the basic techniques of number theory
- Procedure for determining the greatest common divisor of two positive integers
- Two integers are **relatively prime** if their only common positive integer factor is 1

Greatest Common Divisor (GCD)

- The greatest common divisor of *a* and *b* is the largest integer that divides both *a* and *b*
- We can use the notation gcd*(a,b)* to mean the **greatest common divisor** of *a* and *b*
- We also define gcd(0,0) = 0
- Positive integer *c* is said to be the gcd of *a* and *b* if:
  - *c* is a divisor of *a* and *b*
  - Any divisor of *a* and *b* is a divisor of *c*
- An equivalent definition is:

$$\gcd(a,b) = \max[k, \text{ such that } k \mid a \text{ and } k \mid b]$$

GCD

- Because we require that the greatest common divisor be positive, gcd(*a,b)* = gcd*(a,-b)* = gcd*(-a,b)* = gcd*(-a,-b)*
- In general, gcd(*a,b)* = gcd(| *a* |, | *b* |)

gcd(60, 24) = gcd(60, - 24) = 12

- Also, because all nonzero integers divide 0, we have gcd(a,0) = | a |
- We stated that two integers *a* and *b* are relatively prime if their only common positive integer factor is 1; this is equivalent to saying that *a* and *b* are relatively prime if gcd(*a,b) = 1*

8 and 15 are relatively prime because the positive divisors of 8 are 1, 2, 4, and 8, and the positive divisors of 15 are 1, 3, 5, and 15. So 1 is the only integer on both lists.

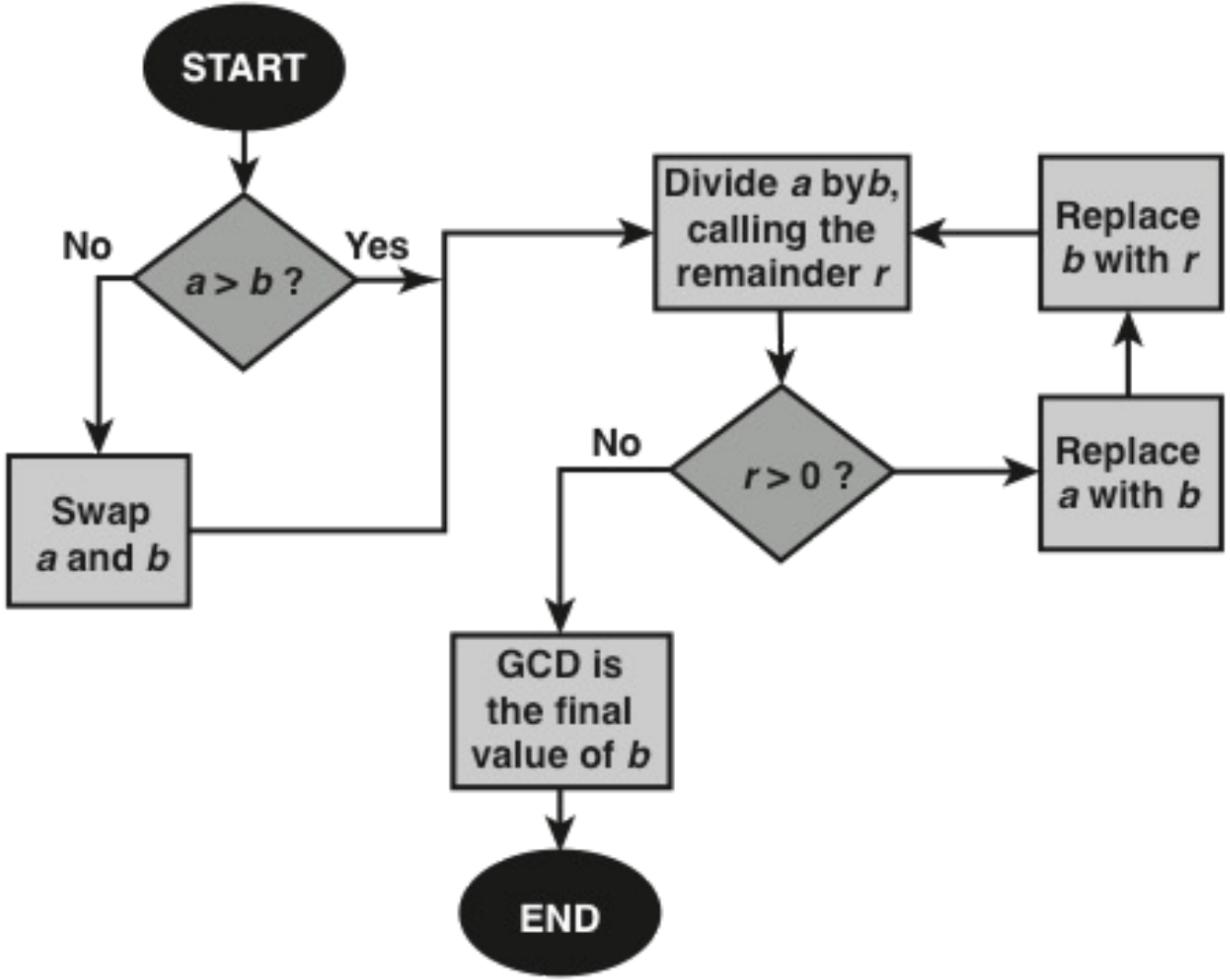


Figure 2.2 Euclidean Algorithm

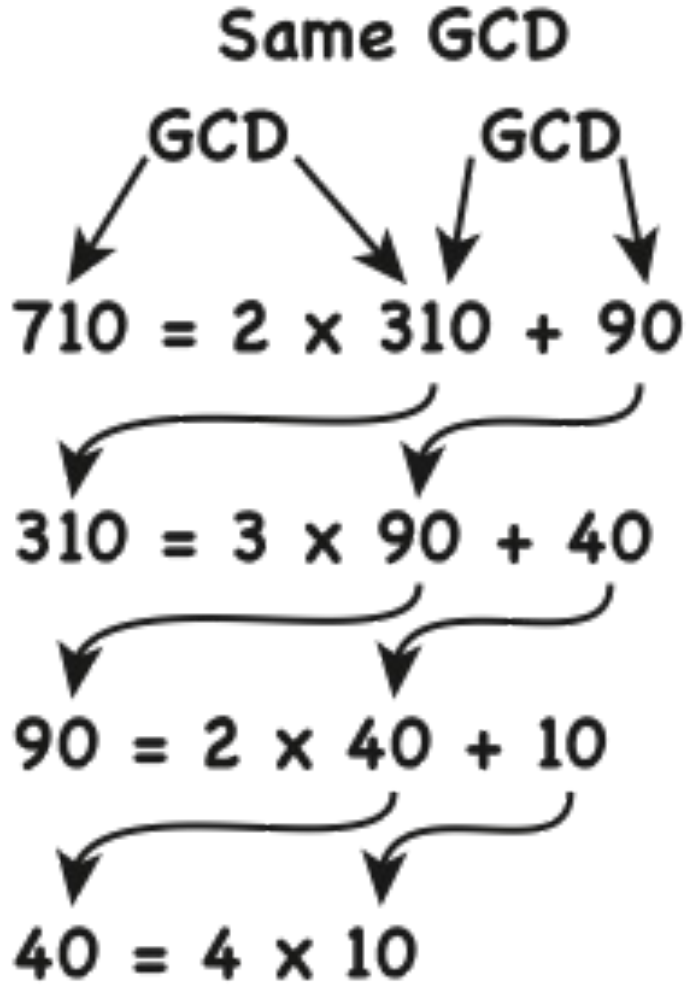


Figure 2.3 Euclidean Algorithm Example: gcd(710, 310)

Table 2.1
Euclidean Algorithm Example

Dividend	Divisor	Quotient	Remainder
$a = 1160718174$	$b = 316258250$	$q_1 = 3$	$r_1 = 211943424$
$b = 316258250$	$r_1 = 211943424$	$q_2 = 1$	$r_2 = 104314826$
$r_1 = 211943424$	$r_2 = 104314826$	$q_3 = 2$	$r_3 = 3313772$
$r_2 = 104314826$	$r_3 = 3313772$	$q_4 = 31$	$r_4 = 1587894$
$r_3 = 3313772$	$r_4 = 1587894$	$q_5 = 2$	$r_5 = 137984$
$r_4 = 1587894$	$r_5 = 137984$	$q_6 = 11$	$r_6 = 70070$
$r_5 = 137984$	$r_6 = 70070$	$q_7 = 1$	$r_7 = 67914$
$r_6 = 70070$	$r_7 = 67914$	$q_8 = 1$	$r_8 = 2156$
$r_7 = 67914$	$r_8 = 2156$	$q_9 = 31$	$r_9 = 1078$
$r_8 = 2156$	$r_9 = 1078$	$q_{10} = 2$	$r_{10} = 0$

(This table can be found on page 34 in the textbook)

Modular Arithmetic

- The modulus
  - If *a* is an integer and *n* is a positive integer, we define *a* mod *n* to be the remainder when *a* is divided by *n;* the integer *n* is called the **modulus**
  - Thus, for any integer *a:*

$$a = qn + r \qquad 0 \leq r < n; \; q = [a/n]$$

$$a = [a/n] * n + (a \bmod n)$$

11 mod 7 = 4; - 11 mod 7 = 3

Modular Arithmetic

- Congruent modulo *n*
  - Two integers *a* and *b* are said to be **congruent modulo *n*** if (*a* mod *n*) = (*b* mod *n*)
  - This is written as $a = b \pmod{n}$[2]
  - Note that if $a = 0 \pmod{n}$, then $n \mid a$

$73 = 4 \pmod{23}$; $21 = -9 \pmod{10}$

Properties of Congruences

- Congruences have the following properties:
  1. $a = b \pmod{n}$ if $n \mid (a - b)$
  2. $a = b \pmod{n}$ implies $b = a \pmod{n}$
  3. $a = b \pmod{n}$ and $b = c \pmod{n}$ imply $a = c \pmod{n}$
- To demonstrate the first point, if $n \mid (a - b)$, then $(a - b) = kn$ for some k
  - So we can write $a = b + kn$
  - Therefore, (a mod n) = (remainder when $b + kn$ is divided by n) = (remainder when b is divided by n) = (b mod n)

$23 = 8 \pmod{5}$ because $23 - 8 = 15 = 5 * 3$
$-11 = 5 \pmod{8}$ because $-11 - 5 = -16 = 8 * (-2)$
$81 = 0 \pmod{27}$ because $81 - 0 = 81 = 27 * 3$

