Cryptographic Techniques

Marco Mellia

Symmetric Encryption

Symmetric encryption is also referred to a conventional encryption or **single-key encryption**

It remains the **most widely used** of the two types of encryption

- Here we revisit some key ideas of symmetric encryption
- The goal is to highlight the design principles and properties

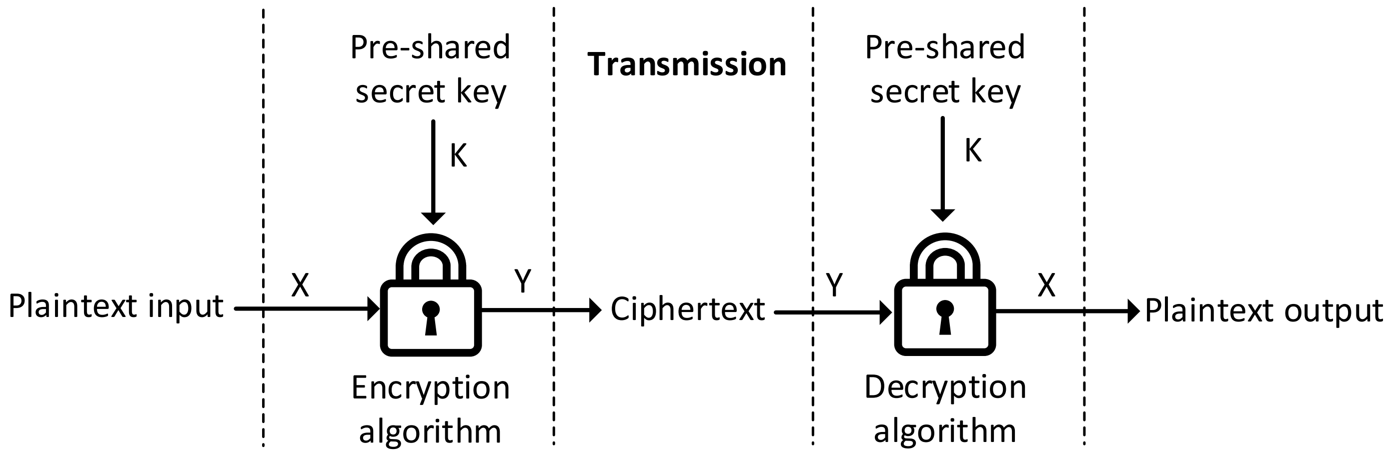
A single **key is pre-shared** between the sender and the receiver

Both parties can either **encrypt** or **decrypt** messages using the **same pre-shared key**

Symmetric Cipher Model

- If the pre-shared key is K, the **plaintext** is X, the **ciphertext** is Y, the encryption and decryption **algorithms** are E and D, respectively. Then the encryption and decryption processes are

$$E_K(X) = Y \text{ and } D_K(Y) = X$$



Classical Ciphers

Caesar Cipher

Caesar cipher is one of the earliest known substitution ciphers

The original Caesar cipher replaces each letter with the **third letter after**

Given a plaintext consists of English letters, its corresponding ciphertext can be found by substituting the letters

Example

- **Plaintext:** see you in the midnight
- **Ciphertext:** VHH BRX LQ WKH PLGQLJKW

Caesar Cipher

Alternatively, Caesar cipher can be defined as modulo arithmetic mathematically

- Each letter or character can be mapped to a number, such that $a = 0, b = 1, \dots, z = 25$. Let $\boldsymbol{k}$ **be the number of letters to shift** (i.e. the symmetric key), then the encryption and decryption of Caesar cipher are calculated as:

$$c = E_k(p) = (p + k) \bmod 26$$

- And

$$p = D_k(c) = (c - k) \bmod 26$$

Question: how many keys are possible?

Monoalphabetic Cipher

Monoalphabetic cipher maps each plaintext letter to a distinct ciphertext letter

Hence the key of monoalphabetic cipher is 26-letter long

Example:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Z	A	E	R	T	Y	U	I	O	P	Q	S	D	F	G	H	J	K	L	M	W	X	C	V	B	N

- Plaintext: if we wish to replace letters
- Ciphertext: OY CT COLI MG KTHSZET STMMTKL

