CRYPTOGRAPHY AND NETWORK SECURITY
UNIT-3

Syllabus: Number Theory & Asymmetric Key Cryptography

Number Theory: Prime and Relatively Prime Numbers, Modular Arithmetic, Fermat's and Euler's Theorems, The Chinese Remainder theorem, Discrete logarithms. Public Key Cryptography: Principles, public key cryptography algorithms, RSA Algorithms, Diffie Hellman Key Exchange, Elgamal encryption & decryption, Elliptic Curve Cryptography.

3.1 PRIME NUMBER:

Prime numbers only have divisors of 1 and self they cannot be written as a product of other numbers.

eg. 2,3,5,7 are prime, 4,6,8,9,10 are not

prime numbers are central to number theory

List of prime number less than 200 is:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199

An integer $p > 1$ is a prime number if and only if its only divisors are ± 1 and $\pm p$.

Any integer $a > 1$ can be factored in a unique way as

where $p1 < p_2 < ... < p_t$ are prime numbers and where each is a positive integer. This is known as the fundamental theorem of arithmetic

91	$= 7 \times 13$
3600	$= 2^4 \times 3^2 \times 5^2$
11011	$= 7 \times 11^2 \times 13$

If P is the set of all prime numbers, then any positive integer a can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a_p} \quad \text{where each } a_p \geq 0$$

The right-hand side is the product over all possible prime numbers p; for any particular value of a, most of the exponents a_p will be 0.

3.2 RELATIVELY PRIME NUMBERS:

Two numbers a,b are relatively prime (coprime) if they have no common divisors apart from 1.

– eg. 8 and 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor.

3.3 MODULAR ARITHMETIC:

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

$$a = qn + r \qquad 0 \leq r < n; q = \lfloor a/n \rfloor$$

3.4 THE EUCLIDEAN ALGORITHM

One of the basic techniques of number theory is the Euclidean algorithm, which is a simple procedure for determining the **greatest common divisor** of two positive integers.

Greatest Common Divisor:

- **The greatest common divisor of a and b is the largest integer that** divides both a and b . We also define gcd(0, 0) = 0.
- The positive integer c is said to be the greatest common divisor of a and b if
  1. c is a divisor of a and of b;
  2. any divisor of a and b is a divisor of c.
- An equivalent definition is the following:
  gcd(a, b) = max[k, such that k|a and k|b]
  gcd(60, 24) = gcd(60,-24) = 12

In general, gcd(a, b) = gcd(|a|, |b|).

Finding the Greatest Common Divisor:

The Euclidean algorithm is based on the following theorem:

For any nonnegative integer a and any positive integer b, **gcd(a,b)=gcd(b,a mod b)**

gcd(55, 22) = gcd(22, 55 mod 22) = gcd(22, 11) = 11

$$
\begin{aligned}
a &= q_1 b + r_1 && 0 < r_1 < b \\
b &= q_2 r_1 + r_2 && 0 < r_2 < r_1 \\
r_1 &= q_3 r_2 + r_3 && 0 < r_3 < r_2 \\
&\vdots && \vdots \\
r_{n-2} &= q_n r_{n-1} + r_n && 0 < r_n < r_{n-1} \\
r_{n-1} &= q_{n+1} r_n + 0 \\
d &= \gcd(a, b) = r_n
\end{aligned}
$$

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)
GCD(1970,1066)=2

CONGRUENT MODULO:

Two integers a and b are said to be congruent modulo of n if

a mod n= b mod n.

then this is written as **a ≡ b mod n.**

Ex: a=73 b=4 and n=23
73 mod 23 =4
4 mod 23 =4
So 73 *≡ 4 mod 23*

Properties of Congruences:

Congruences have the following properties:

1. $a \equiv b \pmod{n}$ if $n|(a - b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$.
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

Modular Arithmetic Operations:

Many complex cryptographic algorithms are actually based on simple arithmetic. In modular arithmetic the numbers which going to deal are just integers and operations are addition, subtraction, multiplication and division.