Modular Arithmetic

- Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) * (b \bmod n)] \bmod n = (a * b) \bmod n$

- We demonstrate the first property:
  - Define $(a \bmod n) = r_a$ and $(b \bmod n) = r_b$. Then we can write $a = r_a + jn$ for some integer j and $b = r_b + kn$ for some integer k
  - Then:

$$(a + b) \bmod n = (ra + jn + rb + kn) \bmod n$$
$$= (ra + rb + (k + j)n) \bmod n$$
$$= (ra + rb) \bmod n$$
$$= [(a \bmod n) + (b \bmod n)] \bmod n$$

Remaining Properties:

- Examples of the three remaining properties:

11 mod 8 = 3; 15 mod 8 = 7

[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2

(11 + 15) mod 8 = 26 mod 8 = 2

[(11 mod 8) - (15 mod 8)] mod 8 = - 4 mod 8 = 4

(11 - 15) mod 8 = - 4 mod 8 = 4

[(11 mod 8) * (15 mod 8)] mod 8 = 21 mod 8 = 5

(11 * 15) mod 8 = 165 mod 8 = 5

Arithmetic Modulo 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(This table can be found on page 37 in the textbook)

Multiplication Modulo 8

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

(This table can be found on page 37 in the textbook)

Additive and Multiplicative Inverse Modulo 8

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(This table can be found on page 37 in the textbook)

Properties of Modular Arithmetic for Integers in Z_n

Property	Expression
Commutative Laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative Laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive Law	$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 \times w) \bmod n = w \bmod n$
Additive Inverse $(-w)$	For each $w \in Z_n$, there exists a z such that $w + z \equiv 0 \bmod n$

(This table can be found on page 38 in the textbook)

Extended Euclidean Algorithm Example

i	r_i	q_i	x_i	Y_i
–1	1759		1	0
0	550		0	1
1	109	3	1	–3
2	5	5	–5	16
3	4	21	106	–339
4	1	1	–111	355
5	0	4		

Result: $d = 1$; $x = -111$; $y = 355$

(This table can be found on page 43 in the textbook)

Prime Numbers

- Prime numbers only have divisors of 1 and itself
  - They cannot be written as a product of other numbers
- Prime numbers are central to number theory
- Any integer a > 1 can be factored in a unique way as

$$a = {}_{p1}{}^{a1} * p_2{}^{a2} * \ldots * p_{p1}{}^{a1}$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each a_i is a positive integer

- This is known as the fundamental theorem of arithmetic

Primes Under 2000

2	101	211	307	401	503	601	701	809	907	1009	1103	1201	1301	1409	1511	1601	1709	1801	1901
3	103	223	311	409	509	607	709	811	911	1013	1109	1213	1303	1423	1523	1607	1721	1811	1907
5	107	227	313	419	521	613	719	821	919	1019	1117	1217	1307	1427	1531	1609	1723	1823	1913
7	109	229	317	421	523	617	727	823	929	1021	1123	1223	1319	1429	1543	1613	1733	1831	1931
11	113	233	331	431	541	619	733	827	937	1031	1129	1229	1321	1433	1549	1619	1741	1847	1933
13	127	239	337	433	547	631	739	829	941	1033	1151	1231	1327	1439	1553	1621	1747	1861	1949
17	131	241	347	439	557	641	743	839	947	1039	1153	1237	1361	1447	1559	1627	1753	1867	1951
19	137	251	349	443	563	643	751	853	953	1049	1163	1249	1367	1451	1567	1637	1759	1871	1973
23	139	257	353	449	569	647	757	857	967	1051	1171	1259	1373	1453	1571	1657	1777	1873	1979
29	149	263	359	457	571	653	761	859	971	1061	1181	1277	1381	1459	1579	1663	1783	1877	1987
31	151	269	367	461	577	659	769	863	977	1063	1187	1279	1399	1471	1583	1667	1787	1879	1993
37	157	271	373	463	587	661	773	877	983	1069	1193	1283		1481	1597	1669	1789	1889	1997
41	163	277	379	467	593	673	787	881	991	1087		1289		1483		1693			1999
43	167	281	383	479	599	677	797	883	997	1091		1291		1487		1697			
47	173	283	389	487		683		887		1093		1297		1489		1699			
53	179	293	397	491		691				1097				1493					
59	181			499										1499					
61	191																		
67	193																		
71	197																		
73	199																		
79																			
83																			
89																			
97																			

(This table can be found on page 44 in the textbook)

Fermat's Theorem

- States the following:
  - If p is prime and a is a positive integer not divisible by p then

$$a^{p-1} = 1 \pmod{p}$$

- An alternate form is:
  - If p is prime and a is a positive integer then

$$a^{p} = a \pmod{p}$$

Some Values of Euler's Totient Function *ø(n)*

n	$\phi(n)$
1	1
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4

n	$\phi(n)$
11	10
12	4
13	12
14	6
15	8
16	8
17	16
18	6
19	18
20	8

n	$\phi(n)$
21	12
22	10
23	22
24	8
25	20
26	12
27	18
28	12
29	28
30	8

(This table can be found on page 48 in the textbook)

Euler's Theorem

- States that for every *a* and *n* that are relatively prime:

$$a^{\phi(n)} = 1 \pmod{n}$$

- An alternative form is:

$$a^{\phi(n)+1} = a \pmod{n}$$

Miller-Rabin Algorithm

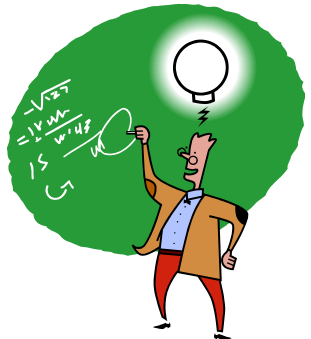
- Typically used to test a large number for primality
- Algorithm is:

TEST (*n*)

1. Find integers k, q, with $k > 0$, q odd, so that $(n - 1) = 2^k q$;
2. Select a random integer a, $1 < a < n - 1$;
3. **if** $a^q \bmod n = 1$ **then** return ("inconclusive") ;
4. **for** $j = 0$ **to** $k - 1$ **do**
5. **if** $(a^{2jq} \bmod n = n - 1)$ **then** return ("inconclusive") ;
6. return ("composite") ;

Deterministic Primality Algorithm

- Prior to 2002 there was no known method of efficiently proving the primality of very large numbers
- All of the algorithms in use produced a probabilistic result
- In 2002 Agrawal, Kayal, and Saxena developed an algorithm that efficiently determines whether a given large number is prime
  - Known as the AKS algorithm
  - Does not appear to be as efficient as the Miller-Rabin algorithm



Chinese Remainder Theorem (CRT)

- Believed to have been discovered by the Chinese mathematician Sun-Tsu in around 100 A.D.
- One of the most useful results of number theory
- Says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli
- Can be stated in several ways

Provides a way to manipulate (potentially very large) numbers mod *M* in terms of tuples of smaller numbers

- This can be useful when *M* is 150 digits or more
- However, it is necessary to know beforehand the factorization of *M*

Powers of Integers, Modulo 19

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

(This table can be found on page 57 in the textbook)

Tables of Discrete Logarithms, Modulo 19

(a) Discrete logarithms to the base 2, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{2,19}(a)$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

(b) Discrete logarithms to the base 3, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

(c) Discrete logarithms to the base 10, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{10,19}(a)$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9

(d) Discrete logarithms to the base 13, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{13,19}(a)$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9

(e) Discrete logarithms to the base 14, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9

(f) Discrete logarithms to the base 15, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{15,19}(a)$	18	5	11	10	8	16	12	15	4	13	6	3	7	17	1	2	14	9

(This table can be found on page 60 in the textbook)

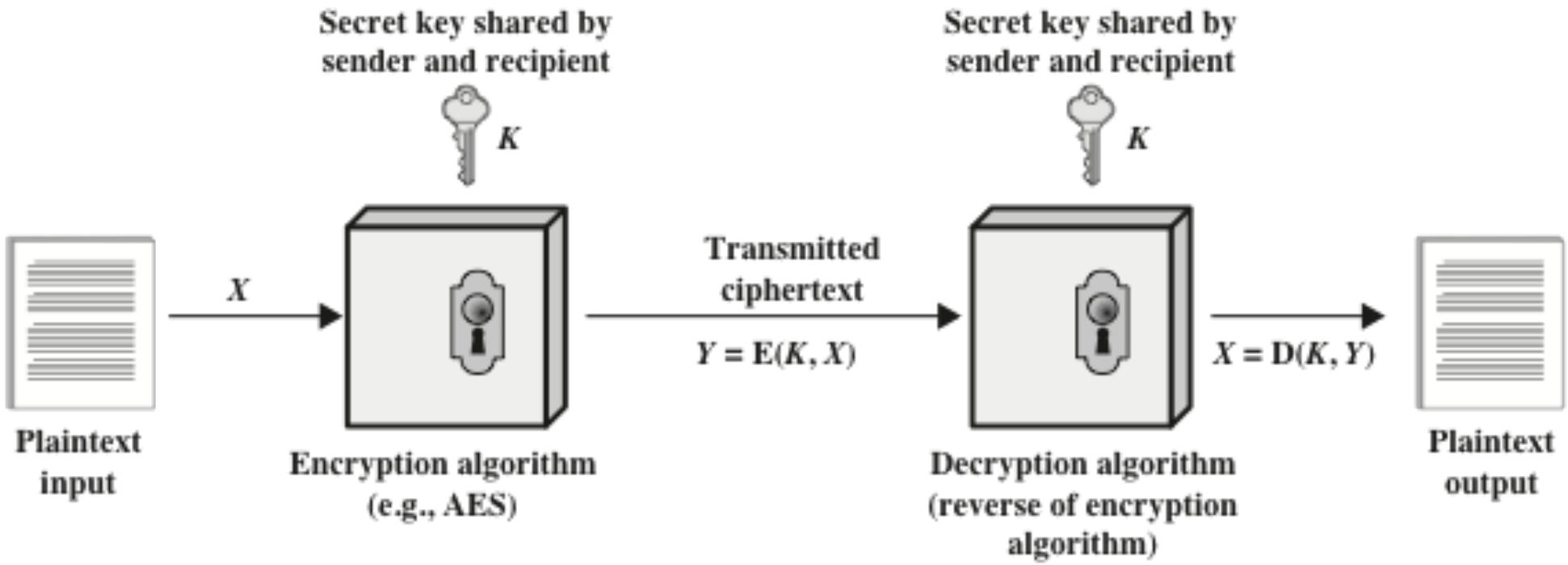
Cryptography

Classical Encryption Techniques

Definitions

Plaintext	An original message
Ciphertext	The coded message
Enciphering/encryption	The process of converting from plaintext to ciphertext
Deciphering/decryption	Restoring the plaintext from the ciphertext
Cryptography	The area of study of the many schemes used for encryption
Cryptographic system/cipher	A scheme
Cryptanalysis	Techniques used for deciphering a message without any knowledge of the enciphering details
Cryptology	The areas of cryptography and cryptanalysis