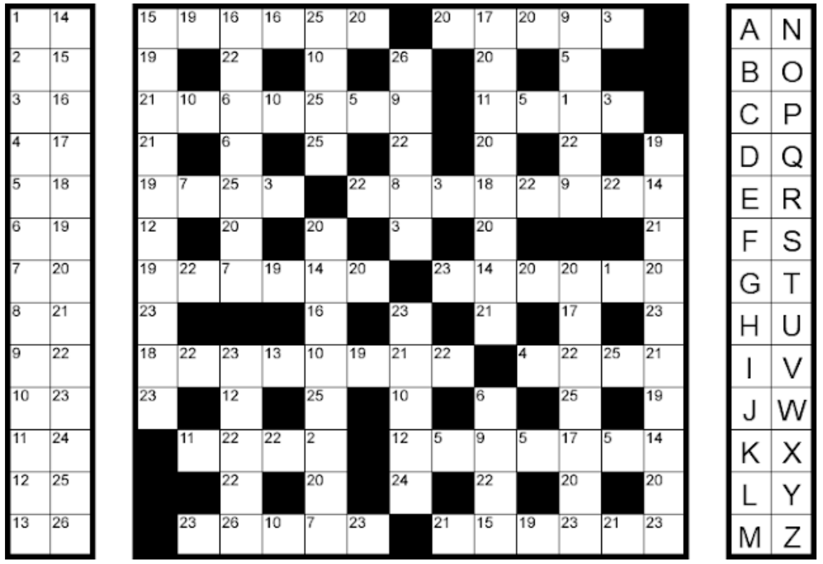
Question: how many keys are possible?

Cryptanalysis of Monoalphabetic Cipher

Monoalphabetic cipher has a total of $26! - 1 \approx 4\times1026$ keys

With so many keys, **brute force is not feasible** to crack the key in general

Question: what attack would make it possible to break this cypher?



Cryptanalysis of Monoalphabetic Cipher

Question: what attack would make it possible to break this cypher?

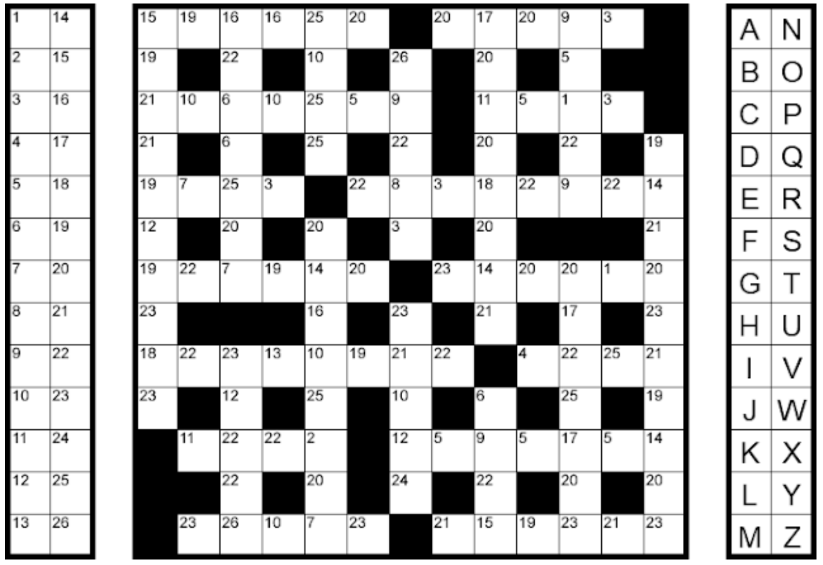
Monoalphabetic cipher is not secure due to the weakness of language characteristics

- For example, “ths s nt dffclt” (“this is not difficult”) can be understood although the vowels were removed

