Week 10 Lecture 2

Network Security

## Authentication

sender, receiver want to confirm identity of each other. In many applications, both end points may need to authenticate each other.

We need to confirm that the user is actually online to avoid playback attack. To achieve this, we use nonce

Diagram

Description automatically generated

However, the assumption here is we assume two parties have the shared symmetric key, but we are talking about two parties trying to authenticate each other, so the process becomes tricky.

Diagram

Description automatically generated

However, this results in a security hole called man/woman/person in the middle

Diagram

Description automatically generated

We come back and solve this problem later

## Message integrity

sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

* Confidentiality: message private and secret
* Integrity: protection against message tampering, encryption alone may not guarantee integrity

**Digital signatures**

Diagram

Description automatically generated

1. Graphical user interface, text

   Description automatically generatedsuppose Alice receives msg m, with signature: m, KB­-(m)
2. Alice verifies m signed by Bob by applying Bob’s public key KB to KB(m) then checks, KB+ (KB­-(m)) = m.
3. If it is m, whoever signed m must have used Bob’ s private key

**Putting everything together:**

m is sent to hash function, we get the hash, then we sign the hash H(m). The sender applies private key to the hash and generate his digital signature (Signed message digest). Then the message is sent over the network. The receiver verifies signature, integrity of digitally signed message. If H(m) don’t match, the receiver knows it is tampered.

Diagram

Description automatically generated

## Certification authorities

binds public key to particular entity, this is used to solve the playback attack. The CA does so by adding a digital signature (signed by the CA's private key) to the public key of the user. Others can verify the certificate by applying the CA's public key.

Diagram

Description automatically generated

Suppose a CA creates Bob’s certificate, which binds Bob’s public key to Bob. This certificate is signed with Bob’s private key. when Alice wants Bob’s public key: she gets Bob’s certificate (Bob or elsewhere), and apply CA’s public key to Bob’s certificate, get Bob’ s public key

Diagram

Description automatically generated

**Public-Key Infrastructure (PKI)**

* Certificates and certification authorities
* Often considered “heavy

## Securing Email

**Alice wants to send confidential e-mail, m, to Bob**.

Diagram, schematic

Description automatically generated

1. Alice encrypts the message with symmetric key KS, but recall KS is randomly generated. She needs to convey the symmetric key to Bob. She does this by encrypting it with Bob’s public key. So this message has two parts, the Ks encrypted ciphertext and public key encrypted symmetric key.

2. Bob first needs to extract the key first, so he applies his private key to get the symmetric key, then use the symmetric key to decrypt the message.

**Alice wants to provide sender authentication, message integrity**

1. Alice digitally signs message, sends both message (in the clear) and digital signature. Bob takes the Alice private key encrypted hash function text, applies Alice’s public key which is known to everyone, gets the hash functioned message.

2. Bob takes a message and applies hash function.

3. He compares the hash functions he gets in 1 and 2

Diagram

Description automatically generated

**Alice wants to provide confidentiality, sender authentication, message integrity (both).**

Alice uses three keys: her private key, Bob’s public key, newly created symmetric key

Diagram, schematic

Description automatically generated