NETWORK AND SYSTEM SECURITY (CORE ELECTIVE - 5) CS424

Unit 2 : Review of Cryptographic Tools

Syllabus

- **INTRODUCTION** **(04 Hours)**

  Introduction to Network and System Security, Security Attacks, Security Requirements, Confidentiality, Integrity, and Availability, Security Mechanisms, NIST Security Standards, Assets and Threat Models.

- **REVIEW OF CRYPTOGRAPHIC TOOLS** **(04 Hours)**

  Number Theory, Prime Numbers, Modular Arithmetic, Confidentiality with Symmetric Encryption, Message Authentication and Hash Functions, Public-Key Encryption, Digital Signatures and Key Management, Random and Pseudorandom Numbers.

- **SYSTEM SECURITY** **(10 Hours)**

  User Authentication - Means of Authentication, Password-Based Authentication, Token-Based Authentication, Biometric Authentication, Remote User Authentication, Access Control-Access Control Principles, Subjects, Objects, and Access Rights, Discretionary Access Control, Example: UNIX File Access Control, Role-Based Access Control, Database Security-The Need for Database Security, Database Access Control, Inference, Statistical Databases, Database Encryption, Cloud Security, Malicious Software, Intruders, Denial of Service and Distributed Denial of Service attacks, Intrusion Detection and Prevention.

Syllabus...

- **SOFTWARE SECURITY AND TRUSTED SYSTEMS** **(12 Hours)**

  Buffer Overflow-Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks, Software Security-Software Security Issues, Handling Program Input, Writing Safe Program Code, Interacting with the Operating System and Other Programs, Handling Program Output, Operating System Security-System Security Planning, Operating Systems Hardening, Application Security, Security Maintenance, Linux/Unix Security, Windows Security, Virtualization Security, Trusted Computing and Multilevel Security-The Bell-LaPadula Model for Computer Security, Other Formal Models for Computer Security,

  The Concept of Trusted Systems, Application of Multilevel Security, Trusted Computing and the Trusted Platform Module, Common Criteria for Information Technology Security Evaluation, Assurance and Evaluation.

Syllabus...

- **NETWORK SECURITY** **(10 Hours)**

  Internet Security Protocols and Standards-Secure E-mail and S/MIME, Pretty Good Privacy (PGP), Domain Keys Identified Mail, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), HTTPS, IPv4 and IPv6 Security, IPSec Protocol, Internet Authentication Applications-Kerberos, X.509, Public-Key Infrastructure, Federated Identity Management, Wireless Network Security-Wireless Security Overview, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security, Network Management Security-SNMP Protocol.

- **ADVANCED TOPICS** **(02 Hours)**

(Total Contact Time = 42 Hours)

Teaching Scheme

	L	T	P	Credit
Scheme	3	0	0	03

Books Recommended

1. William Stallings, Computer Security: Principles and Practice, 2/E, Pearson, 2012.
2. John Vacca, Network and System Security, 2/E, Elsevier, 2013.
3. William Stallings, Network Security Essentials: Applications and Standards, Prentice Hall, 4th edition, 2010.
4. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 2001.
5. William Stallings, Cryptography and Network Security, 7/E, Pearson, 2018.

Number Theory

- The Euclidean Algorithm
- Modular Arithmetic
- Prime Numbers
- Extended Euclidean Algorithm

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, such that k | a and k | 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.

Euclidean Algorithm..

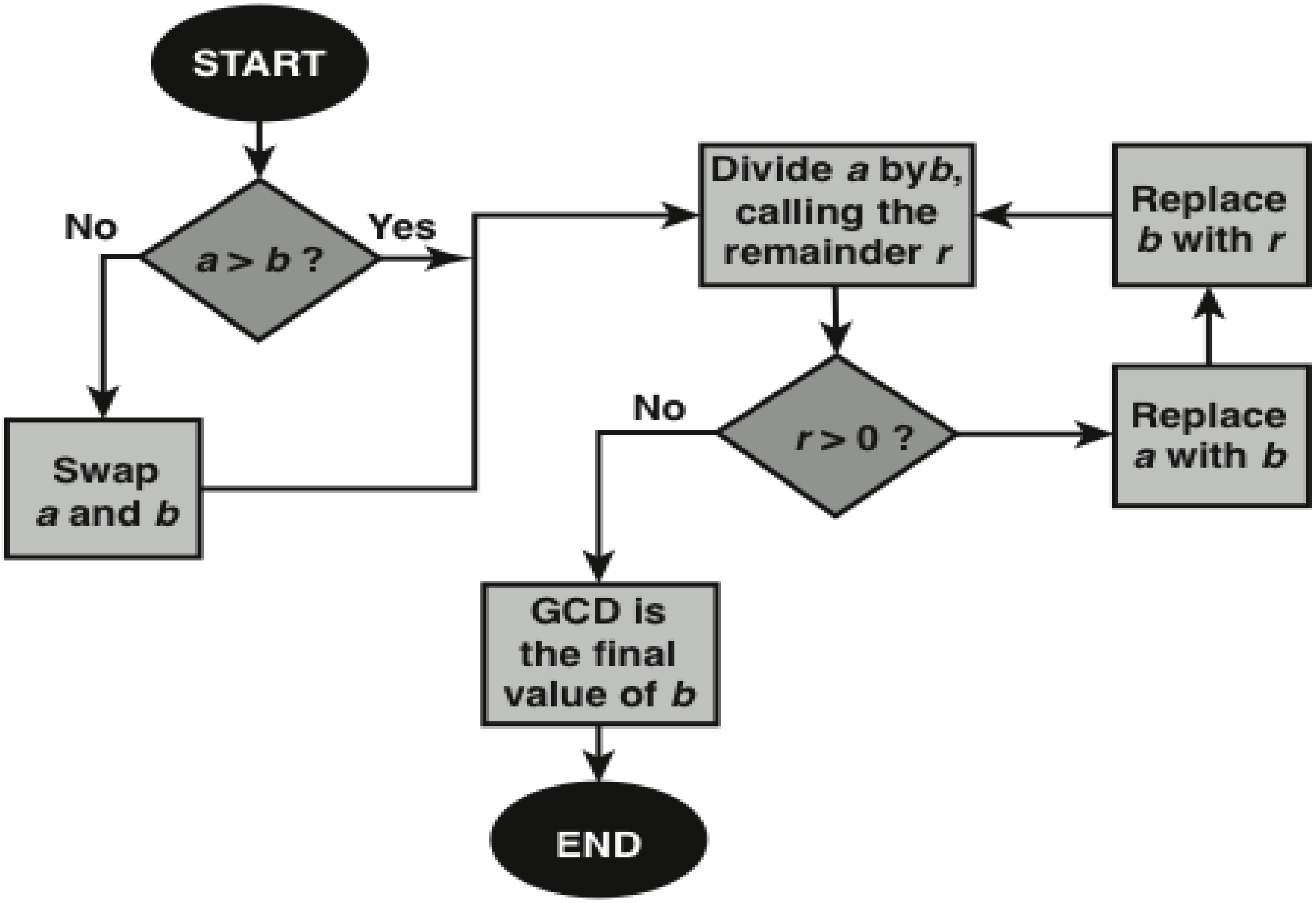


Figure 2.2 Euclidean Algorithm

Euclidean Algorithm…

- an efficient way to find the GCD(a,b)
- uses theorem that:
  - GCD(a,b) = GCD(b, a mod b)
- Euclidean Algorithm to compute GCD(a,b) is:
  - EUCLID(a,b)
  - 1. A = a; B = b
  - 2. if B = 0 return A = gcd(a, b)
  - 3. R = A mod B
  - 4. A = B
  - 5. B = R
  - 6. goto 2