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. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

3.5 FERMAT'S AND EULER'S THEOREMS

Fermat's Theorem:

Fermat's theorem states the following: If 'p' is prime and 'a' is a positive integer not divisible by p, then

$$a^{p-1} \equiv 1 \bmod p$$

Proof: Consider the set of positive integers less than p: $\{1, 2, \ldots, p - 1\}$ and multiply each element by a, modulo p, to get the set $X = \{a \bmod p, 2a \bmod p, \ldots, (p - 1)a \bmod p\}$. None of the elements of X is equal to zero because p does not divide a. Furthermore, no two of the integers in X are equal. To see this, assume that $ja \equiv ka \pmod{p}$), where $1 \leq j < k \leq p - 1$. Because a is relatively prime[5] to p, we can eliminate a from both sides of the equation [see Equation (4.3)] resulting in $j \equiv k \pmod{p}$. This last equality is impossible, because j and k are both positive integers less than p. Therefore, we know that the $(p - 1)$ elements of X are all positive integers with no two elements equal. We can conclude the X consists of the set of integers $\{1, 2, \ldots, p - 1\}$ in some order. Multiplying the numbers in both sets (p and X) and taking the result mod p yields

$$a \times 2a \times \ldots \times (p - 1)a \equiv [(1 \times 2 \times \ldots \times (p - 1)] \pmod{p}$$
$$a^{p-1}(p - 1)! \equiv (p - 1)! \pmod{p}$$

We can cancel the (p-1)! term because it is relatively prime to p then

$$a^{p-1} \equiv 1 \bmod p$$

Euler's Totient Function:

- It is defined as the number of positive integers less than 'n' and relatively prime to 'n' and is written as ø(n). By convention ø(1)=1.
- It should be clear that, for a prime number p,

ø(p) = p – 1

ø(37) = 36

- To determine ø(35), we list all of the positive integers less than 35 that are relatively prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18,19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

There are 24 numbers on the list, so . ø(35) = 24

Now suppose that we have two prime numbers p and q with $p \neq q$. Then we can show that, for $n = pq$,

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p - 1) \times (q - 1)$$

$\phi(21) = \phi(3) \times \phi(7) = (3 - 1) \times (7 - 1) = 2 \times 6 = 12$
where the 12 integers are {1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20}.

Euler's Theorem:

Euler's theorem states that for every a and n that are relatively prime:

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

Proof: Equation (8.4) is true if n is prime, because in that case, $\phi(n) = (n - 1)$ and Fermat's theorem holds. However, it also holds for any integer n. Recall that $\phi(n)$ is the number of positive integers less than n that are relatively prime to n. Consider the set of such integers, labeled as

$$R = \{x_1, x_2, \ldots, x_{\phi(n)}\}$$

That is, each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$. Now multiply each element by a, modulo n:

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots, (ax_{\phi(n)} \bmod n)\}$$

The set S is a permutation[6] of R, by the following line of reasoning:

1. Because a is relatively prime to n and x_i is relatively prime to n, ax_i must also be relatively prime to n. Thus, all the members of S are integers that are less than n and that are relatively prime to n.

2. There are no duplicates in S. Refer to Equation (4.5). If $ax_i \bmod n = ax_j \bmod n$, then $x_i = x_j$.

Therefore,

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} ax_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

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

which completes the proof. This is the same line of reasoning applied to the proof of Fermat's theorem.

3.6. THE CHINESE REMAINDER THEOREM:

The **Chinese remainder theorem** (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\cdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

The Chinese remainder theorem states that the above equations have a unique solution if the moduli are relatively prime.

SOLUTION The solution to the set of equations follows these steps:

1. Find $M = m_1 \times m_2 \times \ldots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1$, $M_2 = M/m_2$, …, $M_k = M/m_k$.
3. Find the multiplicative inverse of M_1, M_2, …, M_k using the corresponding moduli (m_1, m_2, …, m_k). Call the inverses M_1^{-1}, M_2^{-1}, …, M_k^{-1}.
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

Example :

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

1. $M = 3 \times 5 \times 7 = 105$
2. $M_1 = 105/3 = 35$, $M_2 = 105/5 = 21$, $M_3 = 105/7 = 15$
3. The inverses are $M_1^{-1} = 2$, $M_2^{-1} = 1$, $M_3^{-1} = 1$
4. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105 = 23 \bmod 105$

