A (Brief) History of Cryptography

Ozalp Babaoglu

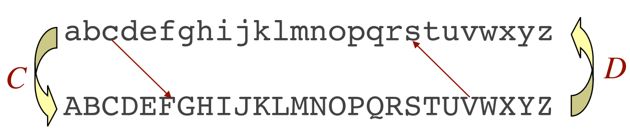
ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

Steganography

- From the Greek ***steganós*** (στεγανός) — "covered", "concealed", and -***graphia*** (γραφή) — "writing"
- The art of concealing information within a file, message, image, or video
- Form of "security through obscurity"
- Can be made "keyless"
- Examples:
  - Message written in secret ink on paper
  - Information contained in the LSB of image or sound files

Caesar Cipher

- A *substitution* cipher
- Each letter of the plaintext is replaced by a unique letter in the ciphertext
- Which letter?
- In the case of Caesar Cipher, the relation between the letter in the plaintext and that in the ciphertext is obtained through a ***cyclic left shift***
- Decryption is obtained through a ***cyclic right shift***
- Example: shift 3



Caesar Cipher

```
ignavi coram morte quidem animam trahunt,
audaces autem illam non saltem advertunt
LJQDYLCFRUDPCPRUWHCTXLGHPCDQLPDPCWUDKXQWC
CDXGDFHVCDXWHPCLOODPCQRQCVDOWHPCDGYHUWXQW
```

- Number of positions to shift becomes the secret key of the cipher
- Let $pos(\alpha)$ be the position of letter α in the alphabet,
- Let $chr(j)$ be the character in the j-th position of the alphabet,
- Let k be the key,
- Let m_i and c_i the i-th characters in the plaintext and ciphertext, respectively

$$C(m_i) = chr\,(pos(m_i) + k) \textbf{ mod } 26$$
$$D(c_i) = chr\,(pos(c_i) - k) \textbf{ mod } 26$$

Caesar Cipher

- Trivial to carry out a brute-force attack because:
  - The encryption and decryption algorithms are known
  - The number of possible keys is very small (only 25 different keys)
  - The language of the plaintext is known and easily recognizable
- Example: Cryptanalysis of

  “`AJSN ANIN ANHN`”

Caesar Cipher

- Brute-force cryptanalysis of ciphertext “`AJSN ANIN ANHN`”

```
Caesar(1) = zirm zmhm zmgm
Caesar(2) = yhql ylgl ylfl
Caesar(3) = xgpk xkfk xkek
Caesar(4) = wfoj wjej wjdj
Caesar(5) = veni vidi vici  ⟵
Caesar(6) = udmh uhch uhbh
Caesar(7) = tclg tgbg tgag
Caesar(8) = sbkf sfaf sfzf
Caesar(9) = raje reze reye
Caesar(10) = qzid qdyd qdxd
        …
```


Substitution Ciphers

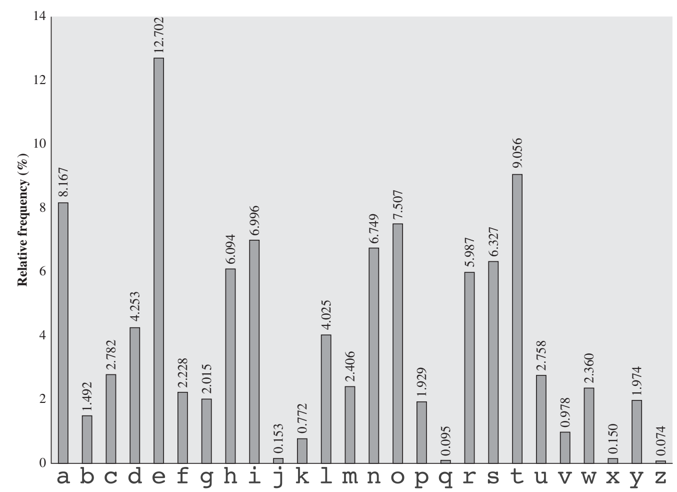
- Instead of substituting letters through a cyclic shift, we can substitute them through a permutation of the alphabet, which becomes the key:

  `abcdefghijklmnopqrstuvwxyz`

  `BFRULMZQWJEASOVKHXPGDTIYCN`
- For an alphabet of 26 letters, there are 26! possible keys since there are 26! possible permutations of 26 letters
- Cryptanalysis through “brute force” becomes non practical
- However, ***statistical*** cryptanalysis is still possible