Example GCD(1970,1066)

1970 = 1 x 1066 + 904	gcd(1066, 904)
1066 = 1 x 904 + 162	gcd(904, 162)
904 = 5 x 162 + 94	gcd(162, 94)
162 = 1 x 94 + 68	gcd(94, 68)
94 = 1 x 68 + 26	gcd(68, 26)
68 = 2 x 26 + 16	gcd(26, 16)
26 = 1 x 16 + 10	gcd(16, 10)
16 = 1 x 10 + 6	gcd(10, 6)
10 = 1 x 6 + 4	gcd(6, 4)
6 = 1 x 4 + 2	gcd(4, 2)
4 = 2 x 2 + 0	gcd(2, 0)

Euclidean Algorithm..

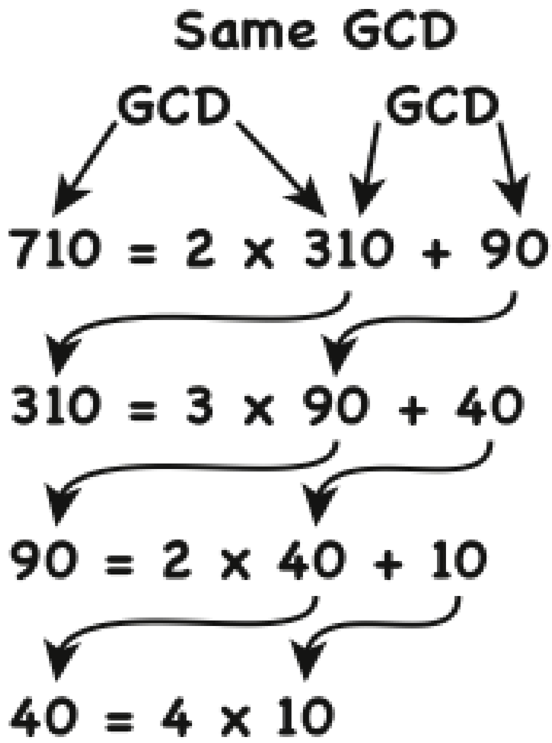


Figure 2.3 Euclidean Algorithm Example: gcd(710, 310)

Euclidean Algorithm Example

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

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:*

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

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

$$11 \bmod 7 = 4; \ -11 \bmod 7 = 3$$

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 $a = b$(mod n)[2]
  - Note that if $a = 0$(mod n), then $n \mid a$

73 = 4 (mod 23); 21 = - 9 (mod 10)

Properties of Congruences

- Congruences have the following properties:

  1. $a = b$ (mod n) if n $(a – b)$
  2. $a = b$ (mod n) implies $b = a$ (mod n)
  3. $a = b$ (mod n) and $b = c$ (mod n) imply $a = c$ (mod n)

- To demonstrate the first point, if n $(a - b)$, 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 (mod 5) because 23 - 8 = 15 = 5 * 3
- 11 = 5 (mod 8) because - 11 - 5 = - 16 = 8 * (- 2)
81 = 0 (mod 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:

$$
\begin{aligned}
(a + b) \bmod n &= (ra + jn + rb + kn) \bmod n \\
&= (ra + rb + (k + j)n) \bmod n \\
&= (ra + rb) \bmod n \\
&= [(a \bmod n) + (b \bmod n)] \bmod n
\end{aligned}
$$

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

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

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

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)

Finding Inverses – Extended Euclidean algorithm

```
EXTENDED_EUCLID(m, b)
1. (A1, A2, A3)=(1, 0, m);
   (B1, B2, B3)=(0, 1, b)
2. if B3 = 0
   return A3 = gcd(m, b); no inverse
3. if B3 = 1
   return B3 = gcd(m, b); B2 = b^-1 mod m
4. Q = A3 div B3
5. (T1, T2, T3)=(A1 – Q B1, A2 – Q B2, A3
   – Q B3)
6. (A1, A2, A3)=(B1, B2, B3)
7. (B1, B2, B3)=(T1, T2, T3)
8. goto 2
```

Inverse of 17 in GF(29)

i.e. calling Extended_Euclid(29, 17)

Q	A1	A2	A3	B1	B2	B3
—	1	0	29	0	1	17
1	0	1	17	1	–1	12
1	1	–1	12	–1	2	5
2	–1	2	5	3	–5	2
2	3	–5	2	-7	12	1

Inverse of 17 in GF(29)

i.e. calling Extended_Euclid(29, 17)

Q	A1	A2	A3	B1	B2	B3
—	1	0	29	0	1	17
1	0	1	17	1	–1	12
1	1	–1	12	–1	2	5
2	–1	2	5	3	–5	2
2	3	–5	2	-7	12	1

Inverse of 37 in GF(49)

i.e. calling Extended_Euclid(49, 37)

Q	A1	A2	A3	B1	B2	B3
—	1	0	49	0	1	37
1	0	1	37	1	-1	12
3	0	1	12	-3	4	1

- Hence $37^{-1} \equiv 4 \bmod 49$ OR $4 = 37^{-1} \bmod 49$

Inverse of 550 in GF(1759)

i.e. calling Extended_Euclid(1759, 550)

Q	A1	A2	A3	B1	B2	B3
—	1	0	1759	0	1	550
3	0	1	550	1	–3	109
5	1	–3	109	–5	16	5
21	–5	16	5	106	–339	4
1	106	–339	4	–111	355	1

Inverse of 49 in GF(37)

i.e. calling Extended_Euclid(37, 49)

Q	A1	A2	A3	B1	B2	B3
—	1	0	37	0	1	49
0	0	1	49	1	0	37
1	1	0	37	-1	1	12
3	-1	1	12	4	-3	1

- Hence $49^{-1} \equiv (-3) \bmod 37$
- But, $-3 \pmod{37} \equiv 34 \pmod{37}$. Hence,
- $34 = 37^{-1} \bmod 49$

