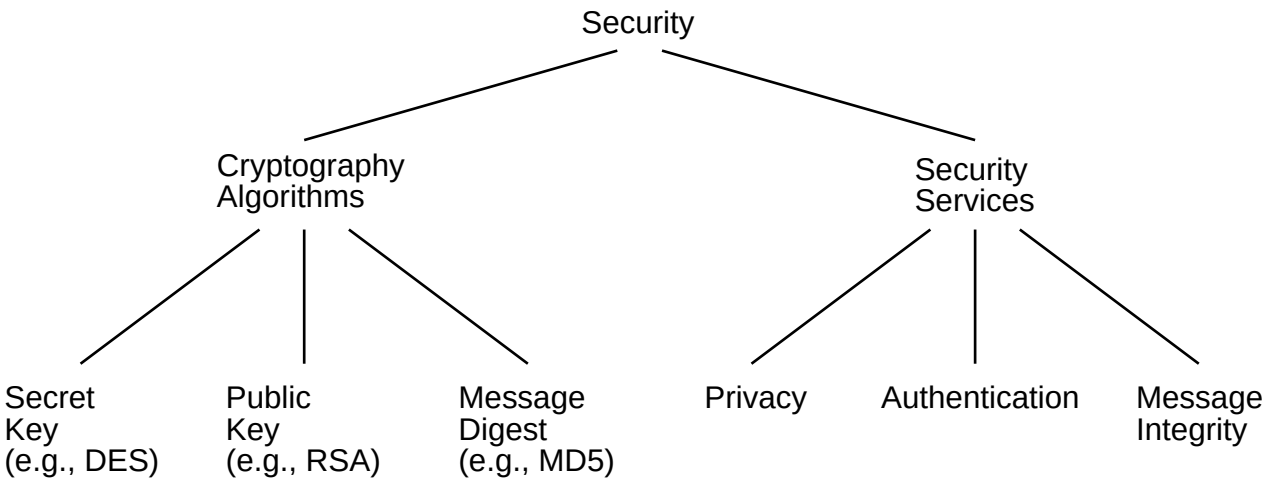
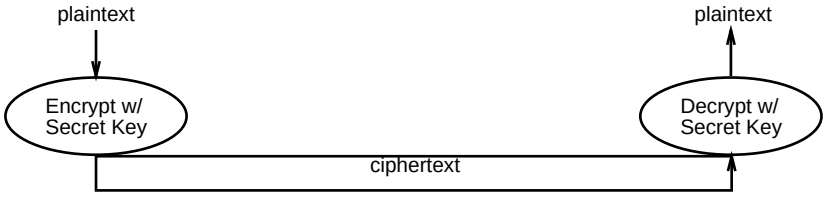
Security

- Cryptography functions
  - Secret key (e.g., DES)
  - Public key (e.g., RSA)
  - Message digest (e.g., MD5)
- Security services
  - Privacy: preventing unauthorized release of information
  - Authentication: verifying identity of the remote participant
  - Integrity: making sure message has not been altered

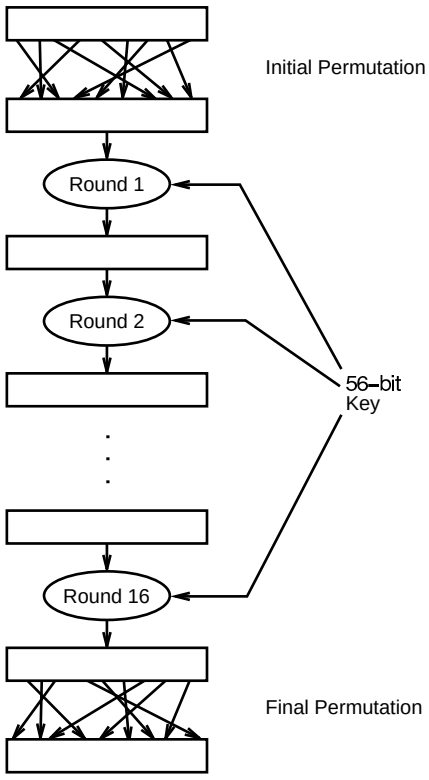


Encryption Algorithms

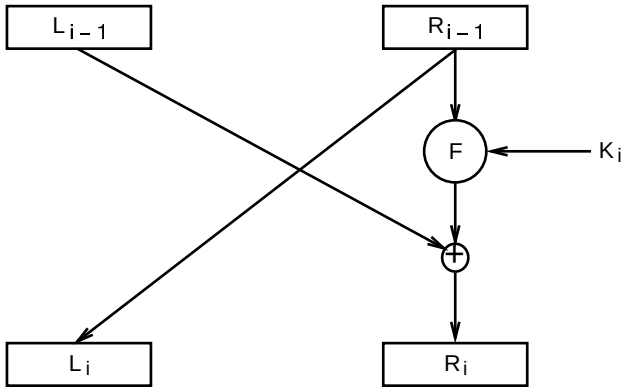
Private Key (DES)



