TY B.Tech Trimester-VIII (AY 2021-2022) Computer Science and Engineering

Disclaimer:

a. Information included in these slides came from multiple sources. We have tried our best to cite the sources. Please refer to the references to learn about the sources, when applicable.

b. The slides should be used only for preparing notes, academic purposes (e.g. in teaching a class), and should not be used for commercial purposes.

Unit II: Mathematical Foundations and Public Key Cryptography

Unit: II	**Mathematical Foundations and Public Key Cryptography**: Mathematics for Security: Modular Arithmetic, Euclidean Algorithm, Chinese Remainder Theorem, Discrete Logarithm, Fermat Theorem, Secret Splitting and Sharing with polynomials, Asymmetric key Cryptography: RSA. Hash algorithms: SHA1, Digital Signatures: Symmetric Key Signatures, Public Key Signatures.	8 Hrs

Laboratory: Lab Assignment

Assign No.	Name of Assignment
B	**API Level - (Using Libraries) (Any two)**
1	To program asymmetric key cryptography such as RSA cryptography using JAVA API, Python or C++ API.
2	To program basic cryptography hash algorithm SHA1 or MD5 Use Java or Python or C++ API. Additionally demonstrate client server authentication using socket programming.
3	Write program for demonstration of digital signature and its verification using Java or Python or C++.

Number Theory

- ❖ **Prime Numbers**
- ❖ **Relative Prime Numbers**

  - Two numbers are called relatively prime if the **greatest common divisor (GCD)** of those numbers is **1**.
  - 8 and 15 are relatively prime number.
  - The factors of 8 are 1, 2, 4, 8 and the factors of 15 are 1, 3, 5, 15.
  - Examples of relatively prime numbers are: (10, 21), (14, 15), (45, 91), ...
  - for more info https://edurev.in/studytube/Prime-Numbers-PPT-PowerPoint-Presentation---Mathem/719baed3-6ca0-42f7-a262-6a882cb16cb6_p

- The greatest common divisor (GCD) of two numbers can be determined by comparing their prime factors and selecting the least powers of the factor.
- For example, the two numbers are 81 and 99.

$$81 = 1 * 9 * 9 = 1 * 3 * 3 * 3 * 3 = 1 * 3^4$$

$$99 = 1 * 3 * 33 = 1 * 3 * 3 * 11 = 1 * 3^2 * 11$$

The GCD is the least power of a number in the factors,

So, $\quad GCD(81, 99) = 1 * 3^2 * 11^0 = 9$

Modular Arithmetic

- *m mod n*
- The mod with respect to *n is (0, 1, 2, ... n - 1).*
- Suppose m = 23 and *n = 9, then*
- 23 mod 9 = 5
- For any value of m, the value of m mod 9 is from (0, 1, 2, ... 8).

1. Addition of modular number

- The addition of two numbers p and q with same modular base n is: (p mod n + q mod n) mod n = (p + q) mod n

2. Subtraction of modular number

- The subtraction of two numbers p and q with same modular base n is: (p mod n - q mod n) mod n = (p - q) mod n

3. Multiplication of modular number

- The multiplication of two numbers p and q with same modular base n is: (p mod n * q mod n) mod n = (p * q) mod n

e.g. p = 11, q = 15

[(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) x (15 mod 8)] mod 8 = 21 mod 8 = 5, (11 x 15) mod 8 = 165 mod 8 = 5

Example 1: Find the value of 7^7 mod 9.

$$7^7 \bmod 9 = (7^2)^3 * 7 \bmod 9$$
$$= (7^2 \bmod 9)^3 \bmod 9 * 7 \bmod 9$$
$$7^2 \bmod 9 = 49 \bmod 9 = 4$$
$$7^6 \bmod 9 = (7^2)^3 \bmod 9 = 4^3 \bmod 9 = 64 \bmod 9 = 1$$
$$7^7 = 7^6 * 7 \bmod 9 = 1 * 7 \bmod 9 = 7$$

Note: *m ^a mod n = m^pq mod n*

*where a = p * q =* $(m^p \bmod n)^q \bmod n$

Example 2: Find the value of 5^117 mod 19.

Answer: 5^117 mod 19 = (5 * 17 * 16 * 9 * 5) mod 19

= 61200 mod 19

= 1

$$117 = (2^0 + 2^2 + 2^4 + 2^5 + 2^6)$$
$$117 = 1 + 4 + 16 + 32 + 64$$
$$5^{117} \bmod 19 = 5^{(1 + 4 + 16 + 32 + 64)} \bmod 19$$
$$5^{117} \bmod 19 = (5^1 * 5^4 * 5^{16} * 5^{32} * 5^{64}) \bmod 19$$

$5^1 \bmod 19 = 5$

$5^2 \bmod 19 = (5^1 * 5^1) \bmod 19 = (5^1 \bmod 19 * 5^1 \bmod 19) \bmod 19$

$5^2 \bmod 19 = (5 * 5) \bmod 19 = 25 \bmod 19$

$5^2 \bmod 19 = 6$

$5^4 \bmod 19 = (5^2 * 5^2) \bmod 19 = (5^2 \bmod 19 * 5^2 \bmod 19) \bmod 19$

$5^4 \bmod 19 = (6 * 6) \bmod 19 = 36 \bmod 19$

$5^4 \bmod 19 = 17$

$5^8 \bmod 19 = (5^4 * 5^4) \bmod 19 = (5^4 \bmod 19 * 5^4 \bmod 19) \bmod 19$

$5^8 \bmod 19 = (17 * 17) \bmod 19 = 289 \bmod 19$

$5^8 \bmod 19 = 4$

$5^{16} \bmod 19 = (5^8 * 5^8) \bmod 19 = (5^8 \bmod 19 * 5^8 \bmod 19) \bmod 19$

$5^{16} \bmod 19 = (4 * 4) \bmod 19 = 16 \bmod 19$

$5^{16} \bmod 19 = 16$

$5^{32} \bmod 19 = (5^{16} * 5^{16}) \bmod 19 = (5^{16} \bmod 19 * 5^{16} \bmod 19) \bmod 19$
$5^{32} \bmod 19 = (16 * 16) \bmod 19 = 256 \bmod 19$
$5^{32} \bmod 19 = 9$

$5^{64} \bmod 19 = (5^{32} * 5^{32}) \bmod 19 = (5^{32} \bmod 19 * 5^{32} \bmod 19) \bmod 19$
$5^{64} \bmod 19 = (9 * 9) \bmod 19 = 81 \bmod 19$
$5^{64} \bmod 19 = 5$

$5^{117} \bmod 19 = (5^1 * 5^4 * 5^{16} * 5^{32} * 5^{64}) \bmod 19$

$5^{117} \bmod 19 = (5^1 \bmod 19 * 5^4 \bmod 19 * 5^{16} \bmod 19 * 5^{32} \bmod 19 * 5^{64} \bmod 19) \bmod 19$

$5^{117} \bmod 19 = (5 * 17 * 16 * 9 * 5) \bmod 19$

$5^{117} \bmod 19 = 61200 \bmod 19 = 1$

$5^{117} \bmod 19 = 1$

Fermat's Little Theorem

- If p is prime and a is an integer not divisible by p, then . . .

  $a^p = a \bmod p.$

  $\mathbf{a^{p-1} = 1 \bmod p}$

- **Hence,** $\mathbf{a^{p-1} \bmod p = 1}$ where, p is prime and GCD (a, p) = 1

- E.g. $8^{12} \bmod 13 = 1 \bmod 13 = 1$
- $8^{103} \bmod 103 = 8 \bmod 103 = 8$
- This theorem is useful in public key (RSA) and primality testing.

Example 3: Suppose a = 7 and p = 19 then prove Fermat's Little theorem

Example 4: Compute the value of **12345^23456789 mod 101** using Fermat's theorem

Solution By Fermat's Little theorem $n^{p-1} = 1 \bmod p$ where $n = 12345$ and $p = 101$.

$$12345^{(101-1)} \bmod 101 = 1$$
$$12345^{100} \bmod 101 = 1$$

Therefore, $12345^{23456789} \bmod 101 = (12345^{100})^{234567} * 12345^{89} \bmod 101$

$$= 1 * 12345^{89} \bmod 101$$
$$= 12345^{89} \bmod 101$$

But $\quad 12345 \bmod 101 = 23$

Therefore, $23^{89} \bmod 101$

$$23 \bmod 101 = 23$$
$$23^2 \bmod 101 = 24$$

Therefore, $23^{89} \bmod 101$

$$23 \bmod 101 = 23$$
$$23^2 \bmod 101 = 24$$

$$23^3 \bmod 101 = 47$$
$$23^4 \bmod 101 = 71$$
$$23^5 \bmod 101 = 17$$
$$23^7 \bmod 101 = 4$$
$$23^{89} \bmod 101 = (23^7)^{12}\ 23^5 \bmod 101$$
$$= 4^{12} * 17 \bmod 101$$
$$= 5 * 17 \bmod 101$$
$$= 85$$

Therefore, the value of $12345^{23456789} \bmod 101 = 85$.

Fermat's little theorem and its congruence

- Suppose a positive integer be p and two integers x and y are congruent mod p.

Mathematically, $x \equiv y \bmod p$ if $p \mid (x-y)$

For example:

i) $5 \equiv 2 \bmod 3$

ii) $23 \equiv -1 \bmod 12$

Euler Totient Function ø(n)

https://www.doc.ic.ac.uk/~mrh/330tutor/ch05s02.html

- ø (n) = how many numbers there are between **1 and *n* - 1** that are **relatively prime to *n*.**
- ø (4) = 2 (1, 3 are relatively prime to 4)
- ø (5) = 4 (1, 2, 3, 4 are relatively prime to 5)
- ø (6) = 2 (1, 5 are relatively prime to 6)
- ø (7) = 6 (1, 2, 3, 4, 5, 6 are relatively prime to 7)

For prime p, ø(p) = p -1 e.g. ø(37) = 36

Two prime p, q with p ≠ q, ø(n) = ø(p.q) = (p -1) x (q - 1) e.g. ø(21) = (3–1) x (7–1) = 2 x 6 = 12

Where 12 integers are [1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20]

- This theorem generalizes Fermat's theorem and is an important key to the RSA algorithm.

- Euler's theorem, for every a and p that are relatively prime:

  $$a^{\Phi(p)} = 1 \pmod{p} \quad \text{i.e.} \quad a^{\Phi(p)} \bmod p = 1$$

- In other words, If a and p are relatively prime, with a being the smaller integer, then when we multiply a with itself (p) times and divide the result by p, the remainder will be 1.

Euclidean Algorithm

- Suppose *p and q are two numbers.*
- *GCD (p, q) is the largest number that divides evenly both p and q.*
- *Euclidean algorithm* is used **to compute** the greatest common divisor (**GCD**) of two integer numbers.
- Euclid theorem: **GCD(p, *q*) = *GCD(q, p mod q)***

Example:

1. Compute GCD (997, 366) using Euclid's algorithm
2. Compute GCD (2222, 1234) using Euclid's algorithm.

1. **Compute GCD (997, 366) using Euclid's algorithm**

2.

