# Cryptography and Network Security

Bhaskaran Raman

Department of CSE, IIT Kanpur

**Reference:** Whitfield Diffie and Martin E. Hellman, "Privacy and Authentication: An Introduction to Cryptography", in Proc. IEEE, vol. 67, no.3, pp. 397 - 427, 1979

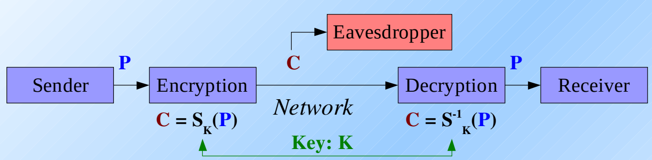


# Cryptography Fundamentals

- Privacy versus Authentication:
  - Privacy: preventing third party from snooping
  - Authentication: preventing impostering
- Two kinds of authentication:
  - Guarantee that no third party has modified data
  - Receiver can prove that only the sender originated the data
    - Digital Signature
    - E.g., for electronic transactions



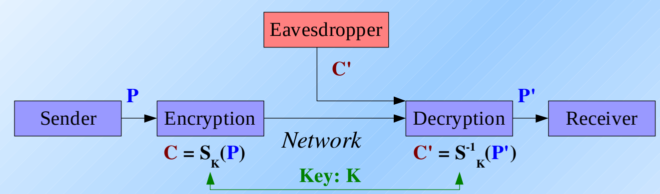
# Cryptographic Privacy




- Encrypt before sending, decrypt on receiving
  - Terms: plain text and cipher text
- Two components: key, and the algorithm
  - Should algorithm be secret?
    - Yes, for military systems; no, for commercial systems
- Key distribution must be secure



# Cryptographic Authentication




- The same system can also be used for authentication

## Cryptanalysis

- Cryptanalysis: attacker tries to break the system
  - E.g., by guessing the plain text for a given cipher text
  - Or, by guessing the cipher text for some plain text
- Possible attacks:
  - Cipher-text only attack
  - Known plain-text attack
  - Chosen plain-text attack
  - Chosen text attack



## Security Guarantees

- Two possibilities:
  - Unconditional
  - Computational security
- Unconditional security: an example
  - One-time tape
- Most systems have computational security
  - How much security to have?
  - Depends on cost-benefit analysis for attacker



## Public-Key Systems

- Shared-key ==> difficulties in key distribution
  - C(n,2) = O(n^2) keys
- Public key system
  - Public component and a private component
  - Two kinds:
    - Public key distribution: establish shared key first
    - Public key cryptography: use public/private keys in encryption/decryption
  - Public key cryptography can also be used for digital signatures



## Some Example Systems

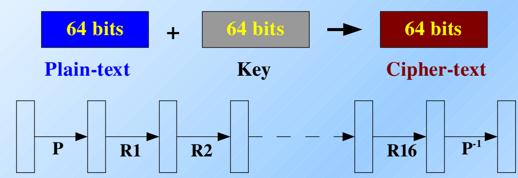
- Permuted alphabet (common puzzle)
  - Can be attacked using frequency analysis, patterns, digrams, trigrams
  - Attack becomes difficult if alphabet size is large
- Transposition
- Poly-alphabetic: periodic or running key
- Codes versus ciphering
  - Codes are stronger, and also achieve data compression

# Some Popular Systems

- Private key systems:
  - DES, 3DES
- Public key systems:
  - RSA: based on difficulty of factoring
  - Galois-Field (GF) system: based on difficulty of finding logarithm
  - Based on knapsack problem

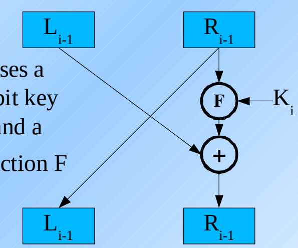


# Digital Encryption Standard (DES)




Permutation, 16 rounds of identical operation, inverse permutation

Each round uses a different 48-bit key $K_i$ (from K) and a combiner function F






# Triple-DES (3DES)

- DES can be broken with 2^55 tries:
  - 4500 years on an Alpha workstation
  - But only 6 months with 9000 Alphas
- Triple-DES:
  - Use DES thrice, with 3 separate keys, or with two keys (K1 first, then K2, then K1 again)



# Rivest, Shamir, Adleman (RSA) Public-Key Crypto-System

- Based on the fact that finding large (e.g. 100 digit) prime numbers is easy, but factoring the product of two such numbers *appears* computationally infeasible
- Choose very large prime numbers P and Q
  - N = P x Q
  - N is public; P, Q are secret
- Euler totient: Phi(N) = (P-1)(Q-1) = Number of integers less than N & relatively prime to N

## RSA (continued)

- Next, choose E in [2, Phi(N)-1], E is public
- A message is represented as a sequence M1, M2, M3..., where each M in [0, N-1]
- Encryption: $C = M^E \bmod N$
- Using the secret Phi(N), A can compute D such that ED = 1 mod Phi(N)
- ED = k x Phi(N) + 1
- Then, for any X < N, $X^{k \times Phi(N)+1} = X \bmod N$



## RSA (Continued)

- Decryption: $C^D = M^{ED} = M^{k \times Phi(N)+1} = M \bmod N$
- Example: Choose P = 17, Q = 31
  - N = 527, Phi(N) = 480
  - Choose E = 7, then D = 343
  - If M = 2, Encryption: C = 128
  - Decryption: $D = C^D \bmod N = 128^{343} \bmod 527 = 2$



## Taxonomy of Ciphers

- **Block ciphers:** divide plain text into blocks and encrypt each independently
- Properties required:
  - No bit of plain text should appear directly in cipher text
  - Changing even one bit in plain text should result in huge (50%) change in cipher text
  - Exact opposite of properties required for systematic error correction codes
- **Stream cipher:** encryption depends on current state



## Key Management

- Keys need to be generated periodically
  - New users
  - Some keys may be compromised
- Addressing the O(n^2) problem with key distribution
  - Link encryption
  - Key Distribution Centre (KDC): all eggs in one basket
  - Multiple KDCs: better security
- Key management easier in public key cryptography

# Some Non-Crypto Attacks

- **Man-in-the-middle attack:** play a trick by being in the middle
- **Traffic analysis:**
  - Can learn information by just looking at presence/absence of traffic, or its volume
  - Can be countered using data padding
- **Playback or replay attacks:**
  - To counter: need to verify *timeliness* of message from sender while authenticating
  - Beware of issues of time synchronization