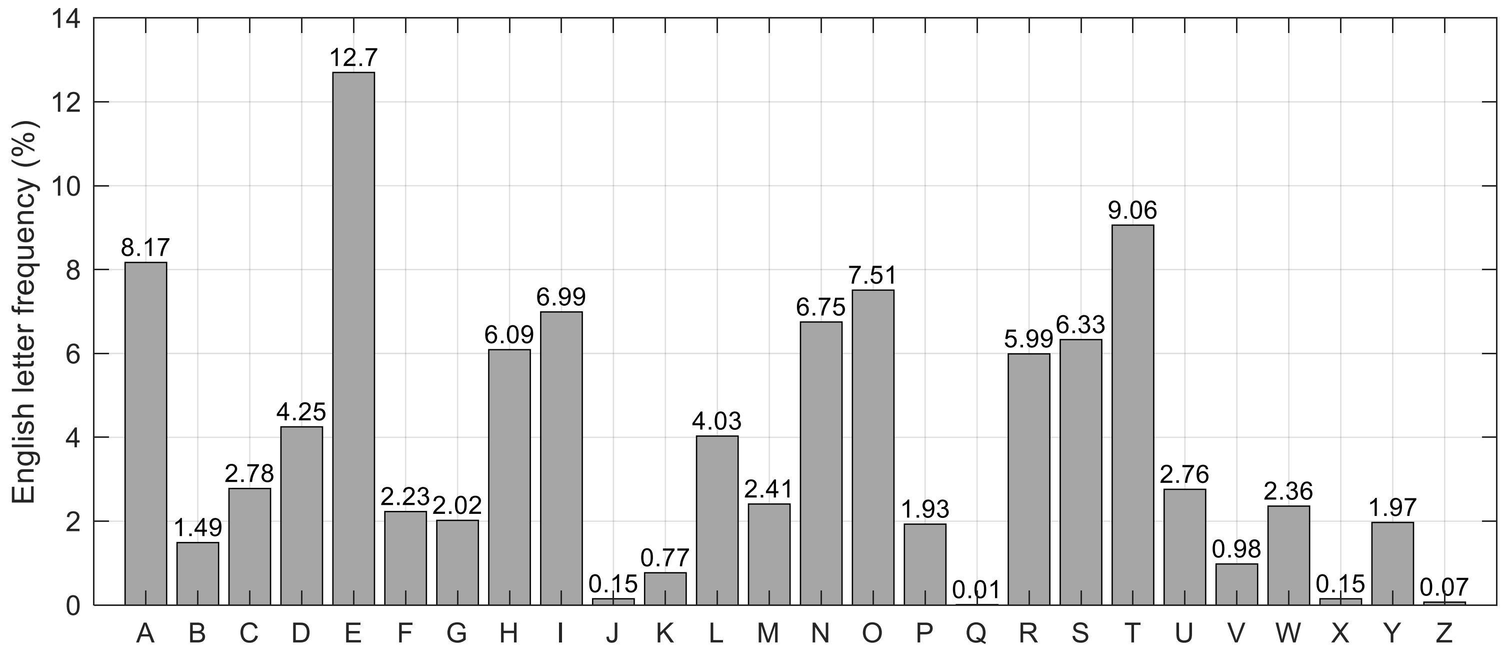
Exploiting the underlying characteristics of the language makes it easy to break the cypher

- Via **Frequency analysis**

Question: how can we make frequency analysis more difficult?



English Letter Frequency Distribution



Playfair Cipher

Playfair cipher treats **two letters** in the plaintext **as a single unit** and translates the unit into ciphertext letters

- Playfair cipher was invented by Charles Wheatstone in 1854, but was named after his friend Baron Playfair

Playfair encryption/decryption is based on the use of a 5×5 matrix of letters constructed using a **keyword**

- The keyword is used as the key

Playfair Cipher

The rules for filling in this 5 × 5 Playfair matrix are listed in the following

- Left to right
- Top to bottom
- First with keyword after duplicate letters have been removed
- Then with the remaining letters in alphabetic order
- “I/J” are used as a single letter

Key idea: make the process apparently complex but in practice easy to implement

For example, given keyword “DECEMBER,” the 5 × 5 matrix is:

D	E	C	M	B
R	A	F	G	H
I/J	K	L	N	O
P	Q	S	T	U
V	W	X	Y	Z

Playfair Cipher

The encryption of Playfair cipher is performed two letters at a time using the 5×5 matrix according to the rules below:

- If a pair is a repeated letter, insert a filler like “X,” e.g. “ballon” is encrypted as “ba lx lo on.”
- If both letters fall in the same row of the matrix
  - replace each with letter to its right (wrapping back to start from end)
- If both letters fall in the same column of the matrix
  - replace each with the letter below it (again wrapping to top from bottom)
- Otherwise
  - each letter is replaced by the one in its row in the column of the other letter of the pair

For instance, using the matrix illustrated earlier

- “mb”→“BD,” “dv”→“RD”, “at”→“GQ,” and “od” → “IB” or “JB.”

Decryption in Playfair cipher works exactly in reverse

Cryptanalysis of Playfair Cipher

Security is much improved over monoalphabetic cipher since it has a total of 26 × 26 = 676 **diagrams** (pair of letters)

It would need a **676-entry frequency table** to analyze verses 26 for a monoalphabetic

However, Playfair cipher can be broken given just a few hundred letters because the ciphertext still has much plaintext structure

Rationale: make it apparently complicated. In practice, this does not work

Polyalphabetic Cipher

Let us generalize the concept: make the frequency attack (and language structure attack) infeasible.

One option: use multiple cipher alphabets!

Polyalphabetic cipher improves security by using **multiple cipher alphabets**

- It makes cryptanalysis harder with more alphabets to guess and **flatters frequency distribution**

Polyalphabetic cipher uses a **key to select which alphabet** is used for each letter of the message