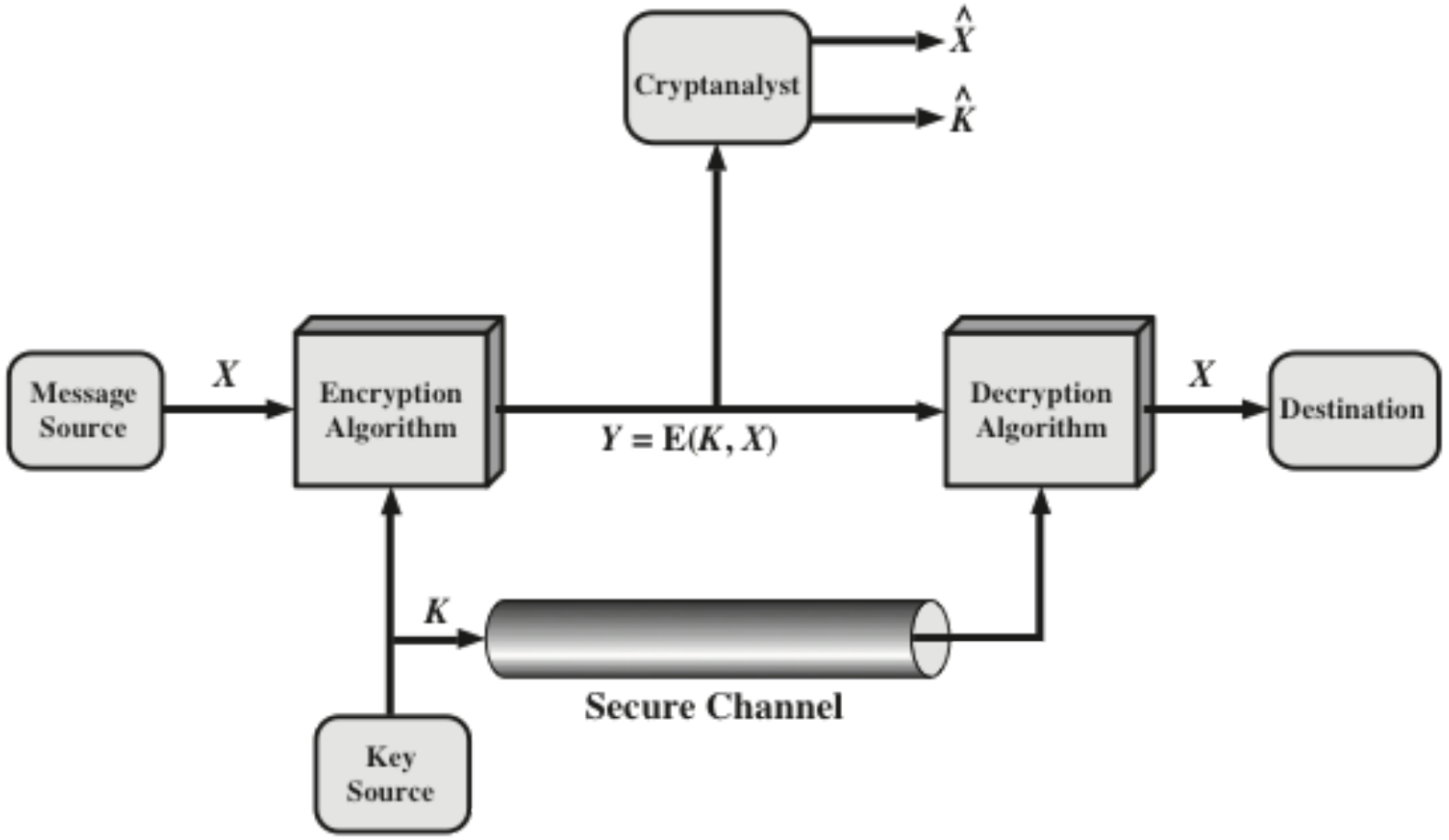
Simplifies Model of Symmetric Encryption



Symmetric Cipher Model

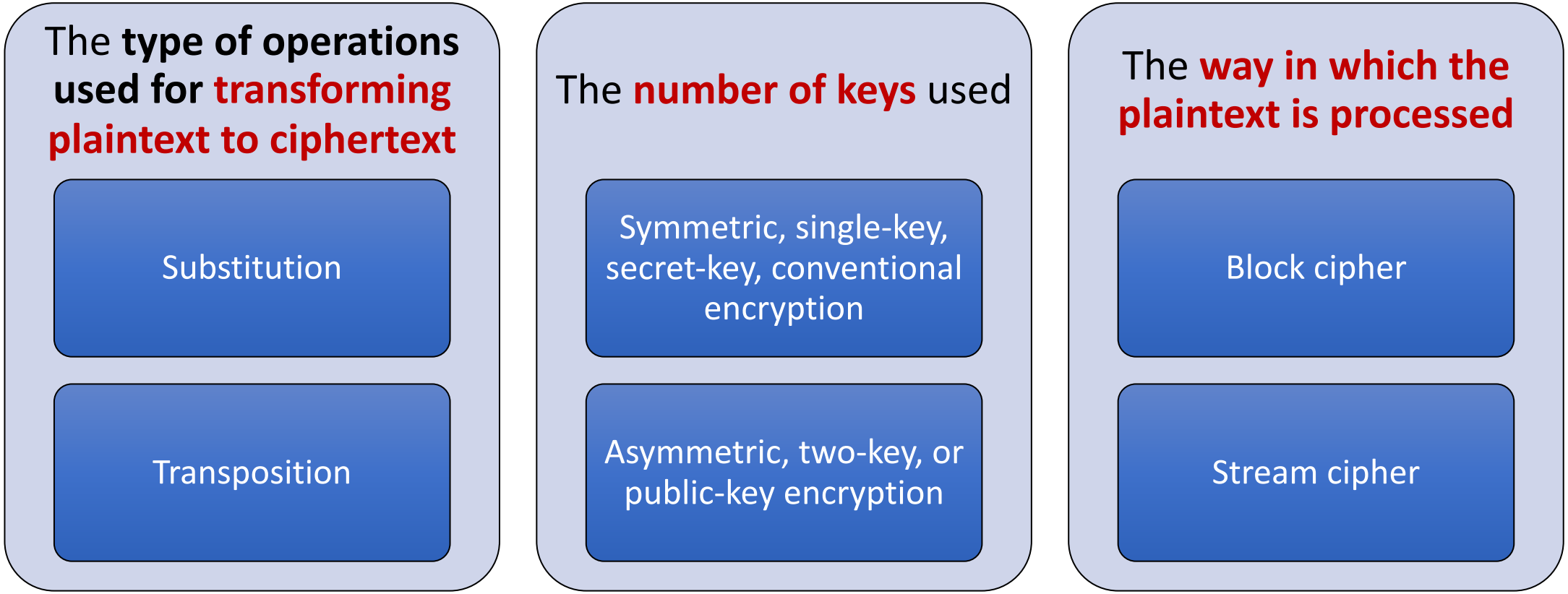
- There are **two requirements** for **secure use** of conventional encryption:
  - A **strong encryption algorithm**
  - Sender and receiver must have obtained copies of the **secret key in a secure fashion** and must keep the key secure

Model of Symmetric Cryptosystem



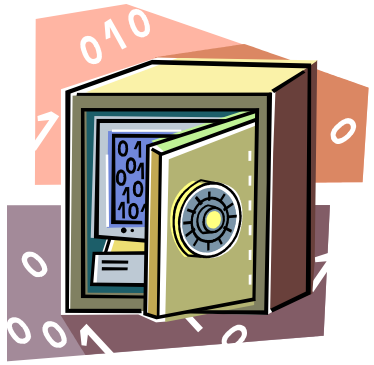
Cryptographic Systems

- Characterized along three independent dimensions:



Encryption Scheme Security

- Unconditionally secure
  - No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there
- Computationally secure
  - The cost of breaking the cipher **exceeds the value of the encrypted information**
  - The time required to break the cipher **exceeds the useful lifetime of the information**



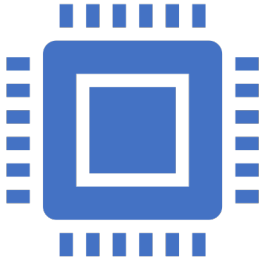
Brute-Force Attack

Involves **trying every possible key** until an intelligible translation of the ciphertext into plaintext is obtained

On average, half of all possible keys must be tried to achieve success

To supplement the brute-force approach, **some degree of knowledge about the expected plaintext** is needed, and some means of automatically distinguishing plaintext from garbage is also needed

Cryptanalysis and Brute-Force Attack



Cryptanalysis

Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext

Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

Brute-force attack

Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained

On average, half of all possible keys must be tried to achieve success

Transforming plaintext to ciphertext: Substitution Technique

- The letters of plaintext are **replaced** by other letters or by numbers or symbols

- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- Simplest and earliest known use of a substitution cipher Used by Julius Caesar

- Involves replacing each letter of the alphabet with the letter standing three (3 is the key) places further down the alphabet (a becomes D)

- Alphabet is wrapped around so that the letter following Z is A

plain: **meet me after the toga party**

cipher: **PHHW PH DIWHU WKH WRJD SDUWB**

Brute-Force or Cryptanalysis of Caesar Cipher

KEY						
	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

Caesar Cipher Algorithm

- Can define transformation as:

```
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
```