997 = 2 * 366 + 265

366 = 1 * 265 + 101

265 = 2 * 101 + 63

101 = 1 * 63 + 38

63 = 1 * 38 + 25

38 = 1 * 25 + 13

25 = 1 * 13 + 12

13 = 1 * 12 + 1

12 = 12 * 1 + 0

GCD (997, 366) = 1

Note: Every time divide the divisor by remainder

2. Compute GCD (2222, 1234) using Euclid's algorithm

- 2222 = 1 * 1234 + 988

1234 = 1 * 998 + 246

998 = 4 * 246 + 4

246 = 61 * 4 + 2

4 = 2 * 2 + 0 GCD (2222, 1234) = 2

Extended Euclidean Algorithm

- Suppose *p and q are two integer numbers. There exist two integers x and y such that $xp + yq = GCD(p, q)$.*

Extended Euclidean algorithm is used to find the value of *x and y.*

- Write the two linear combinations vertically as shown below and apply Euclid's algorithm to get $g = GCD(p, q)$ *and the values of x and the y to satisfy the equation*
  - $xp + yq = g.$
  - $x = 1.x + 0.y$
  - $y = 0.x+1.y$
  - $r = 1.x + (-z) .y$

- Find integers *p and q such that 51p + 36q = 3. Also find the GCD* (51, 36)

51 = 36(1) + 15	15 = 51 – 36(1)
36 = 15(2) + 6	6 = 36 – 15(2)
15 = 6(2) + 3(GCD)	3 = 15 – 6(2)
6 = 3(2) + 0	

- 3 = 15 - 6(2)
- 3 = 15 - [36 - 15(2)](2)
- 3 = 15(5) - 36(2)
- 3 = [51 - 36(1)](5) - 36(2)
- 3 = 51(5) - 36(5) - 36(2)
- 3 = 51(5) - 36(7)
- 3 = 51(5) + 36(-7)
- Therefore, the values of *p = 5 and q = -7 and GCD = 3.*

Chinese Remainder Theorem (CRT)

- used to speed up modulo computations if working modulo a product of numbers
  - eg. mod $M = m_1 m_2 .. m_k$
- Chinese Remainder theorem work in each moduli m_i separately
- since computational cost is proportional to size, this is faster than working in the full modulus M

- can implement CRT in several ways
- to compute A (mod M)
  - first compute all $a_i = A \bmod m_i$ separately
  - determine constants c_i below, where $M_i = M/m_i$
  - then combine results to get answer using:

$$c_i = M_i \times \left(M_i^{-1} \bmod m_i\right) \quad \text{for } 1 \le i \le k$$

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i\right) (\bmod M)$$

Chinese Remainder Theorem: If $m_1, m_2, .., m_k$ are pairwise relatively prime positive integers, and if $a_1, a_2, .., a_k$ are any integers, then the simultaneous congruences,

$x \equiv a_1 \pmod{m_1}, x \equiv a_2 \pmod{m_2}, ..., x \equiv a_k \pmod{m_k}$ have a solution, and the solution is unique modulo m, where $m = m_1 m_{2,} \cdots_{..} m_k$

Example: Solve the simultaneous congruences

$$x \equiv 6 \pmod{11}, x \equiv 13 \pmod{16}, x \equiv 9 \pmod{21}, x \equiv 19 \pmod{25}.$$

Ans: 89469

We will construct a solution x.

First, let $M_k = m/m_k$ for $k = 1, 2, \ldots, n$ and $m = m_1 m_{2,} \cdots_{..} m_k$.

Since $\gcd(m_k, M_k) = 1$, the number M_k has a multiplicative inverse y_k modulo m_k.

i.e., $M_k y_k \equiv 1(\bmod m_k)$

Now we let $x = a_1M_1y_1 + a_2M_2y_2 + \cdot \cdot \cdot + a_nM_ny_n$

Why does this x satisfy all the congruences?

If $j \neq k$ then $M_j \equiv 0(\bmod m_k)$, since M_j contains m_k as a factor.

Thus $x \bmod m_k = 0 + a_kM_ky_k \bmod m_k$

$$= (a_k \bmod m_k)(M_k y_k \bmod m_k)$$

$$= (a_k \bmod m_k) \cdot 1$$

$$= a_k \bmod m_k$$

Solution: Since 11, 16, 21, and 25 are pairwise relatively prime, the Chinese Remainder Theorem tells us that there is a unique solution modulo m, where $m = 11 \cdot 16 \cdot 21 \cdot 25 = 92400$.

We apply the technique of the Chinese Remainder Theorem with

$$k = 4, \quad m_1 = 11, \quad m_2 = 16, \quad m_3 = 21, \quad m_4 = 25,$$
$$a_1 = 6, \quad a_2 = 13, \quad a_3 = 9, \quad a_4 = 19,$$

to obtain the solution.

We compute

$$z_1 = m / m_1 = m_2 m_3 m_4 = 16 \cdot 21 \cdot 25 = 8400$$
$$z_2 = m / m_2 = m_1 m_3 m_4 = 11 \cdot 21 \cdot 25 = 5775$$
$$z_3 = m / m_3 = m_1 m_2 m_4 = 11 \cdot 16 \cdot 25 = 4400$$
$$z_4 = m / m_4 = m_1 m_3 m_3 = 11 \cdot 16 \cdot 21 = 3696$$

$$y_1 \equiv z_1^{-1} \pmod{m_1} \equiv 8400^{-1} \pmod{11} \equiv 7^{-1} \pmod{11} \equiv 8 \pmod{11}$$
$$y_2 \equiv z_2^{-1} \pmod{m_2} \equiv 5775^{-1} \pmod{16} \equiv 15^{-1} \pmod{16} \equiv 15 \pmod{16}$$
$$y_3 \equiv z_3^{-1} \pmod{m_3} \equiv 4400^{-1} \pmod{21} \equiv 11^{-1} \pmod{21} \equiv 2 \pmod{21}$$
$$y_4 \equiv z_4^{-1} \pmod{m_4} \equiv 3696^{-1} \pmod{25} \equiv 21^{-1} \pmod{25} \equiv 6 \pmod{25}$$

$$w_1 \equiv y_1 z_1 \pmod{m} \equiv 8 \cdot 8400 \pmod{92400} \equiv 67200 \pmod{92400}$$
$$w_2 \equiv y_2 z_2 \pmod{m} \equiv 15 \cdot 5775 \pmod{92400} \equiv 86625 \pmod{92400}$$
$$w_3 \equiv y_3 z_3 \pmod{m} \equiv 2 \cdot 4400 \pmod{92400} \equiv 8800 \pmod{92400}$$
$$w_4 \equiv y_4 z_4 \pmod{m} \equiv 6 \cdot 3696 \pmod{92400} \equiv 22176 \pmod{92400}$$

The solution, which is unique modulo 92400, is

$$\begin{aligned} x &\equiv a_1 w_1 + a_2 w_2 + a_3 w_3 + a_4 w_4 \pmod{92400} \\ &\equiv 6 \cdot 67200 + 13 \cdot 86625 + 9 \cdot 8800 + 19 \cdot 22176 \pmod{92400} \\ &\equiv 2029869 \pmod{92400} \\ &\equiv \mathbf{51669} \pmod{92400} \end{aligned}$$

Correct Ans:
89469

Chinese Remainder Theorem :

It will determine a no. that will divided by some given divisor leaves given remainder

$x \equiv a_1 \pmod{m_1}$
$x \equiv a_2 \pmod{m_2}$ $\Rightarrow$
$x \equiv a_3 \pmod{m_3}$

$x \equiv 1 \pmod{5}$
$x \equiv 1 \pmod{7}$
$x \equiv 3 \pmod{11}$

$a_1 = 1$ $\quad$ $m_1 = 5$

$a_2 = 1$ $\quad$ $m_2 = 7$

$a_3 = 3$ $\quad$ $m_3 = 11$

$$M_i = \frac{M}{m_i}$$ (Calculation of M_1, M_2, M_3)

$$\therefore M_1 = \frac{M}{m_1} = \frac{385}{5} = 77$$

$$M_2 = \frac{M}{m_2} = \frac{385}{7} = 55$$

$$M_3 = \frac{M}{m_3} = \frac{385}{11} = 35$$

ASSIGNMENT / TEST Page No.

Calculation of x_1, x_2, x_3 $\longrightarrow$ $M_i X_i \equiv 1 \pmod{m_i}$

$M_1 X_1 \equiv 1 \pmod{m_1}$

$77 x_1 \equiv 1 \pmod 5$

$2 x_1 \equiv 1 \pmod 5$

$3\,(2x_1 \equiv 1 \pmod 5)$

$6 x_1 \equiv 3 \pmod 5$

$1 x_1 = 3 \pmod 5$

$\therefore \boxed{x_1 = 3}$

$(77 x_1 \bmod 5) = 1$

$(77/5$ - remainder $= 2$

5 6 ← 2×3

10 11

15 16

20 21

⋮

$M_2 X_2 \equiv 1 \pmod{m_2}$

$55 x_2 \equiv 1 \pmod 7$

$6 x_2 \equiv 1 \pmod 7$

$6\,(6 x_2 \equiv 1 \pmod 7)$

$36 x_2 \equiv 6 \pmod 7$

$1 x_2 \equiv 6 \pmod 7$

$\therefore \boxed{x_2 = 6}$

7 8

14 15

21 22

28 29

35 36

Example: Find the smallest multiple of 10 which has remainder 1 when divide by 3, remainder 6 when divided by 7 and remainder 6 when divided by 11.

Solution The factors of 10 are: 2 and 5.

Problem is now expressed as a system of congruence as:

$$p \equiv b_i (\text{mod } n_i)$$

where n = 2, 3, 5, 7 and 11 which are relatively prime and b = 0, 1, 0, 6 and 6 are the remainders for respective value of n.

$$p = 0 \text{ mod } 2$$
$$p = 1 \text{ mod } 3$$
$$p = 0 \text{ mod } 5$$
$$p = 6 \text{ mod } 7$$
$$p = 6 \text{ mod } 11$$

To solve for p we first calculate the value of N as:

$$N = n_1 * n_2 * \dots * n_r$$
$$N = 2 * 3 * 5 * 7 * 11 = 2310$$

and find the value of $N_i = N/n_i$ as:

$$N_2 = 2310/2 = 1155$$
$$N_3 = 2310/3 = 770$$
$$N_5 = 2310/5 = 462$$
$$N_7 = 2310/7 = 330$$
$$N_{11} = 2310/11 = 210$$

Now, find out the multiplicative inverse as:

$$y_i \equiv (N_i)^{-1} (\text{mod } n_i)$$
$$y_2 = (1155)^{-1} (\text{mod } 2) = 1$$
$$y_3 = (770)^{-1} (\text{mod } 3) = 2$$
$$y_5 = (462)^{-1} (\text{mod } 5) = 3$$
$$y_7 = (330)^{-1} (\text{mod } 7) = 1$$
$$y_{11} = (210)^{-1} (\text{mod } 11) = 1$$

The solution for the above problem is:

$$p \equiv b_1 N_1 y_1 + b_2 N_2 y_2 + \cdots + b_r N_r y_r \ (\text{mod } N),$$
$$p = 0(N_2 * y_2) + 2(N_3 * y_3) + 0(N_5 * y_5) + 6(N_7 * y_7) + 6(N_{11} * y_{11})$$
$$p = 0(1155)(1) + 1(770)(2) + 0(462)(3) + 6(330)(1) + 6(210)(1)$$
$$p = 0 + 1540 + 0 + 1980 + 1260$$
$$p = 4780 \text{ mod } 2310 = 160.$$

Discrete Logarithms

- The inverse problem to exponentiation is to find the **discrete logarithm** of a number modulo p that is to find i such that $\mathbf{b \equiv a^i \ (mod\ p)}$ where, $0 \le i \le (p-1)$
- This is written as $\mathbf{i = dlog_a\, b\ (mod\ p)}$
- if **a is a primitive root** then it always exists, otherwise it may not, eg.
- Ex: p=11, a=2, b=9, x=?