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

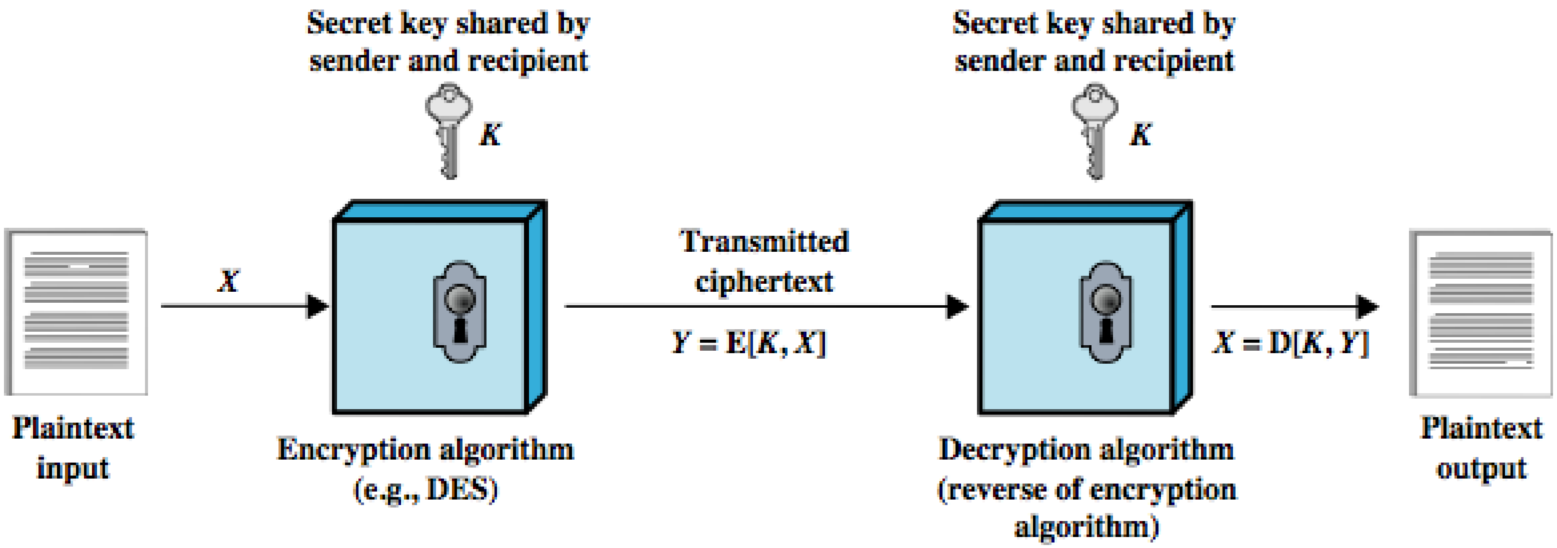
Primes Under 2000

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																			

Cryptographic Tools

- cryptographic algorithms are important element in security services
- review various types of elements
  - symmetric encryption
  - public-key (asymmetric) encryption
  - digital signatures and key management
  - secure hash functions
- example is use to encrypt stored data

Symmetric Encryption



Symmetric Encryption...

A symmetric encryption scheme has five ingredients :

- **Plaintext**: This is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm**: The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secretkey**:Thesecretkeyisalsoinputtotheencryptionalgorithm.Theexact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext**: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm**: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Attacking Symmetric Encryption

- cryptanalysis
  - rely on nature of the algorithm
  - plus some knowledge of plaintext characteristics
  - even some sample plaintext-ciphertext pairs
  - exploits characteristics of algorithm to deduce specific plaintext or key
- brute-force attack
  - try all possible keys on some ciphertext until get an intelligible translation into plaintext

Symmetric Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

DES and Triple-DES

- Data Encryption Standard (DES) is the most widely used encryption scheme
  - uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block
  - concerns about algorithm & use of 56-bit key
- Triple-DES
  - repeats basic DES algorithm three times
  - using either two or three unique keys
  - much more secure but also much slower

Advanced Encryption Standard (AES)

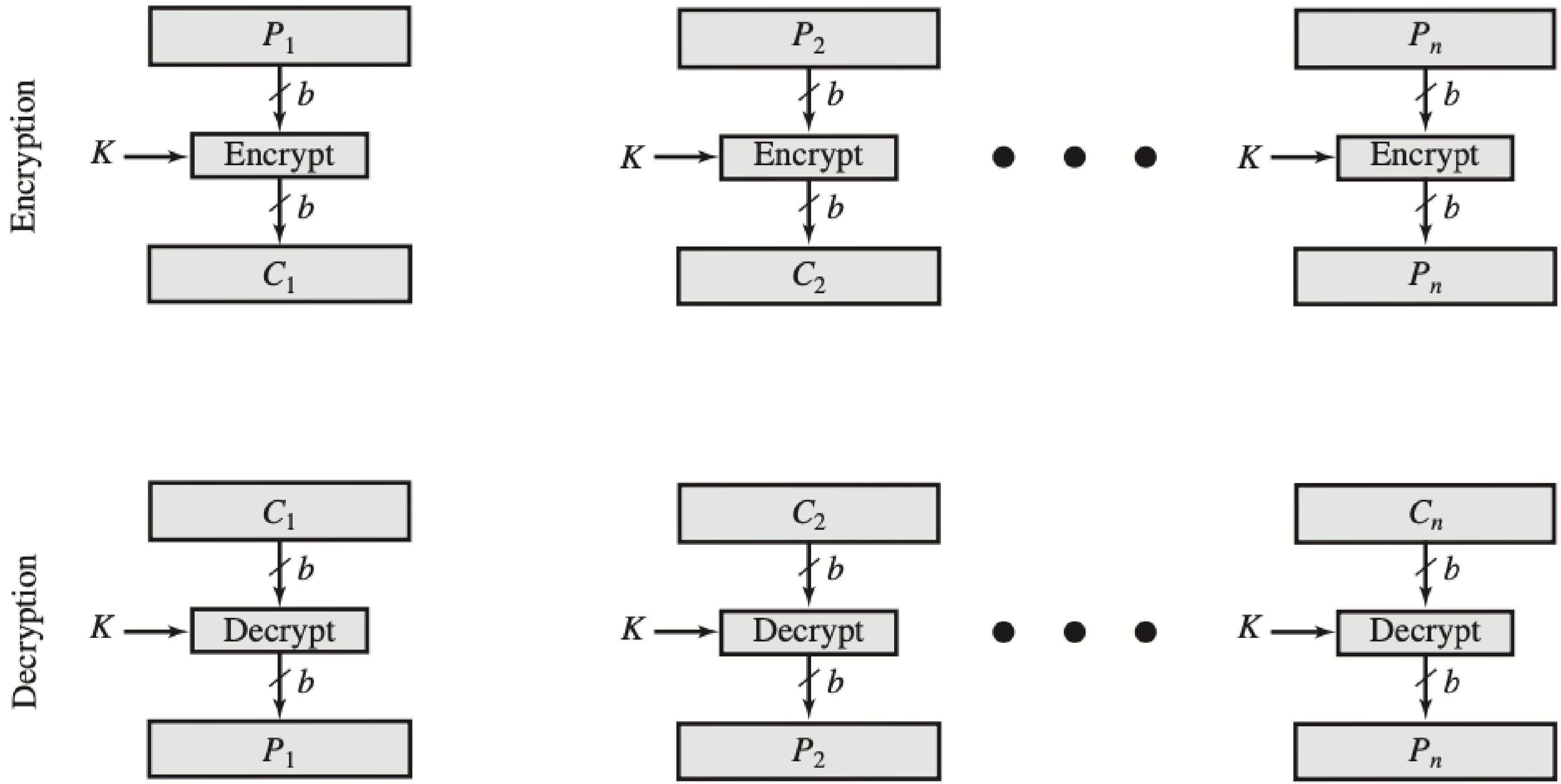
- Needed a better replacement for DES
- NIST called for proposals in 1997
    - efficiency, security, HW/SW suitability, 128, 256, 256 keys
- Selected Rijndael in Nov 2001
- symmetric block cipher
- uses 128 bit data & 128/192/256 bit keys
- now widely available commercially

