Introduction to Cryptography and Information Security

Part 2
Symmetric Encription
Algorithms

Jorge Camargo, PhD

Agenda

- Classic Cryptography
  - Introduction to Notation
  - Block Ciphers
    - Substitution Ciphers
      - Monoalphabetic Substitution Ciphers
        - Polygraphic

          Example: Porta Cipher

          Example: Playfair Cipher
        - Unilateral

          Example: Caesar’s Cipher

          Example: Rot13 Cipher

          Example: Dots Cipher
        - Multilateral

          Example: Garbage-in-between Cipher
      - Polyalphabetic Substitution Ciphers

        Example: Vigenere Cipher

        Example: Hills Cipher
      - Homophonic Substitution Ciphers

        Example: Homophonic Cipher
    - Transposition Ciphers
      - Example: Turning Grille
    - Product Ciphers
      - Example: Lucifer Cipher
      - Example: Hayhanen Cipher
    - Modes of Operations
      - ECB: Electronic Code Block
      - CBC: Cipher Block Chaining
      - CFB: Cipher Feedback
      - OFB: Output Feedback ⋆ CTR: CounTeR

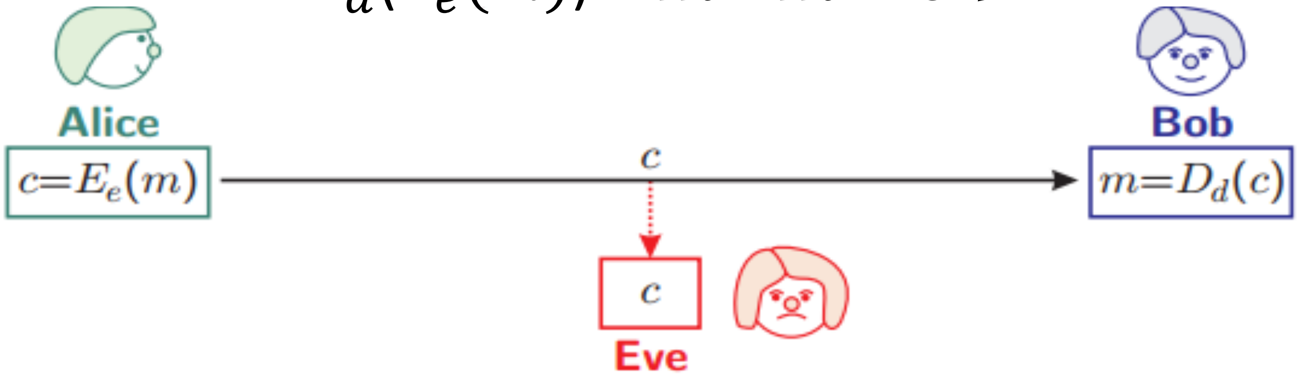
Introduction to Notation

- $\mathcal{A}$, the alphabet of definition, eg. $\mathcal{A}$ = {0, 1},
- $\mathcal{M}$ is the message space; set of strings over $\mathcal{A}$, $\mathcal{M} \subseteq \mathcal{A}*$
- $\mathcal{C}$ is the ciphertext space; set of strings over $\mathcal{A}'$
- $\mathcal{K}$ denotes the key space; each $e \in \mathcal{K}$ uniquely determines a bijection
- m is a plaintext (message), $m \in \mathcal{M}$
- $E_e : \mathcal{M} \rightarrow \mathcal{C}$ is the encryption function (bijection)
- $D_d : \mathcal{C} \rightarrow \mathcal{M}$ is the decryption function (bijection)
- Applying E_e (or D_d) is called encryption (or decryption).

Introduction to Notation (cont.)

- An encryption scheme (or cipher) consist of a set $\{E_e : e \in \mathcal{K}\}$ and a corresponding set $\{D_d : d \in \mathcal{K}\}$ with the property that $\forall e \in \mathcal{K}\ \exists$ a unique $d \in \mathcal{K}$ s.t. $D_d = E_e^{-1}$; i.e.,

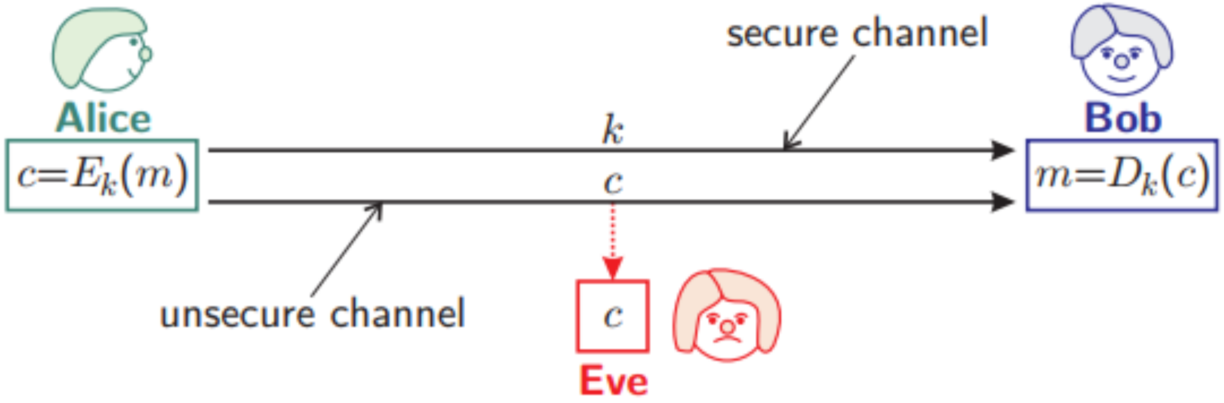
$$D_d(E_e(m)) = m\ \forall m \in \mathcal{M}$$



- To construct an encryption scheme requires fixing a message space $\mathcal{M}$, a cipher space $\mathcal{C}$, and a key space K, as well as encryption transformation $\{E_e : e \in \mathcal{K}\}$ and corresponding $\{D_d : d \in \mathcal{K}\}$.

Symmetric-Key Encryption

Symmetric-key encryption uses the same key (or are easily derived from each other) to encrypt and decrypt. *e* = *d* = *k*.



Is simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way.