- Mathematically give each letter a number

```
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
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

- Algorithm can be expressed as:

  $c = E(3, p) = (p + 3) \bmod (26)$

- A shift may be of any amount, so that the general Caesar algorithm is:

  $$C = E(k, p) = (p + k) \bmod 26$$

- Where k takes on a value in the range 1 to 25
- The decryption algorithm is simply:

  $$p = D(k, C) = (C - k) \bmod 26$$

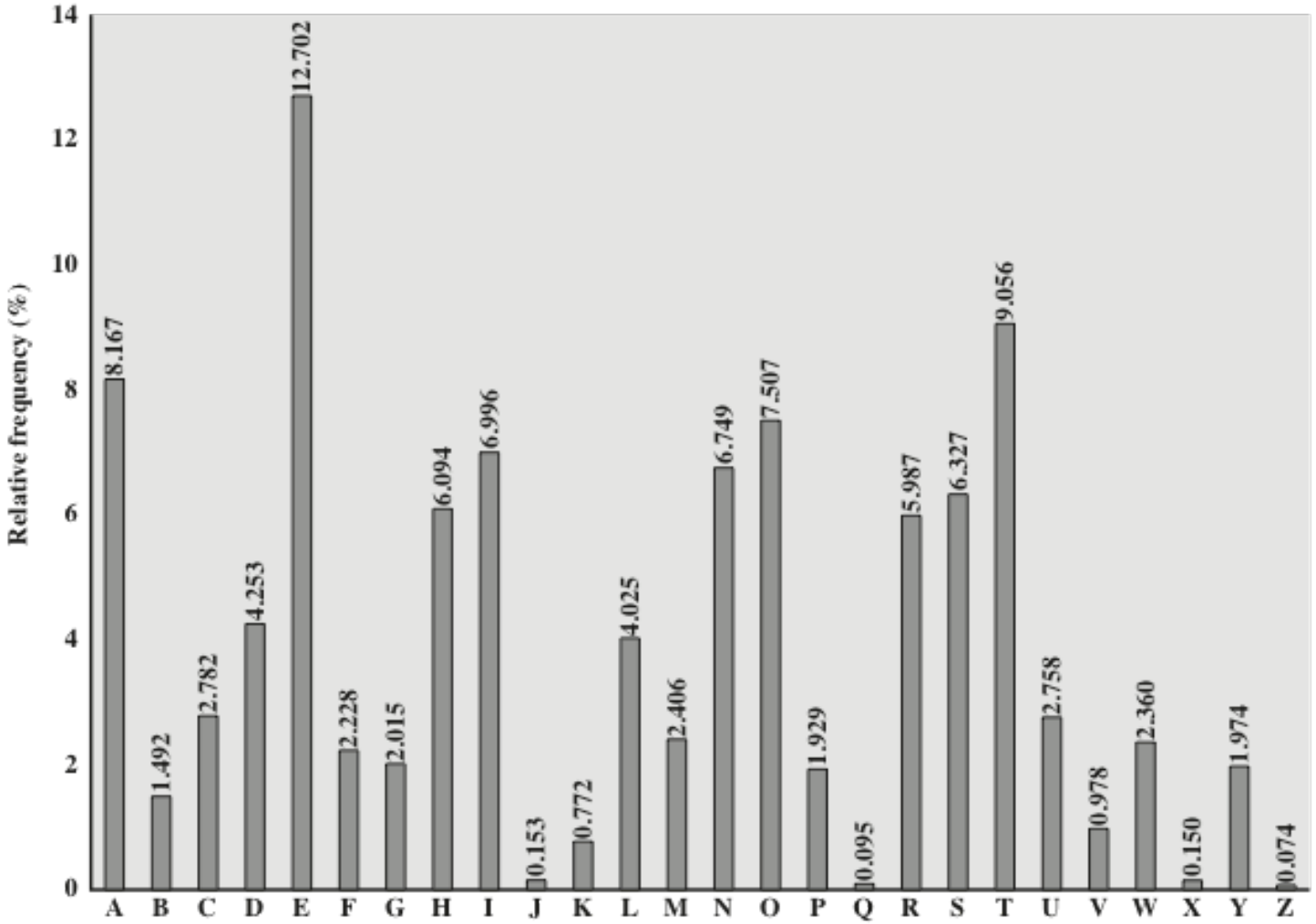
Monoalphabetic Cipher

- With only 25 possible keys, the Caesar cipher is far from secure
- A dramatic **increase in the key space** can be achieved by allowing **an arbitrary substitution**
- Permutation
  - Of a finite set of elements *S* is **an ordered sequence of all the elements of *S*** , with each element **appearing exactly once**, For example, if **S = {a, b, c},** there are six permutations of *S* : **abc, acb, bac, bca, cab, cba**
  - In general, there are **n!** permutations of a set of **n** elements
  - by permutation of the 26 alphabetic characters, there are **26! possible keys**
- Approach is referred to as a ***monoalphabetic substitution cipher*** because a single cipher alphabet is used per message

Monoalphabetic Cipher Limitation

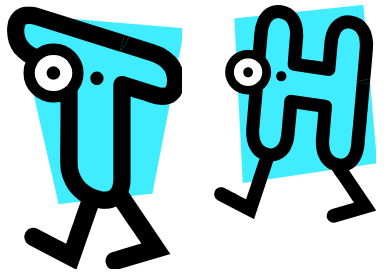
- If the **cryptanalyst** knows the **nature of the plaintext** (e.g., English text)

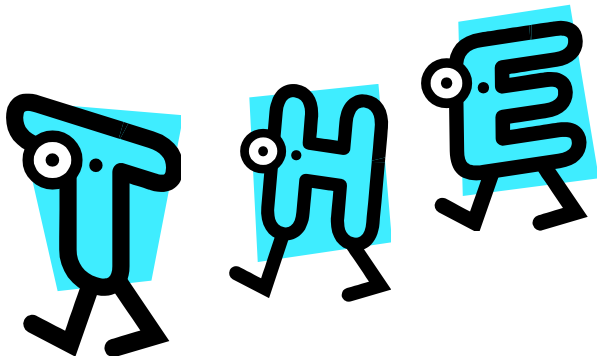
- Relative Frequency of Letters in English Text
  - The relative frequency of the letters can be determined and compared to a standard frequency distribution for English



Monoalphabetic Ciphers Limitation

- Easy to break because they reflect the frequency data of the original alphabet
- Countermeasure is to provide **multiple substitutes (homophones) for a single letter**
- Digram
  - Two-letter combination
  - Most common is ***th***
- Trigram
  - Three-letter combination
  - Most frequent is ***the***





Playfair Cipher

- Best-known **multiple-letter encryption cipher**
- Treats digrams in the plaintext as single units and translates these units into ciphertext digrams
- **Based on the use of a 5 x 5 matrix of letters constructed using a keyword**
- Invented by British scientist Sir Charles Wheatstone in 1854
- Used as the standard field system by the British Army in World War I and the U.S. Army and other Allied forces during World War II

Playfair Key Matrix

- Example:
  - Keyword: MONARCHY
  - **Fill in letters of keyword** (minus duplicates) from left to right and from top to bottom
  - Fill in the remainder of the matrix with **the remaining letters in alphabetic order**

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Key Matrix

- Repeating plaintext letters that are in the same pair are separated with a filler letter, such as **x**, E.g., **balloon** -> **balxloon**
- Two plaintext letters that fall in the **same row** of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. E.g., **ar** -> **RM**
- Two plaintext letters that fall in the **same column** are each replaced by the letter beneath, with the top element of the column circularly following the last. E.g., **mu** -> **CM**
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
  - E.g., **hs** -> **BP** and **ea** -> **IM** (or JM, as the encipherer wishes).

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Polyalphabetic Ciphers

- Polyalphabetic substitution cipher
  - Improves on the simple monoalphabetic technique by using **different monoalphabetic substitutions** as one proceeds through the plaintext message

All these techniques have the following features in common:
• A set of related monoalphabetic substitution rules is used • A key determines which particular rule is chosen for a given transformation

Vigenère Cipher

- Best known and one of the simplest polyalphabetic substitution ciphers
- In this scheme, the **set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25**