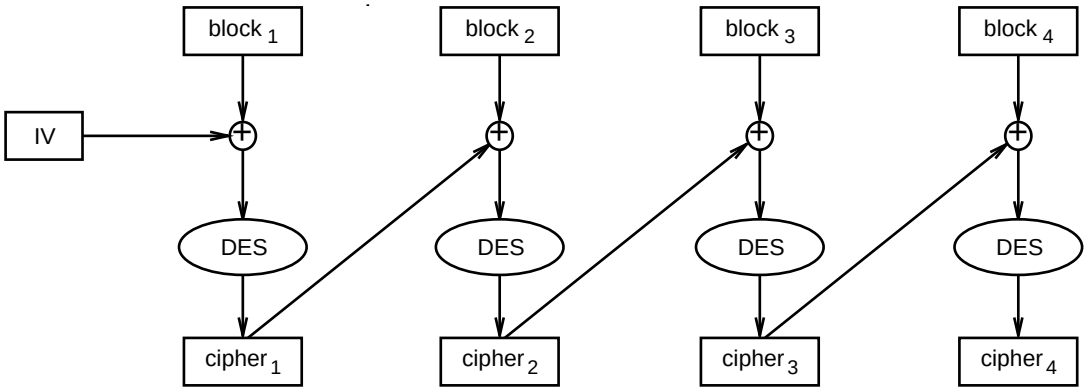
- 64-bit key (56-bits + 8-bit parity)
- 16 rounds



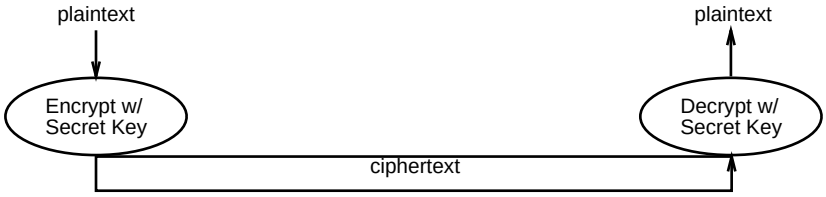
- Each round



- Function F and generation of K_i for each round not shown
- Repeat for larger messages



Public Key (RSA)



- Generate a public and private key
  - choose two large prime numbers p and q (each 256 bits)
  - multiply p and q together to get n
  - chose the encryption key e, such that e and $(p-1) \times (q-1)$ are relatively prime.
  - two numbers are relatively prime if they have no common factor greater than one.
  - compute decryption key d such that

$$d = e^{-1} mod\ ((p-1) \times (q-1))$$

  - construct public key as $\langle e, n \rangle$
  - construct private key as $\langle d, n \rangle$
  - discard (do not disclose) original primes p and q
- Encryption & Decryption

$$c = m^e mod\ n$$

$$m = c^d mod\ n$$

Message Digest

- Cryptographic checksum

  just as a regular checksum protects the receiver from accidental changes to the message, a cryptographic checksum protects the receiver from malicious changes to the message.

- One-way function

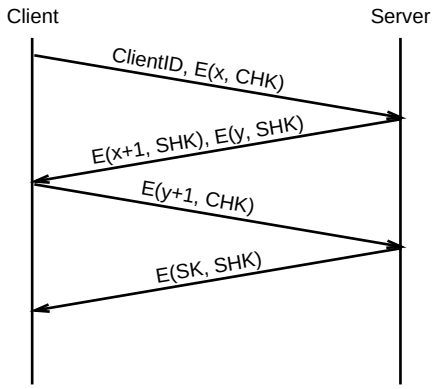
  given a cryptographic checksum for a message, it is virtually impossible to figure out what message produced that checksum; it is not computationally feasible to find two messages that hash to the same cryptographic checksum.

- Relevance

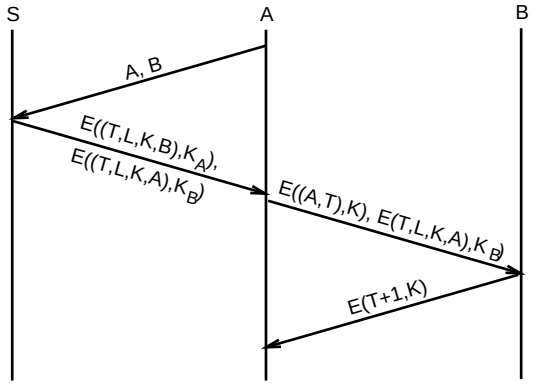
  if you are given a checksum for a message and you are able to compute exactly the same checksum for that message, then it is highly likely this message produced the checksum you were given.

Authentication Protocols

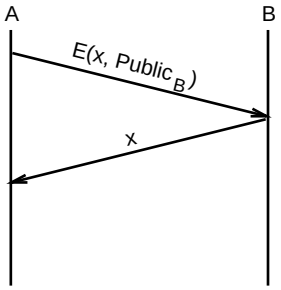
- Three-way handshake



- Trusted third party (Kerberos)



- Public key authentication



Message Integrity Protocols

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signiture with private key and verify with public key
- Keyed MD5
  - sender

    m + MD5(m + k) + E(k, *private*)
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
    - compares result with checksum sent with message
- MD5 with RSA signature
  - sender

    m + MD5(m) + E(MD5(m), *private*)
  - receiver
    - decrypts signature with sender's pubic key
    - compares result with MD5 checksum sent with message