- It uses each alphabet in turn and repeats from start after the key is used up

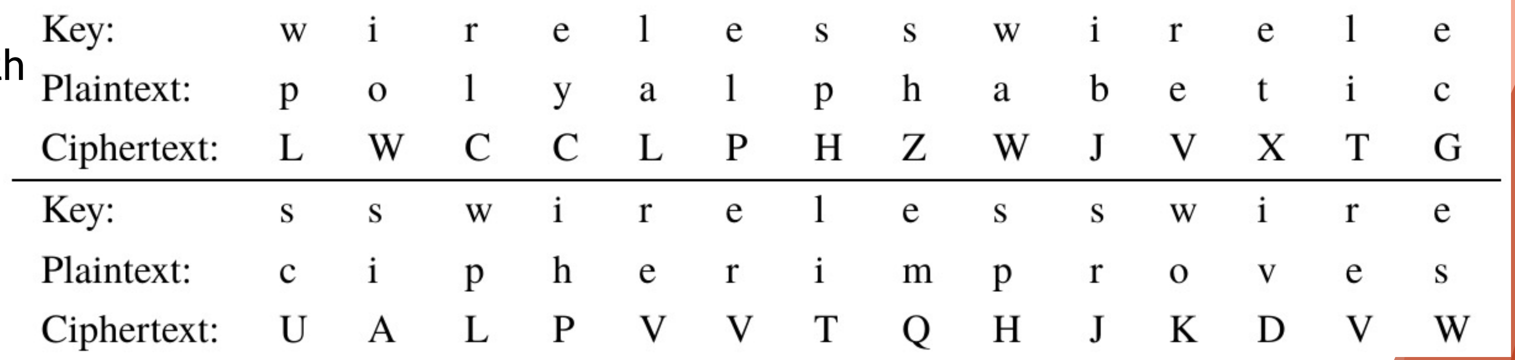
Vigenère Cipher

- Vigenère cipher is the simplest polyalphabetic substitution cipher
- It effectively uses **multiple Caesar ciphers**
- The key of Vigenère cipher is multiple letters long, s.t.

$$K = k_1 k_2, \ldots, k_d,$$

  - The ith letter specifies the ith alphabet to use
  - After the first d letters in a message, it **repeats** from the start

For example, with keyword "wireless":



Key:	w	i	r	e	l	e	s	s	w	i	r	e	l	e
Plaintext:	p	o	l	y	a	l	p	h	a	b	e	t	i	c
Ciphertext:	L	W	C	C	L	P	H	Z	W	J	V	X	T	G
Key:	s	s	w	i	r	e	l	e	s	s	w	i	r	e
Plaintext:	c	i	p	h	e	r	i	m	p	r	o	v	e	s
Ciphertext:	U	A	L	P	V	V	T	Q	H	J	K	D	V	W

Cryptanalysis of Polyalphabetic Cipher

The Vigenère cipher had been le chiffre indéchiffrable (the unbreakable cipher) for several centuries

As a result of a challenge, it was broken by the British cryptographer Charles Babbage in 1854. However, the cracking technique was not published during that time. In fact, the technique was finally made public in the 1970s

The Vigenère and related polyalphabetic ciphers have **multiple ciphertext letters for each plaintext letter**, thus letter frequencies are obscured. However, it does not completely obscure the underlying language characteristics

Key ides: identify first the number of translation alphabets, and then attack each separately

Question: why not publish the algorithm to break the cypher?

One-Time Pad

Taking the idea of multiple cyphers to the extremes: One-Time Pad

- Use a single-use pre-shared key that is larger than or equal to the size of the message being sent

Produces a **random output** that bears no statistical relationship to the plaintext

Because the ciphertext contains no information about the plaintext, there is no way to break the code since any plaintext can be mapped to any ciphertext given some key

Although one-time pad offers complete security, it has two fundamental difficulties in practice

- the practical problem of making large quantities of random keys
- problem of key distribution and protection, where for every message to be sent, a **key of equal length** is needed by both sender and receiver

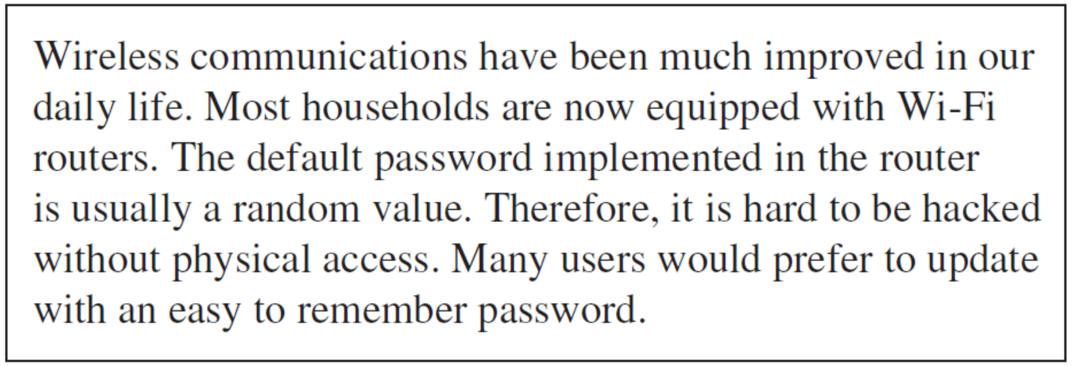
Because of these difficulties, one-time pad is of limited usage, and is only useful for low-bandwidth channels requiring very high security