Exhaustive Key Search

Table 2.2 Average Time Required for Exhaustive Key Search

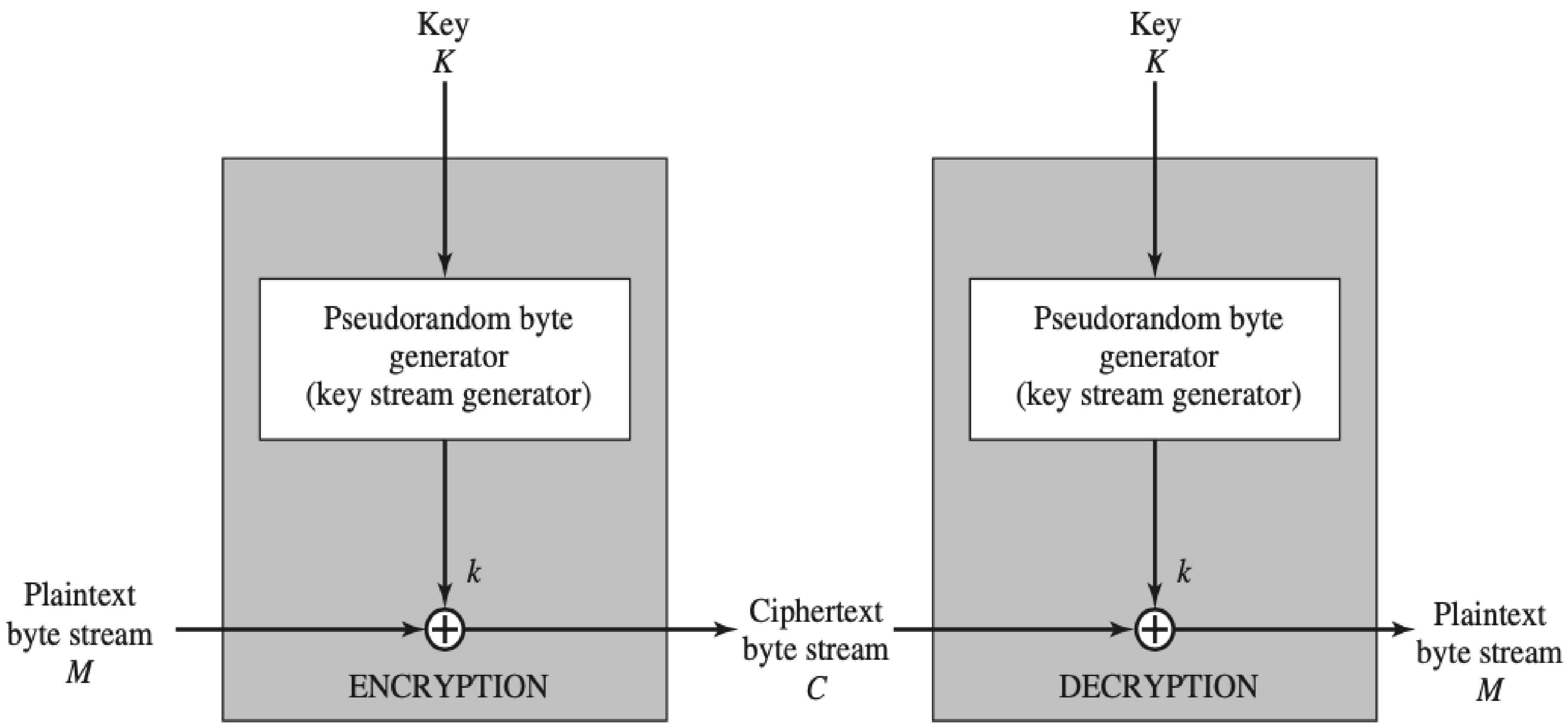
Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/μs	Time Required at 10^{13} decryptions/μs
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55}\,\mu s = 1.125$ years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127}\,\mu s = 5.3 \times 10^{21}$ years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167}\,\mu s = 5.8 \times 10^{33}$ years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191}\,\mu s = 9.8 \times 10^{40}$ years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255}\,\mu s = 1.8 \times 10^{60}$ years	1.8×10^{56} years

Block verses Stream Ciphers



(a) Block cipher encryption (electronic codebook mode)

