

Today's Lecture

- ▶ Guest lecture tomorrow
- ▶ Encryption
- ▶ Public and Private keys
- ▶ RSA

Required Security Properties

There are four main properties required of any secure communication system:

- ▶ **Confidentiality** - message cannot be understood by anyone but the intended sender and receiver
- ▶ **Authentication** - confirm the identity of the other party
- ▶ **Integrity** - the message has not been altered
- ▶ **Availability** - resources are restricted to the designated users

Cryptography

Encryption is the process of disguising original message text so that it can only be deciphered by the intended recipient.

An unencrypted message is known as a **plain-text** message, while an encrypted message is a **cipher-text** message.

There are many ways of encrypting messages, for example:

- ▶ $A \rightarrow 1, B \rightarrow 2, \dots, Z \rightarrow 26$
- ▶ “Hello World” = “8 5 12 12 15 23 15 18 12 4”
- ▶ In order to decipher this, you need to know how it was encrypted!

Encryption Techniques

The method for encrypting a message is known as the **encryption method** or **cipher**. The secret to decoding it is known as the **key**. In the above example, the key would be $A = 1, \dots, Z = 26$. This might be shortened to $A = 1$.

The use is as follows:

- ▶ Alice encrypts using a key K_A as input to an encryption algorithm
- ▶ This produces an encrypted message $K_A(m)$
- ▶ Bob has a key K_B
- ▶ Bob receives the encrypted message and decrypts it by computing $K_B(K_A(m))$

Symmetric Cryptography

In **symmetric key cryptography** Alice and Bob use the same key. The example above uses symmetric key cryptography.

- ▶ Often use multiple ciphers
- ▶ **Monoalphabetic ciphers** - Caesar cipher, ROT-13
- ▶ Both users must know the key being used - how could this be agreed upon?

The **Data Encryption Standard** (DES) is a symmetric key encryption standard first published in 1977. It is relatively secure (a challenge posed in 1997 to crack a test message encrypted using DES took 4 months to solve).

<http://scrs.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
describes DES formally.

Symmetric cryptography

How would you tell the recipient what key you used to encrypt the message?

- ▶ Send the key in plain-text?
- ▶ Agree on a key beforehand?
- ▶ Send the key partitioned into multiple messages?

A **key distribution centre** (KDC) is often used as a trusted intermediary. If Bob and Alice both contact a KDC, it will establish a **different** shared key with both of them. Bob will send encrypted messages to the KDC which decrypts them using his shared key. It then encrypts them with Alice's shared key and sends them on to her.

Asymmetric cryptography methods can be used to avoid trusting messages to a KDC.

Asymmetric Cryptography

Otherwise known as **public key encryption**.

- ▶ Bob has **two** keys, a **public key** (K_B^+) and a **private key** (K_B^-).
- ▶ The public key is known to everyone, the private key is known only to Bob.
- ▶ Alice gets Bob's public key from his website
- ▶ Alice encrypts her message m using Bob's public key to get $K_B^+(m)$ and a standard encryption method
- ▶ Bob receives the message and decrypts the message by computing $K_B^-(K_B^+(m))$ using a standard encryption method

The success of this depends on the fact that Bob's private key and public key decrypt each other i.e. $K_B^-(K_B^+(m)) = m$

Asymmetric Cryptography

Standard encryption methods are described in RFC 1321, RFC 2437, RFC 2420.

The encryption/decryption methods must be chosen so that Bob's private key will actually decrypt messages encoded using his public key!

Mathematically, we require:

- ▶ $K_B^-(K_B^+(m)) = m$ and
- ▶ $K_B^+(K_B^-(m)) = m$

How do you know which encryption method to use? How do you choose your public and private keys so that this will work?

Integrity is also a problem.

Asymmetric cryptography

There are two interrelated concerns:

- ▶ How should you select the public key and the private key?
- ▶ What encryption and decryption algorithms should you use?

The **RSA** algorithm for encryption is the most common method. It was first described in 1977 and named after its inventors (Rivest, Shamir, Adleman) at MIT.

RSA

- ▶ Choose two large prime numbers p, q (e.g. $p = 5, q = 7$)
- ▶ Compute $n = pq$ and $z = (p - 1)(q - 1)$ (e.g. $n = 35, z = 24$)
- ▶ Choose $e < n$ where e and z are relatively prime (e.g. $e = 25$)
- ▶ Find d such that $e * d - 1$ is divisible by z (e.g. $d = 1$)
- ▶ The public key is then the pair n, e and the private key is the pair n, d

RSA continued

If Alice wants to send Bob a number m (the message), she does so as follows:

- ▶ Get Bob's public key n, e from his website
- ▶ Calculate $m^e \text{ MOD } n$ (modulo, or MOD, means the remainder when divided)
- ▶ Send Bob the encrypted message c where $c = m^e \text{ MOD } n$

Bob then calculates $c^d \text{ MOD } n$ to get m .

This relies on relatively tricky number theory.

RSA continued

Try this yourself!

To send “Hi”, you must convert it into a number m (in reality, the ASCII values can be used). You can use the numbers “8” (=‘H’) and “9” (=‘I’) instead.

RSA relies on the fact that it is very difficult to factor prime numbers quickly. This makes it hard to crack.

RSA as described has some drawbacks:

- ▶ **Integrity** - since everyone knows Bob’s public key, how does he know the message hasn’t been altered (MITM attack)
- ▶ **Authentication** - how does Bob know the message really came from Alice?

These can be guaranteed by using Alice’s private key and a **nonce**.

RSA Authentication

A **nonce** is a number used only once in a lifetime by a protocol.

- ▶ Bob sends Alice a **nonce** R .
- ▶ Alice encrypts R using her private key to get $K_A^-(R)$ and sends this to Bob.
- ▶ Bob decrypts this using Alice's public key K_A^+ (i.e. calculates $K_A^+(K_A^-(R))$).
- ▶ This lets Bob authenticate Alice - only Alice's public key and private key decrypt each other

RSA Integrity

Alice can also use a **Digital signature** to ensure integrity and authentication.

- ▶ Alice produces a **hash value** h of her message m
- ▶ Alice encrypts the hash value h using her private key to get $K_A^-(h)$, and sends this (along with her encrypted message) to Bob
- ▶ Bob gets Alice's public key K_A^+ from her website, and applies it to both the encrypted message and encrypted hash value he received
- ▶ Bob calculates the hash value h' of the decrypted message. If this matches the decrypted hash value he received, the message has not been tampered with.

General cryptography

Not every protocol uses encryption (**TELNET** does not, for example). Thus, it is possible for an eavesdropper to obtain a user's password. Other security concerns include the sharing of a public key - how might Alice be sure that she is really getting the key from Bob's website and not from a spoofed site?

We have already discussed some aspects of web security and network security. These provide a means for Bob to secure his website and hence his public key.

References:

- ▶ Kurose & Ross, Chapter 7

Coming Lectures:

- ▶ Web security