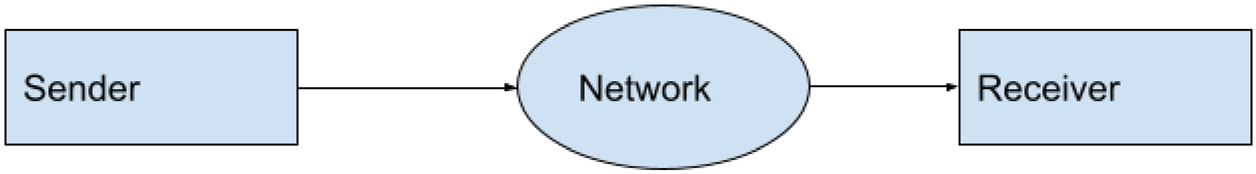


Three security goals:

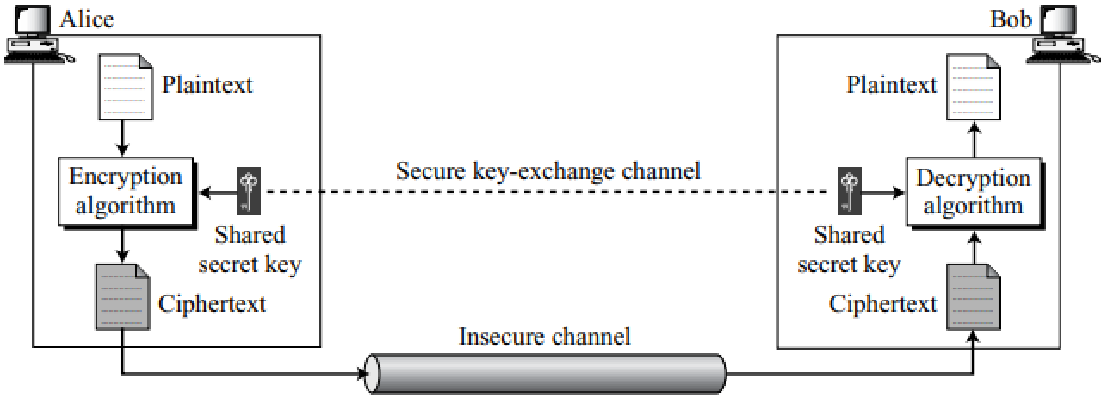
Confidentiality: When we send a piece of information to be stored in a remote computer or when we retrieve a piece of information from a remote computer, we need to conceal it during transmission.
Integrity: Integrity means that changes need to be done only by authorized entities and through authorized mechanisms.
Availability: The information created and stored by an organization needs to be available to authorized entities.

Techniques for achieving security goals:

Cryptography - secret writing, the art of **transforming** messages to make them secure and immune to attacks

Symmetric-Key Cryptography - chapter 3

Encryption/decryption can be thought of as electronic locking. In symmetric key enciphering, Alice puts the message in a box and locks the box using the shared secret key; Bob unlocks the box with the same key and takes out the message.