Symmetric-key encryption schemes are often characterized as block ciphers or stream ciphers although the distinction can be fuzzy.

Block Ciphers

A block cipher is a symmetric-key encryption scheme that breaks up the plaintext message into strings (blocks) of a fixed length t over an alphabet $\mathcal{A}$ and encrypts one block at a time.

There are three classes of block ciphers

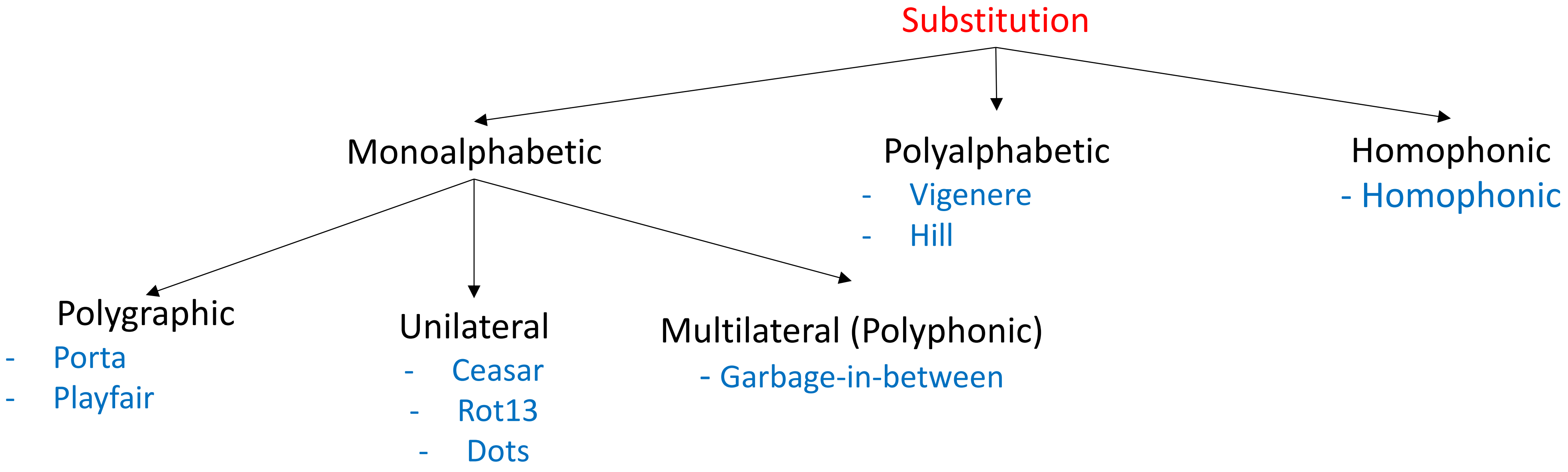
- Substitution ciphers: replace symbols (or groups of symbols) by other symbols or groups of symbols.
- Transposition ciphers: simply permutes the symbols in a block.
- Product ciphers: is a composite of substitution and transposition ciphers.

Substitution Ciphers

A substitution cipher is a cipher that replaces each plaintext symbol (or group of symbols) with another ciphertext symbol. The receiver deciphers using the inverse substitution.

Encryption using substitution cipher is one of the simplest used methods even by ancient cryptographers.

There are three basic types of substitution ciphers:



Monoalphabetic Substitution Ciphers

The main idea of these ciphers is an one by one substitution of a symbol from plaintext to corresponding symbol in ciphertext.

There are three types of monoalphabetic ciphers

- Polygraphic: the plaintext units are consistently more than one plaintext letter long.
- Unilateral: the ciphertext unit is always one character long.
- Multilateral (Polyphonic): the ciphertext unit is more than one character in length. The ciphertext characters may be letters, numbers, or special characters.

Some examples of monoalphabetic ciphers are:

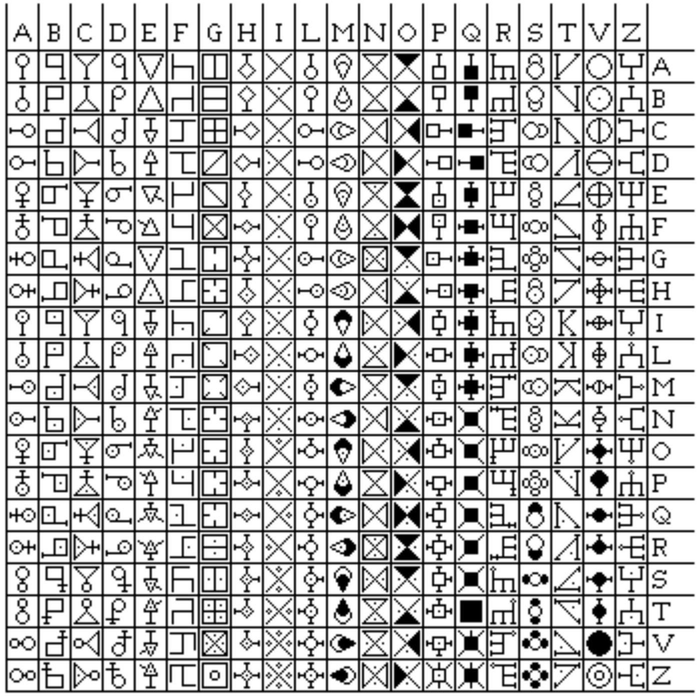
- Porta Cipher: Polygraphic monoalphabetic substitution cipher.
- Playfair Cipher: Polygraphic monoalphabetic substitution cipher.
- Caesar’s Cipher: Unilateral monoalphabetic substitution cipher.
- Rot13 Cipher: Unilateral monoalphabetic substitution cipher.
- Dots Cipher: Unilateral monoalphabetic substitution cipher.
- Garbage-in-between Cipher: Multilateral monoalphabetic substitution cipher.

The prefixes mono- and uni- mean one, and graphic and literal refer to letters or other characters.

Porta Cipher (Polygraphic)

(Giovanni Battista della Porta, 1563)

Based on the following 20×20 tableau filled with 400 unique glyphs.


A B C D E F G H I L M N O P Q R S T V Z
A B C D E F G H I L M N O P Q R S T V Z

Use the table to give different symbols for every pair of letters.

