Security

- Introduction
- Cryptography
- Authentication
- Message integrity and confidentiality
- Key distribution
- Security in practice

Reference:

Network Security

- Intruder may
  - Eavesdrop
  - Remove messages
  - Modify messages
  - Insert, replay messages
- Security threats
  - Interception, Interruption, Modification, Fabrication
Security Mechanisms

- **Encryption**: ensure secrecy of info being transmitted
- **Authentication**: prove who you are and have correspondent prove his/her/its identity
- **Authorization**: verify you have rights to perform requested action
- **Auditing**: log actions and do post-facto analysis (forensics)

Encryption

- plaintext: unencrypted message
- ciphertext: encrypted form of message
A Simple Encryption Algorithm

Substitution cipher:

Abcdefghijklmnopqrstuvwxyz
poiuytrewqasdfghjklmnbvcxz

• Replace each plaintext character in message with matching ciphertext character:

plaintext: Charlotte, my dear
ciphertext: iepkgmy, dz uypk

A Simple Encryption Algorithm

• key is pairing between plaintext characters and ciphertext characters
• Symmetric key: sender and receiver use same key
• 26! (approx 10^26) different possible keys: unlikely to be broken by random trials
• Substitution cipher subject to decryption using observed frequency of letters
  – ‘e’ and ‘t’ are the most frequently occurring letters
  – Particular 2 letters and 3 letters appear quite often together: ‘in’, ‘it’, ‘the’, ‘ion’, ‘ing’
DES: Data Encryption Standard

- Encryption/decryption algorithm is a published standard
- The same key is used to encrypt and decrypt a message
  - DES is used for **symmetric cryptosystems**
- Encryption: operate on 64-bit chunks, 56-bit key
  - A chunk is transformed into an encrypted chunk of output in 16 rounds, plus an initial and final permutation
  - Each round uses a different 48-bit subkey derived from the 56-bit key
- Decryption is similar to encryption
  - The subkeys are applied in the reverse order when decrypting.
- Replacements: 3DES and AES
  - A machine that could crack 56-bit DES in one second would take 149 trillion years to crack a 128-bit AES key!
Key Distribution Problem

- Problem: how do communicant agree on symmetric key?
- Trusted agent distribution
  - Keys distributed by centralized trusted agent
  - Any communicant need only know key to communicate with trusted agent
  - For communication between i and j, trusted agent will provide a key
- We will cover in more detail shortly
Public-Key Systems

• Separate encryption/decryption keys
  – Receiver makes known its encryption key
  – Receiver keeps its decryption key secret
• To send to B, encrypt message M using B’s publicly available key, EB
  – Send EB(M)
• To decrypt, B applies its private decryption key DB to received message
  – Computing DB(EB(M)) gives M

RSA: Public-Key Encryption/Decryption

RSA is used for public-key systems

Entity wanting to receive encrypted messages:
• Choose two very large prime numbers, $p$ and $q$
• Compute $n=pq$ and $z=(p-1)(q-1)$
• Choose number $d$ which has no common factors with $z$
• Compute $e$ such that $ed = 1 \mod z$, i.e., $ed = kz+1$
• Three numbers:
  – $e, n$ made public
  – $d$ kept secret
RSA (continued)

To encrypt:
• Divide message into blocks, \(\{b_i\}\) of size \(j: 2^j < n\)
• Encrypt: \(c_i = b_i^e \mod n\)

To decrypt:
• \(b_i = c_i^d \mod n\)

To break RSA:
• Need to know \(p, q\), given \(pq=n\) (\(n\) known)
• Factoring 200 digit \(n\) into primes takes 4 billion years using known methods

RSA Example

• Choose \(p=3, q=11\), gives \(n=33\), \((p-1)(q-1)=20\)
• Choose \(d = 7\) since \(7\ and\ 20\) have no common factors
• Compute \(e = 3\), so that \(ed = kz+1\) (note: \(k=1\ here\))
• \(b=2: c=2^3 \mod 33=8, b=8^7 \mod 33 = 2\)
Why Does RSA Work?

- Crucial number theory result: if \( p, q \) prime then
  \[ b^{((p-1)(q-1)) \text{ mod } pq} = 1 \]
- Using mod \( pq \) arithmetic:
  \[(b^e)^d = b^{ed}\]
  \[= b^{k(p-1)(q-1)+1} \text{ for some } k \]
  \[= b \ b^{(p-1)(q-1)} \ b^{(p-1)(q-1)} \ ... \ b^{(p-1)(q-1)} \]
  \[= b 1 1 ... 1 \]
  \[= b \]
- **Note:** we can also encrypt with \( d \) and decrypt with \( e \).
  - This will be useful shortly

How to Break RSA?

Brute force: get B's public key

- For each possible \( b_i \) in plaintext, compute \( b_i^e \)
- For each observed \( b_i^e \), we then know \( b_i \)
- Moral: choose size of \( b_i \) "big enough"

![Diagram showing the process of RSA encryption and decryption.](image)
Breaking RSA

- Man-in-the-middle attack: intercept keys, spoof identity

![Diagram showing the man-in-the-middle attack in RSA encryption]