Substitution Ciphers

- Relative frequency of letters in English text



Substitution Ciphers

- Consider the ciphertext

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

- Frequency of the letters in the ciphertext

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

Substitution Ciphers

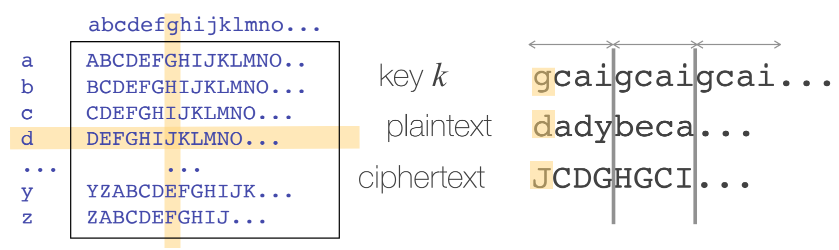
- The two most-frequent cipher letters `P` and `Z` probably correspond to the two most-frequent plain letters `e` and `t`
- Cipher letters `S,U,O,M,H,D` probably correspond to plain letters `a,o,i,n,s,h`
- The least frequent cipher letters `A,B,G,Y,I,J` probably correspond to the least frequent plain letters `v,k,j,x,q,z`

Substitution Ciphers

- To resolve ambiguities, we can look at two-letter combinations
- In ciphertext, the most common 2-letter sequence is `ZW`
- In English language texts, the most common 2-letter sequence is `th`
- So, `Z` is most likely `t` and `W` is `h` meaning `P` is `e`
- Thus, the sequence `ZWP` in the ciphertext is probably `the`

Polyalfabetic Ciphers

- Use multiple substitution ciphers depending on the position of the letter in the plaintext



- Monoalfabetic for every | k | characters
- Statistical attack still possible but becomes more difficult
- Basis for “rotor machines” like *Enigma* and *Purple* that were used during world war 2

Secret-Key Cryptography
Polyalfabetic Ciphers

- Instead of substituting single letters of the plaintext, substitute blocks of letters
- Example (blocks of 3)
  - `AAA → SOM`
  - `AAB → PLW`
  - `ABA → RTQ`
  - `ABB → SLL`
  - …
- Doing so hides information regarding the frequency of single letters and pairs of letters

Secret-Key Cryptography
Permutation Ciphers

- Maintain the same letters in the ciphertext as in the plaintext, but change their order
- For example,

```
4312567   key
attackp
ostpone   plaintext
duntilt
hreepmx
```

Ciphertext: `ttne aptetsuraodhcoipknlmpetx`

Secret-Key Cryptography
Permutation Ciphers

- Can be repeated multiple times

```
4312567   key
ttneapt
etsurao   plaintext
dhcoipk
nlmpetx
```

output: `nscmeuoptthltednariepapttokx`

Secret-Key Cryptography
Permutation Ciphers

`01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28`

After one permutation:

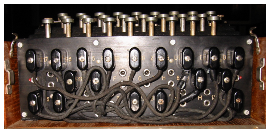
`03 10 17 24 04 11 18 25 02 09 16 23 01 08 15 22 05 12 19 26 06 13 20 27 07 14 21 28`

After two permutations:

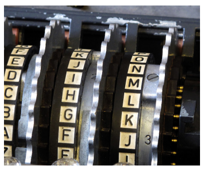
`17 09 05 27 24 16 12 07 10 02 22 20 03 25 15 13 04 23 19 14 11 01 26 21 18 08 06 28`

Enigma

Portable electro-mechanical device invented after WW I and used extensively by Germany to encode and decode messages exchanged with troops and with U-Boats during WW II