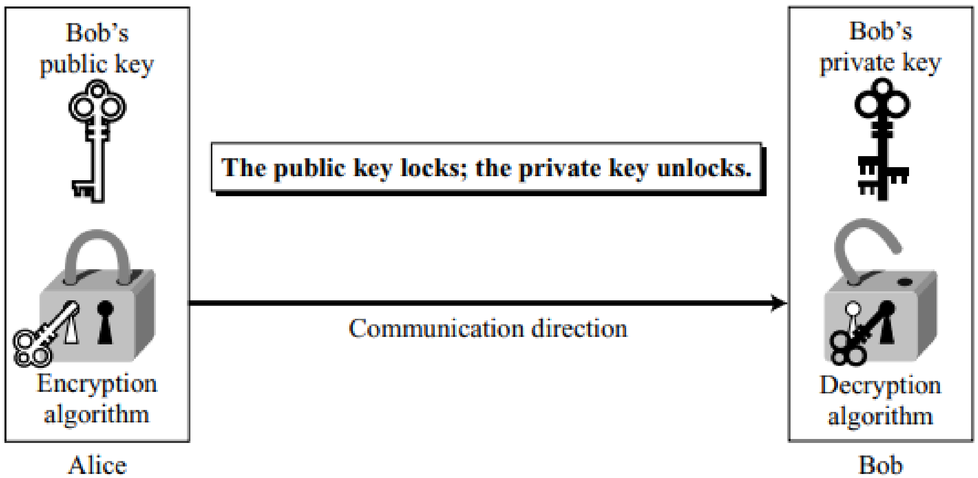


The original message from Alice to Bob is called plaintext; the message that is sent through the channel is called the ciphertext. To create the ciphertext from the plaintext, Alice uses an encryption algorithm and a shared secret key. To create the plaintext from ciphertext, Bob uses a decryption algorithm and the same secret key. We refer to encryption and decryption algorithms as ciphers. A key is a set of values (numbers) that the cipher, as an algorithm, operates on.

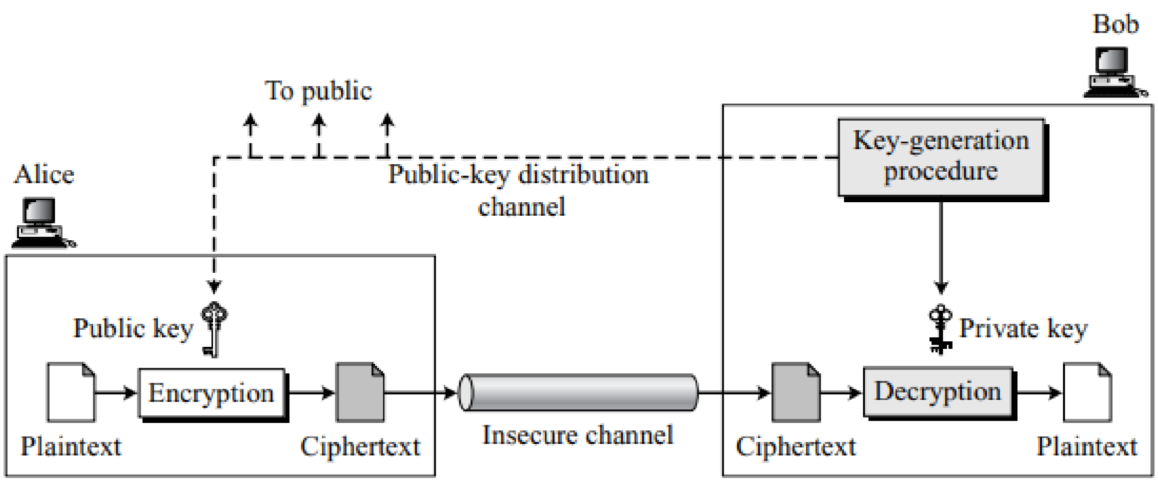
Note that the symmetric-key encipherment uses a single key (the key itself may be a set of values) for both encryption and decryption. In addition, the encryption and decryption algorithms are inverses of each other. If P is the plaintext, C is the ciphertext, and K is the key, the encryption algorithm Ek(x) creates the ciphertext from the plaintext; the decryption algorithm Dk(x) creates the plaintext from the ciphertext. We assume that Ek(x) and Dk(x) are inverses of each other: they cancel the effect of each other if they are applied one after the other on the same input. We have
Encryption: C = Ek(P)
Decryption: P = Dk(C)

Asymmetric-Key Cryptography - chapter 10

Unlike symmetric-key cryptography, there are distinctive keys in asymmetric-key cryptography: a private key and a public key. To send a secured message to Bob, Alice first encrypts the message using Bob's public key. To decrypt the message, Bob uses his own private key.



Locking and unlocking in asymmetric-key cryptosystem



General idea of asymmetric-key cryptosystem

- It emphasizes the asymmetric nature of the cryptosystem. The burden of providing security is mostly on the shoulders of the receiver (Bob, in this case). Bob needs to create two keys: one private and one public. Bob is responsible for distributing the public key to the community. This can be done through a public-key distribution channel.