  $x = \log_3 4 \bmod 13$ has no answer

  $x = \log_2 3 \bmod 13 = 4$ by trying successive powers
- whilst exponentiation is relatively easy, finding discrete logarithms is generally a **hard** problem
- used in Diffe-Hellman and the digital signature algorithm.

(Ex) p=11, a=2, b=9, since $b^{(p-1)/2} \equiv 9^5 \equiv 1$,then check for even numbers {0,2,4,6,8,10} only to find x=6 such that $2^6 \equiv 9 \pmod{11}$

Power of integers, Modulo 13

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	$\|a\|_{13}$
1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	3	6	12	11	9	5	10	7	1	12
3	9	1	3	9	1	3	9	1	3	9	1	3
4	3	12	9	10	1	4	3	12	9	10	1	6
5	11	8	1	5	11	8	1	5	11	8	1	4
6	10	8	9	2	12	7	3	5	4	11	1	12
7	10	5	9	11	12	6	3	8	4	2	1	12
8	11	5	1	8	11	5	1	8	11	5	1	4
9	3	1	9	3	1	9	3	1	9	3	1	3
10	9	12	3	4	1	10	9	12	3	4	1	6
11	4	5	3	7	12	2	9	8	10	4	1	12
12	1	12	1	12	1	12	1	12	1	12	1	2

- ❖ Check 3 is primitive root of 5?
- ❖ Check 4 is primitive root of 5?

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

Discrete Logarithms mod 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

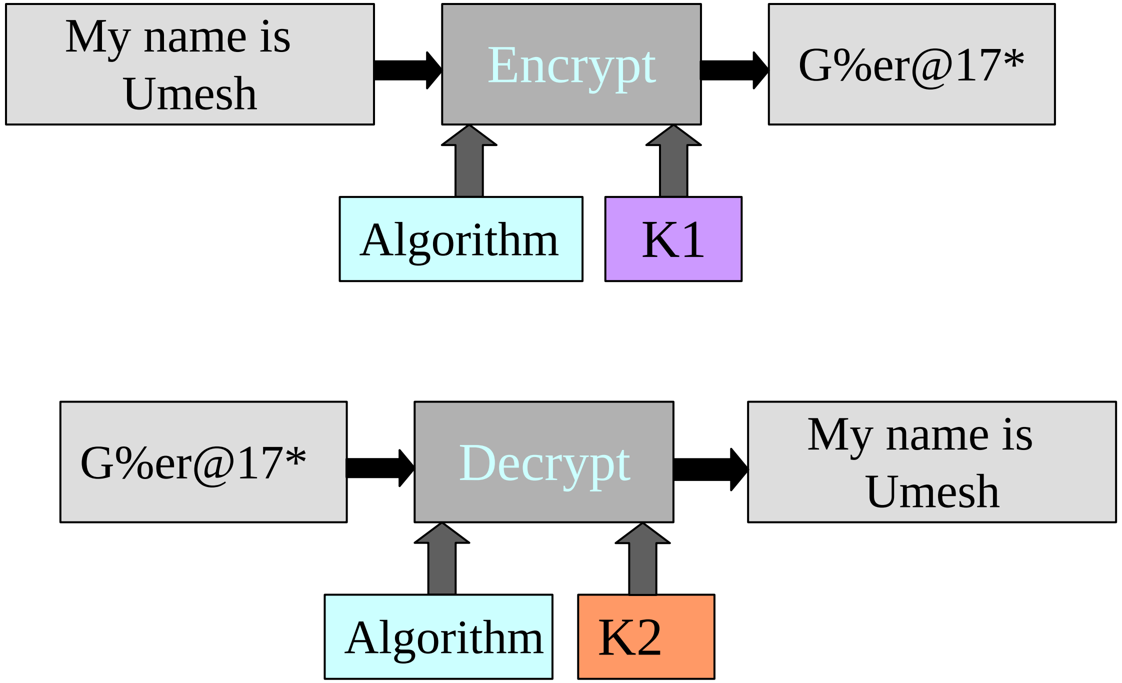
a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9

(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

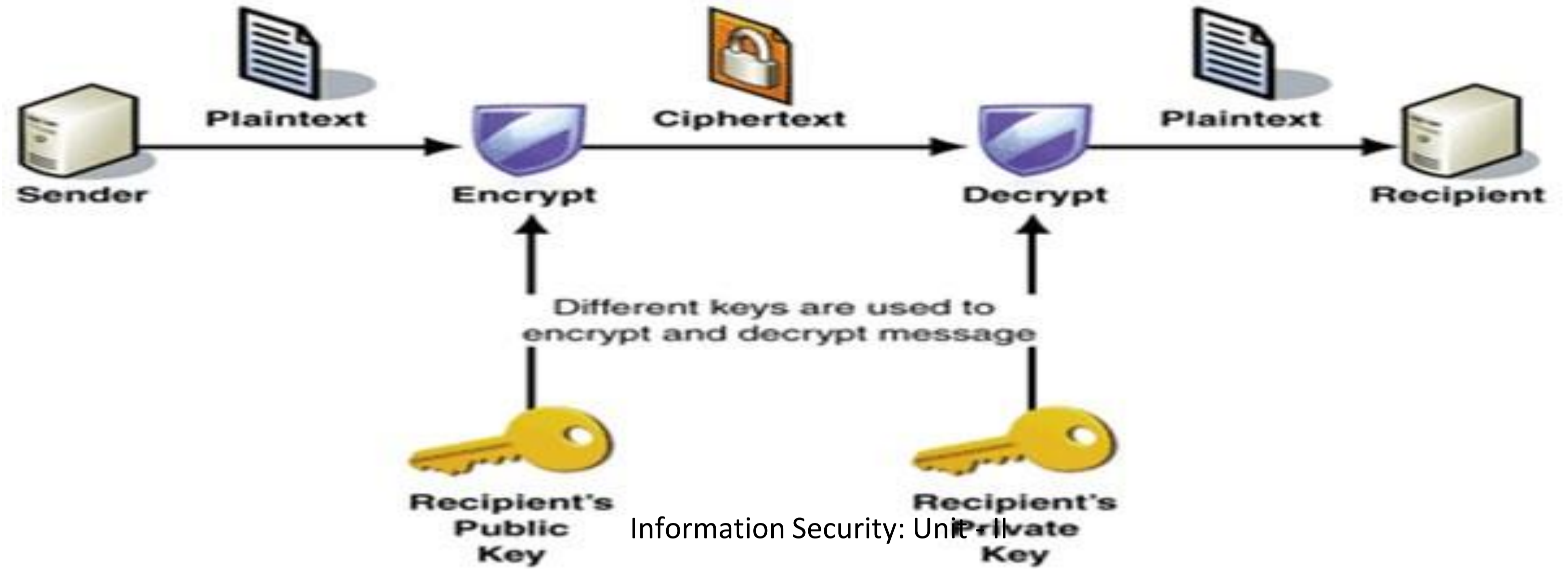
Public Key Cryptography

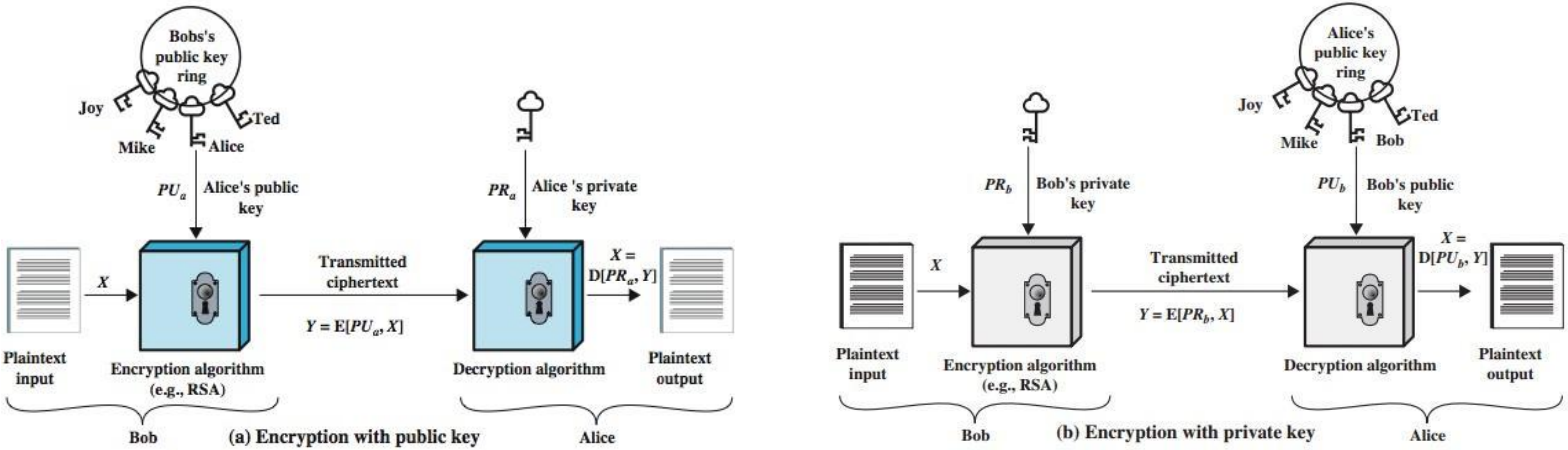
Asymmetric Key Encryption: Example



Matrix of Keys

Key details	*A* should know	*B* should know
A’s private key	Yes	No
A’s public key	Yes	Yes
B’s private key	No	Yes
B’s public key	Yes	Yes





(a) Encryption with public key

(b) Encryption with private key

Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)

- some algorithms are suitable for all uses, others are specific to one

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

RSA En/decryption

- by **R**on Rivest, Adi **S**hamir and Leonard **A**dleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
- uses large integers (eg. 1024 bits)

RSA Key Setup

- each user generates a public/private key pair by: selecting two large primes at random: **p, q**
- computing their system modulus **$n = (p * q)$**
- Compute: **$ø(n) = (p - 1)(q - 1)$**
- selecting at random the **encryption key** (public) e, where $1 < e < ø(n)$, $\gcd(e, ø(n)) = 1$
- solve following equation to find decryption key d

  $d * e = 1 \bmod ø(n)$ and $0 \leq d \leq n$

- publish their public encryption key: **$PU = \{e, n\}$**
- keep secret private decryption key: **$PR = \{d, n\}$**

- to encrypt a message M the sender:
  - obtains **public key** of recipient PU = {e, n}
  - computes Ciphertext : $\mathbf{C = M^e \bmod n}$, where $0 \le M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key PR = {d, n}
  - computes: $\mathbf{M = C^d \bmod n}$
- note that the message M must be smaller than the modulus n (block if needed)

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption

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

Decryption

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

Why RSA Works

- because of Euler's Theorem:
  - $a^{ø(n)} \bmod n = 1$ where $\gcd(a, n) = 1$
- in RSA have:
  - $n = p * q$
  - $ø(n) = (p-1)(q-1)$
  - carefully chose e & d to be inverses mod $ø(n)$
  - hence $e * d = 1 + k * ø(n)$ for some k
- hence :

$$C^d = (M^{e.})^d = M^{e.d} = M^{1+k.ø(n)} = M^1.(M^{ø(n)})^k$$

$$= M^1.(1)^k = M^1 = M \bmod n$$

RSA Example - Key Setup

1. Select primes: $p = 17$ & $q = 11$
2. Calculate $n = pq = 17 \times 11 = 187$
3. Calculate $ø(n) = (p - 1)(q - 1) = 16 \times 10 = 160$
4. Select e: gcd(e,160) = 1; choose $e = 7$
5. Determine d: $de = 1 \bmod 160$ and $d < 160$

   Value is d = 23 since 23 x 7 =161 = 10 x 160 + 1
6. Publish public key PU = {7,187}
7. Keep secret private key PR = {23,187}

RSA Example - En/Decryption

- sample RSA encryption/decryption is:

8. given message M = 88 (nb. 88 < 187)

9. encryption:

   $C = 88^7 \bmod 187 = 11$

10. decryption:

    $M = 11^{23} \bmod 187 = 88$

- Choosing the right keys is the real challenge.
- Knowing **e and n** an attacker could find the value of private key d by trial and error.
- Attacker needs to find out values of p and q using n as n = p x q.
- For small value it may be easy to find out p and q out of n.
- But in actual practice p and q are chosen as very large numbers.
- Therefore factoring n to get p and q is not at all easy. It is complex and time consuming.