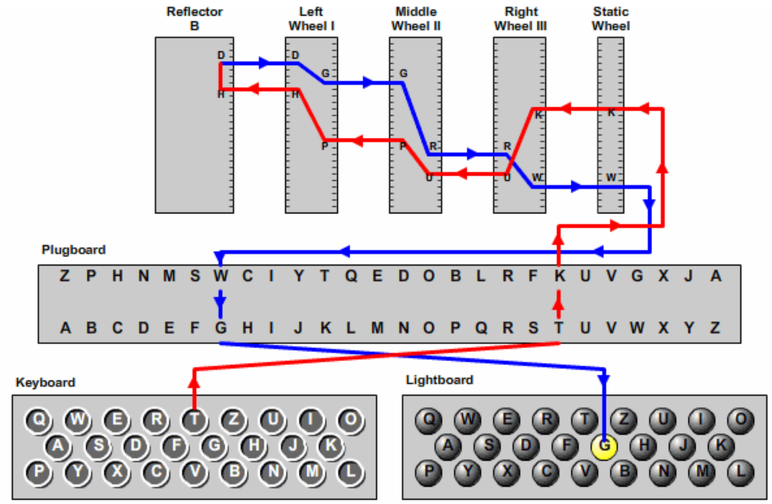


Plugboard: wired to correspond to a specific initial substitution



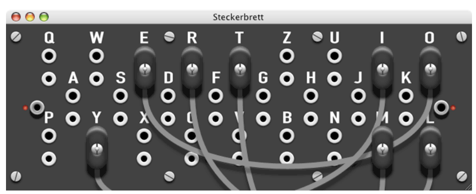
3 Rotors initialized to a specific setting, one or more rotors "step" with each key press

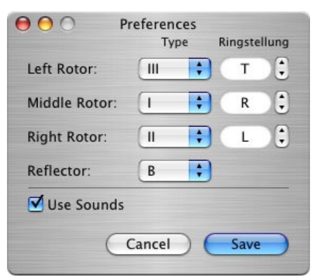
How Enigma Worked



How Enigma Worked

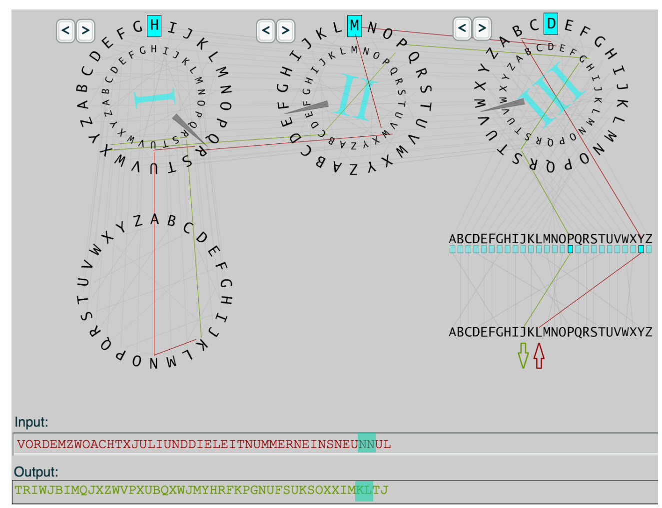
- Enigma Rotor Machine Simulator (MacOSX executable)





How Enigma Worked

- Enigma Cyphering Simulator (Adobe Flash based)



Breaking Enigma

- The plugboard and the rotors define the "key" with 158,962,555,217,826,360,000 (~10^{21}) possible settings
- By the early 1940's, a team of British cryptologists led by Alan Turing assembled at Bletchley Park, Buckinghamshire UK were able to decode thousands of intercepted messages per day
- Relied on earlier work by Polish cryptologists, Marian Rejewski, Jerzy Różycki and Henryk Zygalski
- And on electro-mechanical US Navy "Bombes"

- Breaking Enigma is widely considered to have been decisive to the Allied victory of WW2

"Perfect" Ciphers: One-Time Pad

One-time pad

- *Symmetric* cipher that achieves "perfect computational" secrecy
- *Stream* cipher in that each bit of the ciphertext is determined solely by the corresponding bit of the plaintext and the key
- Based on random strings and modular arithmetic operations
- More of a theoretical concept than a practical solution

One-time pad: example

Plaintext:	1	1	0	1	0	0	0	1	
Key (Pad):	1	0	0	0	1	1	1	0	
⊕	0	1	0	1	1	1	1	1	Ciphertext
⊕	1	1	0	1	0	0	0	1	Plaintext

Based on modular arithmetic:

$c_i = m_i + k_i$ **mod** 2 (also called "exclusive or")

For textual messages: $c_i = m_i + k_i$ **mod** 26

Advantages and Defects

- Advantages:
  - Since each bit of the key is generated at random, knowing one bit of the ciphertext does not provide any information beyond guessing regarding the corresponding bit of the plaintext: guarantees ***computational secrecy***
- Defects:
  - The key is as long as the plaintext message,
  - Self destructs (one-time),
  - Needs to be agreed upon

DES
Data Encryption Standard

History

- In 1973, the National Bureau of Standards (NBS) publishes a "call for proposals"
- IBM submits a proposal for a system similar to an internal product called "Lucifer"
- Soon after, NSA adopts Lucifer under the name DES
- After further studies, DES is certified and made public in 1977
- First example of a robust cipher (with NSA certification) that the research community can study
- Thereafter certified every 5 years

