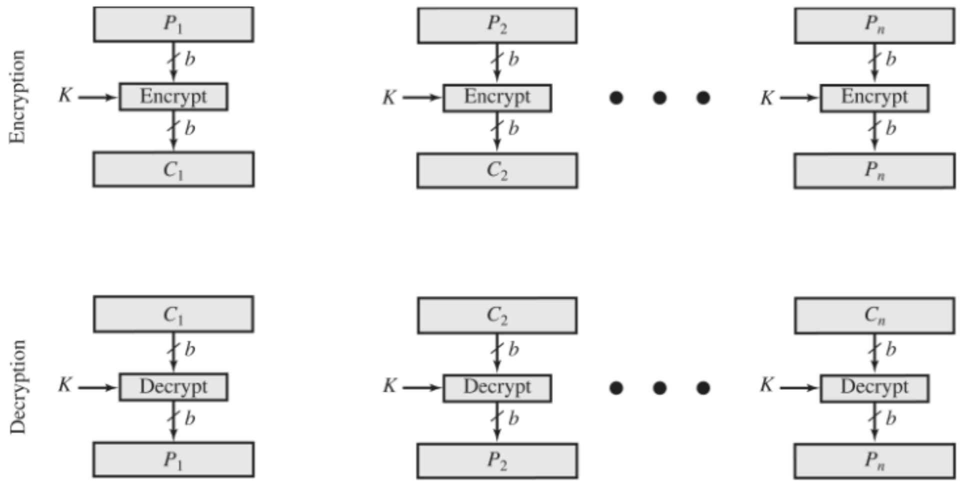
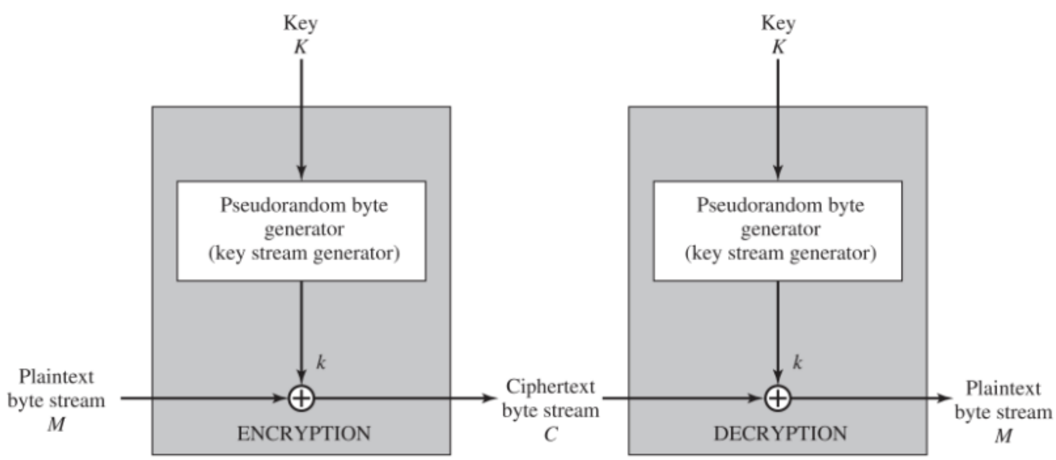
- Confidentiality with symmetric encryption
    - Symmetric encryption
        - strong algorithm that if an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key
        - Sender and receiver must have obtained copies of secret key in a secure fashion and must keep the key secure
    - Two approaches to attacking a symmetric encryption
        - cryptanalysis: relies on the nature of the algorithm plus perhaps knowledge of general characteristics of the plaintext, or even some sample plaintext-ciphertext pairs
        - brute-force attack: try every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
    - Systemic block encryption algorithm:
        - Block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size blocks
            - DES: Data Encryption Standard
                - Takes plaintext block of 64 bits and a key of 56 bits to produces a ciphertext block of 64 bits
            - Triple DES:
                - involves repeating the basic DES algorithm three times, using either two or three unique keys for a size of 112 or 168 bits
            - AES: Advanced Encryption Standard
                - must be a symmetric block cipher with a block length of 128 bits and support key lengths of 128, 192, and 256 bits
    - Practical safety issues:
        - Email messages, network packets, database records, and other plaintext resources must be broken up into a series of fixed-length block for encryption by a symmetric block cipher