3.7 DISCRETE LOGARITHMS:

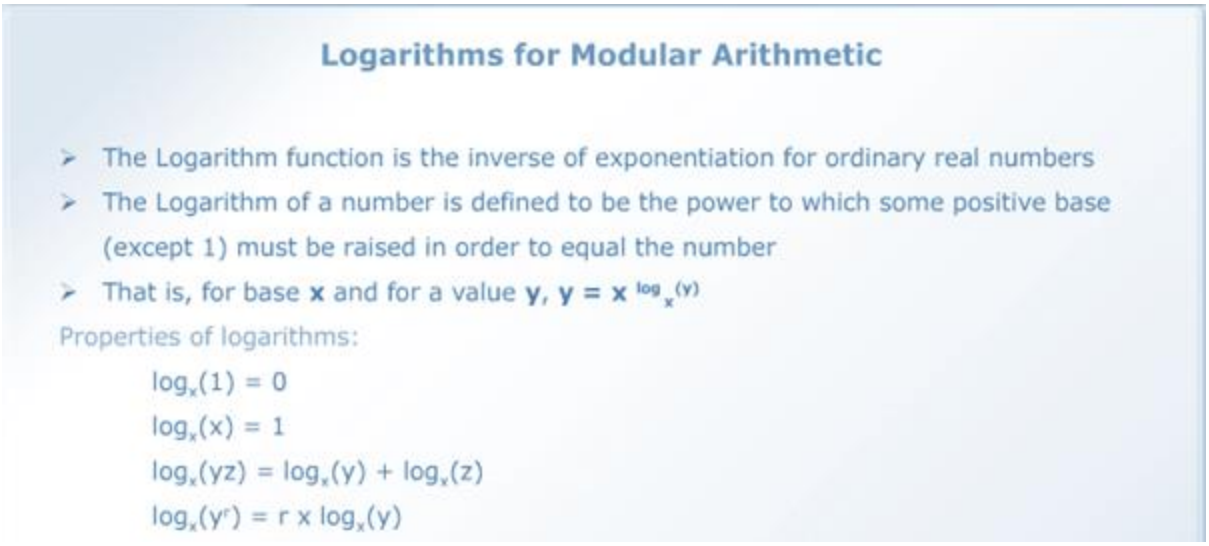
Discrete Logarithms

- From Euler's theorem that, **a** and **n** which are relatively prime, We have, $a^{\phi(n)} \equiv 1 \pmod{n}$ where **ø(n)** is Euler's totient function that gives the number of positive integers less than **n** and relatively prime to **n**
- Let **a^m =1 (mod n)**
- If **a** and **n** are relatively prime, then there is at least one integer **m** that satisfies the given equation
- Least positive exponent **m** for which the above equation holds is referred to in several ways:
  - Order of **a** (mod n)
  - Exponent to which **a** belongs (mod n)
  - Length of the period generated by **a**

7^1		**7(mod 19)**
7^2	**= 49 = 2 x 19 + 11**	**11(mod 19)**
7^3	**= 343 = 18 x 19 + 1**	**1(mod 19)**
7^4	**= 2401 = 126 x 19 + 7**	**7(mod 19)**
7^5	**= 16807 = 884 x 19 + 11**	**11(mod 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

Table. Powers of Integers, Modulo 19



3.8 PUBLIC KEY CRYPTOGRAPHY:

Introduction:

- Asymmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys - one a public key and one a private key. It is also known as public-key encryption.
- Asymmetric encryption transforms plaintext into ciphertext using a one of two keys and an encryption algorithm. Using the paired key and a decryption algorithm, the plaintext is recovered from the ciphertext.
- Asymmetric encryption can be used for confidentiality, authentication, or both.
- The most widely used public-key cryptosystem is RSA.

Principles of Public-Key Cryptosystems:

The concept of public key cryptography in invented for two most difficult problems of Symmetric key encryption.

- **key distribution** – how to have secure communications in general without having to trust a KDC (key distribution center) with your key.
- **digital signatures** – how to verify a message comes intact from the claimed sender.

Public-Key Cryptosystems:

A public-key encryption scheme has six ingredients

- **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.
- **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.

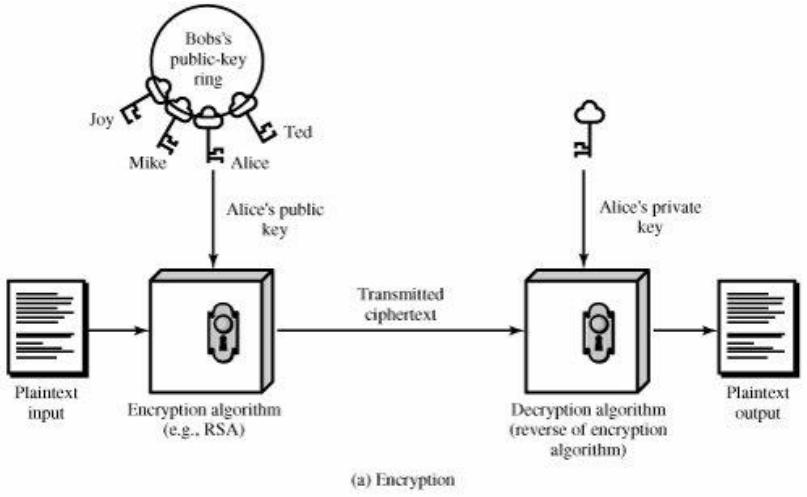
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. Each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

ENCRYPTION:

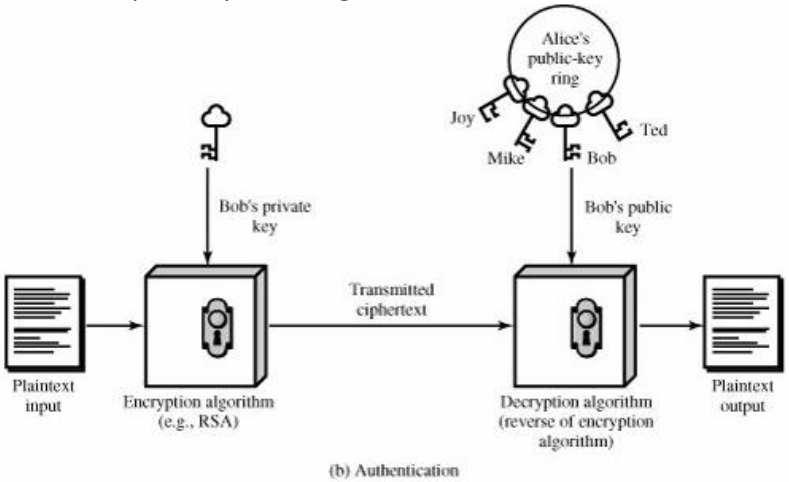
The plaintext is encrypted with receiver's public key and decrypted using receiver private key.



(a) Encryption

AUTHENTICATION:

- Plaintext is encrypted is sender's private key and decrypted using sender's public key.
- The act of messages ciphertext getting decrypted by sender's public key is the proof that the message is actually sent by the designated sender.



(b) Authentication

Difference between symmetric and public key encryption:

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.

- Examples for conventional encryption are DES, AES, IDEA and Blowfish.
- Examples for public key encryption are RSA, Diffie-Hellman, Elliptic Curve Cryptography.

There is some source A that produces a message in plaintext, $X=[X_1,X_2,...,X_M,]$. The M elements of X are letters in some finite alphabet. The message is intended for destination B. B generates a related pair of keys: a public key, PU_b, and a private key, PR_b. PR_b is known only to B, whereas PU_b is publicly available and therefore accessible by A.

With the message X and the encryption key PU_b as input, A forms the ciphertext $Y=[Y_1, Y_2,..., Y_N]$:

$$Y = E(PU_b, X)$$

The intended receiver, in possession of the matching private key, is able to invert the transformation:

$$X = D(PR_b, Y)$$

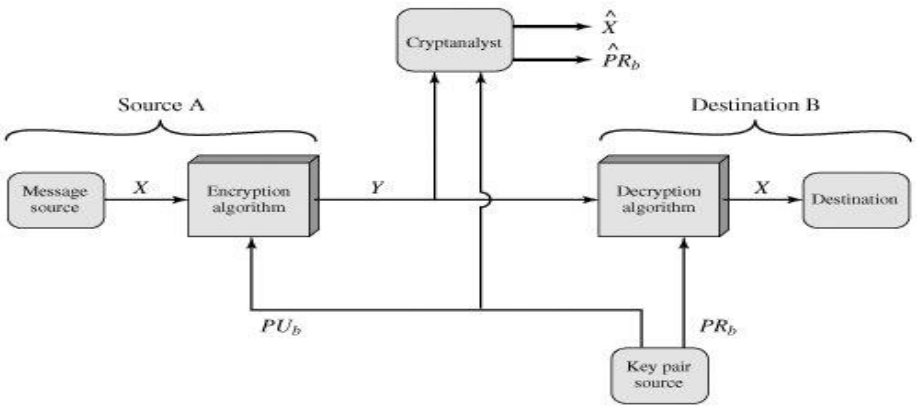


Figure: public key cryptosystems: Secrecy (or) confidentiality

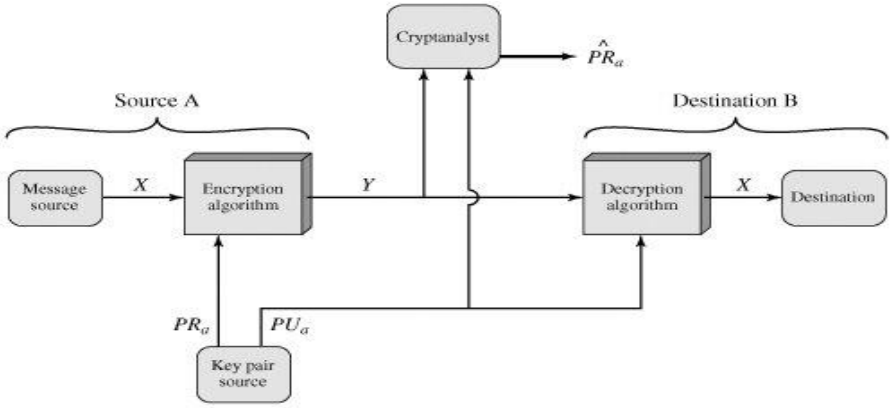


Figure: Public-Key Cryptosystem: Authentication

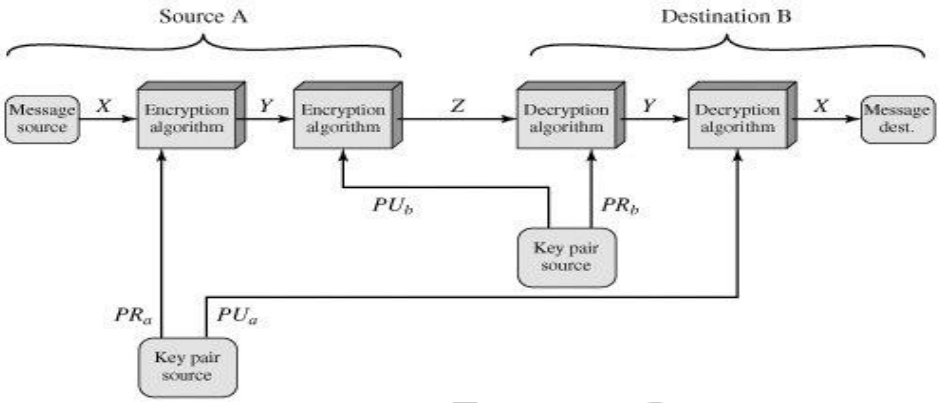


Figure: Public-Key Cryptosystem: Authentication and Secrecy

Applications for Public-Key Cryptosystems:

- **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
- Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message.
- **Key exchange:** Two sides cooperate to exchange a session key.

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

Requirements for Public-Key Cryptography:

1. It is computationally easy for a party B to generate a pair (public key PU_b, private key PR_b).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext: $C = E(PU_b, M)$
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$

4. It is computationally infeasible for an adversary, knowing the public key, PU_b, to determine the private key, PR_b.
5. It is computationally infeasible for an adversary, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M.

We can add a sixth requirement that, although useful, is not necessary for all public-key applications:
The two keys can be applied in either order:
$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$

3.9 THE RSA ALGORITHM

RSA is a public key encryption algorithm developed by Rivert(R) , Shamir(S) and Adleman (A) in year 1977. The RSA scheme is a block cipher in which the plaintext & 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 uses an expression with exponentials.

- In RSA plaintext is encrypted in blocks, with each block having a binary value less than some number n. that is, the block size must be less than or equal to log2(n)+1
- RSA uses two exponents 'e' and 'd' where e → public and d → private.
- Encryption and decryption are of following form, for some **PlainText 'M'** and **CipherText block 'C'**

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

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

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

- Both sender and receiver must know the value of n.
- The sender knows the value of 'e' & only the reviver knows the value of 'd' thus this is a public key encryption algorithm with a
  Public key PU={e, n}
  Private key PR={d, n}

Requirements:

The RSA algorithm to be satisfactory for public key encryption, the following requirements must be met:

- It is possible to find values of e, d and n such that " $M^{ed} \bmod n = M$ " for all M<n
- It is relatively easy to calculate " M^e mod n " and " C^d mod n "for all values of M<n
- It is infeasible to determine "d" given 'e' & 'n'.

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
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:	$M = C^d \bmod n$

Example

1. Select two prime numbers, $p = 17$ and $q = 11$.
2. Calculate $n = pq = 17 \times 11 = 187$.
3. Calculate $\emptyset(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.
4. Select e such that e is relatively prime to $\emptyset(n) = 160$ and less than $\emptyset(n)$; we choose $e = 7$.
5. Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$. The correct value is $d = 23$,

because $23 * 7 = 161 = (1 \times 160) + 1$; d can be calculated using the extended Euclid's algorithm

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$.

The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$.

Exploiting the properties of modular arithmetic, we can do this as follows.

