CS 372 Lecture #42

Security

- encryption

Note: Many of the lecture slides are based on presentations that accompany *Computer Networking: A Top Down Approach,* 6th edition, by Jim Kurose & Keith Ross, Addison-Wesley, 2013.

Password / Data encryption

- Messages are encoded by the sending protocol
- Encoded messages are decoded by the receiving protocol
- Many encryption algorithms (functions)
  - simple substitution to very complex computations
  - most use *mod* with large prime numbers

- *Private key* encryption
- *Public key* encryption

Private key encryption

- Only sender and receiver have the key and the encrypt/decrypt algorithms
- (sender) For message *M*, with key *K*, the encrypted message *E* is

  E = *encrypt* (K, M)
- (receiver) For encrypted message *E*, the original message is produced by the inverse of *encrypt*

  M = *decrypt* (K, E)

- Many algorithms
- Separate key for each correspondent
- Easy to change key
- Difficult to ensure confidentiality of key

Private key encryption example

- Both sender and receiver have key

Example key (K): `abcdefghijklmnopqrstuvwxyz`
`bdfhjlnprtvxzacegikmoqsuwy`

- Sender:

  M = secret sends E = *encrypt*(K, M) = kjfijm

- Receiver:

  receives E = kjfijm

  decodes to get M = *decrypt*(K, E) = secret

Public key encryption

- Each user has two keys and the encrypt/decrypt algorithms
  - one *public* key, one *private* key
- (sender) For message *M*, with the destination user's public key *Kpublic*, the encrypted message *E* is

  E = *encrypt* (Kpublic, M)
- (receiver) For a message *E* (encrypted with the destination user's public key) the original message can be produced only by the destination user's private key *Kprivate*

  M = *decrypt* (Kprivate, E)

- Easy to change key
- Easy to ensure confidentiality of private key

Public key encryption example (RSA*)

- Kpublic = <3, 187>
- Kprivate = <107, 187>
- Message = 25
- E = *encrypt*(Kpublic, Message)

  = Message^3 mod 187

  = 25^3 mod 187 = 104
- M = *decrypt*(Kprivate, E)

  = E^{107} mod 187

  = 104^{107} mod 187 = 25 = Message

- *RSA: Rivest, Shamir, Adleman algorithm

RSA: Choosing keys

1. Choose two large prime numbers $p, q.$ (e.g., 1024 bits each)

2. Compute $n = pq$, $z = (p-1)(q-1)$

3. Choose e $(e<n)$ such that e has no common factors with z. (e, z are "relatively prime").

4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).

5. *Public* key is $\underbrace{(e,n)}_{K_B^+}$. *Private* key is $\underbrace{(d,n)}_{K_B^-}$.

RSA: Encryption, decryption

Given (e,n) and (d,n) as computed above

1. To encrypt bit pattern, m, compute

   $c = m^e \bmod n$

2. To decrypt received bit pattern, c, compute

   $m = c^d \bmod n$

Magic happens!

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

OSU Oregon State UNIVERSITY Extended Campus

Another RSA example:

Let *p=5, q=7* Then *n=35, z=24*

Choose *e=5* (so *e, z* relatively prime)

d=29 (so *ed-1* = 144 is exactly divisible by z)

Suppose message m = 12

encrypt:	m	m^e	$c = m^e \bmod n$
	12	248832	17
decrypt:	c	c^d	$m = c^d \bmod n$
	17	481968572106750915091411825223071697	12

RSA: Why $m = (m^e \bmod n)^d \bmod n$?

Useful result from number theory

If p,q prime and $n = pq$, then:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(using number theory result above)

$$= m^1 \bmod n$$

(since we <u>chose</u> ed to be divisible by $(p-1)(q-1)$ with remainder 1)

$$= m$$

RSA: another important property

$$\underbrace{K_B^-(K_B^+(m))}_{} = m = \underbrace{K_B^+(K_B^-(m))}_{}$$

apply public key first, then apply private key

Apply private key first, then apply public key

Result is the same!

Summary Lecture #42

- “Security” must be defined by an organization
  - Determine value of information and define a security policy
  - Aspects to consider include
    - privacy
    - data integrity
    - availability
    - confidentiality
- Mechanisms to provide aspects of security
  - Firewalls: packet filtering
  - Encryption: private and public key cryptosystems
  - Virtual private networks
  - etc.