Playfair Cipher (Polygraphic)

(Sir Charles Wheatstone, popularized by Baron Lyon Playfair, 1854)

To encipher, pick a keyword and write it into a 5×5 square, omitting repeated letters and combining I and J in one cell.

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

We need to break the plaintext up into two-letter groups. If letters in a pair are the same, insert X between them. If there is only one letter in the last group, add X to it. To encrypt find each two-letter group in the square and if they are:

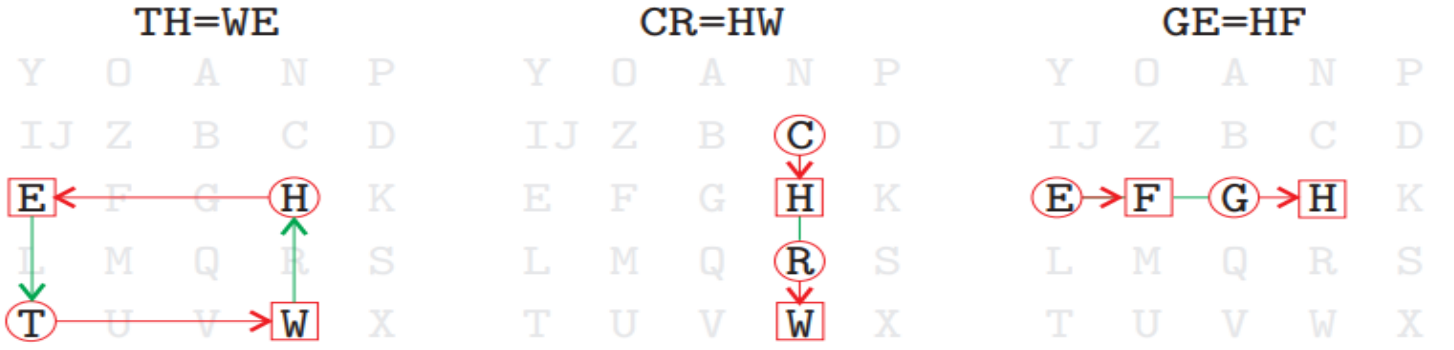
- in the same column – use the letter below it as the cipher text,
- in the same row – use the letter to the right as the cipher text,
- neither – each letter is exchanged with the letter at the intersection of its own row and the other column.

For deciphering, the rules are the exact opposite.

Example: Encrypt “THIS SECRET MESSAGE IS ENCRYPTED”

```
TH IS SE CR ET ME SX SA GE IS EN CR YP TE DX
```

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



```
TH IS SE CR ET ME SX SA GE IS EN CR YP TE DX
WE DL LK HW LY LF XP QP HF DL HY HW OY YL KP
```

It was in military use from the Boer War through World War II.

Exercise: Using the same key above, decipher the following ciphertext:

```
ZO MH LC HY ZK MN SO NQ DL KT OQ CY KI EC LK SO YI EQ PQ RX EY KR WM NS
DL GY LD GF AB YA QN YE AP GN IX PG HY YS NB HT EC TL KF VN RP YT PU PF
CY EB YA WM KI MP LF UZ LH TC YH NP CK KL LY YT KI GB DH CY EC RD GN CL
GO IH YE TY KI XO UY VN SC LX KF MX PW
```

Caesar’s Cipher (Unilateral)

(Julius Caesar, 50-60 BC)

$\mathcal{A}$ = {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}
$\mathcal{M}$ = C = strings of length 5
k = permutations of 3

Example: Cipher the message “RETURN TO ROME”

plaintext = RETUR NTORO ME
ciphertext = UHWXU QWRUR PH

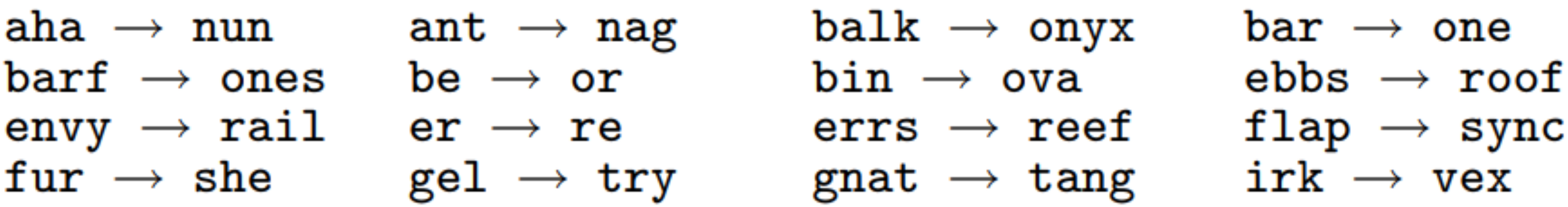
Exercise: Decrypt WKLVL VHAWU HPHOB LQVHF XUHHQ FUBSW LRQGR QRWXV HLWWR SURWH FWYDO XDEOH LQIRU PDWLR Q and LWLVF ODLPH GWKHH DUOLH VWNQR ZQUHI HUHQF HWRWK LVWBS HRIFL SKHUL VLQWK HNDPD VXWUD ZKLFK VDBVZ RPHQV KRXOG OHDUQ WKHDU WRIVH FUHWZ ULWLQ JWRFR QFHDO WKHLU OLDVR QV and Encrypt the message “we will attack at dawn through the left flank”.

This type of cipher is easily defeated by frequency analysis

Rot13 Cipher (Unilateral)
(USENET net.jokes newsgroup, mid-1980s)

Replaces each letter with the letter thirteen places down the alphabet. A becomes N, B becomes O and so on.