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$$

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$$

$11^1 \bmod 187 = 11$

$11^2 \bmod 187 = 121$

$11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$

$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$

$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

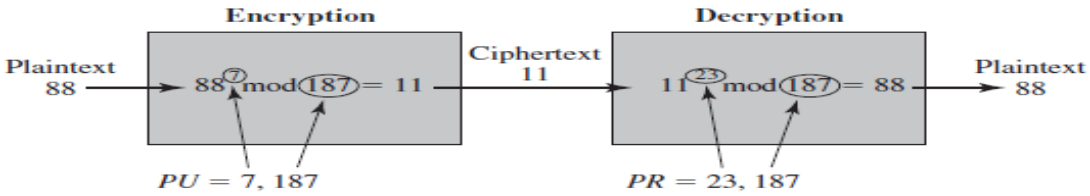


Figure 9.6 Example of RSA Algorithm

3.10. Diffie-Hellman key exchange/Agreement Algorithm:

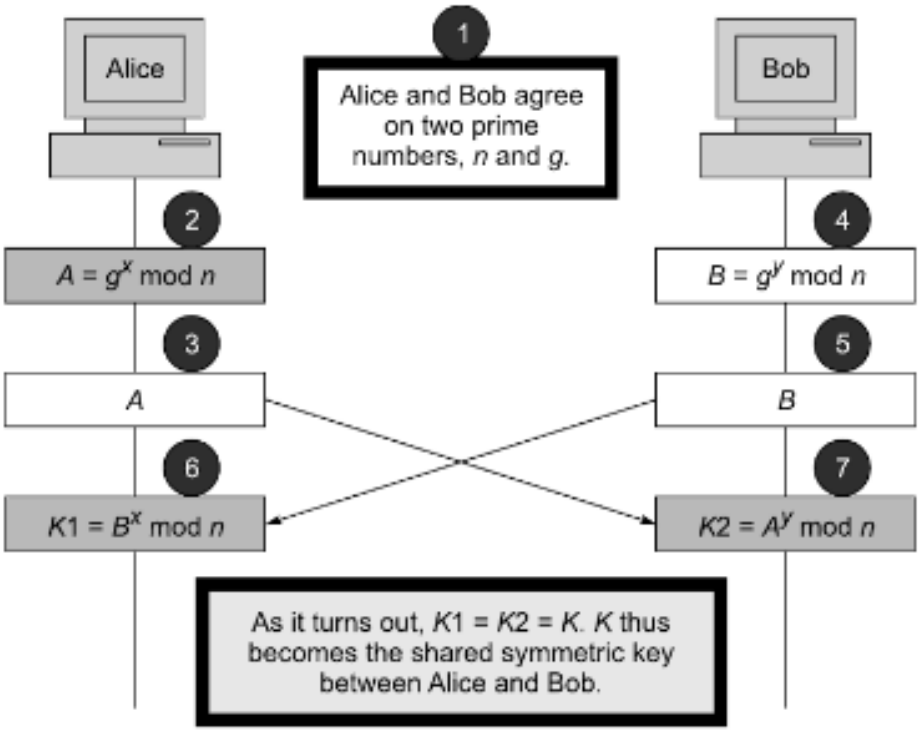
Whitefield Diffie and Martin Hellman devised an amazing solution to the problem of key agreement, or key exchange, in 1976. This solution is called the Diffie—Hellman key exchange/agreement algorithm. The beauty of this scheme is that the two parties, who want to communicate securely, can agree on a symmetric key using this technique. This key can then be used for encryption/decryption. However, we must note that the Diffie-Hellman key exchange algorithm can be used only for key agreement, but not for encryption or decryption of messages. Once both the parties agree on the key to be used, they need to use other symmetric key-encryption algorithms for actual encryption or decryption of messages. Although the Diffie—Hellman key-exchange algorithm is based on mathematical principles, it is quite simple to understand.

Description of the algorithm:

Let us assume that Alice and Bob want to agree upon a key to be used for encrypting/decrypting messages that would be exchanged between them. Then, the Diffie–Hellman key-exchange algorithm works as shown in Fig. 2.52.

1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.
2. Alice chooses another large random number x, and calculates A such that: $A = g^x \bmod n$
3. Alice sends the number A to Bob.
4. Bob independently chooses another large random integer y and calculates B such that: $B = g^y \bmod n$
5. Bob sends the number B to Alice.
6. A now computes the secret key $K1$ as follows: $K1 = B^x \bmod n$
7. B now computes the secret key $K2$ as follows: $K2 = A^y \bmod n$

Fig. 2.52 Diffie-Hellman key-exchange algorithm



Example of the algorithm:

1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.

   Let $n = 11, g = 7$.

2. Alice chooses another large random number x, and calculates A such that:
   $A = g^x \bmod n$

   Let $x = 3$. Then, we have, $A = 7^3 \bmod 11 = 343 \bmod 11 = 2$.

3. Alice sends the number A to Bob.

   Alice sends 2 to Bob.