Advantages of RSA

- Can be used for both encryption as well as for digital signature.
- Trapdoor in RSA is in knowing value of n but not knowing the primes of that are factors of n

Disadvantages of RSA

- If any one value of p, q, Φ(n) and e is known then the other values can be calculated.
- To protect the encryption, the minimum number of bits in n should be of 2048 bits.

Symmetric Encryption	Asymmetric Encryption
Symmetric encryption is fast in execution	Asymmetric Encryption is slow in execution due to the high computational burden
Symmetric encryption uses a single key for both encryption and decryption	Asymmetric encryption uses a different key for encryption and decryption
Size of resulting encrypted text usually same or less than original	Size of resulting encrypted text more than original
Problem of Key Exchange	No Problem of Key Exchange
Easier to implement	Practically more difficult
Exemple: DES, 3DES, AES, and RC4	Exemple: Diffie-Hellman, RSA.

Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Example 1: The parameters given are $p = 5$, $q = 17$. Find out the possible public keys and private key for RSA algorithm. Also encrypt the message "4".

Example 2: Using RSA algorithm to encrypt the message m = "6" use parameters $p = 3$, $q = 17$, $e = 7$, calculate decryption key.

Secret Splitting and Sharing with polynomials:

Shamir’s Secret Sharing Scheme(SSSS)

Problem: Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present.

What is the smallest number of locks needed?

What is the smallest number of keys to the locks each scientist must carry?

Minimal solution uses 462 locks and 252 keys per scientist.

Drawbacks:

- These numbers are clearly impractical
- Becomes exponentially worse when the number of scientists increases

❖Secret Sharing is an algorithm in cryptography created by Adi Shamir. It is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part.

❖ More particularly **Shamir Secret Sharing Scheme (SSSS)** enables to split a secret **S** in **n** parts such that with any **k**-out-of-**n** pieces you can reconstruct the original secret S.

❖ But with any **k-1** pieces no information is exposed about S.

❖That is conventionally called a (**n, k) threshold scheme.**

- Suppose using (k, n) threshold scheme to share our secret S. [n = 5, k = 3)
- Divide secret data (D) in to pieces (n)
- Choose at random k-1 coefficients a_1, a_2,.., $a_{(k-1)}$ and let $a_0 = S$.
- Build the polynomial.
  - $f(x) = a_0 + a_1 * x + a_2 *x^2 + ... + a_{(k-1)} * x^{(k-1)}$
- Construct the n pieces that are distributed to the participants.
  - $D_{x-1} = [x, f(x)]$
- Given any subset of k pairs, can find S using interpolation
- The secret is the constant term a_0.

.from the polynomial:

Example

- Suppose that our secret is 1234
- We wish to divide the secret into 6 parts. (n = 6)
- where any subset of 3 parts, (k = 3) is sufficient to reconstruct the secret.
- At random we obtain two (k - 1) numbers: 166 and 94.
- (a1 = 166; a2 = 94)
- Our polynomial to produce secret shares (points) is therefore:

$$f(x) = 1234 + 166x + 94x^2$$

- We construct 6 points $D_{x-1} = (x, f(x))$

$$D_0 = (1, 1494); D_1 = (2, 1942); D_2 = (3, 2578); D_3 = (4, 3402); D_4 = (5, 4414); D_5 = (6, 5614)$$

We give each participant a different single point (both x and $f(x)$). Because we use D_{x-1} instead of D_x the points start from $(1, f(1))$ and not $(0, f(0))$. This is necessary because if one would have $(0, f(0))$ he would also know the secret ($S = f(0)$)

Reconstruction

In order to reconstruct the secret any 3 points will be enough.

Let us consider $(x_0, y_0) = (2, 1942); (x_1, y_1) = (4, 3402); (x_2, y_2) = (5, 4414)$.

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)},$$

Reconstruction

In order to reconstruct the secret any 3 points will be enough.

Let us consider $(x_0, y_0) = (2, 1942); (x_1, y_1) = (4, 3402); (x_2, y_2) = (5, 4414)$.

We will compute Lagrange basis polynomials:

$$\ell_0 = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x - 4}{2 - 4} \cdot \frac{x - 5}{2 - 5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}$$

$$\ell_1 = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} = \frac{x - 2}{4 - 2} \cdot \frac{x - 5}{4 - 5} = -\frac{1}{2}x^2 + \frac{7}{2}x - 5$$

$$\ell_2 = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} = \frac{x - 2}{5 - 2} \cdot \frac{x - 4}{5 - 4} = \frac{1}{3}x^2 - 2x + \frac{8}{3}$$

Therefore

$$f(x) = \sum_{j=0}^{n-1} y_j l_j(x)$$

$$f(x) = \sum_{j=0}^{2} y_j \cdot \ell_j(x)$$

$$= y_0\ell_0 + y_1\ell_1 + y_2\ell_2$$

$$= 1942\left(\frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}\right) + 3402\left(-\frac{1}{2}x^2 + \frac{7}{2}x - 5\right) + 4414\left(\frac{1}{3}x^2 - 2x + \frac{8}{3}\right)$$

$$= 1234 + 166x + 94x^2$$

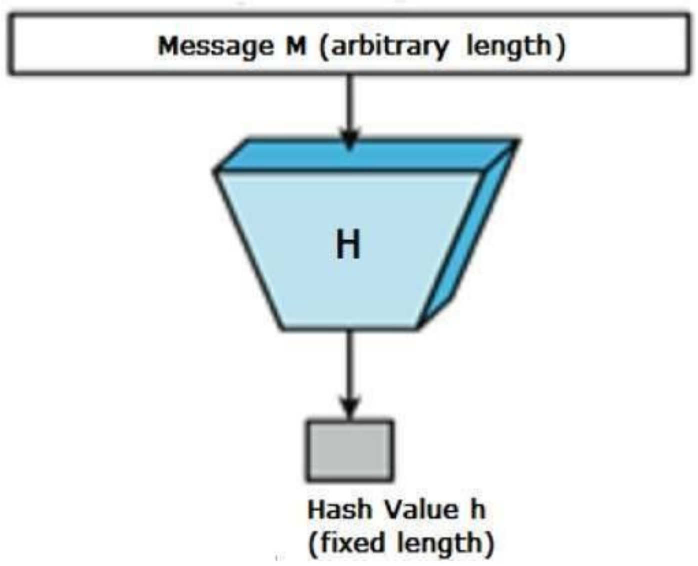
Recall that the secret is the free coefficient, which means that $S = 1234$, and we are done.

Useful properties of (k, n) threshold scheme:

- ❖ Secure.
- ❖ Minimal: The size of each piece does not exceed the size of the original data.
- ❖ Extensible: When k is kept fixed, D_i pieces can be dynamically added or deleted without affecting the other pieces.
- ❖ Dynamic: Security can be easily enhanced without changing the secret, but by changing the polynomial occasionally (keeping the same free term) and constructing new shares to the participants.
- ❖ Flexible: In organizations where hierarchy is important, we can supply each participant different number of pieces according to his importance inside the organization. For instance, the president can unlock the safe alone, whereas 3 secretaries are required together to unlock it.

Message Digest: MD 5 and SHA -1

- ❖ The digest is sometimes called the "hash" or "fingerprint" of the input.
- ❖ Hash value is used to check the integrity of the message
- ❖ MD5 processes a variable-length message into a fixed-length output of 128 bits.



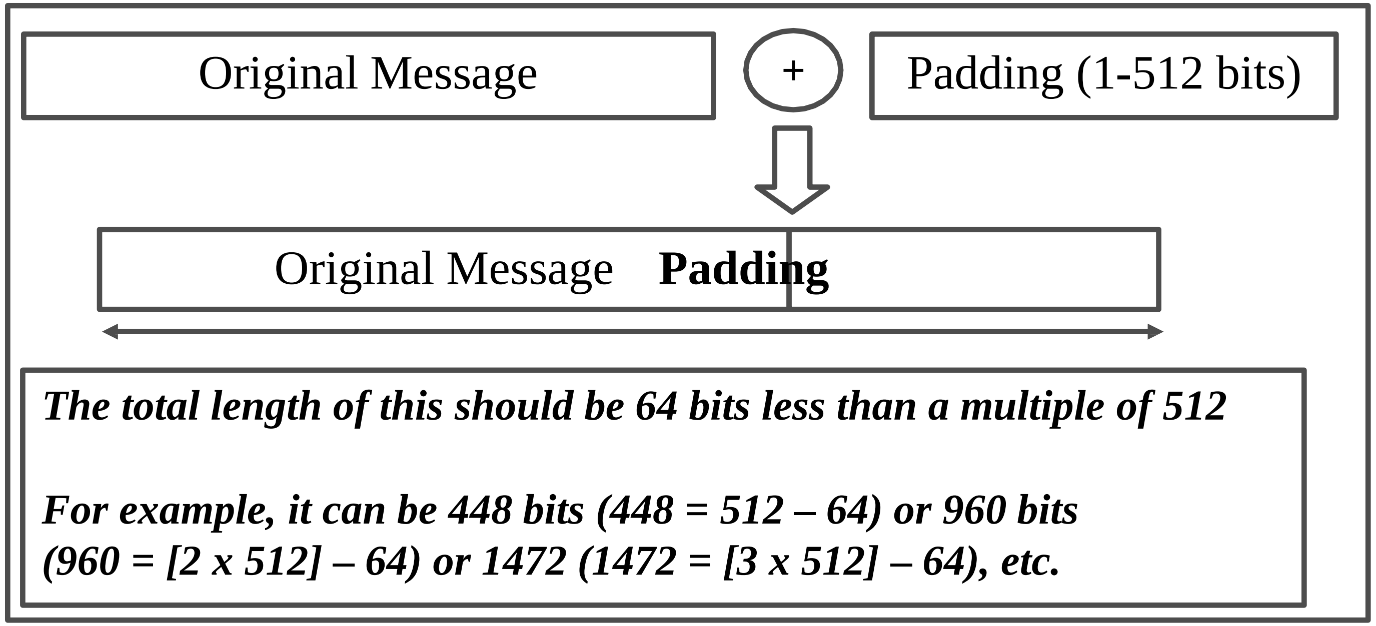
Algorithm:

- ❖ Step -1: Padding
- ❖ Step - 2: Append length
- ❖ Step - 3: Divide the input into 512-bit blocks.
- ❖ Step - 4: Initialize chaining variables (4 variables)
- ❖ Step - 5: Process blocks

http://www.herongyang.com/Cryptography/SHA1-Message-Digest-Algorithm-Overview.html

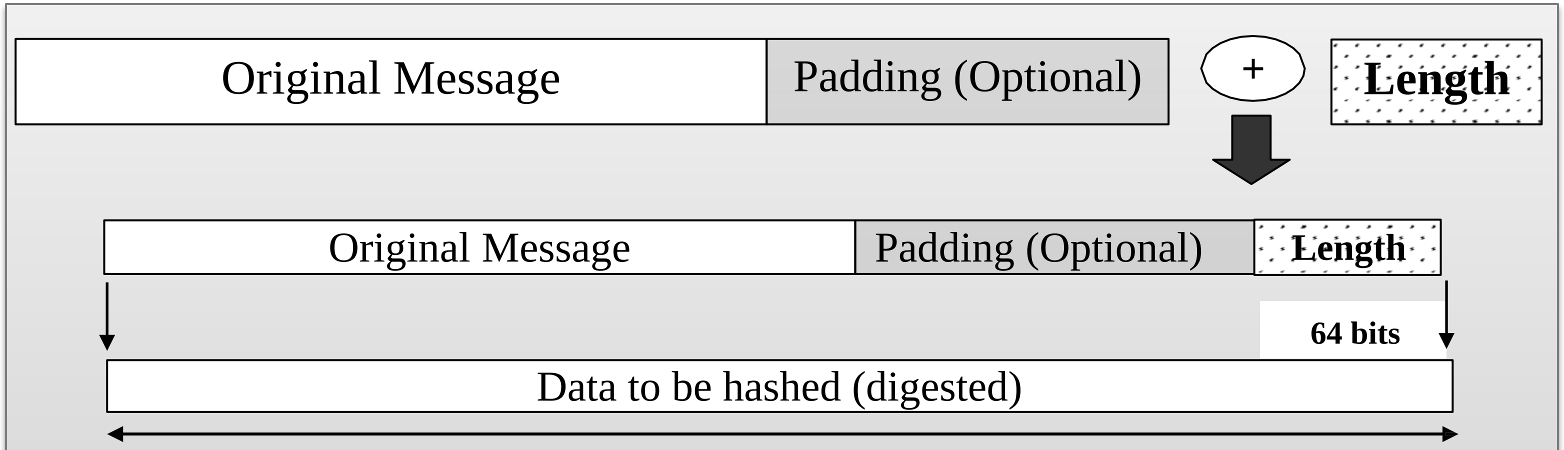
Step 1: Padding

- To make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512
- **Note:** Padding is always added, even if the original message is already 64 bits less than a multiple of 512



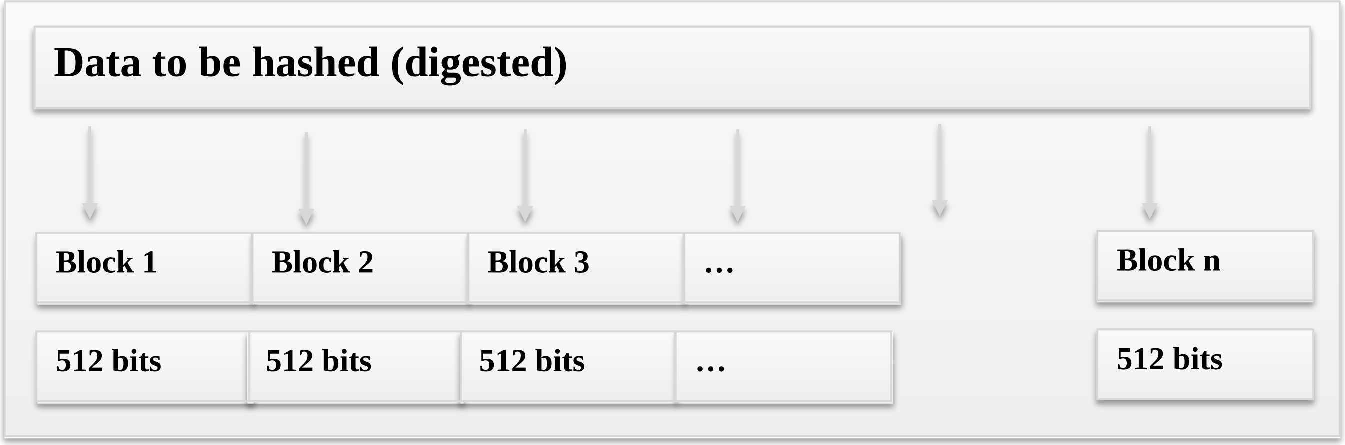
Step 2: Append length

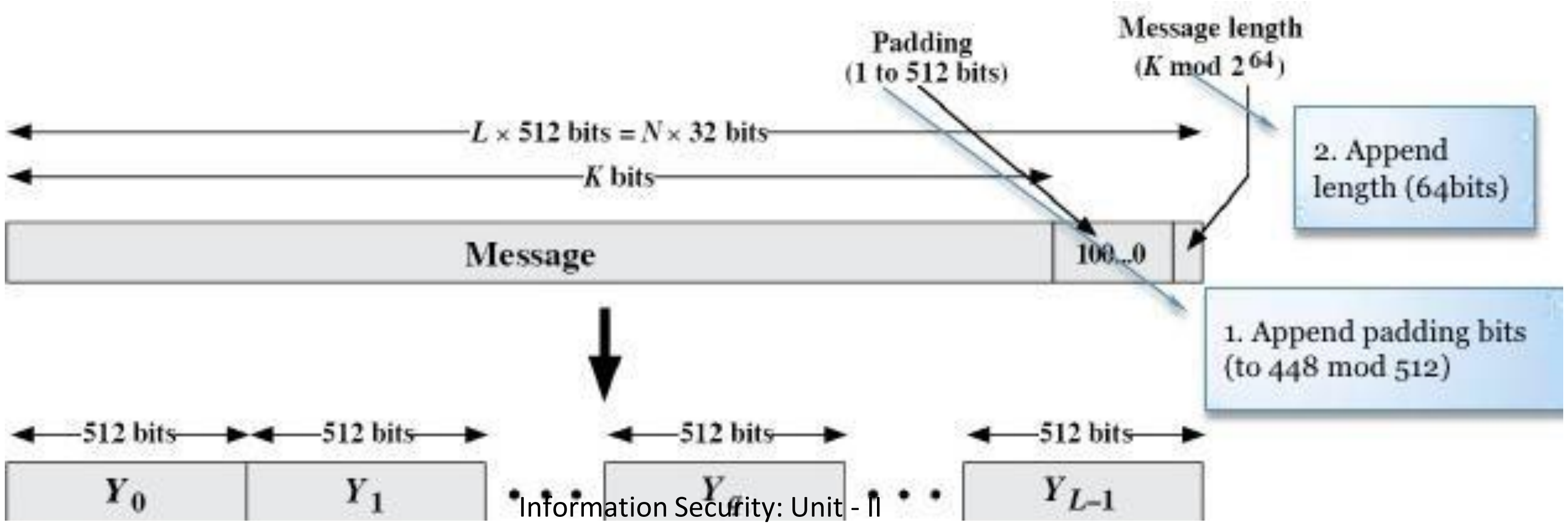
- Add a 64-bit binary-string which is the representation of the message's length
- If the original length is greater than 2^{64}, then only **the low-order 64** bits of the length are used.
- Thus, field contains the length of the original message, modulo 2^{64}.



The length of the original message (excluding the padding) is calculated, and is appended to the end of the message block + padding. This length is expressed in 64 bits. If the length of the message exceeds 64 bits (i.e. it is greater than 2^{64}, then only the last 64 bits of the length are used, i.e. a modulus 64 of the length is taken. After the 64-bit length of the message is appended, this becomes the final message (i.e. the message to be hashed).
The length of the message is now a multiple of 512 bits.

Step 3: Divide the input into 512-bit blocks





Step 4: Initialize MD buffer

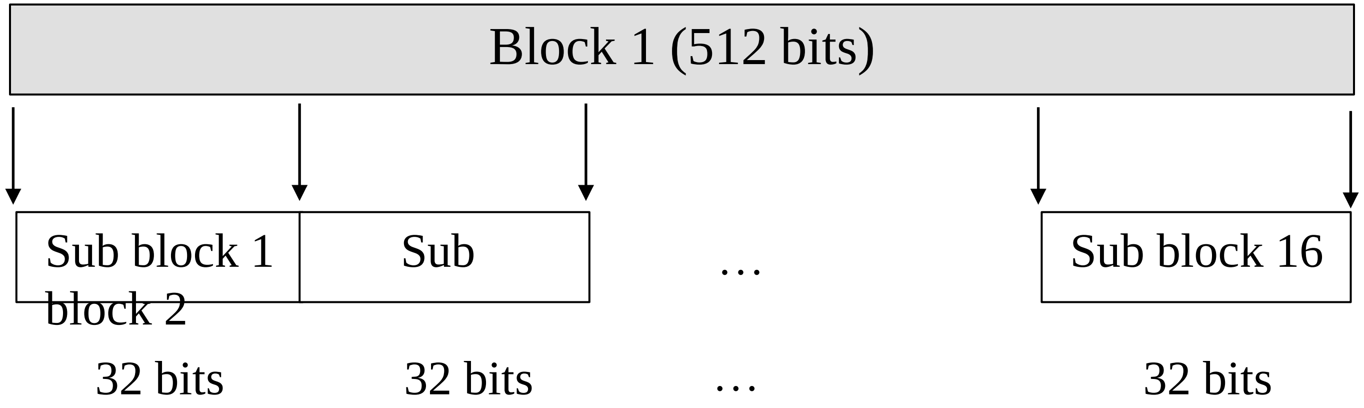
- A four-word buffer (A, B, C, D) is used to compute the message digest.

- Here each of A, B, C, D is a 32 bit register.