Examples:



aha → nun	ant → nag	balk → onyx	bar → one
barf → ones	be → or	bin → ova	ebbs → roof
envy → rail	er → re	errs → reef	flap → sync
fur → she	gel → try	gnat → tang	irk → vex

The algorithm was used in online forums as a means of hiding joke punchlines, puzzle solutions, movie and story spoilers and offensive materials from the casual glance. ROT13 has been described as the“Usenet equivalent of a magazine printing the answer to a quiz upside down”

Exercise: Decrypt VSGUR AFNUN FGVZR GBERN QZLRZ NVYVJ VFUGU RLFRA QZRNO YBBQL

Dots (Unilateral)

Symbols in the plaintext will be replaced by squares with or without points and with or without surrounding lines using the following rules:

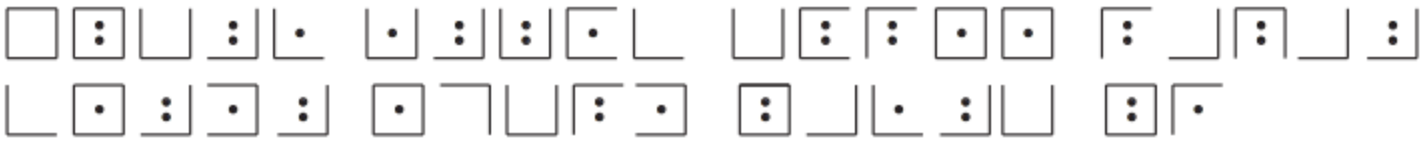
A:	B:	C:
D:	E:	F:
G:	H:	I:

J·	K·	L·
M·	N·	O·
P·	Q·	R·

S	T	U
V	W	X
Y	Z	

Example: Encrypt the message "WE TALK ABOUT FINNISH SAUNA MANY TIMES LATER"

WE TALK ABOUT FINNISH SAUNA MANY TIMES LATER

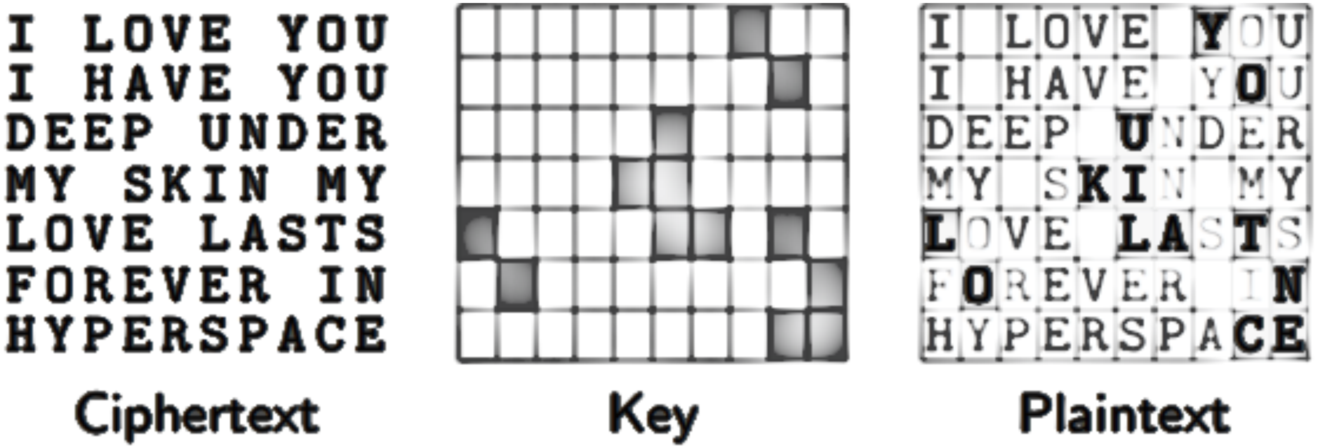


Garbage-in-between Cipher (Multilateral)

(Cardinal Richelieu, 1626)

In garbage-in-between cipher, the message is written in positions determined by the key. After that, empty positions are filled by arbitrary letters, in a misleading way so that the whole message looks like a different story.

Example:



I LOVE YOU
I HAVE YOU
DEEP UNDER
MY SKIN MY
LOVE LASTS
FOREVER IN
HYPERSPACE

Ciphertext

Key

Plaintext

The encrypted message is YOU KILL AT ONCE

Polyalphabetic Substitution Ciphers

The main idea behind the polyalphabetic cipher is that a single symbol can be encrypted to several different symbols instead of just one.

The reason behind using polyalphabetic ciphers is to flatten frequency distributions.

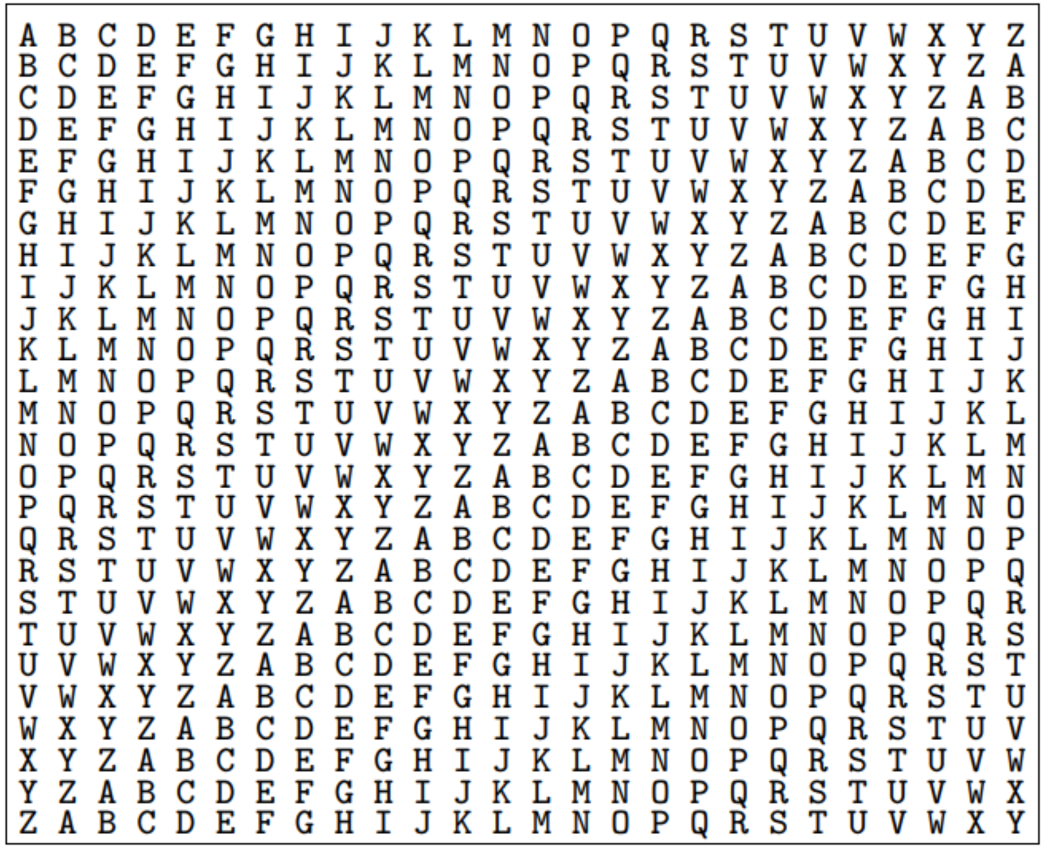
Following slides show two examples of polyalphabetic ciphers:

- Vigènere Cipher
- Hill Cipher

Vigènere Cipher (Polyalphabetic)

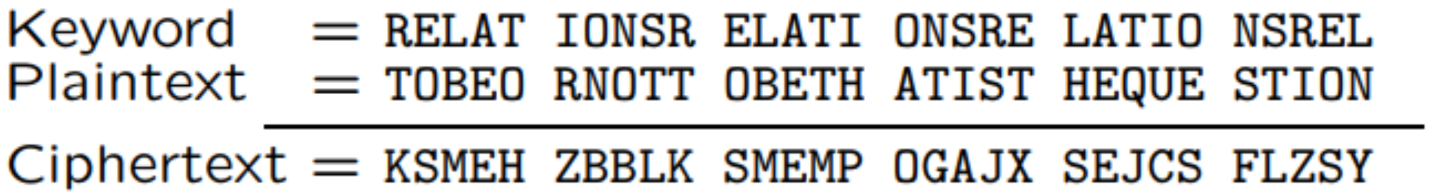
(Blaise de Vigenere, 16-th century)

Based on the tableau shown below and the use of a keyword.



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

Example: Encrypt the message “TO BE OR NOT TO BE THAT IS THE QUESTION”, using the keyword=“RELATIONS” and t = 5.



Keyword	=	RELAT	IONSR	ELATI	ONSRE	LATIO	NSREL
Plaintext	=	TOBEO	RNOTT	OBETH	ATIST	HEQUE	STION
Ciphertext	=	KSMEH	ZBBLK	SMEMP	OGAJX	SEJCS	FLZSY

Exercise: Encrypt the message “THERE IS A SECRET PASSAGE BEHIND THE PICTURE FRAME”, keyword=“CRYPTO”, and *t* = 3.

This cipher is the basic idea behind the Enigma machine, which used three rotors to encode.

For 300 years the Vigènere cipher was considered to be unbreakable. Then, in 1863, a Prussian military devised a method to determine the length of the keyword and then divide the message into a simpler form to which letter frequency analysis could be applied.

Hill Cipher (Polyalphabetic)
(Lester S. Hill, 1929)

Each letter is treated as a digit in base 26: `A = 0, B = 1`, and so on.

Let $\mathcal{M} = C = (\mathbb{Z}^{26})^t$

The idea is to take t linear combinations of the t alphabetic characters in one plaintext element, thus producing the m alphabetic characters in one cipher element.

For example, if $t = 2$, we could write the plaintext element as $m = (m_1, m_2)$ and a ciphertext element as $c = (c_1, c_2)$, Here, c_1 would be a linear combination of m_1 and m_2, as would c_2. We might take

$$c_1 = 11\, m_1 + 3\, m_2$$

$$c_2 = 8\, m_1 + 7\, m_2$$

which can be written more succinctly in matrix notation as follows:

$$(c_1, c_2) = (m_1, m_2) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

We use the inverse matrix K^{-1} to decrypt. The ciphertext is decrypted using the formula $m = cK^{-1}$.

The inverse matrix to a matrix A (if it exists) is the matrix A^{-1} such that AA^{-1} = A^{-1}A = I, where I is the identity matrix (matrix with 1's on the main diagonal and 0's elsewhere).

Not all matrices have inverses, but if an inverse exists, it is unique.

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$$

Since

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} = \begin{pmatrix} 11*7+8*23 & 11*18+8*11 \\ 3*7+7*23 & 3*18+7*11 \end{pmatrix}$$

$$= \begin{pmatrix} 261 & 286 \\ 182 & 131 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Remember that all operations are done modulo 26.

Example: Encrypt the message “JULY” using K =$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$.

(9, 20) → “JU” and (11, 24) → “LY”, then

(9, 20) $\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$= (99 + 60, 72 + 140) = (3, 4)

And

(11, 24) $\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$ = (121 + 72, 88 + 168) = (11, 22)

Hence the encrypted message of “JULY” is “DELW**”**.

Exercise: Decrypt VKFZRVWTIAZSMISGKA using the same key as above.

Modular Inverse of a Matrix

for a 2x2 matrix and mod 26

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$A^{-1} = (\mathrm{det}A)^{-1}AdjA, \quad detA = ad - bc, AdjA = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Where $(\mathrm{det}A)^{-1}$ is the *modular inverse* of *det*A mod 26 (i.e., $(\mathrm{det}A)^{-1}$ detA mod 26 = 1) and *AdjA* is the adjoint matrix of A.

If detA = 0, the matrix is non-invertible.

If *g*cd(detA, 26)≠ 1, the matrix is non-invertible

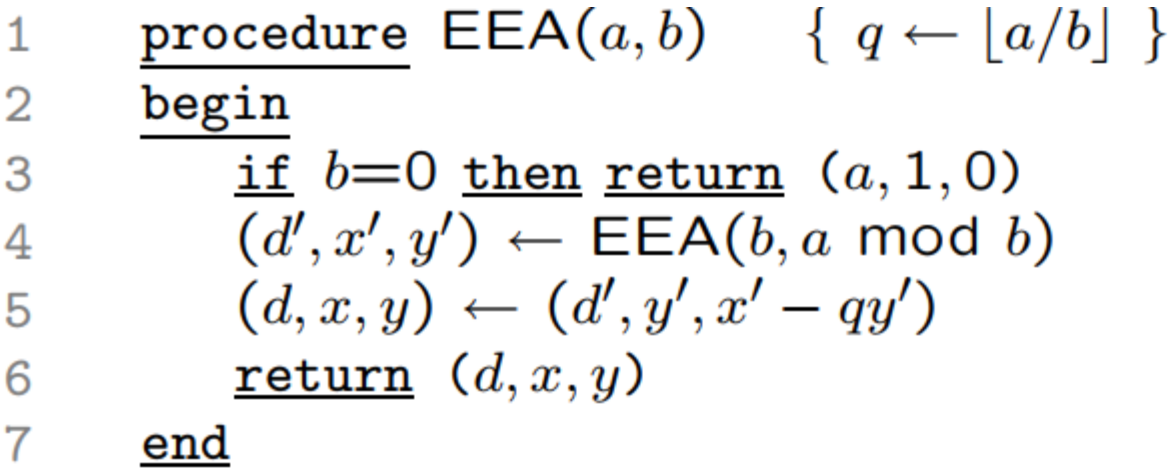
The modular inverse can be computed using the Extended Euclidean Algorithm (EEA)