- Asymmetric-key cryptography means that Bob and Alice cannot use the same set of keys for two-way communication. Each entity in the community should create its own private and public keys.
- Asymmetric-key cryptography means that Bob needs only one private key to receive all correspondence from anyone in the community, but Alice needs n public keys to communicate with n entities in the community, one public key for each entity. In other words, Alice needs a ring of public keys.

Encryption and decryption in asymmetric-key cryptography are mathematical functions applied over the numbers representing the plaintext and ciphertext. The ciphertext can be thought of as C = f (Kpublic, P); the plaintext can be thought of as P = g(Kprivate, C). The decryption function f is used only for encryption; the decryption function g is used only for decryption.

Difference between Symmetric-Key and Asymmetric-Key Cryptography

- In symmetric-key cryptography, the secret must be shared between two persons. In asymmetric-key cryptography, the secret is personal (unshared); each person creates and keeps his or her own secret.

- Symmetric-key cryptography is based on substitution and permutation of symbols (characters or bits), asymmetric-key cryptography is based on applying mathematical functions to numbers.

- Asymmetric-key cryptography is much slower than symmetric-key cryptography.

We can divide traditional symmetric-key ciphers into two broad categories: substitution ciphers and transposition ciphers

SUBSTITUTION CIPHERS

A substitution cipher replaces one symbol with another. If the symbols in the plaintext are alphabetic characters, we replace one character with another. For example, we can replace letter A with letter D, and letter T with letter Z. If the symbols are digits (0 to 9), we can replace 3 with 7, and 2 with 6.

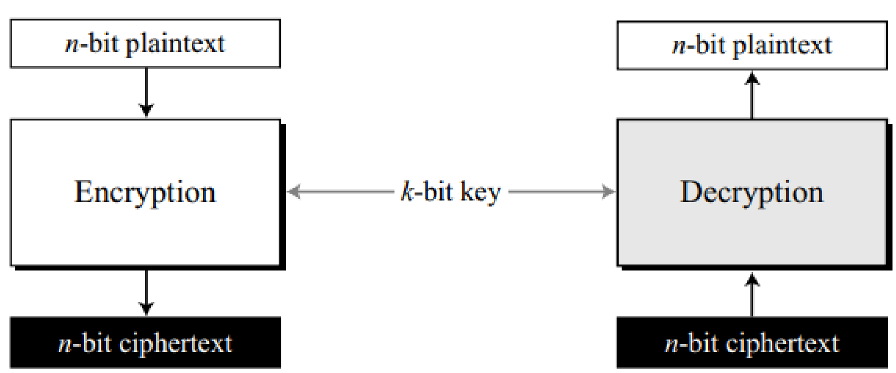
TRANSPOSITION CIPHERS

A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols. A symbol in the first position of the plaintext may appear in the tenth position of the ciphertext. In other words, a transposition cipher reorders (transposes) the symbols.

MODERN BLOCK CIPHERS

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key. The decryption algorithm must be the inverse of the encryption algorithm, and both operations must use the same secret key.

A modern block cipher



Components of a Modern Block Cipher

P-Boxes
A P-box (permutation box),It transposes/permutes bits.We can find three types of P-boxes in modern block ciphers: straight P-boxes, expansion P-boxes, and compression P-boxes.
Straight P-Boxes: A straight P-Box with n inputs and n outputs is a permutation.
Compression P-Boxes: are used when we need to permute bits and the same time decrease the number of bits for the next stage.
Expansion P-boxes are used when we need to permute bits and at the same time increase the number of bits for the next stage.