- Each cipher is denoted by a **key letter** which is the ciphertext letter that substitutes for the plaintext letter.

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
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**

Example of Vigenère Cipher

- To encrypt a message, a key is needed that is as long as the message
- Usually, the key is a repeating **keyword**
- For example, if the keyword is "***deceptive"***, the message *"we are discovered save yourself*" is encrypted as:

key: deceptivedeceptivedeceptive ← Key Letters

3 4 2 4 15 19 8 21 4 ….

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

cipher with a shift of 3 is denoted by the key value 3

W is substituted with the letter Z which is in the third position after w (i.e., w, x, y ,z)

Vigenère Autokey System

- A keyword is concatenated with the plaintext itself to provide a running key
- Example for Keyword "*detective*"

  key: deceptivewearediscoveredsav

  plaintext: wearediscoveredsaveyourself

  ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA
- Even this scheme is vulnerable to cryptanalysis, why?
  - Because the key and the plaintext share the same **frequency distribution** of letters, a statistical technique can be applied

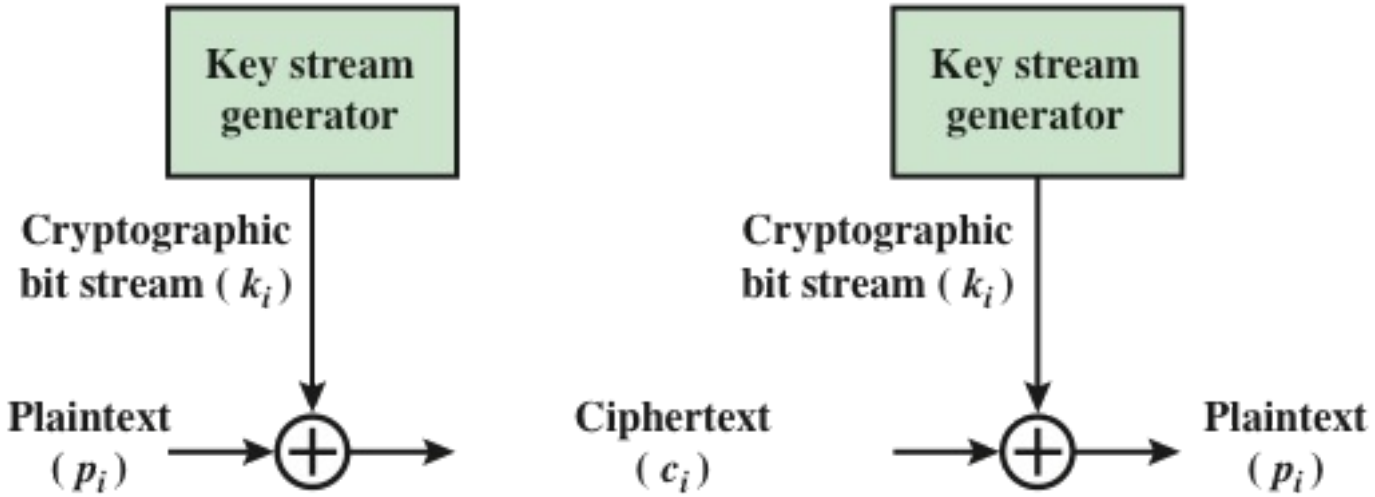
Vernam Cipher

- The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it.

- Such a system was introduced by an AT&T engineer named Gilbert Vernam in 1918. His system **works on binary data (bits) rather than letters.**

Vernam Cipher

- Vernam proposed the use of a running loop of tape that eventually repeated the key, so that in fact the system worked with a **very long but repeating keyword.**
- Can be broken **with sufficient ciphertext**, the use of known or probable plaintext sequences, or both.



One-Time Pad

- **Improvement to Vernam** cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
  - Use a **random key** that is **as long as the message** so that the **key need not be repeated**
  - **Key is used to encrypt and decrypt a single message** and then is discarded
  - **Each new message requires a new key** of the same length as the new message
- **Scheme is unbreakable**
  - Produces random output that bears no statistical relationship to the plaintext
  - Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code

One-Time Pad - Difficulties

- The one-time pad offers **complete security** but, in practice, has two fundamental difficulties:
  - There is **the practical problem of making large quantities of random keys**
    - Any heavily used system might require millions of random characters on a regular basis
  - Mammoth key distribution problem
    - For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time **pad is of limited utility**
  - Useful primarily for low-bandwidth channels requiring very high security
- The one-time pad is the only cryptosystem that exhibits ***perfect secrecy*** (see Appendix F)

Transposition Cipher: Rail Fence Cipher

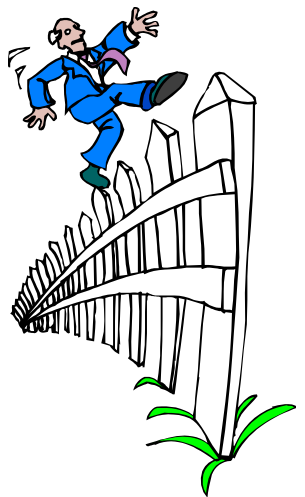
- Simplest **transposition cipher** (a sort of permutation on the plaintext letter)
- Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
- To encipher the message "***meet me after the toga party***" with a rail **fence of depth 2**, we would write:

m e m a t r h t g p r y

e t e f e t e o a a t

Encrypted message is:

MEMATRHTGPRYETEFETEOAAT



Row Transposition Cipher

- Is a more complex transposition
- **Write** the message in a rectangle, **row by row**, and **read** the message off, **column by column**, but permute the order of the columns
  - **The order of the columns then becomes the key to the algorithm**

Key:	**4 3 1 2 5 6 7**
Plaintext:	a t t a c k p
	o s t p o n e
	d u n t i l t
	w o a m x y z
Ciphertext:	TTNAAPTMTSUOAODWCOIXKNLYPETZ

Cryptography

Block Ciphers and the Data Encryption Standard

Stream Cipher

Encrypts a **digital data stream:** one bit or one byte at a time

Examples:

- Autokeyed Vigenère cipher
- Vernam cipher

In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream is as long as the plaintext bit stream

If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream

- Keystream must be provided to both users in advance via some independent and secure channel
- This introduces insurmountable logistical problems if the intended data traffic is very large

For practical reasons the bit-stream generator must be implemented as an **algorithmic procedure** so that the cryptographic bit stream can be produced by both users

It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream

The two users need only share the generating key and each can produce the keystream

Block Cipher

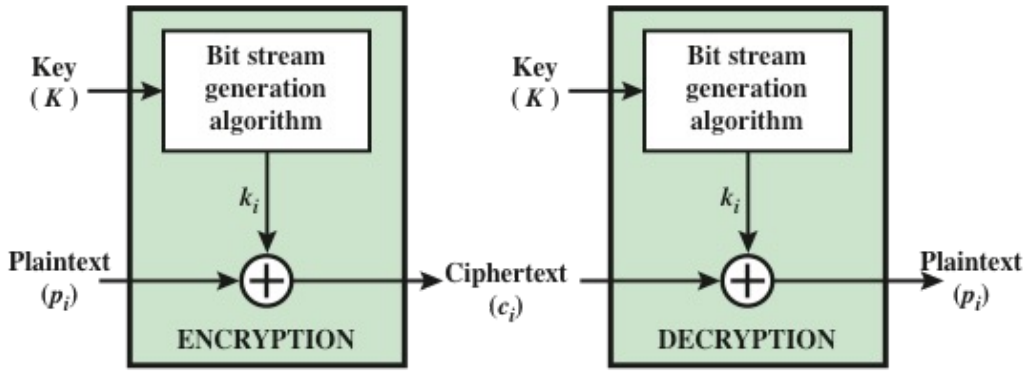
A **block of plaintext** is treated as a whole and used to produce a **ciphertext block** of equal length

Typically a block size of **64 or 128 bits** is used

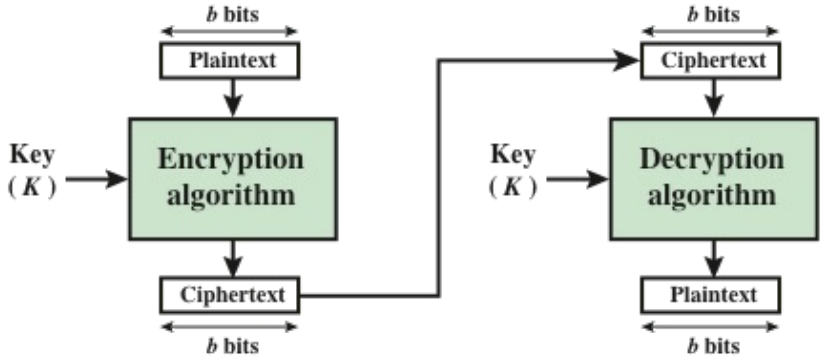
As with a stream cipher, the **two users share a symmetric encryption key**

The **majority of network-based symmetric cryptographic** applications make use of **block ciphers**

Stream Cipher vs. Block Cipher



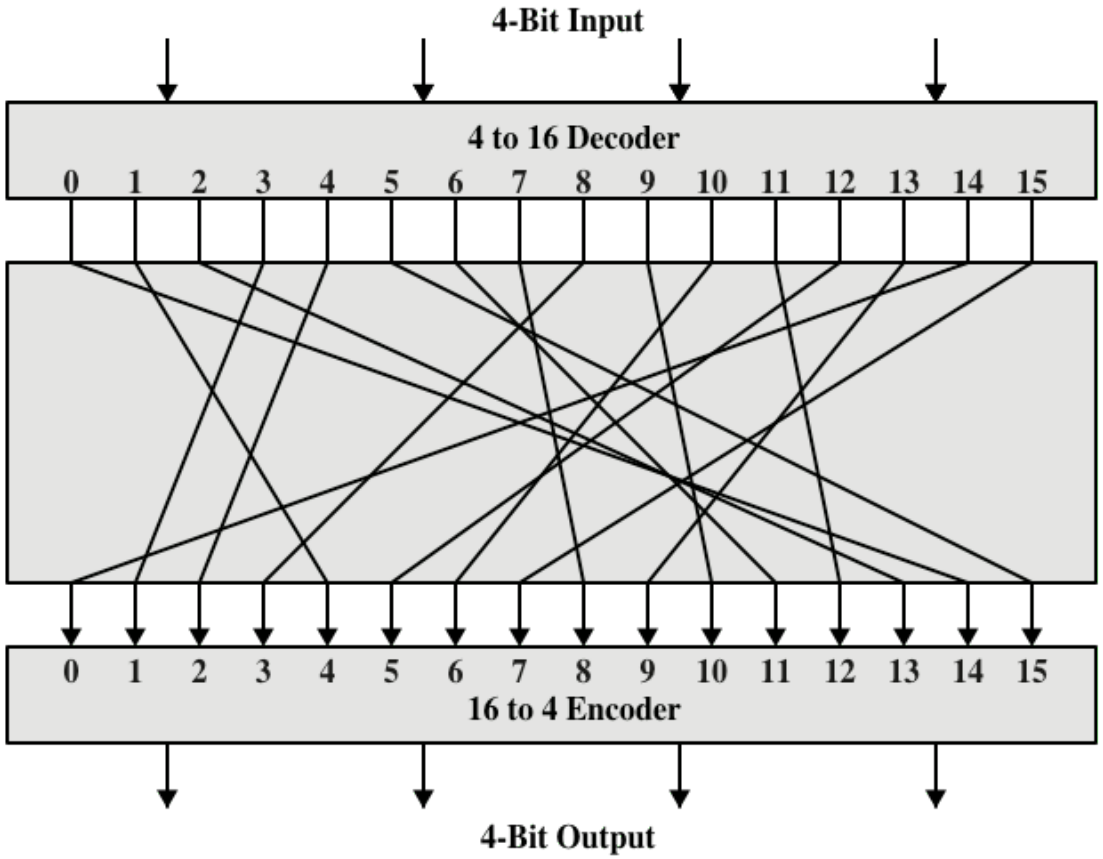
(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

General n-bit-n-bit Block Substitution

- A block cipher operates on a plaintext block of **n bits** to produce a ciphertext block of n bits.
- There are **2^n possible different plaintext blocks**
- For the encryption to be **reversible**, **each must produce a unique ciphertext block**



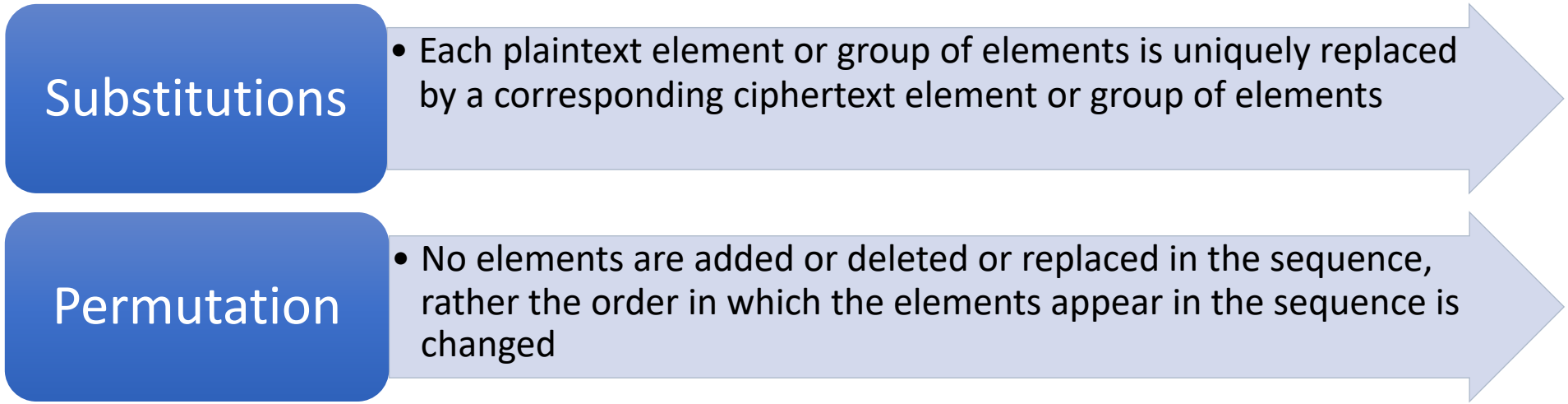
Encryption and Decryption Tables

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Feistel Cipher

- Feistel proposed the use of a cipher that alternates **substitutions** and **permutations**

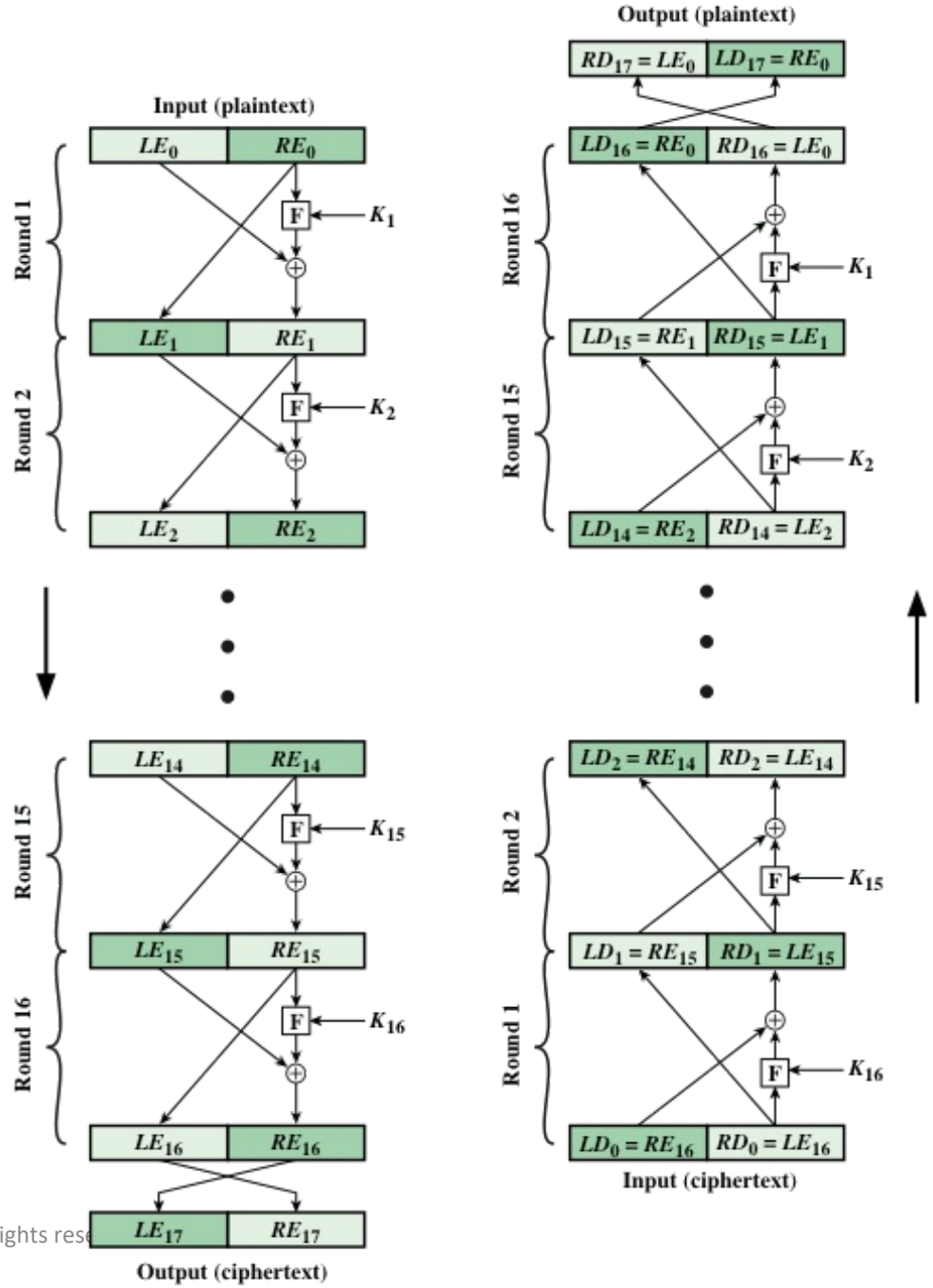


- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions [SHAN49]

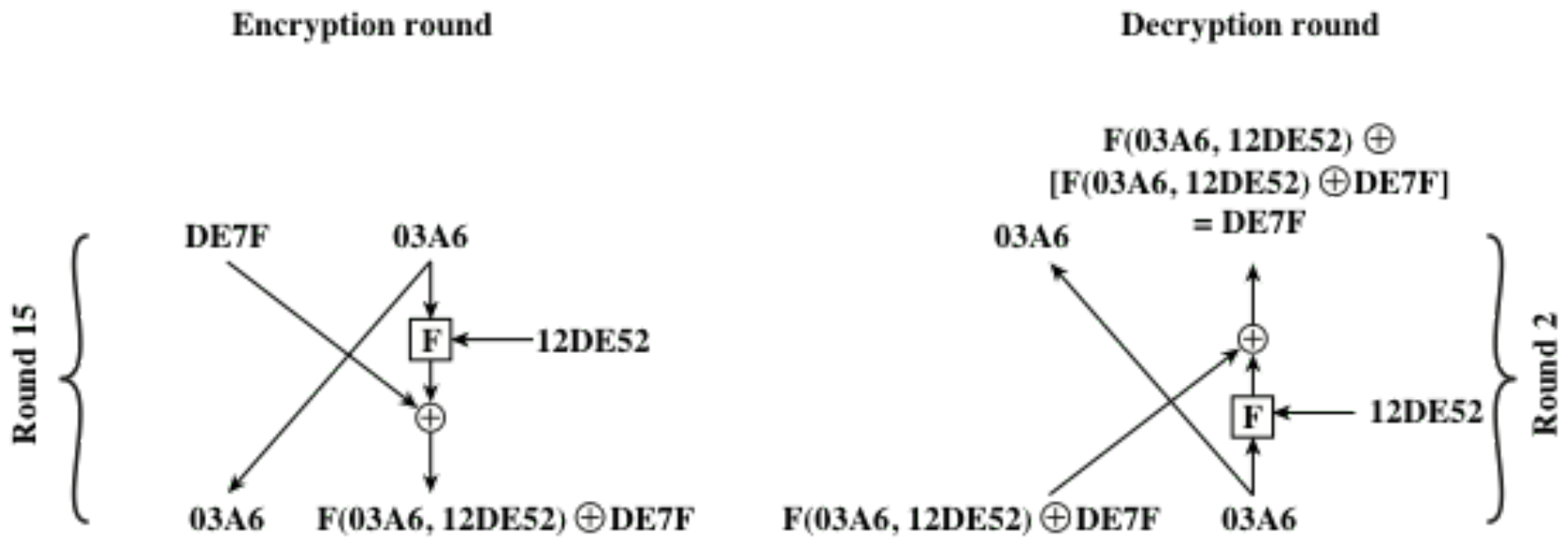
- Is the structure used by many significant symmetric block ciphers currently in use (TDEA and AES)

Feistel Encryption and Decryption

- The plaintext block is **divided** into **two halves**, **LE_0 and RE_0**
- The two halves of the data pass through **16 rounds of processing**
- All rounds have the **same structure**
- Each round i has as inputs LE_{i-1} and RE_{i-1} derived from the previous round, as well as a subkey **K_i**



Feistel Example

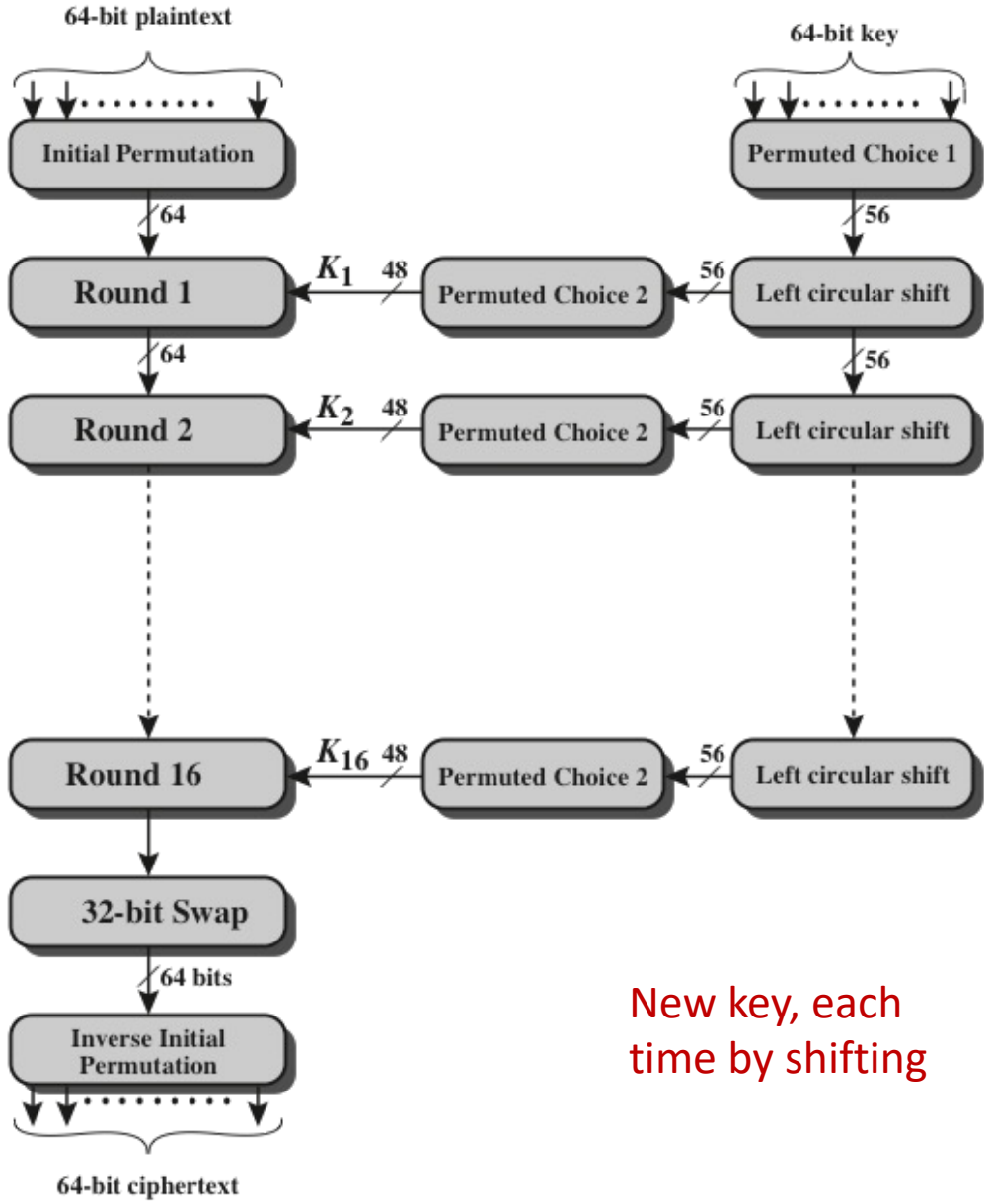


Data Encryption Standard (DES)

- Issued in **1977** by the **National Bureau of Standards (now NIST)** as Federal Information Processing Standard 46
- Was the **most widely used encryption scheme** until the introduction of the **Advanced Encryption Standard (AES) in 2001**
- Algorithm itself is referred to as the **Data Encryption Algorithm (DEA)**
  - Data are encrypted in **64-bit blocks** using a **56-bit key**
  - The algorithm transforms 64-bit input in a series of steps into a 64-bit output
  - **The same steps**, with the **same key**, are used to reverse the encryption

General Depiction of DES Encryption Algorithm

- ẃo inputs: **the plaintext and the key.**
- First, the 64-bit plaintext passes through an **initial permutation (IP)**
- This is followed by a phase consisting of **sixteen rounds** of the same function
- **Then, the left and right halves of the output are swapped**
- Finally, the output is passed through **a permutation [IP $^{-1}$]** that is the inverse of the initial permutation function



New key, each time by shifting

DES Disadvantage

- **Key Length**: DES has a fixed key length of 56 bits. **In today's computing environment**, this key length is considered too short to provide a high level of security. Brute-force attacks, where an attacker systematically tries all possible keys, can be performed relatively quickly with modern computing power.

Block Cipher Design Principles: Number of Rounds