EEA - Extended Euclidean Algorithm

INPUT: a, b ∈ $\mathbb{Z}^+$a ≥ b

OUTPUT: (d, x, y), d = $gcd(\ a, b)$; $x, y \in \mathbb{Z} : ax + by = d$

Pseudo-code:

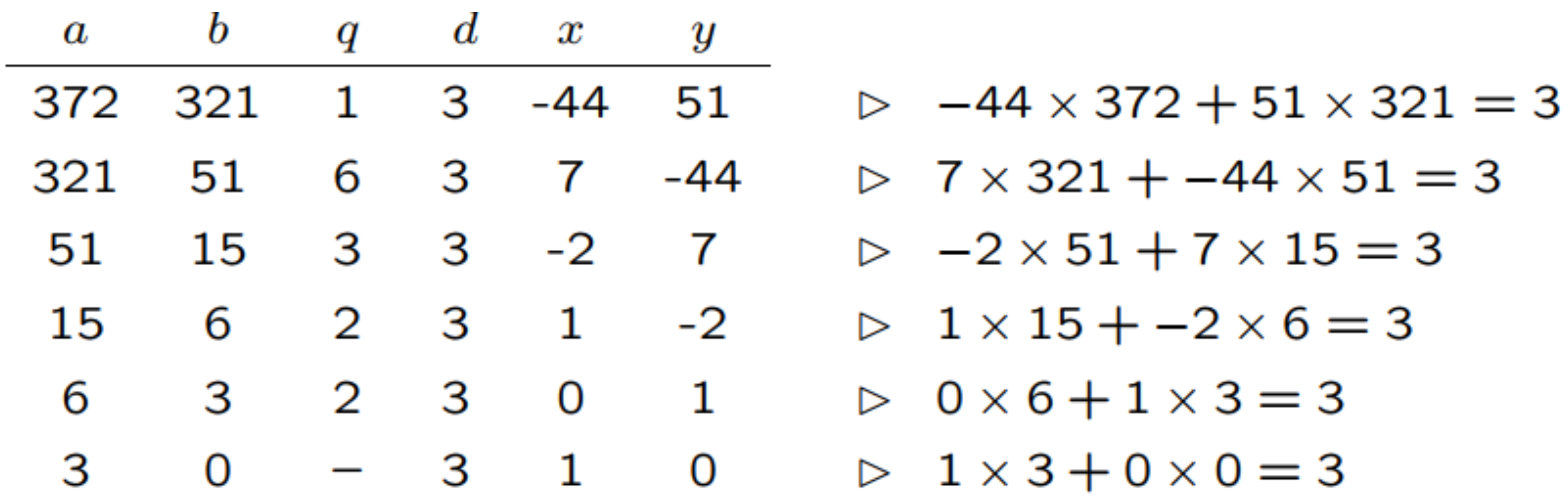


```
procedure EEA($a, b$)    { $q \leftarrow \lfloor a/b \rfloor$ }
begin
    if $b$=0 then return $(a, 1, 0)$
    $(d', x', y') \leftarrow$ EEA($b, a$ mod $b$)
    $(d, x, y) \leftarrow (d', y', x' - qy')$
    return $(d, x, y)$
end
```

EEA - Extended Euclidean Algorithm

Example

- Compute EEA(372, 321)



a	b	q	d	x	y	
372	321	1	3	-44	51	$\triangleright$ $-44 \times 372 + 51 \times 321 = 3$
321	51	6	3	7	-44	$\triangleright$ $7 \times 321 + -44 \times 51 = 3$
51	15	3	3	-2	7	$\triangleright$ $-2 \times 51 + 7 \times 15 = 3$
15	6	2	3	1	-2	$\triangleright$ $1 \times 15 + -2 \times 6 = 3$
6	3	2	3	0	1	$\triangleright$ $0 \times 6 + 1 \times 3 = 3$
3	0	–	3	1	0	$\triangleright$ $1 \times 3 + 0 \times 0 = 3$

Modular Inverse of a Matrix

The modular inverse can be computed using the Extended Euclidean Algorithm (EEA)

$(\det A)^{-1}$ is the *modular inverse* of *det*A mod 26

$(\det A)^{-1}$ (detA) mod 26 = 1

$$ab \equiv 1 \pmod{m}$$

$$ax + my = 1$$

This equation has solution only if $\gcd(a, m) = 1$

Thus, *x* is the modular multiplicative inverse of a modulo *m.*

Modular Inverse of a Matrix

The modular inverse can be computed using the Extended Euclidean Algorithm (EEA)

(d, x, y), d = gcd(a, b); $x, y \in \mathbb{Z}$: $ax + by = d$

Finding the modular inverse of $\mathrm{A} = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$

$A^{-1} = (\det A)^{-1} AdjA \; mod \; 26$

detA = 77 – 24 = 53

$\mathrm{AdjA} = \begin{pmatrix} 7 & -8 \\ -3 & 11 \end{pmatrix}$

A^{-1} = 53^{-1} $\begin{pmatrix} 7 & -8 \\ -3 & 11 \end{pmatrix}$ $mod \; 26$

$53 \cdot \mathrm{x} \equiv 1 \; mod \; 26$

$gcd(53, 26) = ?$

$x = ?$

A^{-1} = 1 · $\begin{pmatrix} 7 & -8 \\ -3 & 11 \end{pmatrix}$ mod 26

= $\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$

Modular Inverse of a Matrix

for a 3x3 matrix and mod 26

$$M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Let $\mathrm{A} = \det\begin{pmatrix} e & f \\ h & i \end{pmatrix}, \mathrm{B} = -\det\begin{pmatrix} d & f \\ g & i \end{pmatrix}, \mathrm{C} = \det\begin{pmatrix} d & e \\ g & h \end{pmatrix}$