Steganography

Steganography is an **alternative** to encryption that hides the very existence of a message by some means

Steganography has a number of drawbacks when compared with encryption

- It requires a lot of overhead to hide a relatively few bits of information
- Once the system is discovered, it becomes virtually worthless, although a message can be first encrypted and then hidden using steganography

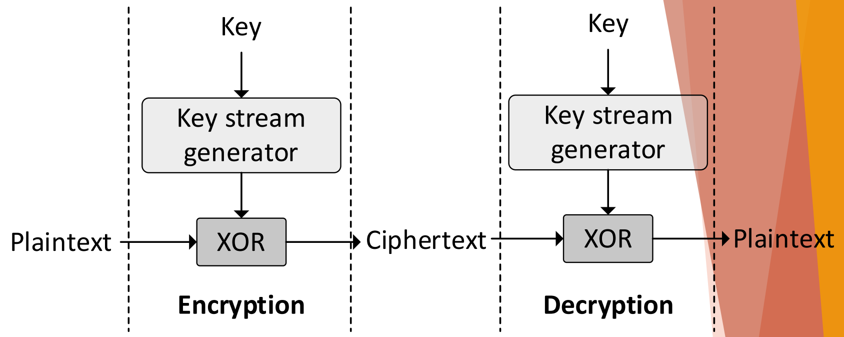


Wireless communications have been much improved in our daily life. Most households are now equipped with Wi-Fi routers. The default password implemented in the router is usually a random value. Therefore, it is hard to be hacked without physical access. Many users would prefer to update with an easy to remember password.

Can you find the original message?

Stream Cipher

Stream Cipher



- A stream cipher is a symmetric key cipher
- Each plaintext digit (in bit or byte) is encrypted one at a time with a corresponding digit from the keystream
- The encryption is generally XOR function, where a digit of the plaintext is XORed with the corresponding digit of the keystream
- The keystream is usually a pseudo-random number generated by an algorithm with a (pseudo) random seed value
- Because a stream cipher is a symmetric key cipher, the seed value is synchronized between a transmitter and a receiver to serve as the cryptographic key
- The keystream is generated at both side instead of being transmitted

Rivest Cipher 4

Rivest Cipher 4 (RC4) is one of the typical and widely used stream ciphers. It was designed by Ron Rivest of RSA Security in 1987

- Because of its speed and simplicity, RC4 has been applied in many applications for years, e.g. Wired Equivalent Privacy (WEP) and Wireless Protected Access (WPA) in wireless local area networks, and Transport Layer Security (TLS) for Internet security

There are two elements in the RC4 algorithm

- Key Scheduling Algorithm (KSA)
- Pseudo-Random Generation Algorithm (PRGA)

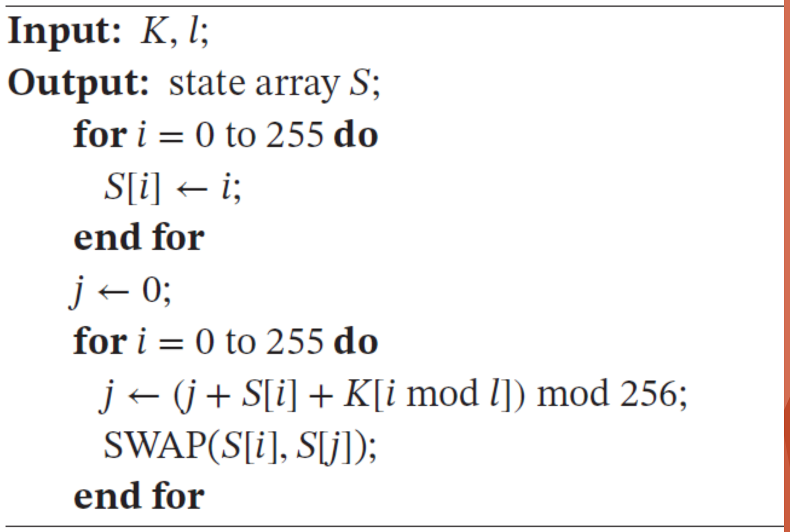
RC4: Key Scheduling Algorithm (KSA)

First, it has the key initialization process

- Takes input the l-byte secret key K
- Generates a random 256-value state array S

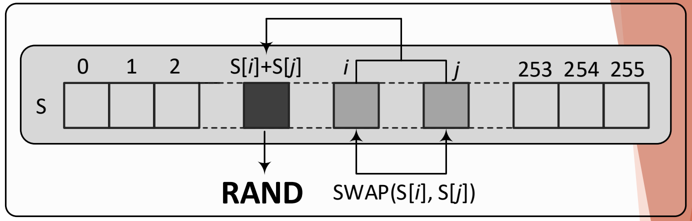
The array S is initialized from 0 to 255

