Asymmetric Encryption

Uses publicprivate key pair

Each user generates a pair of public and private keys

Public Key is known to everyone and is used to encrypt data

Private Key is only known to the key owner and used for decryption

Used in 3 categories

–Encryption/Decryption (provide secrecy)

–Digital signatures (provide authentication)

–Key exchange (of session keys)

Diffie-Hellman key exchange

First public-key type scheme

Proposed by Diffie & Hellman in 1976

A practical method for public exchange of a secret key

Cannot be used to exchange an arbitrary message

Security relies on the difficulty of computing discrete logarithms

Diffie-Hellman algorithm

q	prime number
α	α<q ,α primitive root of q
User A	
Select PR=Xa	Xa<q
Calculate PU=Ya	Ya=α^Xa mod q
User B	
Select PR=Xb	Xb<q
Calculate PU=Yb	Yb=α^Xb mod q
Secret key calculation	
User A	K=(Yb)^Xa mod q
User B	K=(Ya)^Xb mod q

Disadvantages

Cannot be used for asymmetric key exchanges

Man-in-the-Attack

ElGammal-Cryptosystem

Presented in 1984 by Tather Elgammal

Used for encrypting messages

Based on discrete lagarithmic problem

Disadvantages

Decryption is slow

Duplicates message length by factor of two during encryption

ElGammal algorithm

Select large prime	q
Select	p , p is primitive root of q
User A	
Choose private key	Xa , 1 < Xa < q-1
Compute public key	Ya=p^Xa mod q
Similarly User B calculates Xb and Yb	
Encryption from A	
Message M	0<=M<=q-1
Choose k	1<=k<=q-1
Compute	K=Ya^k mod q
Compute	C1=p^k mod q
Compute	C2=KM mod q
---Ciphertext(C1,C2)	
Decryption from B	
Recover key	K=C1^Xa mod q
Compute message	M=C2*K-1 mod q

RSA

Uses large integers (eg.1024 bits)

RSA key generation

Select two large primes p and q	p not equal to q
Calculate	n<--p*q
Calculate	O(n)<--(p-1)*(q-1)
Select e	1<e<O(n) and e is coprime to O(n)
Calculate	d<--e^-1mod O(n)
Public key	PU={e,n}
Private key	PR={d,n}
Encryption	
Plaintext	M<n
Ciphertext	C=M^e(mod n)
Decryption	
Plaintext	C
Ciphertext	M=C^d(mod n)

Key Distribution Techniques

Means of delivering key to two parties who wish to communicate

For symmetric encryption to work,two parties must exchange the same key

Public-key cryptosystems are mostly used to encrypt secret keys

Frequent key exchanges are desirable to limit the amount of data compromised

The strength of any cryptographic system relys on key distribution technique

Advantages and Disadvantages

Hard to crack since it involves factorization of prime numbers

Can be very slow in cases where large data needs to be encrypted

Man-in-the-Middle attack

Key Distribution Models

Model 1

A->B	PUa||IDa
B->A	E(PUa,ks)

-Ensures confidentiality but not authentic-ation

-Vulnerable to man-in-the-middle attack

Model 2

A->B	E(PUb,[N1||IDa])
B->A	E(PUa,[N1||N2])
A->B	E(PUa,N2)
A->B	E(PUb,E(PRa,Ks))

-ensures both confidentiality and authentic-ation

Distribution of public keys:

Public announcement

Feeding in a Publicly available directory

-Both vulnerable to forgery(anyone can claim to be someone)

Public Key Authority

-A trusted third party(KDC)

-Provides session keys to users who wish to communicate

-Requires users to be registered

-Just like a directory composed of users public key

-User interacts with the directory to obtain any desired public key securely

Interaction Model:

A->auth	Request|T1
auth->A	Epr_auth[KPU_b|Request|T1]
A->B	Epu_b[IDA|N1]
B->auth	Request|T2
auth->B	Epr_auth[KPU_a|Request|T2]
B->A	Epu_a[N1|N2]
A->B	Epu_b[N2]

Public-Key Certificates:

Certificates allow key exchange without realtime access to Public-Key Authority

A certificate binds user identity to public key

Certificate contains all necessary details appended by its hash

Helps user claim accountability for a Key

X_509 Certificates:

Issued by a Certification Authority (CA)

Part of CCITT X.500 directory service standards

Defines the framework for authentication

Uses public-key crypto & digital signatures

X.509 certificates are widely used and has 3 versions

Each version with information extended

Certificate contains information such as Public Key, Digital Signature , Issuer, Version, Serial Number, Time Stamp.

X.509 Version 3:

Has been recognised that additional inform-ation is needed in a certificate

-email/URL, policy details, constraints

Rather than explicitly naming new fields defined a general extension model

-Identifier, Criticality Indicator, Value

Hash Functions

Accepts variable length input M and produces fixed-size hash h

h = H(M)

Principal object is data integrity

It is infeasible to find object

- With pre-specified hash **(One-Way Property)**

-Two objects mapping to same hash**(Co-llision -Free Property)**

Message Authentication Code(MAC):

Also known as a keyed hash function

Concerned with integrity and authentication

Calculates hash from the message and encrypts with the secret key->Cryptographic Checksum or MAC or Tag

-Then is appended to the message

-The reciever calcutates the hash of message and compares

-Same hash value confirms that the message came from the stated sender (its authenticity) and has not been changed.

Digital Signatures :

Concerned with integrity , authentication, Non-repudiation

Operation is similar to that of the MAC

Model-1:

Instead the hash value of a message is encrypted with a user's private key

Anyone who knows the user's public key can verify the integrity of the message

An attacker who wishes to alter the message would need to know the user's private key

-This provides authentication.

Model 2 :

Once the encrypted hash of the message is calulated and appended to original message

Once again it is encrypted with the symmetric secret key

Reciever decrypts it with the symmetric key then public key

Then calculates hash of message and compares

This ensures confidentiality as well as authentication