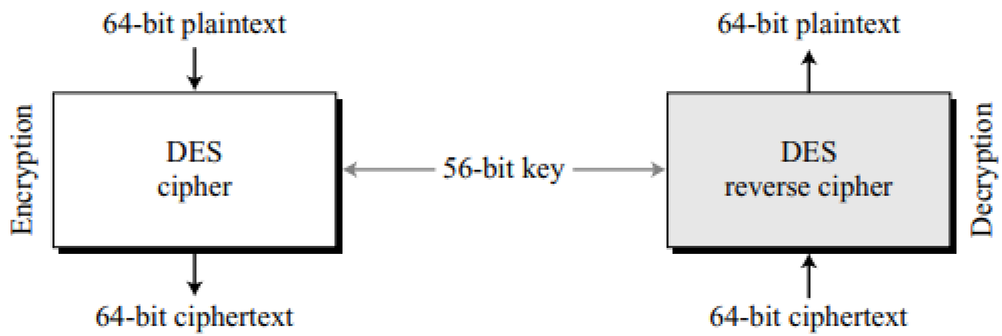
S-Boxes
An S-box (substitution box) are substitution ciphers in which the relationship between input and output is defined by a table or mathematical relation. However, an S-box can have a different number of inputs and outputs.

Rounds
In an N-round cipher, the plaintext is encrypted N times to create the ciphertext; the ciphertext is decrypted N times to create the plaintext.The block cipher uses a key schedule or key generator that creates different keys for each round from the cipher key.

To achieve this goal, divide the plaintext and the ciphertext into two equal-length blocks, left and right. We call the left block L and the right block R. Let the right block be the input to the function, and let the left block be exclusive-ored with the function output.

Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

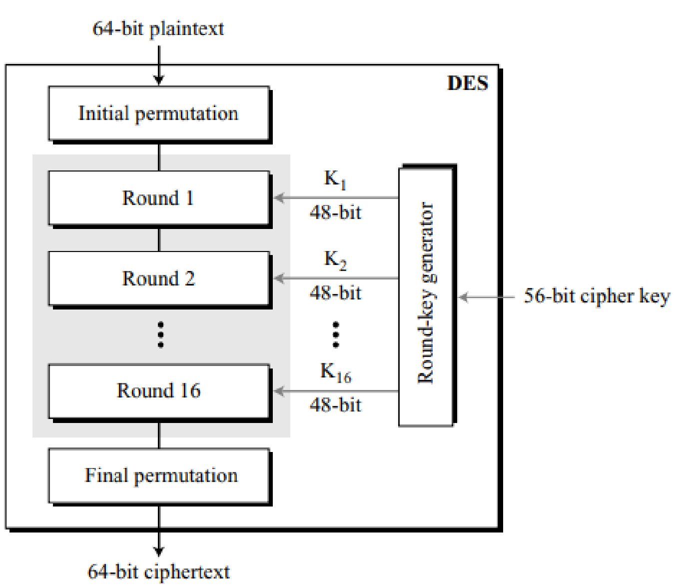


Encryption and Decryption with DES

At the encryption site, DES takes a 64-bit plaintext and creates a 64-bit ciphertext; at the decryption site, DES takes a 64-bit ciphertext and creates a 64-bit block of plaintext. The same 56-bit cipher key is used for both encryption and decryption.

DES STRUCTURE

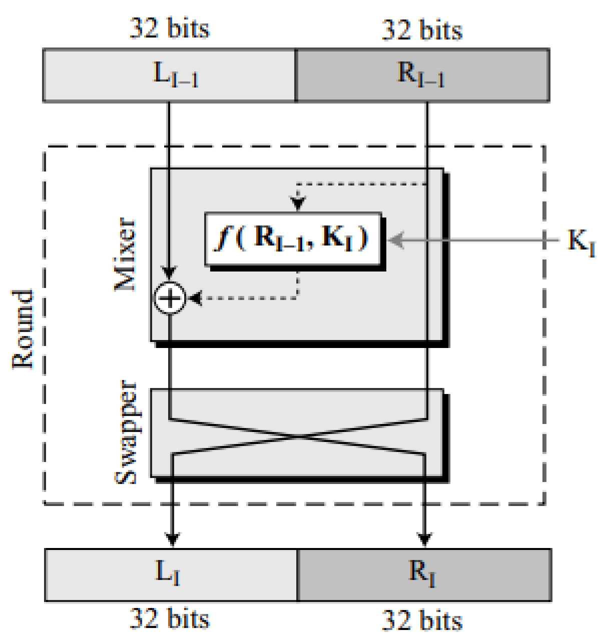
General structure of DES



1. Initial and Final Permutations - The initial and final permutations are straight P-boxes that are inverses of each other. Each of these permutations takes a 64-bit input and permutes them according to a predefined rule.