The greater the number of rounds, the more difficult it is to perform cryptanalysis

In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack

If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- **The more nonlinear F**, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

The algorithm should have good avalanche properties

Strict avalanche criterion (SAC)

States that any output bit j of an S-box should change with probability 1/2 when any single input bit i is inverted for all i , j

Bit independence criterion (BIC)

States that output bits j and k should change independently when any single input bit i is inverted for all i , j , and k

Avalanche Effect

- **A slight change in either Plaintext or Key should result in a significant change in the Ciphertext**

- One of the Desirable property of any encryption algorithm

Block Cipher Design Principles: Key Schedule Algorithm

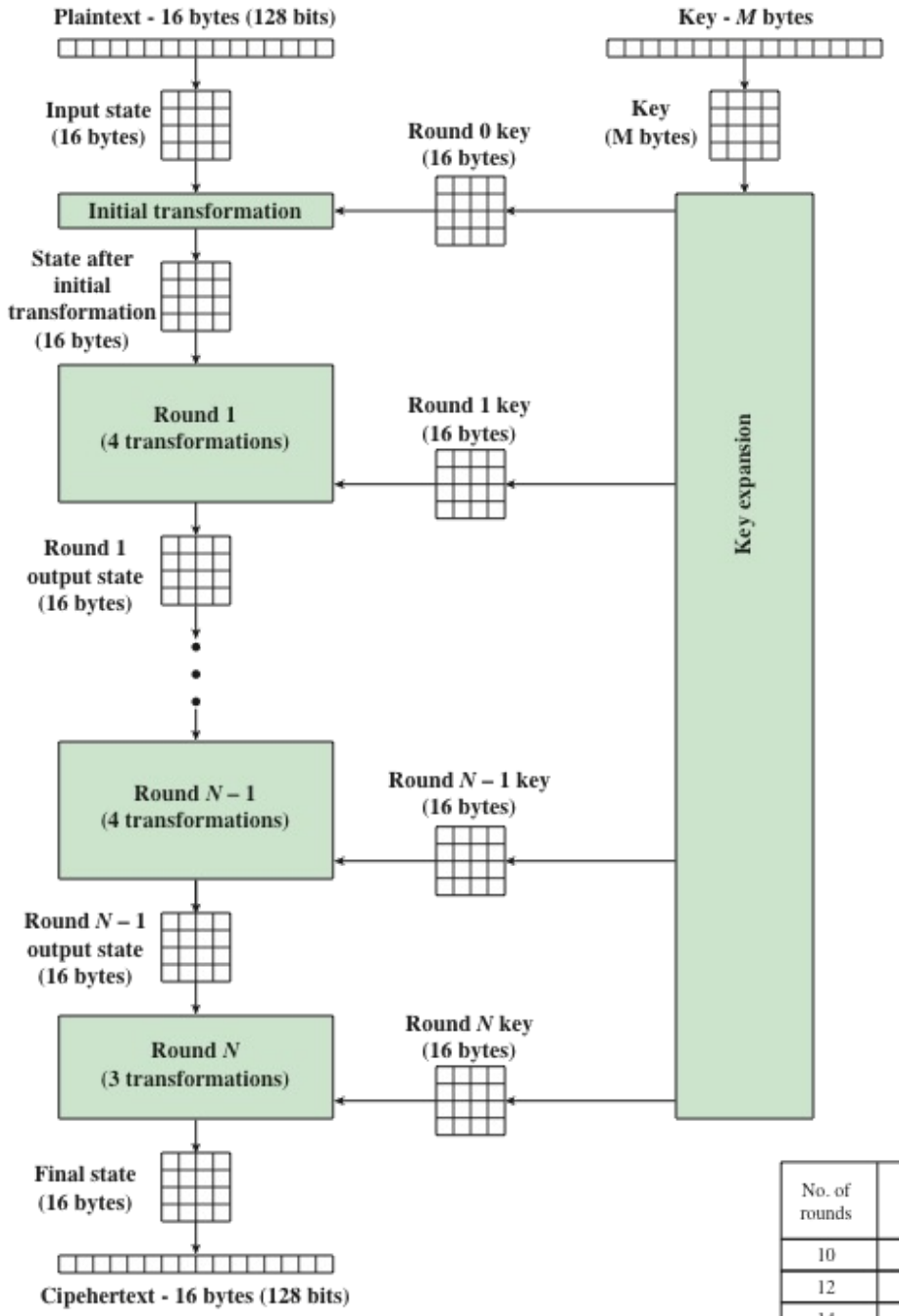
- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to **maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key**
- **It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion**

Advanced Encryption Standard (AES)

- In the Advanced Encryption Standard (AES) all operations are performed on 8-bit bytes
- The arithmetic operations of addition, multiplication, and division are performed over the **finite field**
  - A finite field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set
- Division is defined with the following rule:
  - $a / b = a (b^{-1})$
- An example of a finite field (one with a finite number of elements) is the set Z_p consisting of all the integers $\{0, 1, \ldots, p - 1\}$, where p is a prime number and in which arithmetic is carried out modulo p

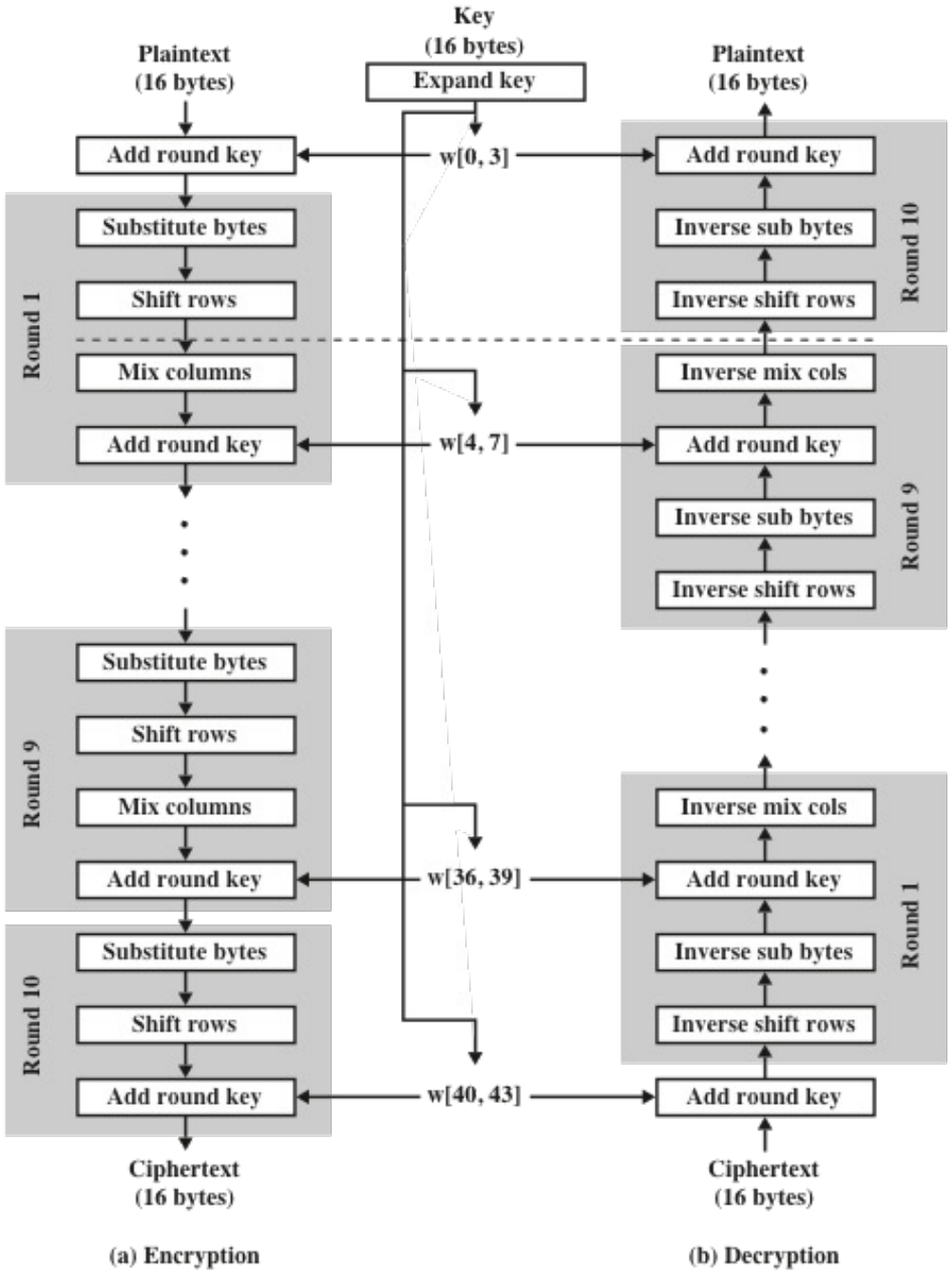
AES Encryption Process

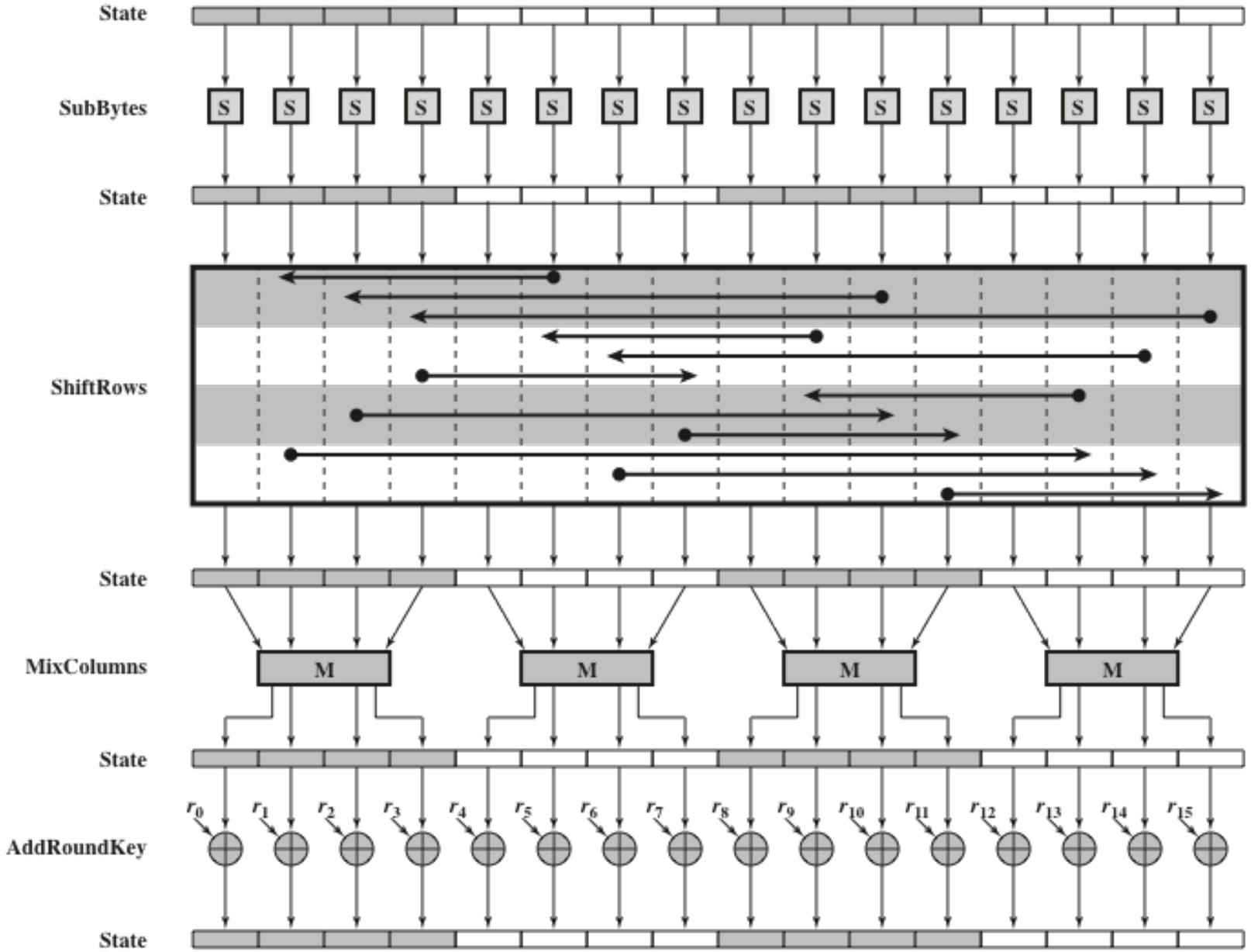
- The cipher takes a plaintext block size of 128 bits, or 16 bytes.
- The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits).
- The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.
- The plaintext and the key are transformed into squared matrix



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

- Key is expanded and in each round a part of that is used
- Several Processing in each round
- At the end of each round the key is applied





Rivest-Shamir-Adleman (RSA) Algorithm

- Developed in **1977** at MIT by Ron Rivest, Adi Shamir & Len Adleman
- **Most widely used general-purpose approach to public-key encryption**
- Is a cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n
  - A typical size for n is 1024 bits, or 309 decimal digits

RSA Algorithm

- RSA makes use of an expression with exponentials
- Plaintext is **encrypted in blocks** with each **block having a binary value less than some number *n***
- Encryption and decryption are of the following form, for some plaintext block *M* and ciphertext block C

  $$C = M^e \bmod n$$

  $$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

- Both sender and receiver must know the **value of *n***
- The sender knows the value of ***e***, and only the receiver knows the value of ***s***
- This is a public-key encryption algorithm with a public key of ***PU={e,n}*** and a private key of ***PR={d,n}***

Algorithm Requirements

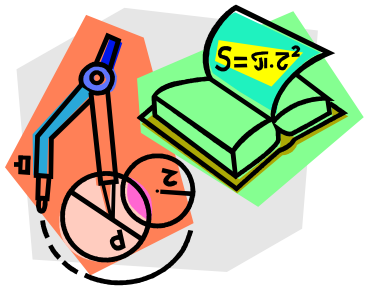
- For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

    1. It is possible to find values of ***e, d, n***

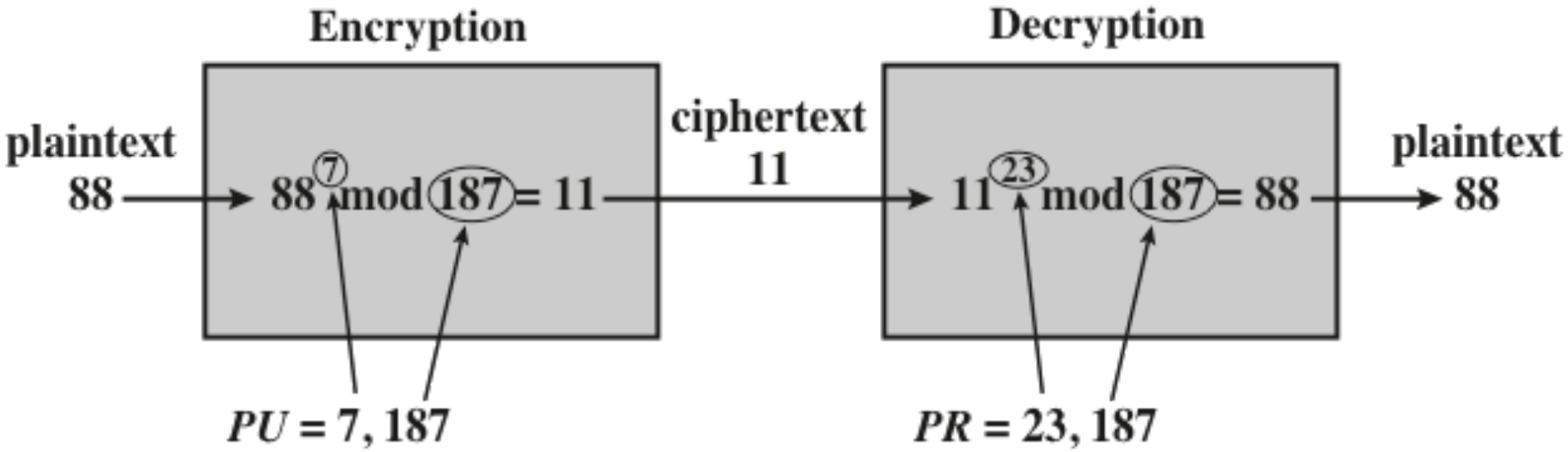
        such that $M^{ed} \bmod n = M$ for all $M < n$

    2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$

    3. It is **infeasible** to determine *d* given *e* and *n*



Example of RSA Algorithm



Practical Use of Symmetric Crypto

- **Libraries and tools to use**
  - Keys creation
  - Key storage
  - Choosing the best algorithm for the task

- Write programs that exchange encrypted messages
  - for the PL classes

These recommendations are only valid for limited time, you should keep an eye of the news

Python Libraries