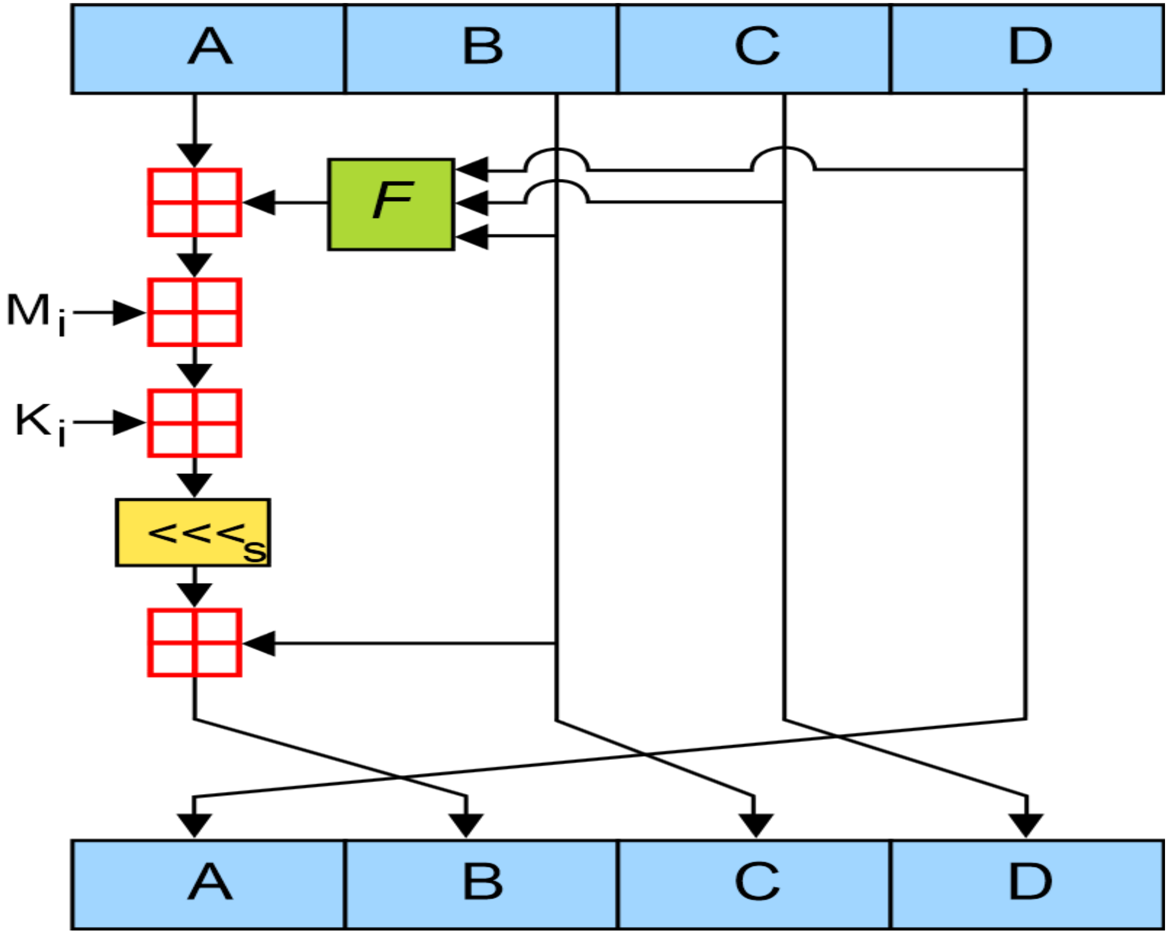
A	01	23	45	67
B	89	AB	CD	EF
C	FE	DC	BA	98
D	76	54	32	10

Step 5: Process Blocks (or message)

- Divide the 512- bit block into 16 sub-blocks.
- Each sub-block undergoes 4 rounds of operations. Total 16 operations are performed.



$$A = B + ((A + \text{Process F}(B, C, D) + M_i + K_i) <<< s)$$



- There are four possible functions F; a different one is used in each round:

Round	Process F
1	(B AND C) OR ((NOT B) AND (D))
2	(B AND D) OR (C AND (NOT D))
3	B XOR C XOR D
4	C XOR (B OR (NOT D))

Types of Attack on Hashes

- **Preimage:** An attacker has an output and finds an input that hashes to that output
- **2nd Preimage:** An attacker has an output and an input x and finds a 2nd input that produces the same output as x
- **Collision:** An attacker finds two inputs that hash to the same output
- **Length Extension:** An attacker, knowing the length of message M and a digest of M signed by a sender can extend M with an additional message N and can compute the digest of M || N even without the key used to sign the digest of M

Secure Hash Algorithm (SHA)

- SHA is a modified version of MD5. (Published in 1993)
- SHA works any input message less than 2^{64} bits and produces a hash value of 160 bits.
- SHA is designed to be computationally infeasible to:
  - Obtain the original message
  - Find two message producing the same MD.

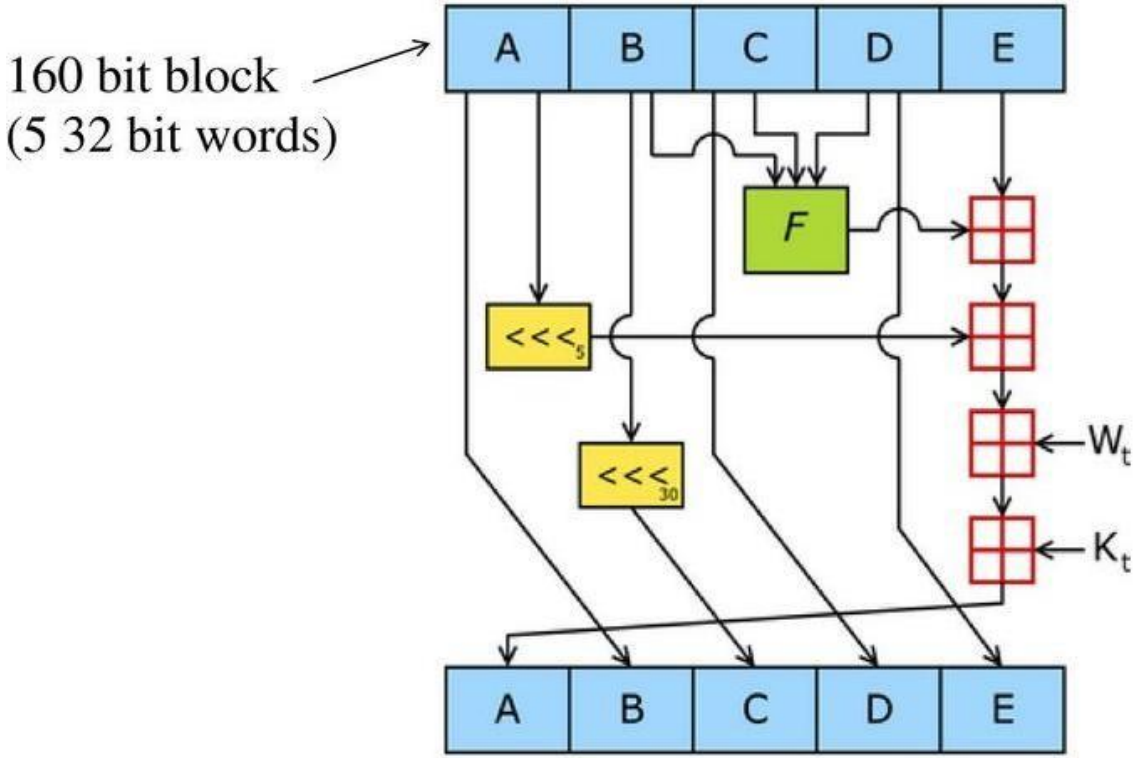
	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	64	80	80
Security	80	128	192	256

Algorithm:

- Step -1: Padding
- Step - 2: Append length
- Step - 3: Divide the input into 512-bit blocks.
- Step - 4: Initialize chaining variables (5 variables)
- Step - 5: Process blocks

A	01	23	45	67
B	89	AB	CD	EF
C	FE	DC	BA	98
D	76	54	32	10
E	C3	D2	E1	F0

Process each block with A, B, C, D, E




```
temp = (A <<<5) + F + E + Kt + wt
E = D
D = C
C = B <<<30
B = A
A = temp
```

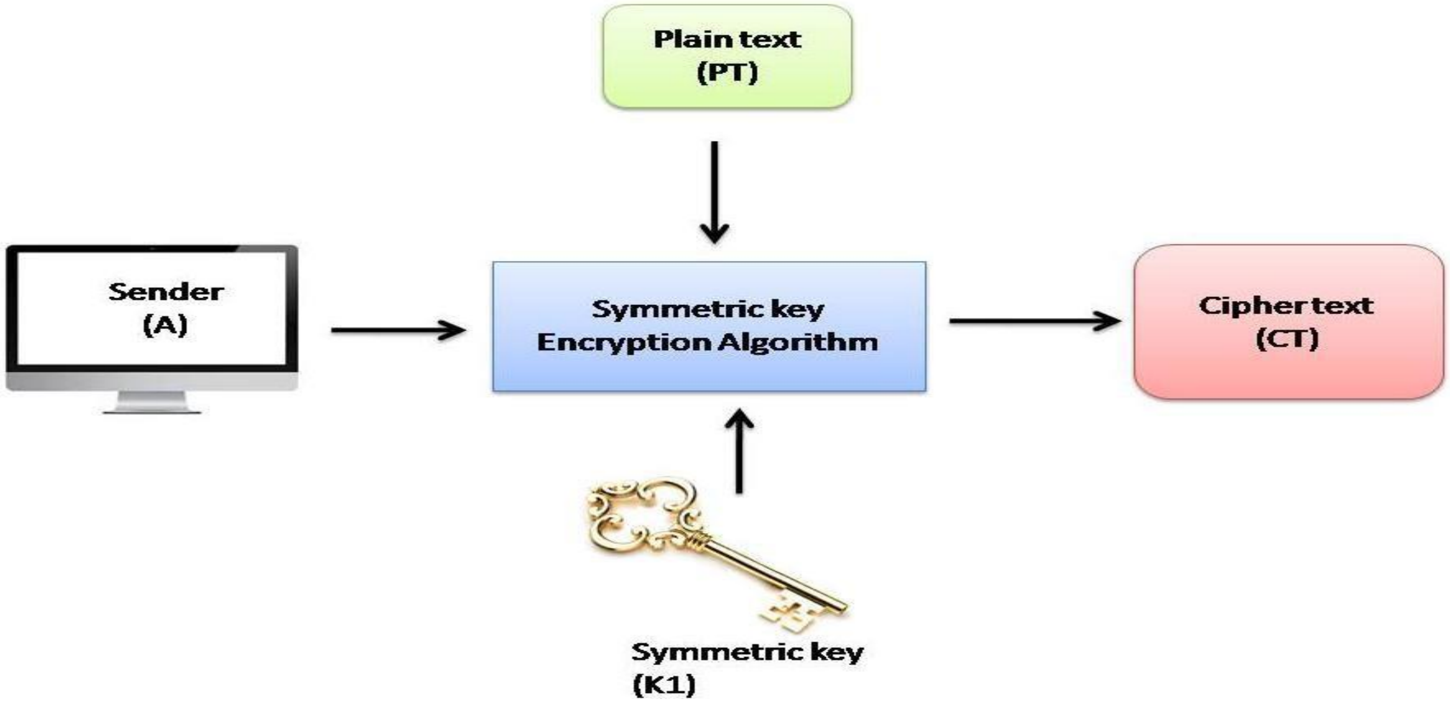
Last round:
A-E is the digest

Round	Process P
1	(b AND c) OR ((NOT b) AND (d))
2	b XOR c XOR d
3	(b AND c) OR (b AND d) OR (c AND d)
4	b XOR c XOR d

