UNIT-2

PKC-Public-Key Cryptosystems (Asymmetric Cryptosystem)

Principles of Public-Key Cryptosystems

Key Points

- 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. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

Asymmetric algorithms have the following important characteristic:

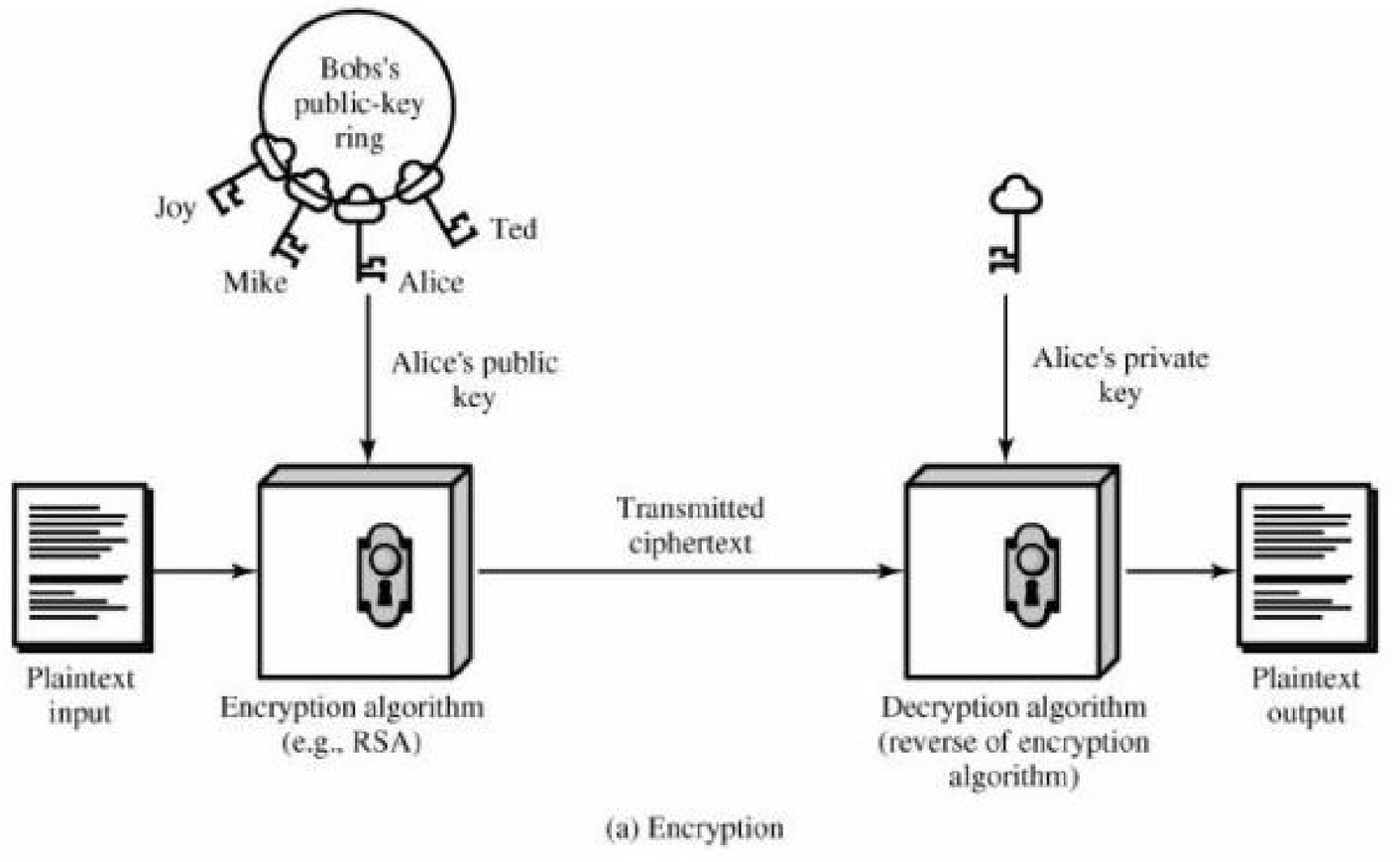
- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

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 of PKC:

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. As Figure suggests, each user maintains a collection of public keys obtained from others.