```
$ pip install cryptography
```

- recipes - provides a simple API for proper symmetric encryption
- hazmat (hazardous materials) - provides low-level cryptographic primitives

```
$ pip install pynacl
```

- Based on NaCl / libsodium
- Designed to be easy-to-use and high-speed

Java Library

- Java Cryptography Extension
  - SunJCE included since Java 1.4

```
java.security
java.security.cert
java.security.spec
java.security.interfaces
javax.crypto
javax.crypto.spec
javax.crypto.interfaces
```

- BouncyCastle

```
Security.addProvider(new BouncyCastleProvider());
cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", “BC”);
byte[] keyBytes = new byte[]{0,1,2,3,4,5,6,7,8,9};

SecretKeySpec key = new SecretKeySpec(keyBytes, "RawBytes");
cipher.init(Cipher.ENCRYPT_MODE, key);
cipher.init(Cipher.DECRYPT_MODE, key);

byte[] plainText = "abcdefghijklmnopqrstuvwxyz".getBytes("UTF-8");
byte[] cipherText = cipher.doFinal(plainText);
```

.NET Libraries

- Very similar to Java
- BouncyCastle

C Library

- OpenSSL
  - Standard
  - Well established
- libsodium
  - Security and ease of use → less flexibility
  - Simple interface, designed to be *less error prone*
  - Good performance
- Libgcrypt
  - Well established, used in GnuTLS
  - More complete in terms of algorithms and

BIBLIOGRAPHY