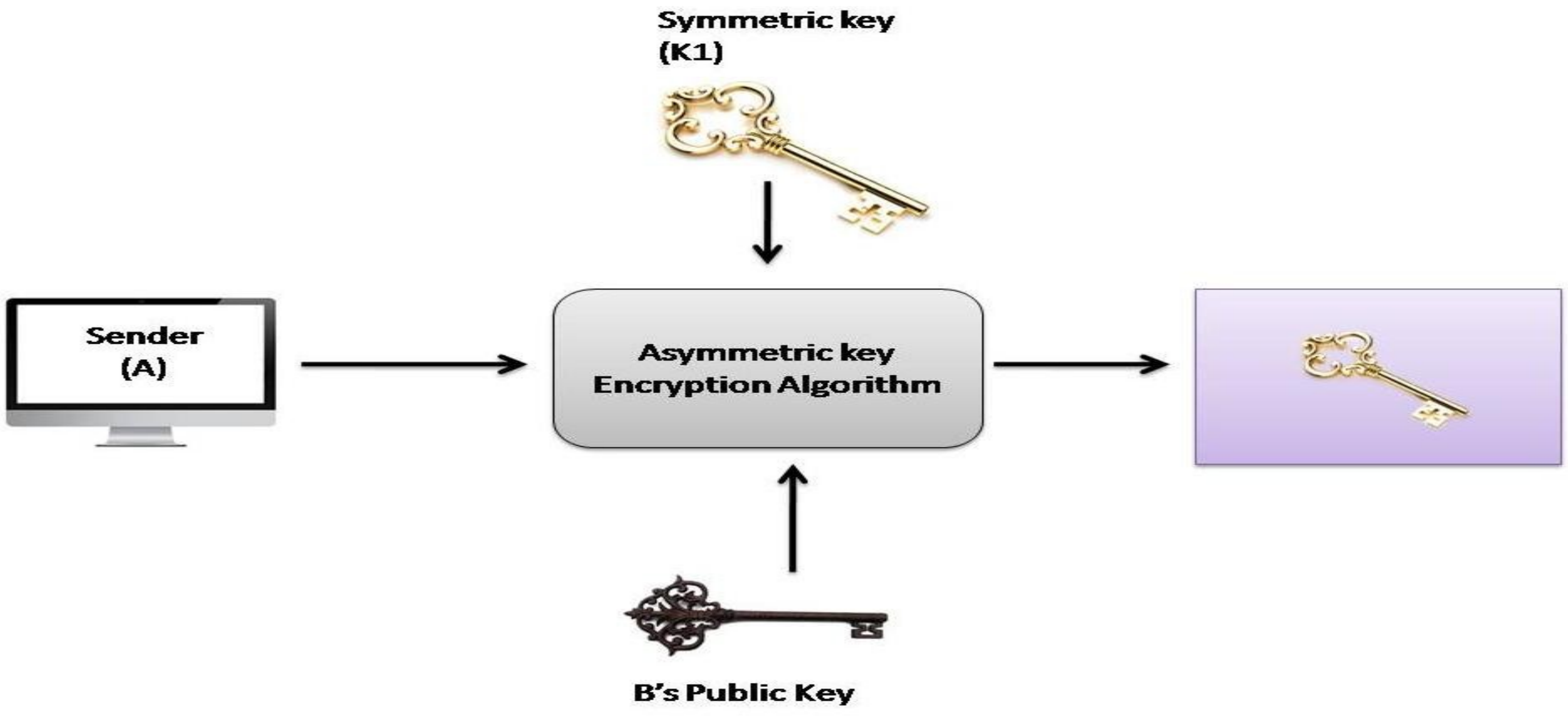
Comparison of MD5 and SHA

Point of discussion	MD5	SHA
Message digest length in bits	128	160
Attack to try and find the original message given a message digest	Requires 2^{128} operations to break in	Requires 2^{160} operations to break in, therefore more secure
Attack to try and find two messages producing the same message digest	Requires 2^{64} operations to break in	Requires 2^{80} operations to break in
Successful attacks so far	There have been reported attempts to some extent	No such claims so far
Speed	Faster (64 iterations, and 128-bit buffer)	Slower (80 iterations, and 160-bit buffer)
Software implementation	Simple, does not need any large programs or complex tables	Simple, does not need any large programs or complex tables

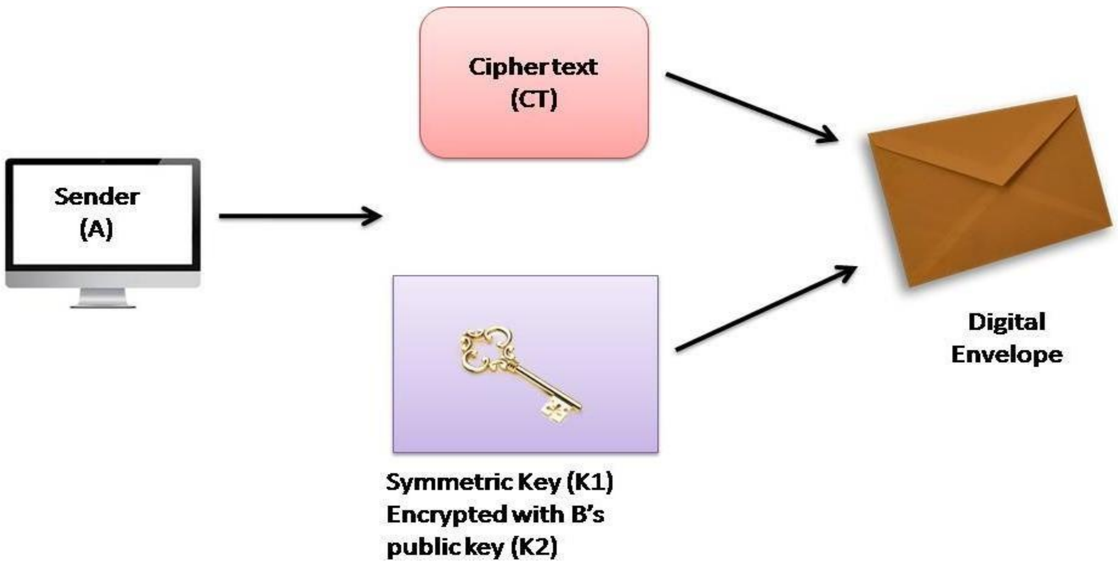
Symmetric and Asymmetric Key Cryptography Together

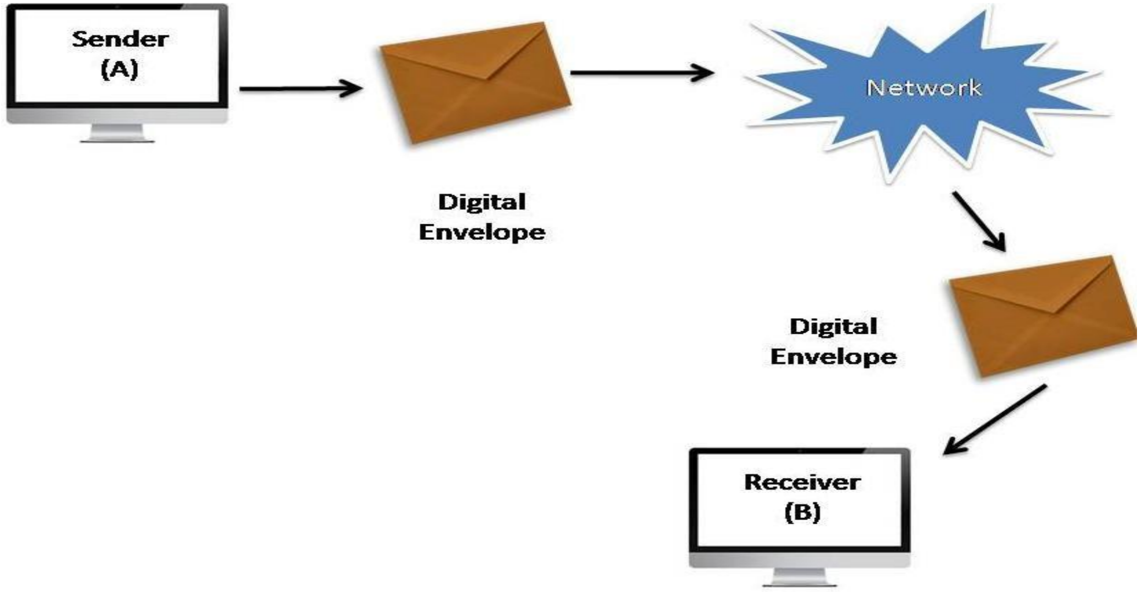


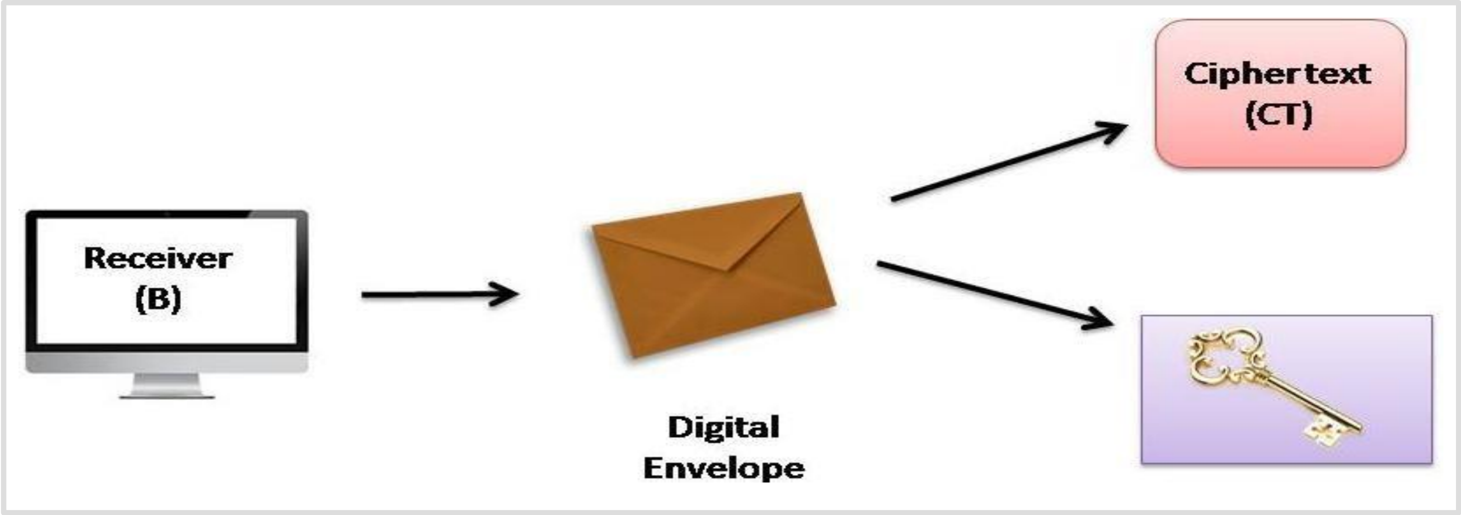
❖ Takes the one-time symmetric key (i.e. K1), and encrypts K1 with B's public key (K2). This process is called **key wrapping** of the symmetric key