- It is shuffled based on the input key



```
Input: $K, l$;
Output: state array $S$;
    for $i = 0$ to 255 do
        $S[i] \leftarrow i$;
    end for
    $j \leftarrow 0$;
    for $i = 0$ to 255 do
        $j \leftarrow (j + S[i] + K[i \bmod l]) \bmod 256$;
        SWAP($S[i], S[j]$);
    end for
```

RC4: PRGA



PRGA is the pseudo-random generation algorithm that runs after KSA

- Two states in the array S are first chosen, for example, $S[i]$ and $S[j]$
- Swap $S[i]$ and $S[j]$
- Then, a new index is calculated as $(S[i] + S[j])$ mod 256
- Lastly, the element in array S from this index is selected as the keystream byte k

The keystream k generated from PRGA algorithm is XORed with the plaintext byte to produce the ciphertext byte in the sender side

Rationale: make it apparently complicated. In practice, this does not work

```
Input: $K, l, S$;
Output: $k$;
    $i = 0, j = 0$;
    while Generating output do
        $i \leftarrow (i + 1) \bmod 256$;
        $j \leftarrow (j + S[i]) \bmod 256$;
        SWAP($S[i], S[j]$);
        $k \leftarrow S[(S[i] + S[j]) \bmod 256]$;
    end while
```

Modern Block Ciphers

Feistel Block Cipher

Feistel cipher structure was devised by Horst Feistel

- One of Feistel's main contributions was the invention of a suitable structure, which adapted Shannon's S-P network in an easily inverted structure

The input data in the encryption process is split into two halves (i.e. L_0 and R_0) and processed through a number of rounds

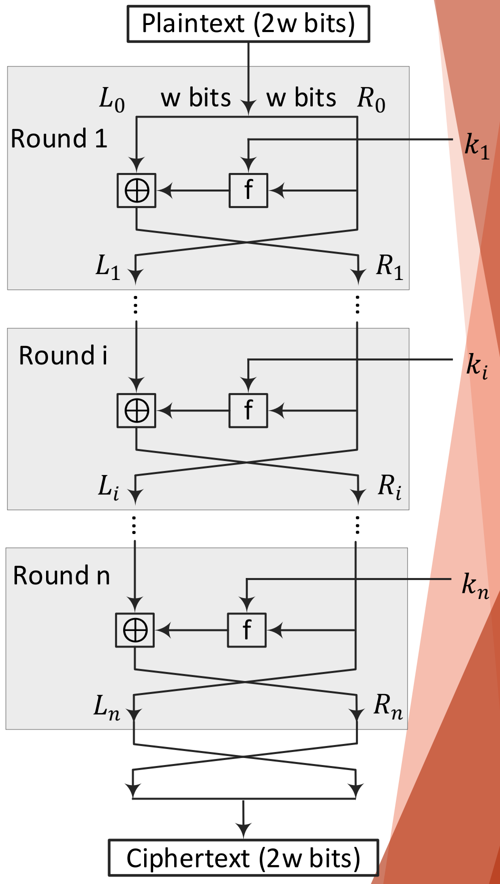
Rationale: make it apparently complicated

Encryption Structure of Feistel Cipher

In each round

- a substitution is performed on the left half (i.e. L_{i-1} in the ith round) using the output of the round function $f(\cdot)$
- Round function f $(\cdot)$ performs a series of substitution and permutation on right half (i.e. R_{i-1} in the ith round) and a subkey (i.e. k_i in the ith round)
- The final stage of a round is a permutation function, which swaps the two halves. The output of the left half is input into the right side of the next round, where the output of the right half is input into to left side of the next round

The output of the final round gets swapped such that $R_n||L_n$ as the ciphertext