(a) Block cipher encryption (electronic codebook mode)



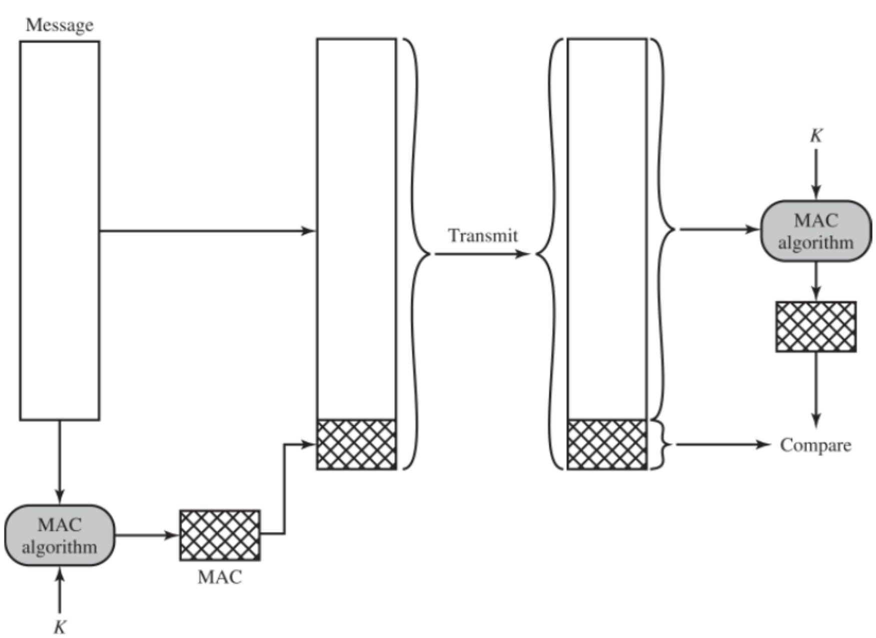
(b) Stream encryption

        - ECB may not be best practice for long messages
    - Stream Ciphers:
        - processes the input elements continuously, producing output one element at a time
        - key is input into pseudo random bit generator that produces a stream of 8-bit numbers, output, keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive OR (XOR) operation

2.2 Message Authentication and Hash Functions:

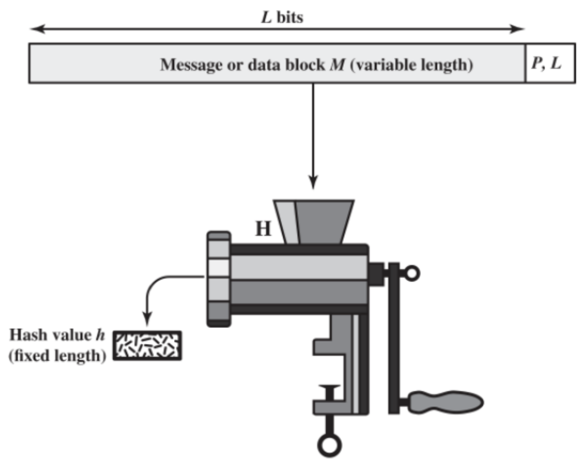
- Authentication Using Symmetric Encryption:
    - In ECB mode an attacker may alter the order of the blocks altering the meaning of the data
- Message Authentication without Message Encryption:

- Authentication tags may be generated and appended to each message in transmission
- Normally authentication and encryption are done separately and not apart of the same process
- 3 instances of when message confidentiality is preferable:
    - when messages are sent using a broadcast there can be an authentication tag provided, the responsible system performs the authentication and if there is a violation other destination systems are alerted
    - Authentication is carried out on a selective basis, with messages being chosen at random for checking. Usually with high volume loads
    - computer programs may also use authentication tags, if one were attached to the program, it could be checked whenever assurance is required of the integrity of the program
- Message Authentication with Code:
    - Small block of data that generates a secret key
        - two communicating parties must share a common secret key KAB
        - calculates message authentication code as complex function: MACM = F(KAB, M)
        - recipients perform the same calculation using the secret key to generate a new message authentication code



- Receiver assures message has not been altered, even if message was altered the code would not be and result in a different calculation
- Receiver assures that the message is from the alleged sender
- If the message includes a sequence number (like TCP), then the receiver can be assured of the proper sequence, because an attacker cannot successfully alter the sequence number

  - Authentication is less vulnerable to being broken than encryption because of mathematical properties
- One-way Hash Function:
  - Accepts a variable-size message M as input and produces a fixed-size message digest H(M) as output



P, L = padding plus length field
Figure 2.4 Cryptographic Hash Function; h=H(M)

  - Hash function does not take a secret key as input
  - Can be encrypted using symmetric encryption
  - can be encrypted using public-key encryption
  - public key has two advantages: digital signature and message authentication.

