




## Network Security - Basics



## Goals

- Confidentiality
- Authentication
- Integrity
- Non-repudiation
- Access Control
- Availability



### • Confidentiality

- Information in system or transmitted over network should only be accessible to authorized users
  - Encryption/decryption
- Ideally should protect even the existence of data
  - Unauthorized users should not even know about the existence of a message or its properties (sender, receiver, frequency etc.)



### • Authentication

- Should be able to verify that an user is indeed who it claims to be
  - User = a human being, a program accessing a service, two communicating parties...
- ### • Integrity
- Messages are received as sent, with no modification, duplication, insertion, reordering, replays



- Non-repudiation
  - Sender or receiver cannot deny their role in a communication
- Access Control
  - Limit and control access to system services and applications
- Availability
  - Services must be available to authorized users



## Cryptography Basics

- Encryption – convert a *plaintext* message to *ciphertext*
  - Applies a parameterized function (an algorithm and a key) over the plaintext
- Decryption – convert a *ciphertext* back to its original *plaintext* form
  - Applies another parameterized function over the ciphertext that gives the plaintext back
- Parameters used in the function – *Key*
- Key used in encryption may or may not be the same as the one used for decryption
- Decryption should be very very difficult (computationally intensive) in general – unless you have the secret information *key*



## Private Key Cryptography

- Sender and receiver share the same key
- Sender encrypts message using key
- Receiver decrypts message using the same key
- The key must be kept secret – how to share the key in the first place? (*key management problem*)
- Also called symmetric cryptosystem
- Typical key length –128 bit
- Fast, simple
- Example: DES (Data Encryption Standard), AES etc.



## Public Key Cryptography

- Pair of keys for each user – private key known only to the user, and a public key known to everyone
- To send to X, encrypt using X's public key and send
- X will decrypt using its private key (known to it)
- Typical key length = 512 bits, 1024 bits etc.
- No problem of sharing keys
- Stronger security than private key systems
- Slower than private cryptosystems
- Ex. – RSA etc.



## Message Authentication Code (MAC)

- Sender and receiver share a secret key
- Apply a function and the key on a message to get a fixed length value (*authenticator*)
- Send authenticator with message
- Receiver applies same function and key on message and matches the MAC. If match, no change in message
- Actual message may or may not be encrypted depending on need of confidentiality



## Hash Functions

- Applies a function on message to produce fixed size *message digest*
- Similar to MAC, but no secret key
- One-way hash functions
  - $F(x) = h$  easy to calculate, but computationally infeasible to compute  $x$  given  $h$
- Hash functions can be used in different ways
- Ex. MD5, SHA-1



## Digital Signature

- Used to protect sender-receiver from each other!
- Sender encrypts hash code of the message using his own private key (can encrypt whole message, but hardly used)
- Receiver can decrypt it using sender's public key, and verify the hash code
- If same, receiver is sure that the sender sent it and the message is not modified
- But receiver cannot modify the message and forge a signature since it does not know sender's private key
- Message + digital signature can be further encrypted using receiver's public key or shared private key if confidentiality needed



Think how we can use one or more of the techniques (encryption/decryption, MAC, hash functions, digital signatures) in combination to achieve some of the goals mentioned



## Digital Certificates

- An electronic id used to prove that the user/sender is who it claims to be
- Certificates are given by Certificate Authorities (CA, ex. Verisign) after extensive identification check
- Most common format – CCITT X.509
- Certificate contains, among other things
  - Owner's name
  - Owner's public key
  - Issuer's id
  - Issuer's digital signature (verifies that the certificate came from the issuer and is not modified)
  - Expiry
  - Serial no.



- To authenticate himself, user sends his certificate along with message
- Receiver does the following
  - Retrieves CA's public key and decrypts the digital signature to find the hash
  - Computes the hash of the certificate to make sure the certificate is authentic and issued by a trusted CA
  - User is authenticated since it has been issued a Certificate by a trusted CA



### • Common uses

- Certificate of web server sent to client browser to authenticate server to client (ex., secure transactions over web)
- Code-signing certificates – certificates sent with code to indicate that the developer of the code has been identified to be genuine by a trusted CA
- Go to <http://www.verisign.com.au/repository/> to learn more about certificates in action (follow Tutorials links)
- Question – what CAs to trust?