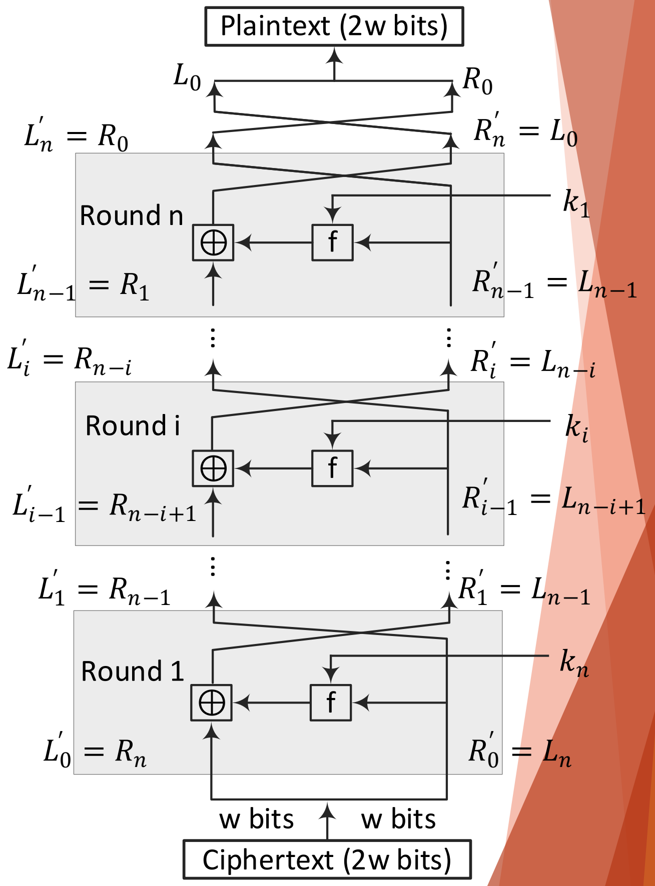


Decryption Structure of Feistel Cipher

The decryption process with a Feistel cipher is essentially the same as the encryption process

- The input to the ith decryption round is $L'_{i-1}||R'_{i-1}$, or equivalently, $R_{n-i+1}||L_{n-i+1}$
- The subkey k_i is applied in reverse order, that is, k_n is used in the first round, $k_{n-1}$is used in the second round, and so on until k_1 is used in the last round

A final swap recovers the original plaintext $R'_n||L'_n = L_0||R_0$



Parameters and Design Features of Feistel Cipher Structure

Parameters and features	Description
Block size	Increasing size improves security, but slows cipher
Key size	Increasing size improves security, makes exhaustive key searching harder, but may slow cipher
Number of rounds	Increasing number improves security, but slows cipher
Subkey generation	Greater complexity can make analysis harder, but slows cipher
Round function	Greater complexity can make analysis harder, but slows cipher
Software implementation	More recent concern for practical use
Ease of analysis	For easier validation and testing of strength

Data Encryption Standards (DES)

In 1973, the National Bureau of Standards (NBS) issued a request for proposals for a national cipher standard

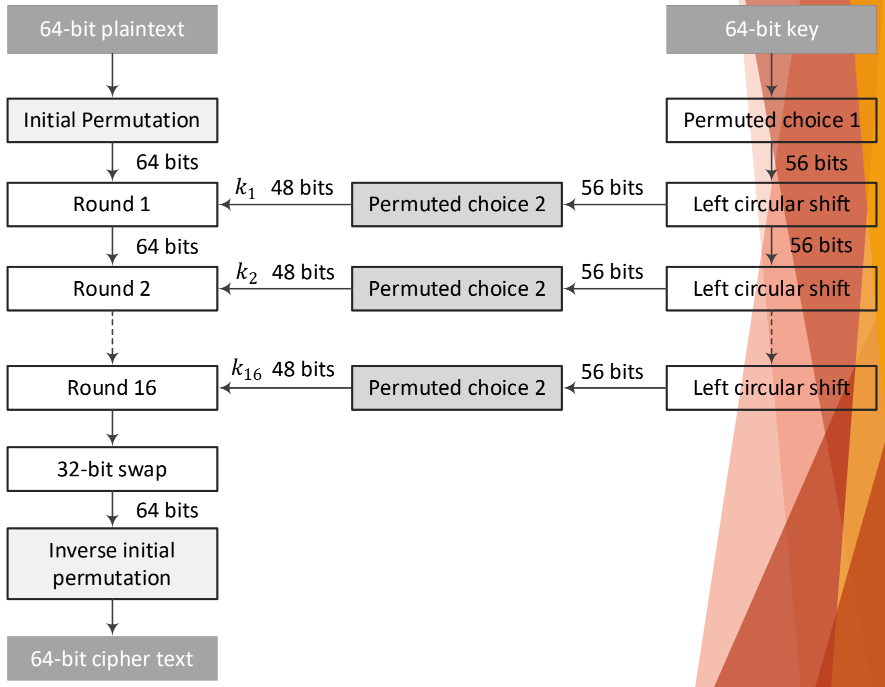
The modified LUCIFER from IBM was adopted in 1977 by the National Bureau of Standards as Federal Information Processing Standard 46 (FIPS PUB 46), also known as the DES

- DES is theoretically broken using Differential or Linear Cryptanalysis (requires 2^{47} and 2^{43} plaintext – thus these attacks are impractical)
- DES is subject to brute force attacks (2^{56} keys to check). With specialized hardware, it takes less than 1 day.