Block verses Stream Ciphers…

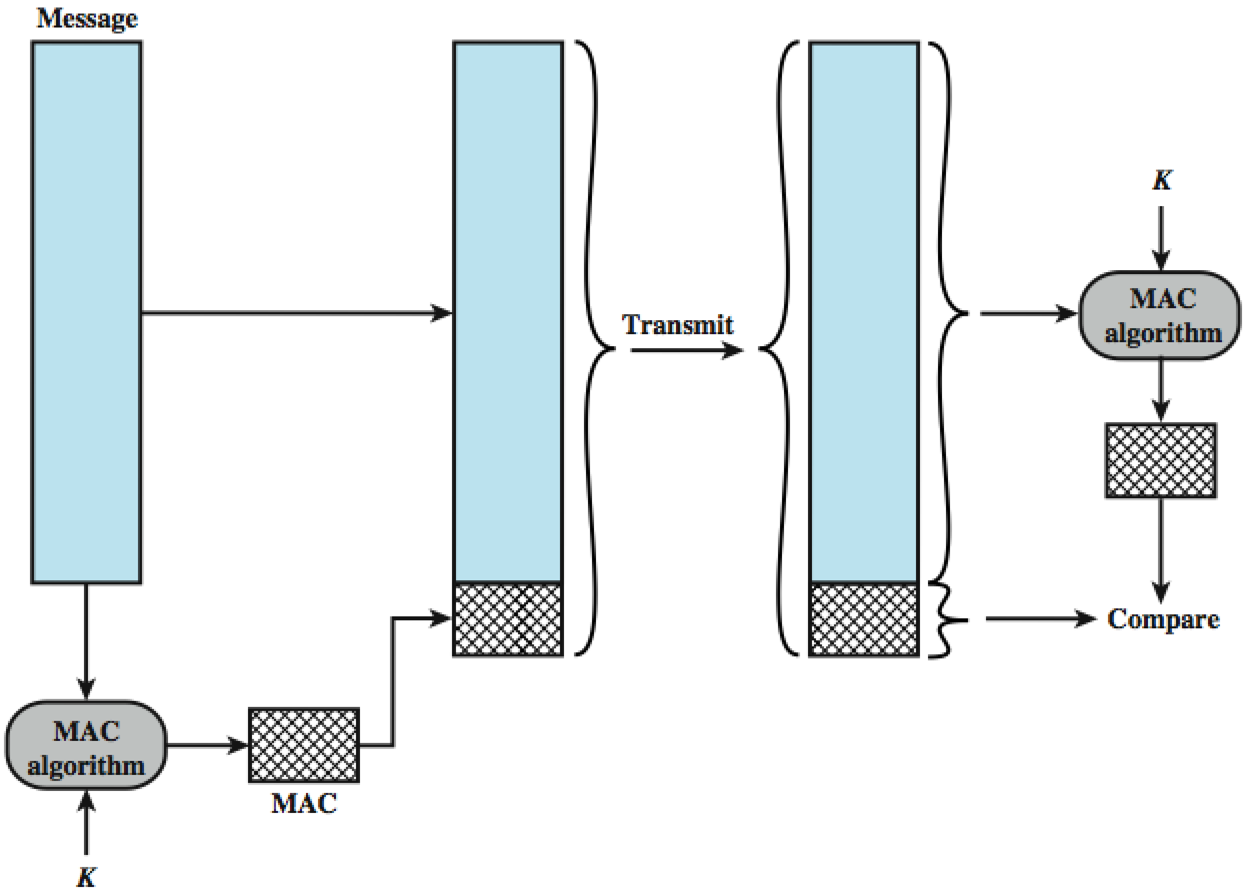


(b) Stream encryption

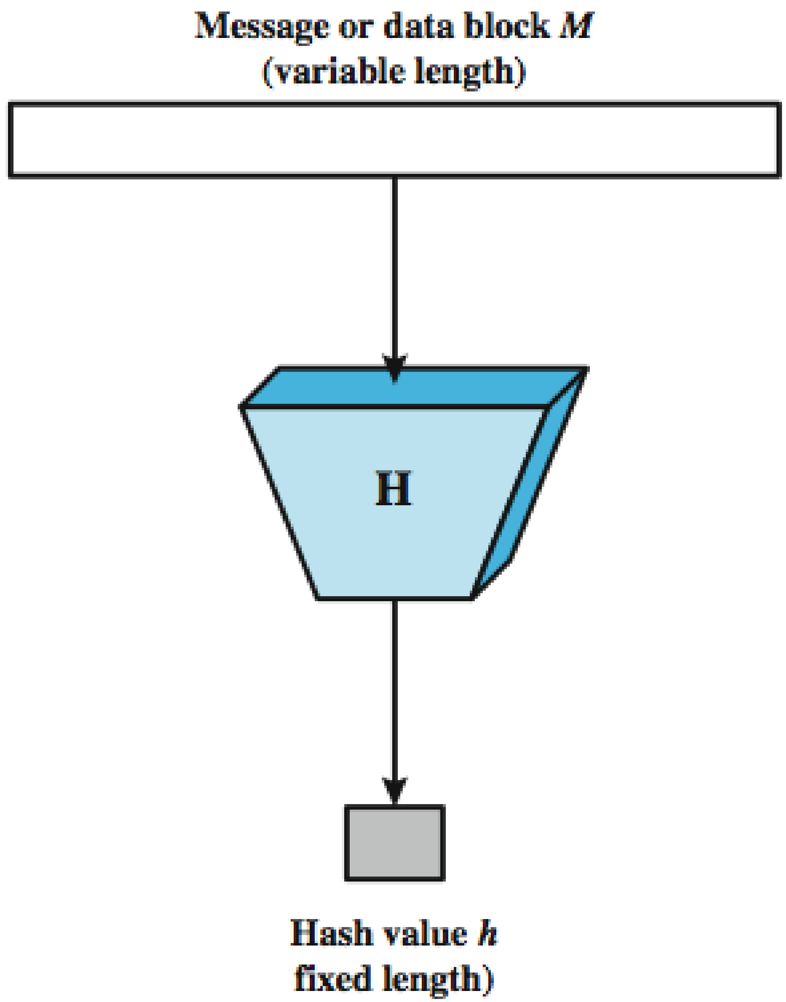
Message Authentication

- Encryption protects against passive attack (eavesdropping).
- A different requirement is to protect against active attack (falsification of data and transactions).
- Protection against such attacks is known as message or data authentication.
- verifies received message is authentic
  - contents unaltered
  - from authentic source
  - timely and in correct sequence
- can use conventional encryption
  - only sender & receiver have key needed
- or separate authentication mechanisms
  - append authentication tag to cleartext message