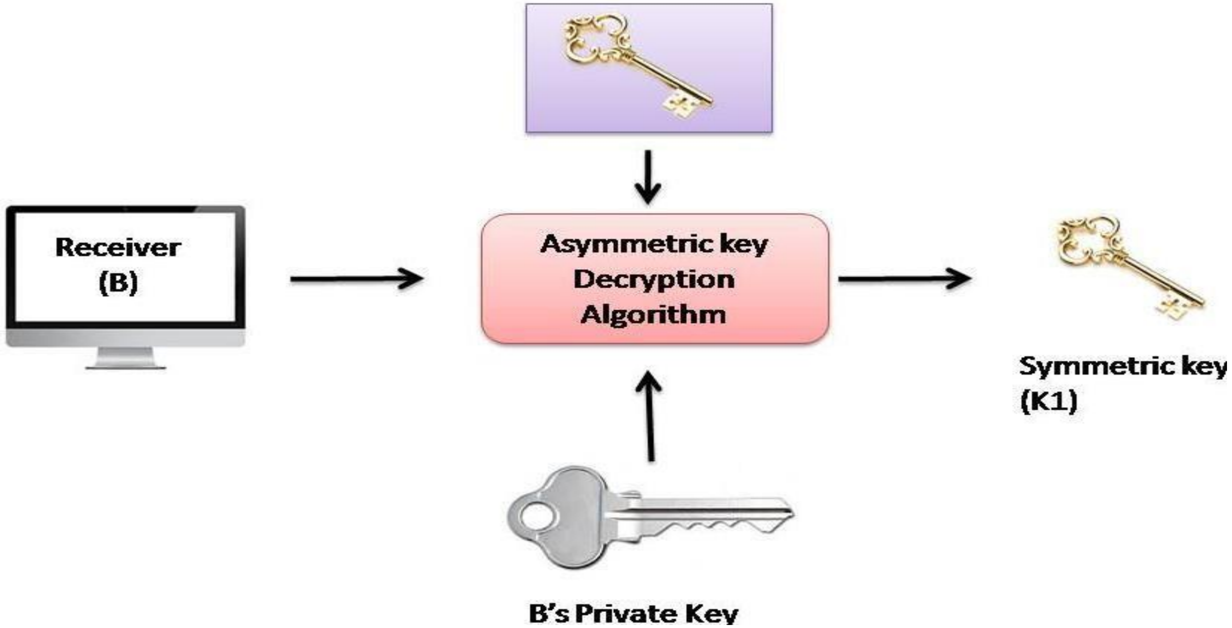


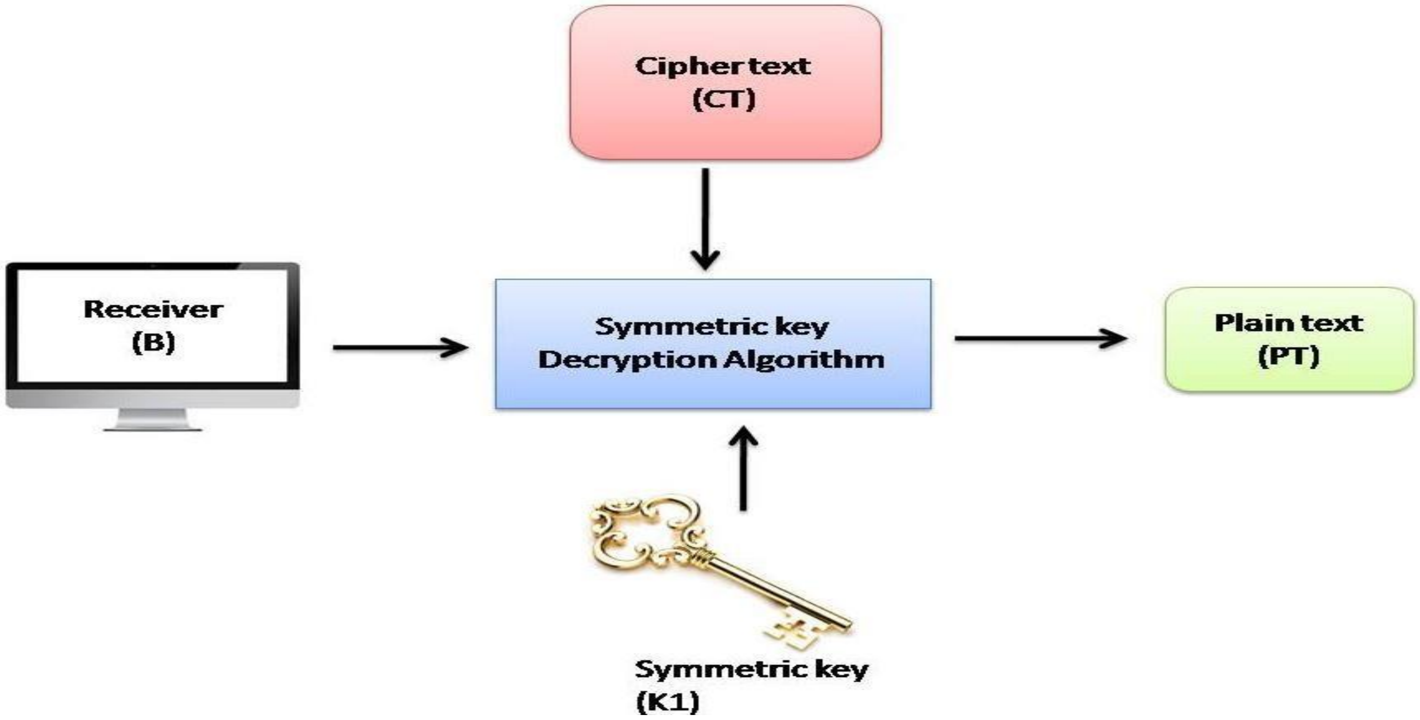
- A puts the cipher text CT and the encrypted symmetric key together inside a digital envelope.






Receiver
(B)
Digital
Envelope
Cipher text
(CT)

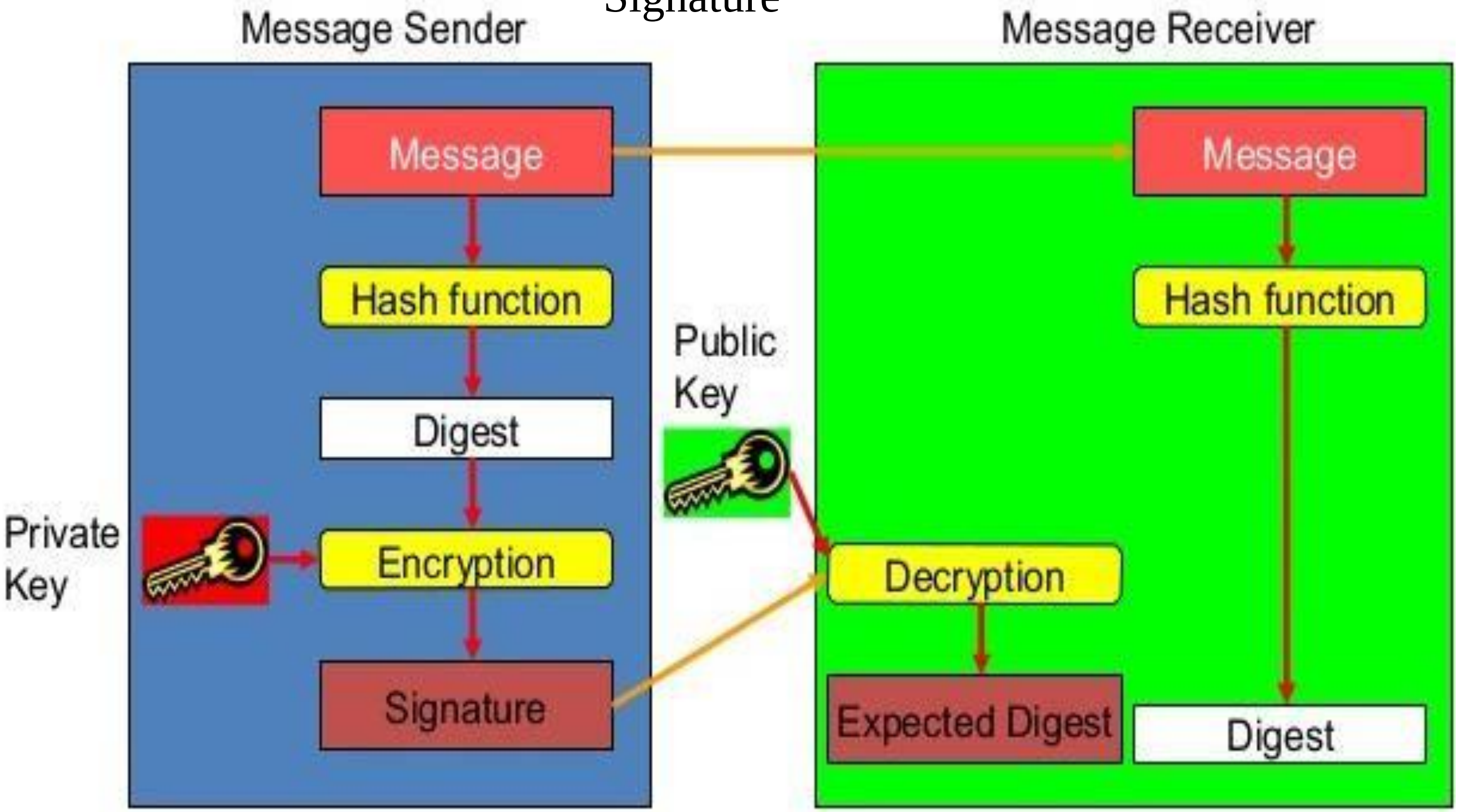

Receiver
(B)
Asymmetric key
Decryption
Algorithm
Symmetric key
(K1)
B's Private Key


Cipher text
(CT)
Receiver
(B)
Symmetric key
Decryption Algorithm
Plain text
(PT)
Symmetric key
(K1)

Digital Signature Techniques

- DSS uses SHA-1
- RSA and DSA

RSA and Digital Signature



Digital Signature Algorithm (DSA)

- creates a 320 bit signature with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal & Schnorr schemes

DSA Key Generation

Note: L will be one member of the set {512, 576, 640, 704, 768, 832, 896, 960, 1024}

- have shared global public key values (**p, q, g**):
  - choose a large prime **p** with $\mathbf{2^{L-1} < p < 2^L}$
    - where L= 512 to 1024 bits and is a multiple of 64
  - choose 160-bit prime number q
    - such that q is a 160 bit prime divisor of (p-1)
  - choose $\mathbf{g = h^{(p-1)/q}}$
    - where $1 < h < p-1$ and $h^{(p-1)/q} \bmod p > 1$
- users choose **private** & compute **public key**:
  - choose random private key: $\mathbf{x < q}$
  - compute public key: $\mathbf{y = g^x \bmod p}$

DSA Signature Creation

- to **sign** a message M the sender:
  - generates a random signature key k, $k < q$
- then computes signature pair:

  $$\mathbf{r = (g^k \bmod p) \bmod q}$$

  $$\mathbf{s = [k^{-1}(SHA(M) + x * r)] \bmod q}$$

- sends **signature (r, s)** with message M

DSA Signature Verification

- having received M & signature (r, s)
- to **verify** a signature, recipient computes:

$w = s^{-1} \bmod q$

$u_1 = [SHA(M) * w] \bmod q$

$u_2 = (r * w) \bmod q$

$v = [(g^{u1} * y^{u2}) \bmod p] \bmod q$

- if $v = r$ then signature is verified

Thank You !!!!!!