It is still standardized for legacy systems, with either AES or triple DES for new applications

DES Encryption Overview

- It takes input of 64-bit data and a 56-bit key
- The basic process for enciphering a 64-bit data block consists of
  - an Initial Permutation (IP) of the 64 bit plaintext,
  - a Permuted Choice 1 (PC1) of the key, which selects 56 bits out of the 64-bit input in two 28-bit halves;
  - 16 rounds of the same function that involves both permutation and substitution functions, with a generated 48-bit subkey in each of the 16 rounds, and
  - a final permutation (i.e. the inverse IP) before the left and right swap of the 32-bits after the round 16
- Except for the initial and final permutations, DES has the exact structure of a Feistel cipher



DES Decryption

- Decryption must unwind steps of data computation
- With Feistel design, do encryption steps again using subkeys in reverse order (k_{16}, k_{15}, …, k_2, k_1)
  - IP undoes final FP step of encryption
  - 1st round with k_{16} undoes 16th encrypt round
  - ….
  - 16th round with k_1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

DES Security

Avalanche effect:

- It is a desirable property for an encryption algorithm
- A small change in either the plaintext or the key should produce a significant change in the ciphertext
- A change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext
- It make attempts to “home-in” by guessing keys impossible
- It is proved that DES exhibits a strong avalanche effect

Key size:

- DES has a key size of 56 bits. There are $2^{56} \approx 7.2 \times 10^{16}$ keys
- DES could be broken within a day as shown in 1999 with the capability of an average computer and some assumptions
  - The time is much shortened nowadays

DES Security

Analytic attacks:

- Cryptanalysis is possible by exploiting the characteristics of DES

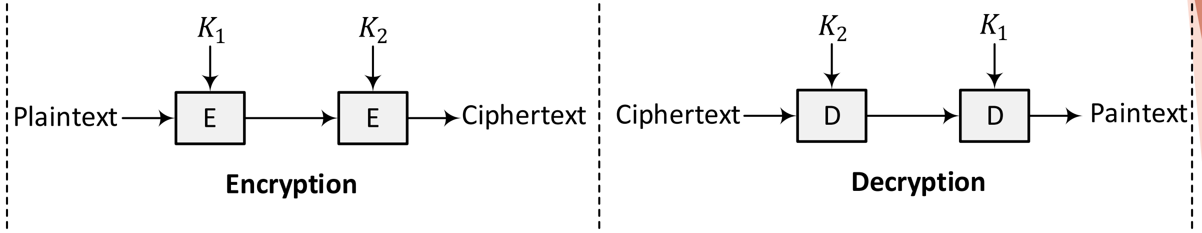
Timing attacks:

- A timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts
- DES appears to be fairly resistant to a successful timing attack

Multiple Encryption and DES

Double DES (2-DES):

- Two keys (e.g., K_1 and K_2) are applied to 2-DES encryptions and decryptions
- $C = E_{K_2}\left(E_{K_1}(P)\right)$ and
- $P = D_{K_1}\left(D_{K_2}(C)\right)$



One would expect the key length to be $\mathrm{len}(K_1) + \mathrm{len}(K_2) = 2\mathrm{n}$

- n=56 => 2-DES has an equivalent key of 112 bits

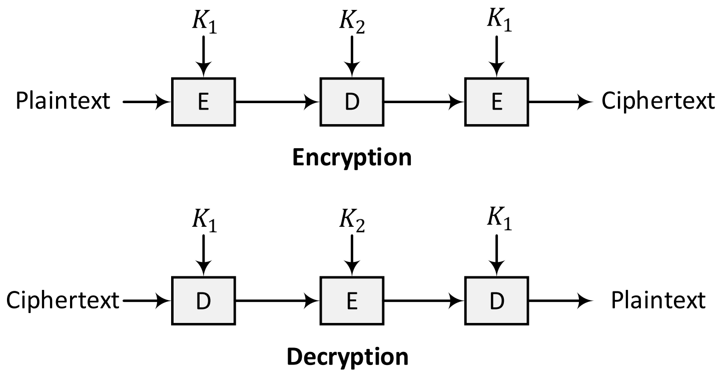
“Meet in the middle” attack makes the actual 2-DES key length just n+1

- MITM attack attempts to find the keys by using both the range (ciphertext) and domain (plaintext) of the composition of several functions (or block ciphers) such that the forward mapping through the first functions is the same as the backward mapping (inverse image) through the last functions, quite literally meeting in the middle of the composed function

Multiple Encryption and DES

Triple DES (3-DES)

- Two keys (e.g., K_1 and K_2) are applied to 3-DES
- $C = E_{K_2}\left(D_{K_2}\left(E_{K_1}(P)\right)\right)$ and
- $P = D_{K_1}\left(E_{K_2}\left(D_{K_1}(C)\right)\right)$



Triple DES with three independent keys has a key length of 168 bits (three 56-bit DES keys), but due to the meet-in-the-middle attack, the effective security it provides is only 112 bits

Thus, just use 2 keys (which are also shorter to share)