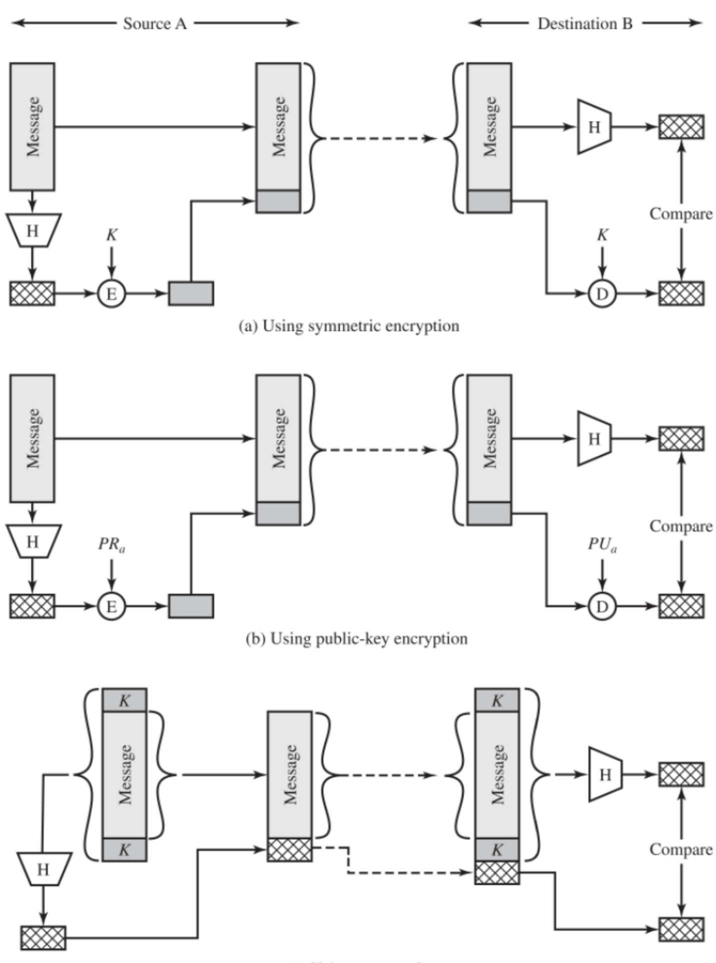


Figure 2.5 Message Authentication Using a One-Way Hash Function

  - Sometimes encryption is not necessarily the best method, for example it is slow software
  - encryption hardware is not cheap
  - hardware is optimized toward large data sizes
  - encryption algorithm may be protected by a patent

  - c. assumes that both parties share a secret key, which is incorporated into the process of generating a hash code

- Secure Hash Functions:
  - Hash function requirements:
    - H can be applied to a block of data of any size
    - H produces a fixed-length output
    - H(x) is relatively easy to compute for any given x
    - For any given code h, it is computationally infeasible to find x, such that H(x) = x (one way hash)
      - generates a code given a message, but virtually impossible to generate a massage given a code
    - For any given block x, it is computationally infeasible to find y != x with H(y) = H(x) (weak collision resistant)
      - guarantees that it is impossible to find an alternative message with the same hash value
    - Computationally infeasible to find any pair (x, y) such that H(y) = H(x) (collision restraint)
- Security of Hash Functions:
  - Strength of hash depends on length of the hash code produced by the algorithm
    - Preimage resistant: 2n
    - Second preimage resistant: 2n
    - Collision Resistant: 2n/2
- Secure Hash Function Algorithms:
  - SHA (Secure Hash Algorithm)
- Other Applications of Hash Functions:
  - Passwords: when a user enters a password the hash of that password is compared to the stored hash value for verification
  - intrusion detection: store the hash value for a file for each file on a system and secure the hash values