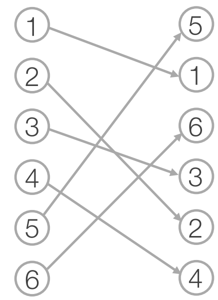
Characteristics of DES

- Symmetric cipher (secret-key cryptography)
- Works in 64-bit *blocks* (*not* a stream cipher)
- 64-bit keys, of which only 56 bits are used (other 8 serve as parity checks)

Basic Operations

- Permutation
- Substitution
- Expansion
- Choice (contraction)
- Circular shift (left or right)

Permutation



P=(5,1,6,3,2,4)

One bit of input determines one bit of output

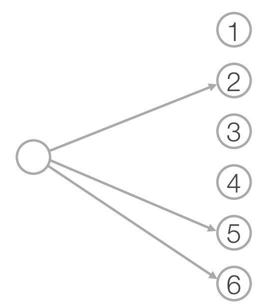
Substitution

- Block of input bits replaced by a unique block of output bits

000		010
001		011
010		100
011	⇨	111
100		110
101		000
110		001
111		101

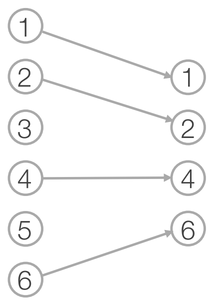
Expansion

- Certain bits of the input are repeated multiple times in the output
- Example:

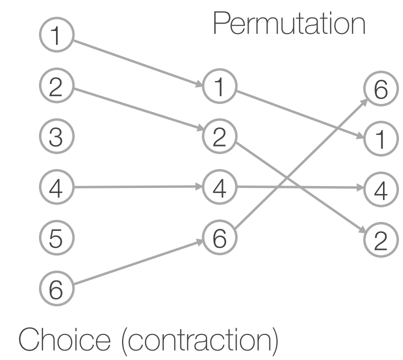


Choice (Contraction)

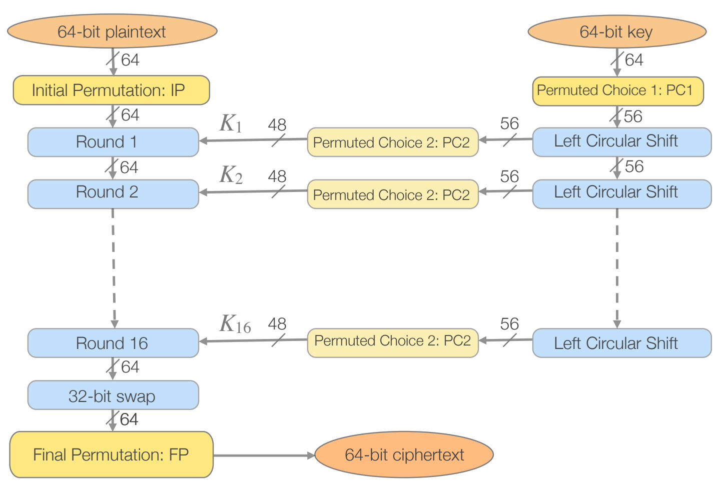
- Certain input bits do not appear int the output (they are ignored)
- Example:



Permuted Choice



DES Overview



DES: IP and FP boxes

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

FP

- IP and FP are inverses

DES: PC1 and PC2 boxes

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

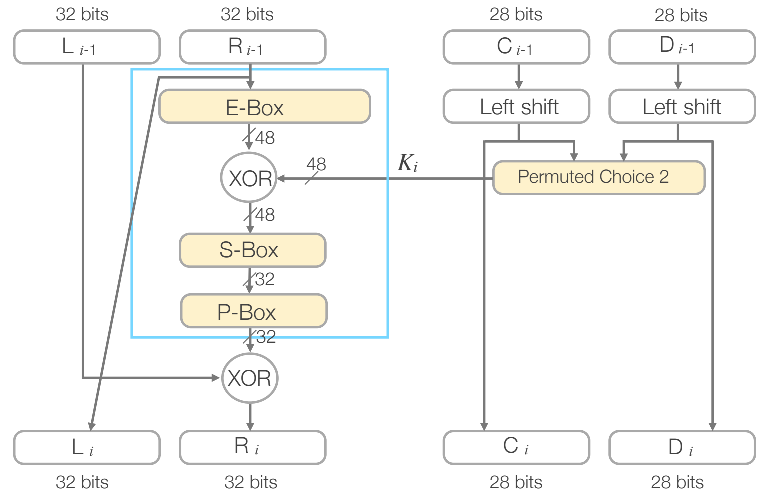
PC1 (64 bits in, 56 bits out)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