Let $\mathrm{D} = -\det\begin{pmatrix} b & c \\ h & i \end{pmatrix}, \mathrm{E} = \det\begin{pmatrix} a & c \\ g & i \end{pmatrix}, \mathrm{F} = -\det\begin{pmatrix} a & b \\ g & h \end{pmatrix}$

Let $\mathrm{G} = \det\begin{pmatrix} b & c \\ e & f \end{pmatrix}, \mathrm{H} = -\det\begin{pmatrix} a & c \\ d & f \end{pmatrix}, \mathrm{I} = \det\begin{pmatrix} a & b \\ d & e \end{pmatrix}$

The signs used are in a chess board pattern

$$\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$$

$detM = aA + bB + cC$, or $detM = aA + dD + gG$

$detM = dD + eE + fF$, or $detM = bB + eE + hH$

$detM = gG + hH + iI$, or $detM = cC + fF + iI$

$$A^{-1} = (detM)^{-1} \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix}$$

$(\mathrm{det}M)^{-1}$ is the modular inverse of detM mod 26 s.t. $(\mathrm{det}M)^{-1}$ detM mod 26 = 1

Homophonic Substitution Ciphers

Homophonic substitution ciphers are ciphers in which several different ciphertext letters can be used to stand for some plaintext letters.

Typically in a homophonic substitution cipher, the cipher alphabet (the symbols which can appear in a ciphertext) is larger than the plaintext alphabet.

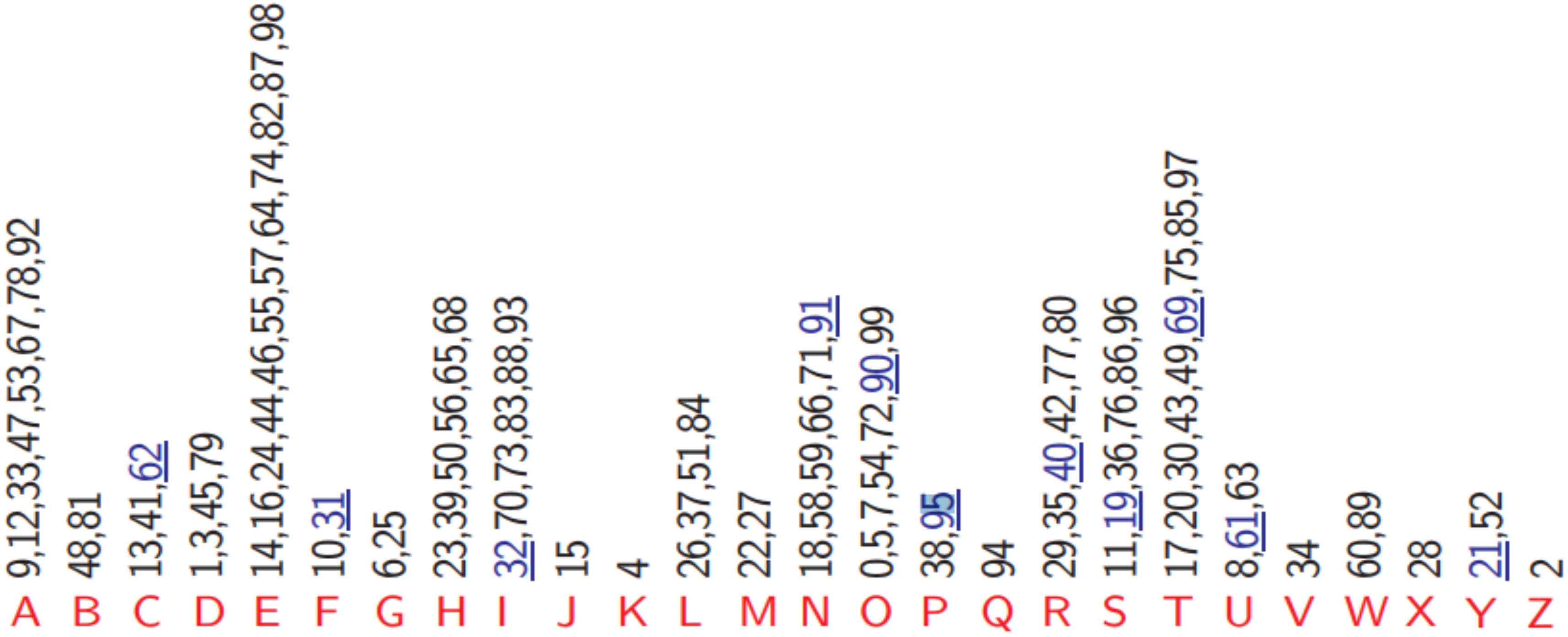
Coding symbols are assigned to each plain letter based on their frequency.

Following slide show an example of homophonic cipher:

- Homophonic Cipher

Homophonic Cipher

Example: Encrypt the message: “Crypto is fun”



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
9,12,33,47,53,67,78,92	48,81	13,41,62	1,3,45,79	14,16,24,44,46,55,57,64,74,82,87,98	10,31	6,25	23,39,50,56,65,68	32,70,73,83,88,93	15	4	26,37,51,84	22,27	18,58,59,66,71,91	0,5,7,54,72,90,99	38,95	94	29,35,40,42,77,80	11,19,36,76,86,96	17,20,30,43,49,69,75,85,97	8,61,63	34	60,89	28	21,52	2

Ciphertext : 62 40 21 95 69 90 32 19 31 61 91

Exercise: Decrypt “13 5 26 0 22 81 88 47”

Transposition Ciphers

A transposition ciphers are where the letters are jumbled up together.

Instead of replacing characters with other characters, this cipher just changes the order of the characters.

Mathematically a bijective function is used on the characters' positions to encrypt and an inverse function to decrypt.