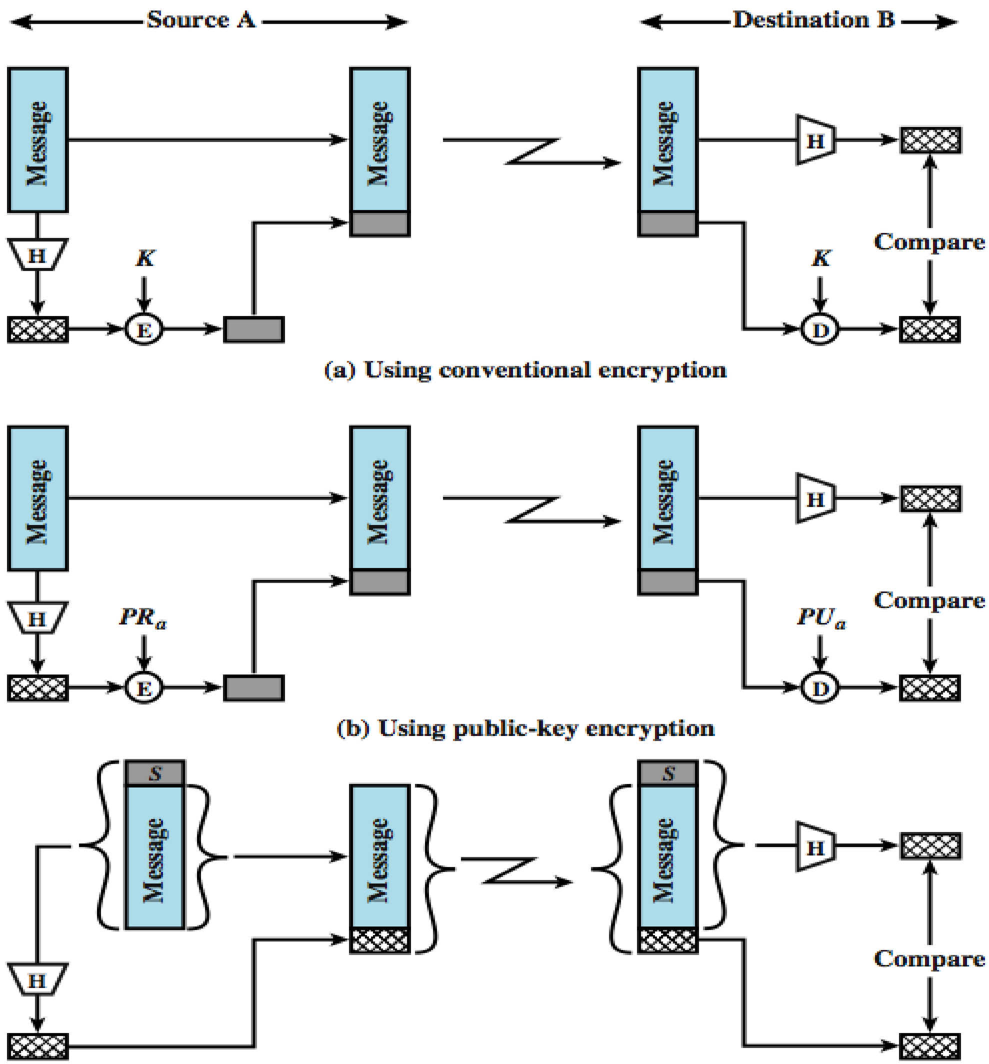
Message Authentication Codes



Secure Hash Functions



Message Authentication



(a) Using conventional encryption

(b) Using public-key encryption

(c) Using secret value

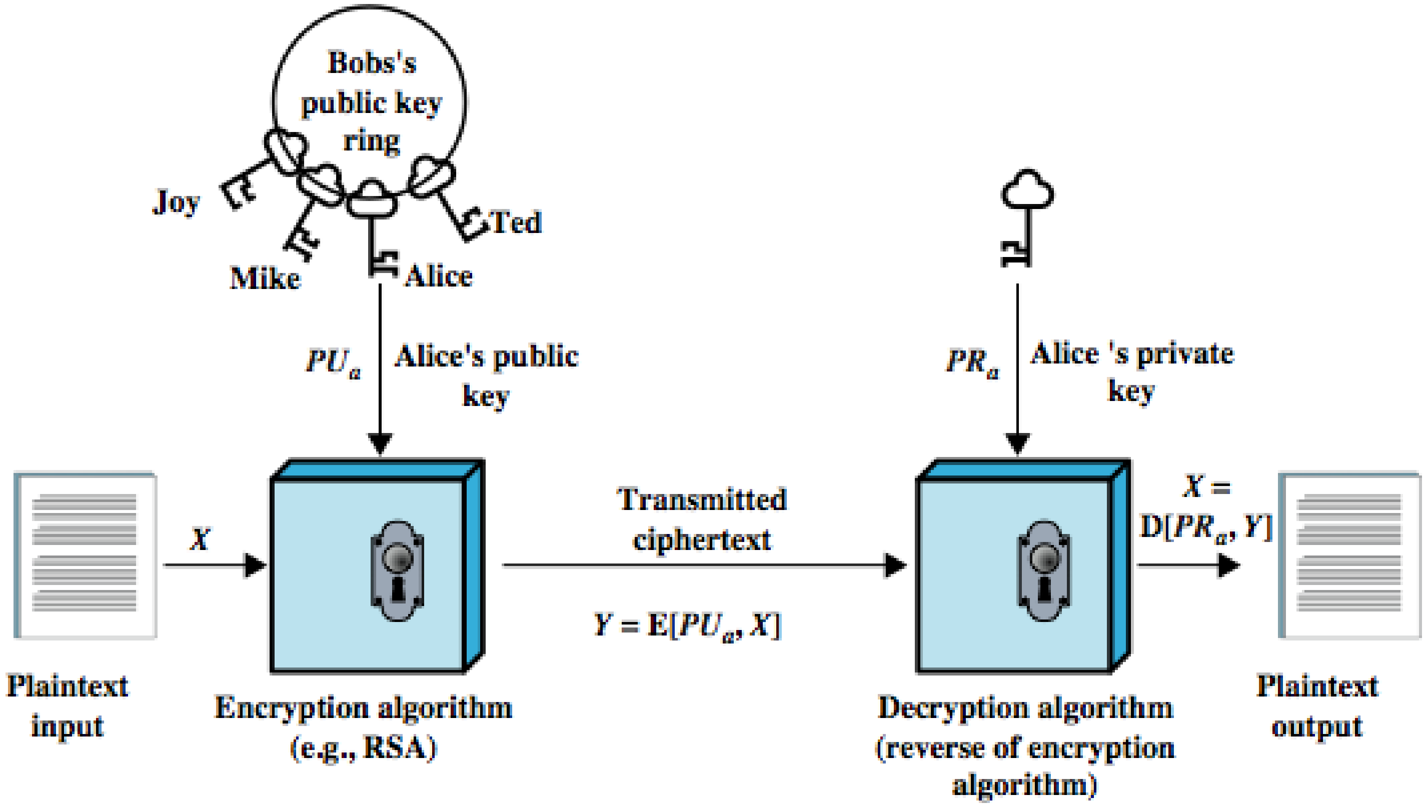
Hash Function Requirements

- applied to any size data
- H produces a fixed-length output.
- $H(x)$ is relatively easy to compute for any given x
- one-way property
  - computationally infeasible to find x such that $H(x) = h$
- weak collision resistance
  - computationally infeasible to find $y \neq x$ such tha $H(y) = H(x)$
- strong collision resistance
  - computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$

