COMP232 - Cyber Security

Week 1

Goals of Network Security

Confidentiality, Integrity, and Availability

- Security Architecture for OSI
  - ITU-T Recommendation X.800
    - Concepts
      - **Security attack** - any actions compromising security of information
        - **Passive** - make use of data but does not affect system resources
        - **Active** - attempts to alter system resources or alter operation
        - Attacks
          - Interruption -availability
          - Interception - confidentiality
          - Modification - integrity
          - Fabrication - authenticity
      - **Security mechanism** - a mechanism to detect, prevent or recover from a security attack
        - Used to implement security services, including

        - Encipherment or encryption
        - Digital signature
        - Access control
        - Data integrity
        - Authentication exchange
        - Traffic padding
        - Routing control
        - Notarisation
    - **Security service** - a service that enhances security of the data processing and transfer
        - Categories
            - Authentication
            - Access Control
            - Data confidentiality
            - Data integrity
            - Non-repudiation
            - Availability

Week 2

- **Identification -** Associating identity with a subject (who are you)
- **Authentication**- Establishing validity of identity (are you who you claim you are)
- **Authorisation -** Associating rights with a subject (what can you do)

Authentication

- Password-based authentication
  - Based on what you know
- Token-based authentication
  - Based on what you have
- Biometrics-based authentication
  - Based on what you are

Cryptography

- Two types
  - Symmetric key - Same key used to decrypt and encrypt message
  - Asymmetric key - One key used to encrypt and one to decrypt
- Types of operations used
  - Substitutions - Each element of text is mapped to another
  - Transposition - Rearrange elements in text
- Way plaintext is processed
  - Block cipher - input block to be transformed at once
  - Stream cipher - continuous processing of input
- **Cryptanalysis -** The process of attempting to discover plaintext or key
- Feistel cipher - most blocks algorithms have similar structure to it (symmetric key)
  - Input is divided into blocks of even number of elements
  - Multiple stages of substitutions and transpositions applied to it with different keys (derived from master key)
- Symmetric encryption algorithms
  - DES

        - Block - 64 bits
        - Key - 56 bits
        - Rounds - 16
        - Sub keys - 16
        - Only way to bypass is by brute force, but relatively easy to due to key size
    - 3DES
        - DES but done three times, with different keys each time
    - AES
        - Blocks of 128 bits
        - Every round
            - Bytes substituted
            - Rows shifted
            - Columns mixed
            - Each byte combined with round key

Week 3

Symmetric encryption

- Electronic codebook mode (ECB)
    - Each block is encrypted with the same key
- Cipher block chaining (CBC)
    - A block of the plaintext of the current block is XOR'ed with the ciphertext of the previous block

    - Each cipher is unique
    - First block encrypted by initialisation vector
- Cipher feedback mode (CFB)
    - Transform a block cipher to stream cipher

 Key distribution is tricky, since it needs to be secure also

Asymmetric encryption

- Public and private key
- Public key used for encryption and private for decryption
- Private key can also be used as a mean of digital signature
- More computationally expensive than symmetric key
    - Normally symmetric key encrypted by public key and used for exchange of messages (solve key distribution problem)

Week 4

RSA (Asymmetric encryption)

- **Encryption:** A message M (represented as a number) is encrypted as

$$C = M^e \ mod \ n$$

- **Decryption:** To retrieve M, compute

$$M' = C^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$$

- Key generation
  - Select two prime numbers p and q
  - Calc n = p * q
  - Calc phi(n) = (p-1)*(q-1)
  - Select e less that phi of n and relatively prime with it
  - Calculate d s.t d*e = 1mod(phi(n))
  - Public key = {e, n}, Private key = {d, n}
- How to break
  - Brute force (Too long)
  - Try to find p and q given n
    - N should be a number very hard to factorize
  - Common factors attack
    - Due to lack of good random number generators, some good amount of keys have common factors

Diffie-Hellman key exchange

- Most known algo for key exchange

- Users generate secret key based on public + private info
- Method
  - Two publicly known numbers:
    prime number q
    primitive root α of q
  - Let A and B wish to exchange a key, then they do the following:

    A selects a random integer $X_A < q$ and keeps it in secret
    B selects a random integer $X_B < q$ and keeps it in secret

    A computes $Y_A = \alpha^{X_A} \; mod \; q$ and sends it to B
    B computes $Y_B = \alpha^{X_B} \; mod \; q$ and sends it to A

- Both know can calculate common secret key

$$\text{A calculates } K = (Y_B)^{X_A} \; mod \; q$$
$$\text{B calculates } K = (Y_A)^{X_B} \; mod \; q$$

Neural Key Exchange Protocol

- Use synchronization of neural networks instead of traditional math
- Method
  - Both parties start neural networks
  - They process shared inputs to generate outputs
  - Match outputs and update weight until identical weights

- Potential resilience to quantum attacks and efficient in resource-constrained devices, but some versions are vulnerable to certain attacks and still in research phase

MAC

- Authentication code generated from a common secret key to prove identity of sender
- Does not need to be reversible (less vulnerable)

One-way hash functions

- Alternative method for message authentication
- Don’t use secret key
- Easy to first compute, but reverse is very hard to compute
- Hash function (H) properties for message authentication
    - Can be applied to any data size
    - Fixed-length outptut
    - Easy to compute
    - Infeasible to compute reverse
    - Hard to try to match the result by matching output from a known input (Weak collision resistance)
    - Hard to find a pair of inputs that give same output (strong collision resistance)
- SHA-1 (Secure Hash Algorithm)
    - Take input with a max length
    - Process input in 512-bit blocks
    - Each bit of output uses all bits of input

- Deprecated, now SHA-2 or 3 is used.