2.3 Public-Key Encryption:

- Public Encryption Structure
  - Public-key algorithms are based on mathematical functions rather than on simple operations on bit patterns
  - asymmetric: involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key
  - security of any encryption depends on two things:
    - length of key
    - computational work involved in breaking the cipher

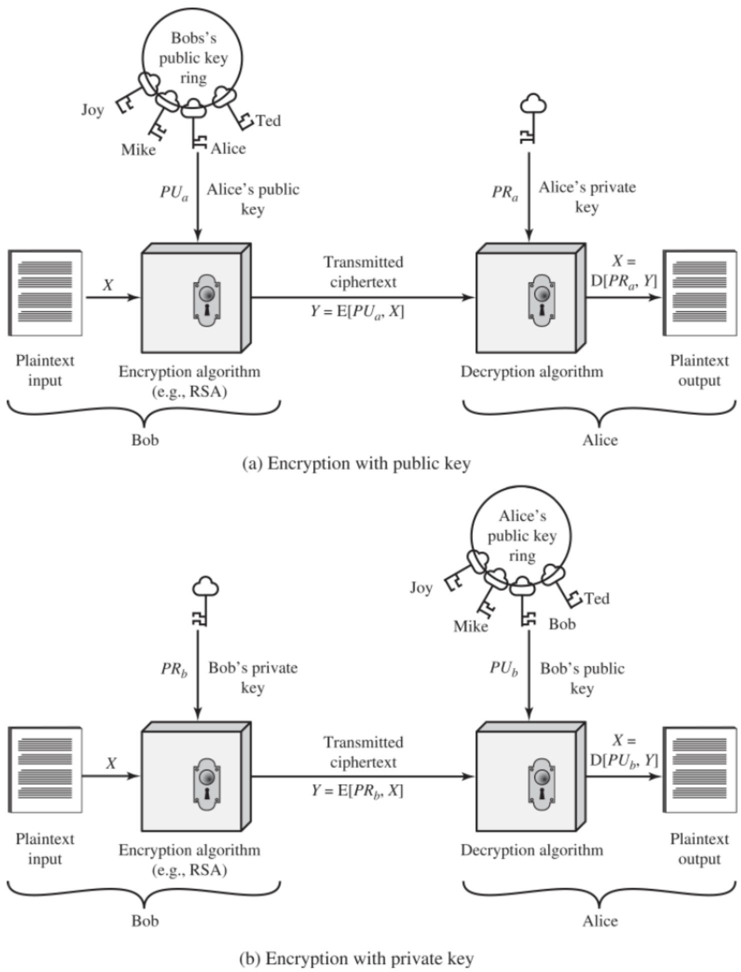


Figure 2.6 Public-Key Cryptography

- Plaintext: readable message or data that is fed into the algorithm as input
- Encryption algorithm: performs various transformations on the plaintext
- public and private key: selected keys so that if one is used for encryption, the other is used for decryption
- ciphertext: scrambled message produced as output. depends on the plaintext and key
- Decryption algorithm: accepts the ciphertext and matching key and produces the original plaintext

- Public-key algorithms rely on:
  - Each user generates a pair of keys to be used for encryption and decryption
  - each user places one of the two keys in a public register or another accessible file and the other is kept private
  - if bob wants to send a private message to alice, then bob encrypts the message using alice's public key
  - when alice receives the message, she decrypts it using her private key, no other recipient can decrypt
- Applications For Public Key Cryptosystems:
  - We classify the use of public-key cryptosystems into three categories: digital signature, symmetric key distribution, and encryption of secret keys

Table 2.3 Applications for Public-Key Cryptosystems

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie–Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

- Asymmetric Encryption Algorithms
    - RSA:
        - most widely accepted and implemented approach to public-key encryption
        - block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n
    - Diffie-Hellman Key Agreement
        - enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption
    - Digital Signature Standard:
        - makes use of the SHA-1 and presents a new digital signature technique, the DSA (digital signature algorithm).
        - uses an algorithm that is designed to provide only the digital signature function
        - cannot be used for encryption or key exchange
    - Elliptic Curve Cryptography:
        - Appears to offer equal security to RSA for a smaller bit size, reducing processing overhead

2.4 Digital Signatures and Key Management

-