(a) Encryption

3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice'spublic key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient candecrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys, and private keys are generated locallyby each participant and therefore need never be distributed. As long as a user's private key remains protected and secret, incoming communication is secure. At any time, a system can change its private key and publish the companion public key to replace its old public key.

Difference between Conventional and Public-Key Encryption

Conventional Encryption	Public-Key Encryption
1. The same algorithm with the same keyis used for encryption and decryption.	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 share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
3. The key must be kept secret.	3. One of the two keys must be kept secret.
4. It must be impossible or at least impractical to decipher a message if no other information is available.	4. It must be impossible or at least impractical to decipher a message if no other information is available.
5. Knowledge of the algorithm plussamples of ciphertext must be insufficient to determine the key.	5. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Applications for Public-Key Cryptosystems

In broad terms, we can classify the use of public-key cryptosystems into three categories:

- **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 or to a small block of data that is a function of the message.
- **Key exchange:** Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

Applications for Public-Key Cryptosystems

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 *PUb*, private key *PRb*).
2. It is computationally easy for a sender A, knowing the public key and the message to beencrypted, *M*, to generate the corresponding ciphertext:
   $C = E(PUb, M)$
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using theprivate key to recover the original message:
   $M = D(PRb, C) = D[PRb, E(PUb, M)]$
4. It is computationally infeasible for an adversary, knowing the public key, *PUb*, to determinethe private key, *PRb*.
5. It is computationally infeasible for an adversary, knowing the public key, *PUb*, and aciphertext, *C*, to recover the original message, *M*.
   We can add a sixth requirement that, although useful, is not necessary for all public-keyapplications:
6. The two keys can be applied in either order:
   $M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]$

The RSA Algorithm

RSA was invented by three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** and hence, itis termed as RSA cryptosystem.We will see two aspects of the RSA cryptosystem,

- firstly generation of key pair and
- Secondly encryption-decryption algorithms.

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

An example of generating RSA Key pair is given below. (For ease of understanding, theprimes p & q taken here are small values. Practically, these values are very high).

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factorof 5 and (p − 1)(q − 1) = 6 × 12 = 72, except for 1.

- The pair of numbers (n, e) = (91, 5) forms the public key and can be made availableto anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output willbe d = 29.
- Check that the d calculated is correct by computing –

$$de = 29 \times 5 = 145 = 1 \bmod 72$$

- Hence, public key is (91, 5) and private keys is (91, 29).

RSA Security Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

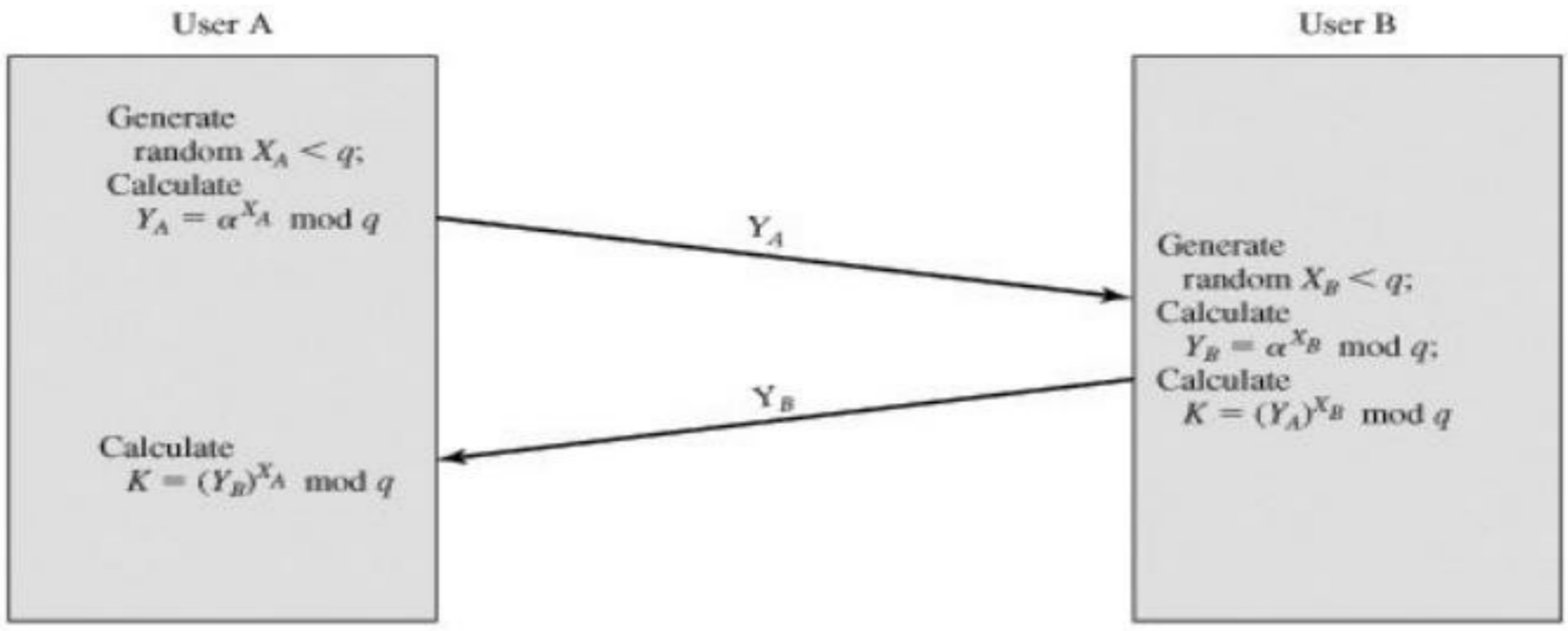
- **Encryption Function** − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.
- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also aone way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