Hash Functions

- two attack approaches
  - cryptanalysis
    - exploit logical weakness in alg
  - brute-force attack
    - trial many inputs
    - strength proportional to size of hash code ($2^{n/2}$)
- SHA most widely used hash algorithm
  - SHA-1 gives 160-bit hash
  - more recent SHA-256, SHA-384, SHA-512 provide improved size and security

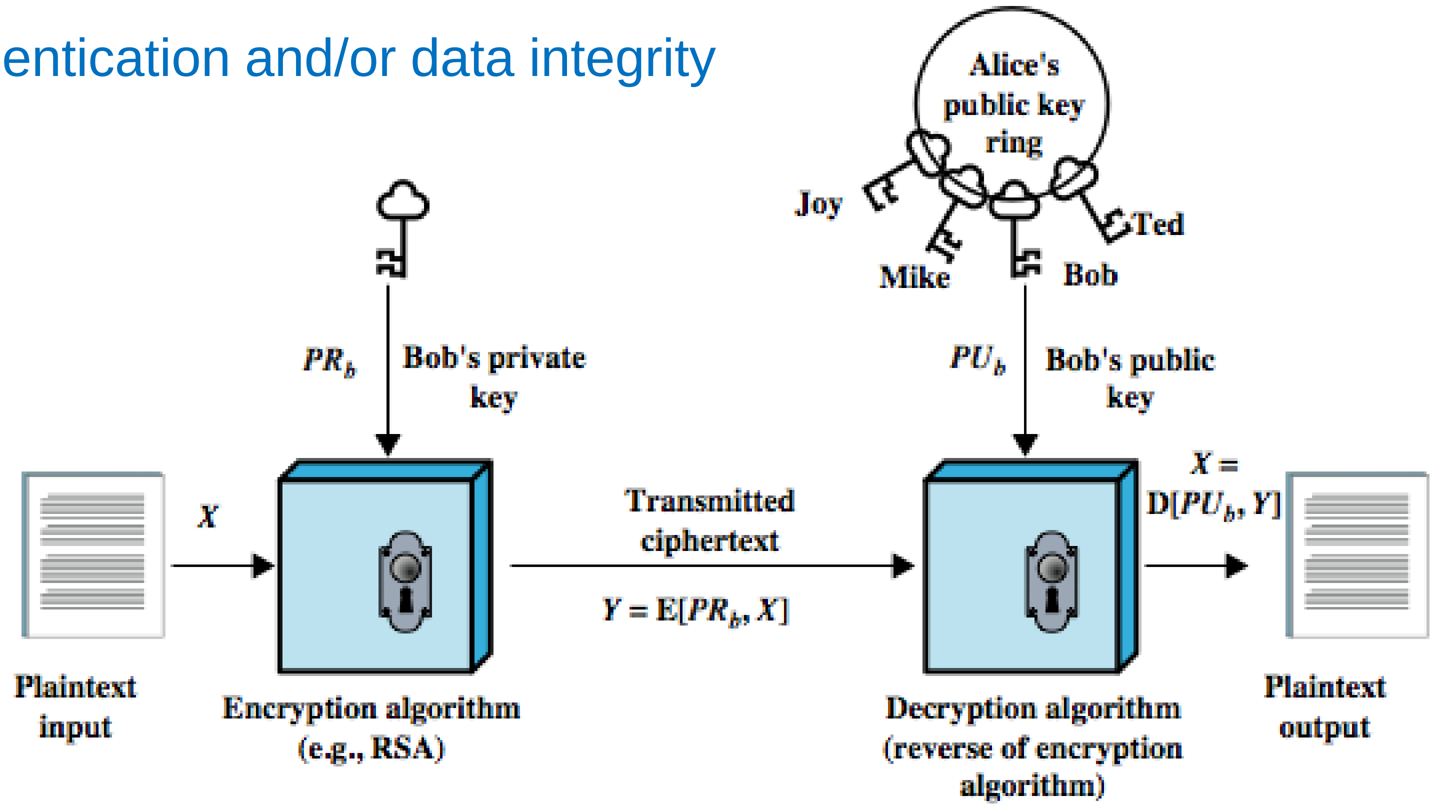
Public Key Encryption



(a) Confidentiality

Public Key Authentication

Authentication and/or data integrity



(b) Authentication

Public Key Requirements

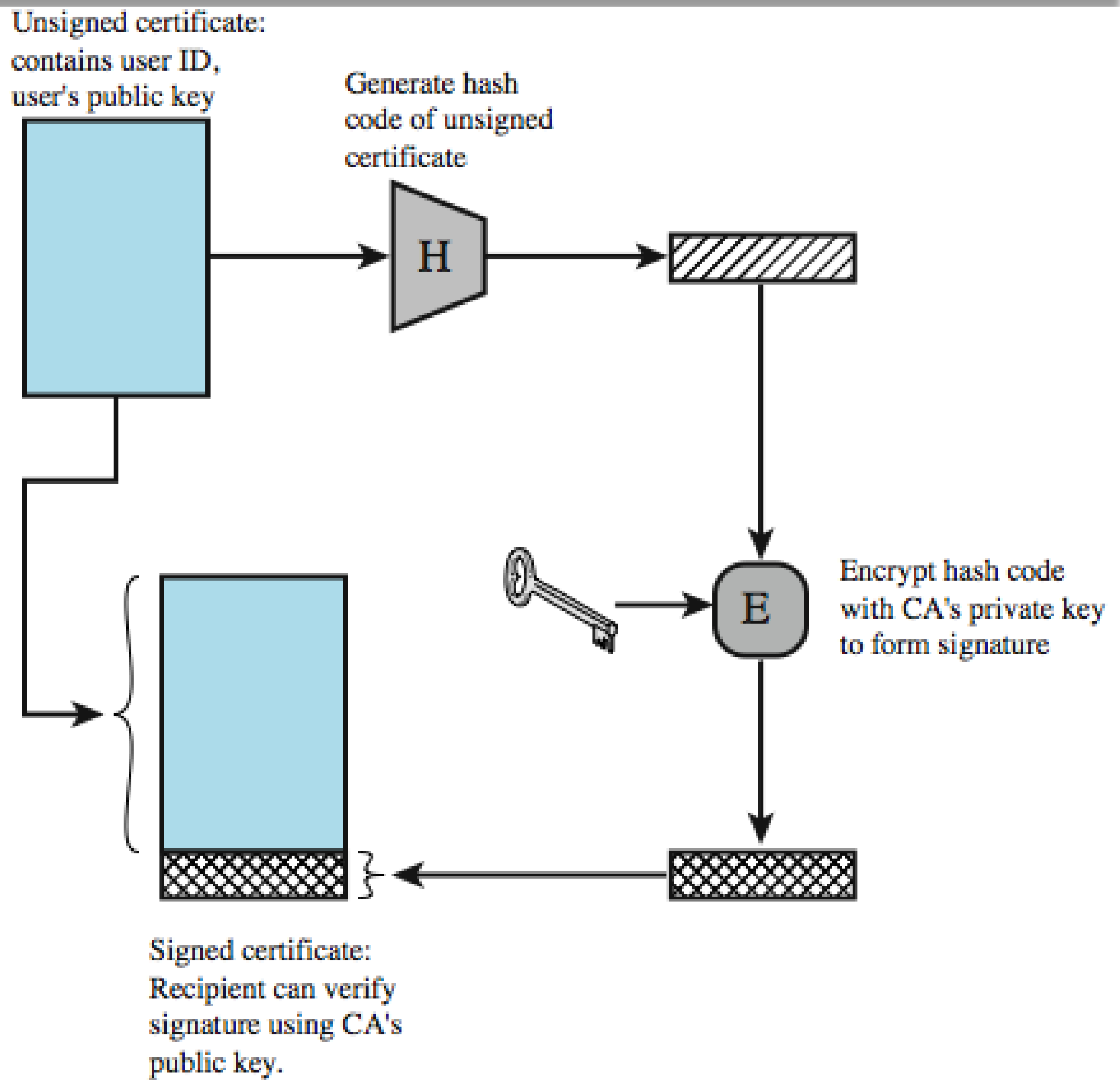
1. computationally easy to create key pairs
2. computationally easy for sender knowing public key to encrypt messages
3. computationally easy for receiver knowing private key to decrypt ciphertext
4. computationally infeasible for opponent to determine private key from public key
5. computationally infeasible for opponent to otherwise recover original message
6. useful if either key can be used for each role