4. Bob independently chooses another large random integer y and calculates B such that
   $B = g^y \bmod n$

   Let $y = 6$. Then, we have, $B = 7^6 \bmod 11 = 117649 \bmod 11 = 4$.

5. Bob sends the number B to Alice.

   Bob sends 4 to Alice.

6. A now computes the secret key $K1$ as follows:
   $K1 = B^x \bmod n$

   We have, $K1 = 4^3 \bmod 11 = 64 \bmod 11 = 9$.

7. B now computes the secret key $K2$ as follows:
   $K2 = A^y \bmod n$

   We have, $K2 = 2^6 \bmod 11 = 64 \bmod 11 = 9$.

PROBLEM WITH THE ALGORITHM (MAN-IN-THE-MIDDLE ATTACK):

Can we now consider that the Diffie—Hellman key-exchange algorithm solves all our problems associated with key exchange? Unfortunately, not quite! The Diffie-Hellman key exchange algorithm can fall pray to the man-in-the-middle attack

1. Alice wants to communicate with Bob securely, and therefore, she first wants to do a Diffie-HeIlman key exchange with him. For this purpose, she sends the values of n and g to Bob, as usual. Let n =11 and g = 7. (As usual, these values Will form the basis of Alice's A and Bob's B, which will be used to calculate the symmetric key K1 = K2 = K)

2. Alice does not realize that the attacker Tom is listening quietly to the conversation between her and Bob. Tom simply picks up the values of n and g, and also forwards them to Bob as they originally were (i.e. n = Il and g = 7).

Alice	Tom	Bob
$n = 11, g = 7$	$n = 11, g = 7$	$n = 11, g = 7$

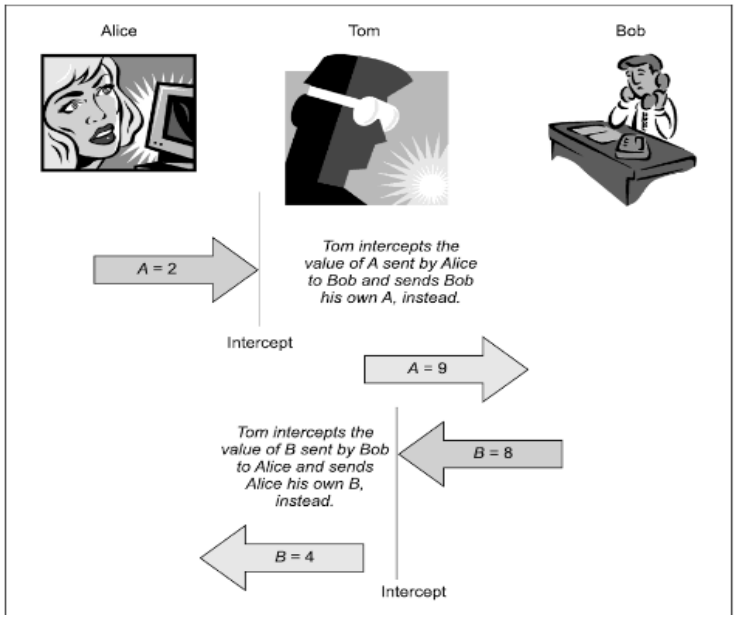
3. Now, let us assume that Alice, Tom and Bob select random numbers x and y as shown in Figure.

Alice	Tom	Bob
$x = 3$	$x = 8, y = 6$	$y = 9$

4. One question at this Stage could be: Why does Tom select both x and y? We shall answer that shortly. Now, based on these values, all the three persons calculate the values of A and B as shown in Figure, note that Alice and Bob calculate only A and B, respectively. However, Tom calculates both A and B.

Alice	Tom	Bob
$A = g^x \bmod n$ $= 7^3 \bmod 11$ $= 343 \bmod 11$ $= 2$	$A = g^x \bmod n$ $= 7^8 \bmod 11$ $= 5764801 \bmod 11$ $= 9$	$B = g^y \bmod n$ $= 7^9 \bmod 11$ $= 40353607 \bmod 11$ $= 8$
	$B = g^y \bmod n$ $= 7^6 \bmod 11$ $= 117649 \bmod 11$ $= 4$	

5. Now the real drama begins as shown in figure,



As shown in the figure, the following things happen:

a) Alice sends her (i.e. 2) to Bob. Tom intercepts it, and instead, sends his A (i.e. 9) to Bob. Bob has no idea that Tom had hijacked Alice's A and has instead given his A to Bob.

b) In return, Bob sends his B (i.e. 8) to Alice. As before, Tom intercepts it, and instead, sends his B (i.e. 4) to Alice. Alice thinks that this B came from Bob. She has no idea that Tom had intercepted the transmission from Bob, and changed B.

c) Therefore, at this juncture, Alice, Tom and Bob have the values Of A and B as shown in following Fig.