Diffie-Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key thatcan then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.
- The Diffie-Hellman algorithm depends for its effectiveness on the difficulty ofcomputing discrete logarithms



The Algorithm

Global Public Elements	
q	prime number
α	$\alpha < q$ *and* α a primitive root of q

User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Assume two communication Parties A for (Anu) and B for (Banu):

First, Anu and Banu agree on a large prime, ***n*** and ***g***, such that ***g*** is *primitive root* mod ***n***. These two integers don't have to be secret; Anu and Banu can agree to them over some insecure channel. They can even be common among a group of users. It doesn't matter.

Then, the algorithm goes as follows:

(1) Anu chooses a random large integer x and sends Banu

$$X = g^x \bmod n$$

(2) Banu chooses a random large integer ***y*** and sends to Anu

$$Y = g^y \bmod n$$

(3) Anu computes

$$k = Y^x \bmod n$$

(4) Banu computes

$$K' = X^y \bmod n$$

Both k and k´ are equal to $g^{xy} \bmod n$.

Example

Assume Anu generated the values as ***n*** =11 and ***g***= 7 and communicated to Banu over a channel.

Anu	Banu
n=11, g=7	n=11, g=7

b. Anu and Banu has the same n and g values.

Anu	Banu
x=3	y=9

c. Anu and Banu computed ***A*** and ***B*** values at their sides.

Anu	Banu
$A=g^x \bmod n$	$B=g^Y \bmod n$
$=7^3 \bmod 11$	$=7^9 \bmod 11$
$=343 \bmod 11$	$= 40353607 \bmod 11$
=2	**=8**

d. Anu and Banu computing their keys where K1 = K2.

Anu	Banu
$K1=B^X \bmod n$	$K2=A^Y \bmod n$
$=8^3 \bmod 11$	$=2^9 \bmod 11$
$=512 \bmod 11$	$=512 \bmod 11$
=6	=6

Man-In-The-Middle Attack (MIMA)

In cryptography and computer security, a man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.

For example,

1. Anu wants to communicate with Banu privately. Then Anu chooses and sends the values of ***n*** and ***g*** to Banu. Let ***n=11 and g=7***.

2. Anu does not realize Cnu that Cnu is listening quietly to the conversation. Cnu simply note down the values of ***n*** and ***g***, and also forwards them to Banu as they originally were.

Anu	Cnu	Banu
n=11, g=7	n=11, g=7	n=11, g=7

3. Now Anu, Cnu and Banu select random numbers say ***x*** and ***y***.

Anu	Cnu	Banu
x=3	x=8,y=6	y=9

4. Anu calculates her **A** value and Banu calculates his **B** value whereas Cnu calculates both **A** and **B** values at his side to play the role of *man in middle*.