Public Key Algorithms

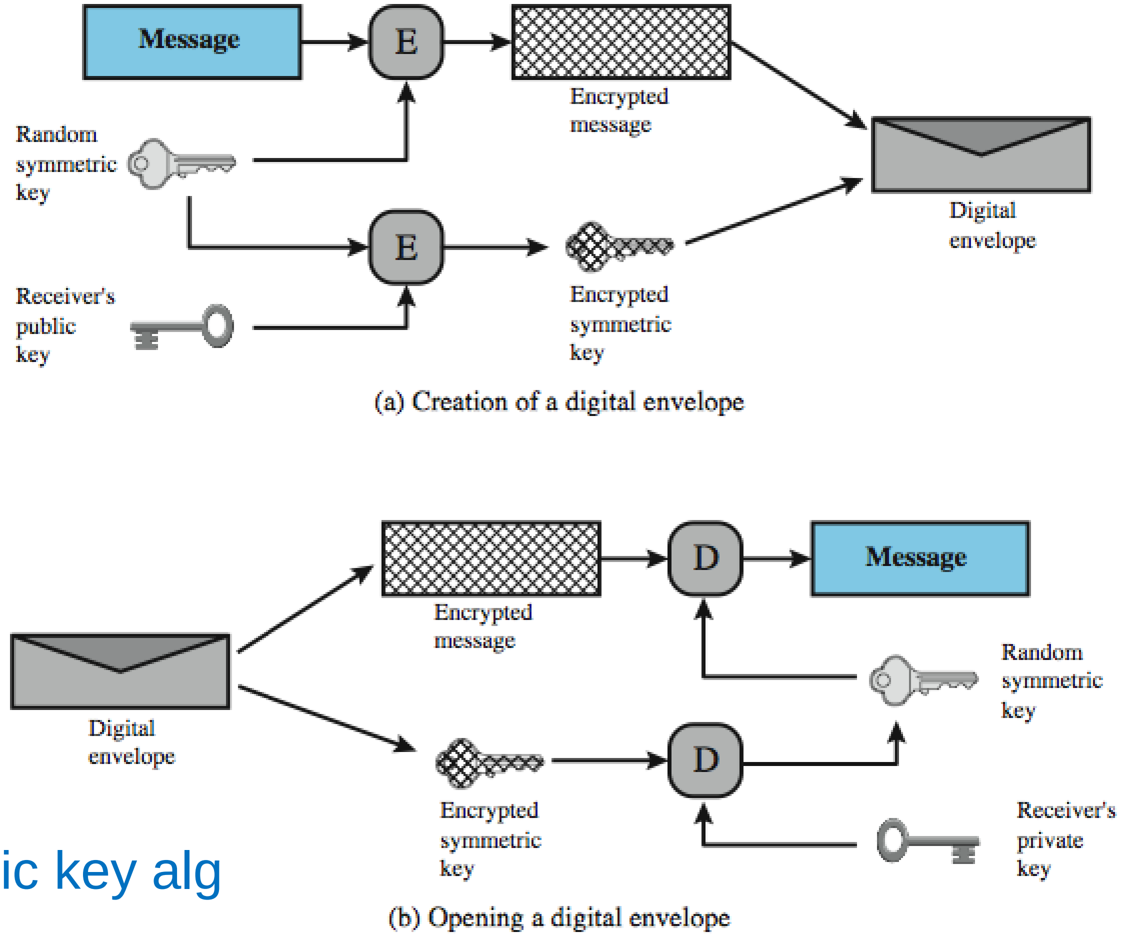
- RSA (Rivest, Shamir, Adleman)
  - developed in 1977
  - only widely accepted public-key encryption alg
  - given tech advances need 1024+ bit keys
- Diffie-Hellman key exchange algorithm
  - only allows exchange of a secret key
- Digital Signature Standard (DSS)
  - provides only a digital signature function with SHA-1
- Elliptic curve cryptography (ECC)
  - new, security like RSA, but with much smaller keys

Public Key Certificates

See textbook figure p.63



Digital Envelopes



Another application of public key alg

Random Numbers

- random numbers have a range of uses
- requirements:
- randomness
  - based on statistical tests for uniform distribution and independence
- unpredictability
  - successive values not related to previous
  - clearly true for truly random numbers
  - but more commonly use generator

Pseudorandom verses Random Numbers

- often use algorithmic technique to create pseudorandom numbers
  - which satisfy statistical randomness tests
  - but likely to be predictable
- true random number generators use a nondeterministic source
  - e.g. radiation, gas discharge, leaky capacitors
  - increasingly provided on modern processors

Practical Application: Encryption of Stored Data

- common to encrypt transmitted data
- much less common for stored data
  - which can be copied, backed up, recovered
- approaches to encrypt stored data:
  - back-end appliance ***(hardware device close to data storage; encrypt close to wire speed)***
  - library based tape encryption ***(co-processor board embedded in tape drive)***
  - background laptop/PC data encryption

Thank You !!!