Alice	Tom	Bob
$A = 2, B = 4^*$	$A = 2, B = 8$	$A = 9^*, B = 8$

(Note: * indicates that these are the values after Tom hijacked and changed them.)

Based on these values, all The three persons now calculate their keys as shown in following Figure. We will notice that Alice calculates only K1 , Bob calculates only K2, whereas Tom calculates both K1 and K2. Why does Tom need to do this?

Alice		Tom		Bob	
$K1$	$= B^x \bmod n$	$K1$	$= B^x \bmod n$	$K2$	$= A^y \bmod n$
	$= 4^3 \bmod 11$		$= 8^8 \bmod 11$		$= 9^9 \bmod 11$
	$= 64 \bmod 11$		$= 16777216 \bmod 11$		$= 387420489 \bmod 11$
	$= 9$		$= 5$		$= 5$
		$K2$	$= A^y \bmod n$		
			$= 2^6 \bmod 11$		
			$= 64 \bmod 11$		
			$= 9$		

Let us now revisit the question as to why Tom needs two keys. This is because at one side, Tom wants to communicate with Alice securely using a shared symmetric key (9), and on the other hand, he wants to communicate with Bob securely using a different shared symmetric key Only then can he receive messages from Alice, view/manipulate them and forward them to Bob, and Vice versa. Unfortunately for Alice and Bob, both will (incorrectly) believe that they are directly communicating with each other. That is, Alice Will feel that the key 9 is shared between her and Bob, whereas Bob Will feel that the key 5 is shared between him and Alice. Actually, what is happening is, Tom is sharing the key 5 with Alice and with Bob!

This is also the reason why Tom needed both sets of the secret variables x and y, as well as later on, the non-secret variables A and B. As we can see, the man-in-middle/attack can work against the Diffie-Hellman key-exchange algorithm, causing it to fail. This is plainly because the man-in-the-middle makes the actual communicators believe that they are talking to each other, whereas they are actually talking to the man-in-the-middle, who is talking to each of them! This attack can be prevented if Alice and Bob authenticate each other before beginning to exchange information. This proves to Alice that Bob is indeed Bob, and not someone else (e.g. Tom) posing as Bob, Similarly, Bob can also get convinced that Alice is genuine as well.

3.11. ELGAMAL CRYPTOGRAPHY:

Taber ElGamal created EIGamal cryptography, more popularly known as EIGamal cryptosystem. There are three aspects that need to be discussed: EIGamal key generation, EIGamal encryption, and EIGamal decryption.

Elgamal key generation:

This involves the following steps:

1. Select a large prime number called P. This is the first part of the encryption key or public key.
2. Select the decryption key or private key D. There are some mathematical rules that need to be followed here, which we are omitting for keeping things simple.
3. Select the second part of the encryption key or public key $E1$.
4. The third part of the encryption key or public key $E2$ is computed as $E2 = E1^D \bmod P$.
5. The public key is ($E1$, $E2$, P) and the private key is D.

For example, $P = 11$, $E1 = 2$, $D = 3$. Then $E2 = E1^D \bmod P = 2^3 \bmod 11 = 8$.

Hence, the public key is (2, 8, 11) and the private key is 3.

Elgamal key encryption:

This involves the following steps:

1. Select a random integer R that fulfills some mathematical properties, which are ignored here.
2. Compute the first part of the cipher text $C1 = E1^R \bmod P$.
3. Compute the second part of the cipher text $C2 = (PT \times E2^R) \bmod P$, where PT is the plain text.
4. The final cipher text is ($C1$, $C2$).

In our example, let $R = 4$ and plain text $PT = 7$. Then we have:

$C1 = E1^R \bmod P = 2^4 \bmod 11 = 16 \bmod 11 = 5$

$C2 = (PT \times E2^R) \bmod P = (7 \times 2^8) \bmod 11 = (7 \times 4096) \bmod 11 = 6$

Hence, our cipher text is (5, 6).

Elgamal key decryption:

This involves the following step:

Compute the plain text PT using the formula $PT = [C2 \times (C1^D)^{-1}] \bmod P$

In our example:

$PT = [C2 \times (C1^D)^{-1}] \bmod P$

$PT = [6 \times (5^3)^{-1}] \bmod 11 = [6 \times 3] \bmod 11 = 7$, which was our original plain text.

3.12. ELLIPTIC CURVE CRYPTOGRAPHY(ECC):

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms that is much more difficult to challenge at equivalent key lengths.

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_A < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$