Anu	Cnu	Banu
$A=g^x \bmod n$	$A=g^x \bmod n$	$B=g^y \bmod n$
$=7^3 \bmod 11$	$=7^8 \bmod 11$	$=7^9 \bmod 11$
$=343 \bmod 11$	$= 5764801 \bmod 11$	$= 40353607 \bmod 11$
=2	**= 9**	**=8**
	$B=g^Y \bmod n$	
	$= 7^6 \bmod 11$	
	$= 117649 \bmod 11$	
	= 4	

5. Anu sends her **A** value to Banu. Cnu intercepts it and send his **A** value 9 instead of Anu's **A** values to him. In response, Banu sends his B value 8 to Anu but Cnu intercepts it and sends his B value 4 to Anu.

Based on these values, all the three persons now calculate their keys.

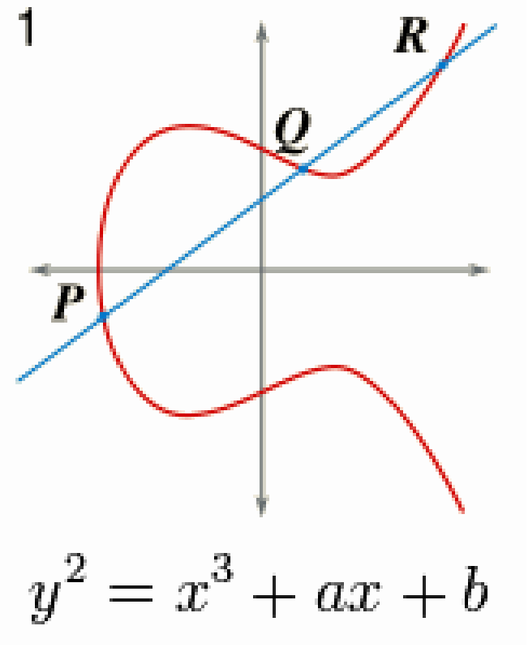
Anu	Cnu	Banu
$K1=B^x$ mod n	$K1=B^Y$ mod n	$K2=A^Y$ mod n
$=4^3$ mod 11	$=8^8$ mod 11	$=9^9$ mod 11
=64 mod 11	= 16777216 mod 11	=387420489 mod 11
=9	=5	=5
	$K2=A^Y$ mod n	
	$=2^6$ mod 11	
	=64 mod 11	
	=9	

Elliptic Curve Cryptographic algorithm (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.
- Elliptical curve cryptography is a public key encryption technique which is based on the theory ofelliptical curves.
- This encryption technique uses the properties of elliptic curve in order to generate keys instead of using the traditional methodology of generation of keys using the product of two very large prime numbers.
- ECC is a public key cryptosystem which is used to generate the public key and the private key in order to encrypt and decrypt the data.
- It is based on the mathematical complexity of solving the elliptic curve discrete logarithm problem which deals with the problem of calculating the number of steps or hops it takes to move from one point to another point on the elliptic curve.
- Elliptic curves are the binary curves and are symmetrical over x- axis. These are defined by the function:

$$y^2 = x^3 + ax + b$$

- Where x and y are the standard variables that define the function while as a and b are the constant coefficients that define the curve.
- As the values of a and b change, elliptical curve also alters.
- For elliptical curves, the discriminant is non zero.
- The operations used on elliptical curves in cryptography arepoint addition, point multiplication and point doubling.
- The important characteristic of elliptic curve is the finite fieldconcept.
- This means that there is a way to limit the values on the curve.



Few terms that will be used,

E -> Elliptic Curve
P -> Point on the curve
n -> Maximum limit (This should be a prime number)

Key Generation

- Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key andthe receiver will decrypt its private key.

- Now, we have to select a number**'d'** within the range of **'n'**.
- Using the following equation we can generate the public key

- Q = d*P
- **d** = The random number that we have selected within the range of (**1 to n-1**). **P** isthe point on the curve.
- ‘ Q’ is the public key and ‘ d’ is the private key.

Encryption

- Let ‘M’ be the message that we are sending. We have to represent this message onthe curve.
- Randomly select ‘k’ from [1 – (n-1)].
- Two cipher texts will be generated let it be **C1** and **C2**.

C1 =
k*P C2 =
M + k*Q

C1 and C2 will be send.

Decryption

- We have to get back the message ‘m’ that was send to us,
- M = C2 – d*C1
- M is the original message that we have send.

Proof

How do we get back the message,

- C2 – d * C1 = (M + k * Q) – d * (k * P) (C2 = M + k * Q and C1 = k * P)
- = M + k * **d** * **P** – d * k *P (Since Q =d * p)
- = M (Original Message)