2. Rounds - DES uses 16 rounds.

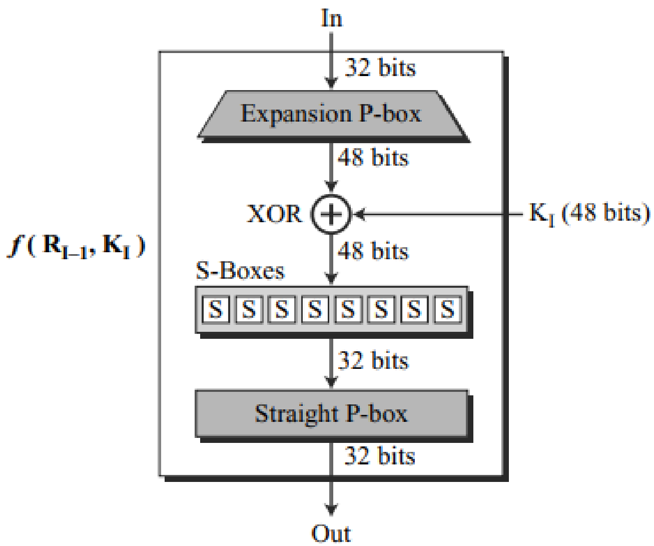
A round in DES (encryption site)



The round takes LI−1 and RI−1 from the previous round (or the initial permutation box) and creates LI and RI, which go to the next round (or final permutation box). We can assume that each round has two cipher elements (mixer and swapper).

DES Function- The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits (RI−1) to produce a 32-bit output. This function is made up of four sections: an expansion P-box, a whitener (that adds key), a group of S-boxes, and a straight P-box.

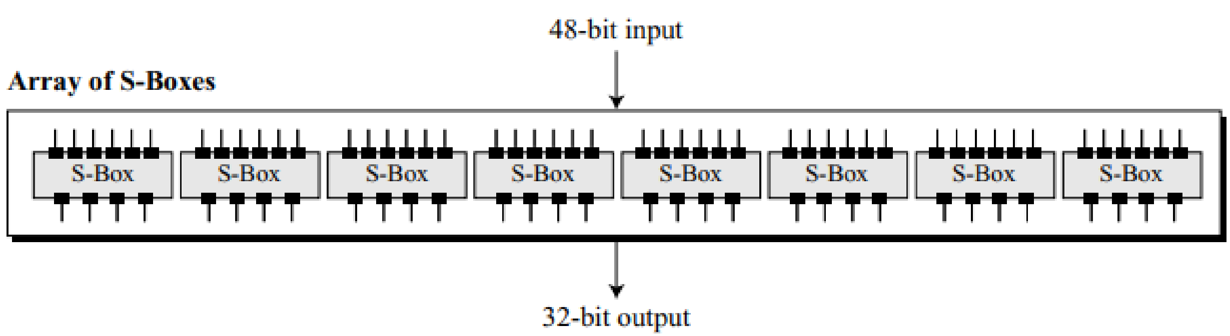
DES function



- **Expansion P-box**- Since RI−1 is a 32-bit input and KI is a 48-bit key, we first need to expand RI−1 to 48 bits. RI−1 is divided into 8 4-bit sections. Each 4-bit section is then expanded to 6 bits. This expansion permutation follows a predetermined rule.

- **Whitener (XOR)** - After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

- **S-Boxes** - The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box. The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text. The substitution in each box follows a pre-determined rule.

- **Straight Permutation** - The last operation in the DES function is a straight permutation with a 32-bit input and a 32-bit output.

3. Key Generation - The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.