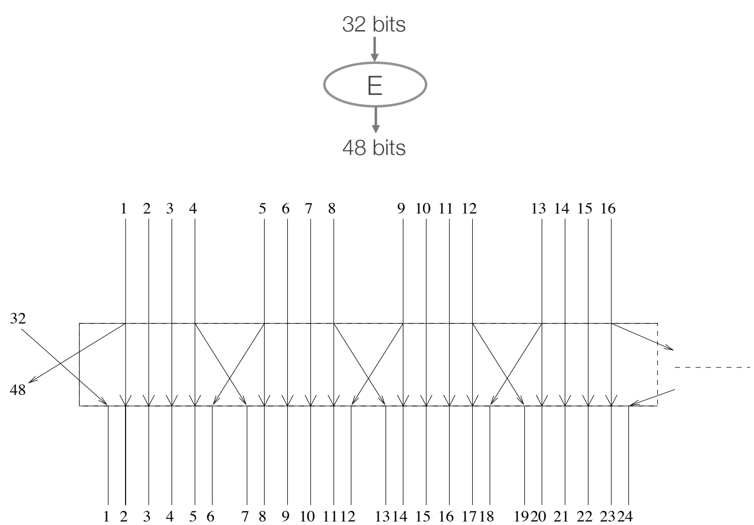
PC2 (56 bits in, 48 bits out)

- Bits 8,16, 24, 32, 40, 48, 56, 64 missing in the PC1 box
- Bits 9,18, 25, 35, 38, 43, 45, 54 missing in the PC2 box

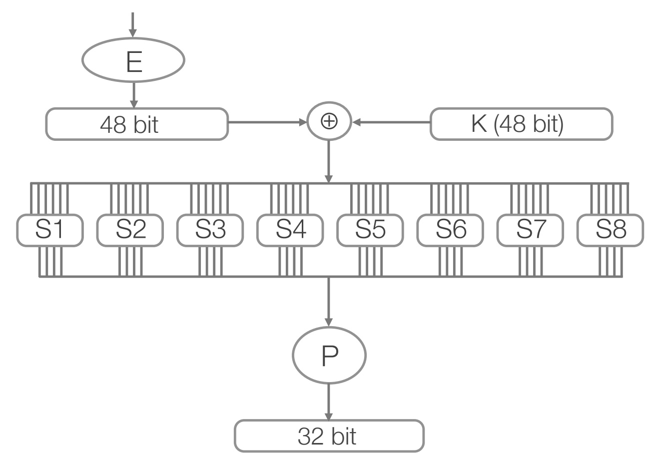
DES: Details of a Round



DES: E-Box

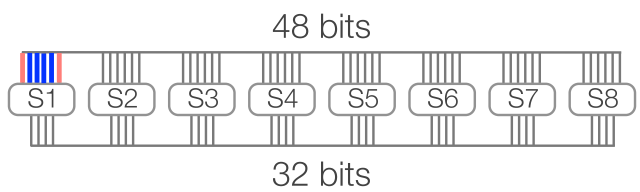


DES: S-Box



DES: S-Box

- Bits 1 and 6 select a row, bits 2-5 select a column to read a 4-bit value from one of eight possible maps



S1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

DES: P-Box

- Straight permutation of 32 bits

```
16 07 20 21 29 12 28 17
01 15 23 26 05 18 31 10
02 08 24 14 32 27 03 09
19 13 30 06 22 11 04 25
```


DES Replacements

- As of 1999, DES is considered *insecure* due to its short key
- More-recent symmetric ciphers that have replaced DES:
  - *Triple-DES* — effectively triples the DES key size
  - *Blowfish* — variable key sizes from 32 bits up to 448 bits
  - *International Data Encryption Algorithm* (IDEA) —128-bit keys
  - *Advanced Encryption Standard* (AES) — key sizes of 128, 192 or 256 bits

Brute-Force Attacks on Symmetric Ciphers

- Average time required for exhaustive key search as a function of key size

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Brute-Force Attacks on Symmetric Ciphers

- A password-cracking expert has unveiled a computer cluster that can cycle through as many as 350 billion guesses per second

Welcome to Radeon City, population: 8. It's one of five servers that make up a high-performance password-cracking cluster.