Security

Introduction

THIS DOCUMENT COVERS

- Introduction

One Way Function

A function with the property that while it is easy to calculate $f(x)$ given x, it is impossible to calculate x given $f(x)$

Cryptography

Cryptographic algorithms are either symmetric or public key (asymmetric). With symmetric key cryptography the same key is used for encryption and decryption. With public key cryptography there are two keys: the public key and the private key. One is used for encryption and the other is use for decryption. In some algorithms, such as RSA, the public or private key can be used for encryption.

Public Key Cryptography

Secure Communication

Alice wants to send a secure message to Bob. This can be achieved using public key cryptography (asymmetric cryptography).

1. Bob sends Alice his public key p^B
2. Alice Encrypt the message using the public key $c = E(m, p^B)$
3. Alice sends the encrypted message c to Bob
4. Bob decrypts the message using his secret key $m = D(c\,, s^B)$

Hybrid

Secure Communication

In practice, because public key cryptography is significantly slower than symmetric cryptography, public key cryptography is only used to exchange keys. Subsequent exchanges can then use symmetric cryptography. Such hybrid systems work as follows.

1. Bob sends Alice his public key p^B
2. Alice generates a random session key k and encrypts it using Bobs public key. $k' = E(k, p^B)$
3. Alice sends the encrypted key k' to Bob
4. Bob decrypts the session key using his private key $k = D(k', s^B)$
5. Both parties then communicate using symmetric cryptography and the session key.

Authentication

Digital Signatures

Alice takes the message she wants to send m and adds some extra information m (such as her name)

$$m' = m + x$$

Now Alice takes the resulting document and encrypts it with her private key giving

$$\sigma = E(m', s^A)$$

Alice sends a packet with both the encrypted and unencrypted versions to Bob.

$$(\sigma, m')$$

Bob uses Alice's public key to decrypt σ If it matches m' then it must have been sent by Alice. Furthermore, it cannot have been tampered with.

This system has the following properties.

- **Authentic:** If Alice's public key decrypts a message it must have come from Alice
- **Unforgeable:** Only Alice knows her private key
- **Unalterable:** If the signed document is tampered with it cannot be decrypted with Alice's public key.

> **NOTE:**
>
> Bob must be sure the public key he has is really Alice's public key

Certificates

A certificate is a data structure that contains a public key and a name. The data structure is then signed to bind the public key to the name. The entity that signs the certificate is know as the certificate authority or issuer.

X509 CERTIFICATES

Most certificates these days are X509 v3 certificates.

HASHING AND DIGITAL SIGNATURES

Public Key algorithms are inefficient for large documents. To save time, one-way hash functions are used. Alice signs a hash of the document, rather than the document itself.

1. Alice hashes the document $h = H(m)$
2. Alice encrypts the hash using her private key $h' = E(h, k_i^a)$
3. Alice sends the encrypted hash and the document to Bob
4. Bob decrypts the hash using Alice's public key $h = E(h', k_u^a)$
5. Bob hashes the document m and compares it with h

All digital signature algorithms are public key based. They use secret information to sign documents and public information to verify the signatures. We denote the signing process with private key K as

$$S(m, k_i)$$

And verifying with public key as

$$V(m, k_u)$$

The bit string that we add to the document when signed is called the digital signature The whole process that convinces the receiver of the identity of the sender is known as authentication.

Digital Signatures and Encryption

We can combine public key cryptography and digital signatures to carry out authentication and privacy

1. Alice signs the message using her private key $S(m, k_i^a)$

2. Alice encrypts the signed message using Bobs public key $E(S(m, k_i^a), k_u^b)$
3. Bob decrypts the encrypted signed message using his private key $D\left(E(S(m, k_i^a), k_u^b), k_i^b\right) = S(m, k_i^a)$
4. Bob verifies with Alice's public key $V(S(m, k_i^a), k_u^a) = m$

Protocols

KEY EXCHANGE

Assume we have a Key Distribution Center. Both Alice and Bob have a private key on the KDC and want to communicate

1. Alice tells KDC she wants to communicate with Bob
2. KDC generates a session key sends two copies to Alice; one encrypted with Alice's private key and the other encrypted with Bobs private key
3. Alice decrypts the session key encypted with her own private key and sends the one encrypted with bobs private key to Bob
4. Bob decrypts the session key encrypted with his private key
5. This communicate using the session keuy