Following slide show an example of transposition cipher:

- Turning Grille

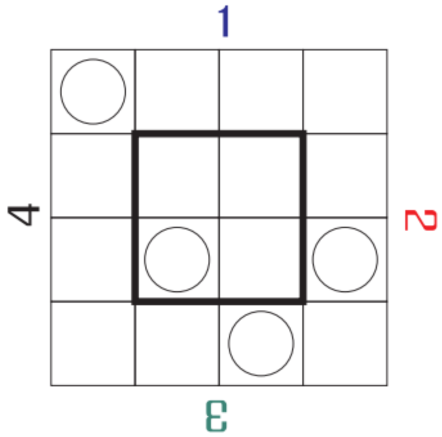
Turning Grille
(Eduard B. Fleissner v. Wostrowitz, 1881)

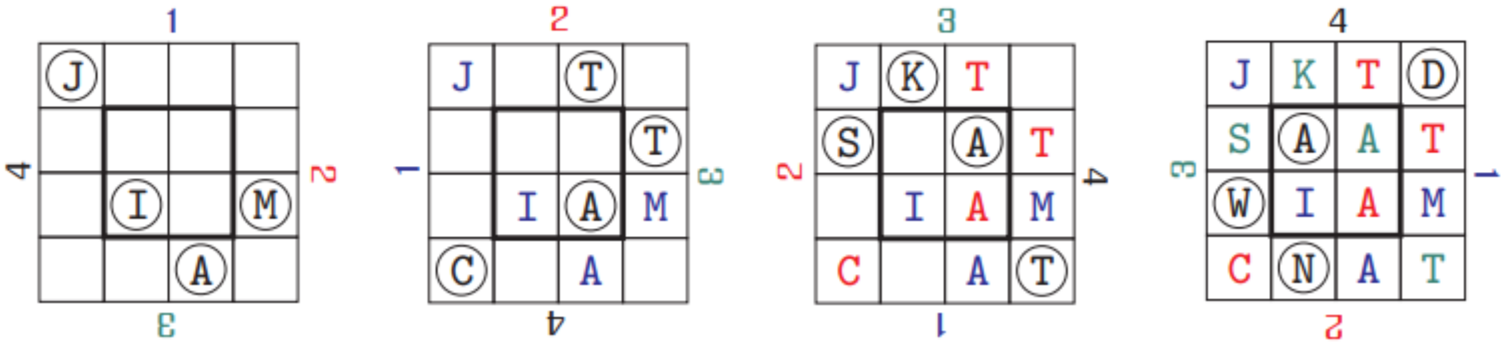
Used by Germany during the First World War.

This is a square piece of cardboard with holes in it such that each cell in the square appears in no more than one position when the grille is rotated to each of its four positions.

As much message as will fit in the grille is written, then it is turned to another position and more message is written.

Example: Encrypt “JIM ATTACKS AT DAWN” using the following 4×4 grille.

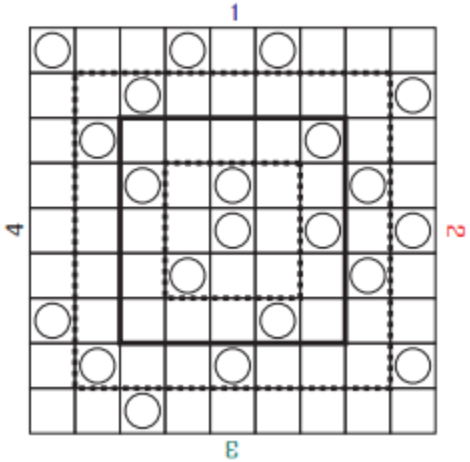




JIMA TTAC KSAT DAWN

JKTD SAAT WIAM CNAT

Exercise: Decrypt the ciphertext TESHN INCIG LSRGY LRIUS PITSA TLILM REENS ATTOG SIAWG IPVER TOTEH HVAEA XITDT UAIME RANPM TLHIE I using the following grille.



Product Ciphers

A product cipher is a composite of substitution (confusion) and transposition (diffusion) ciphers.

Diffusion refers to the dissipation of the statistical properties of the plaintext.

Confusion makes relationship between ciphertext and key as complex as possible.

- two substitutions make a more complex substitution
- two transpositions make more complex transposition
- but a substitution followed by a transposition makes a new much harder cipher

Product ciphers are the bridge from classical to modern ciphers like DES (Data Encryption Standard)

Lucifer Cipher
(Horst Feistel, IBM, 1971)

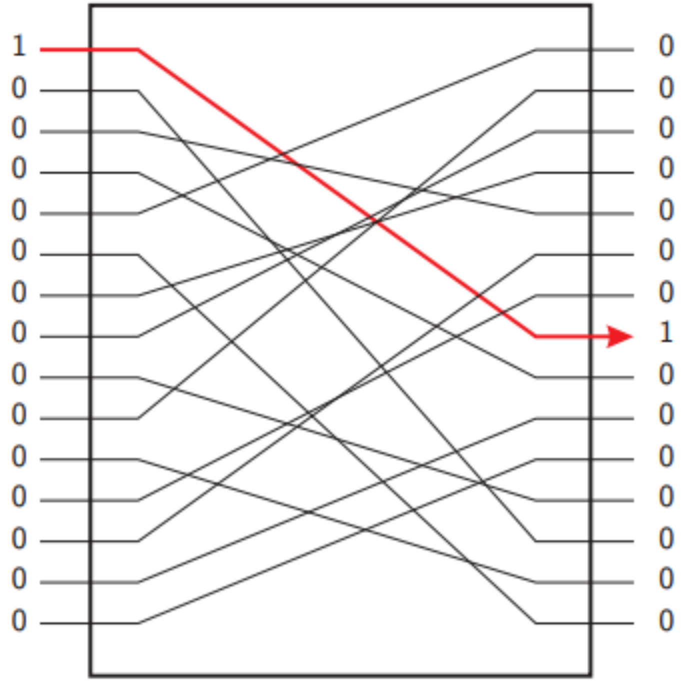
This system uses permutations (transpositions) on large blocks for the mixing transformation, and substitution on small blocks for confusion. Note that there is no key - thus the system is insecure.

Since this system was set up in hardware, they called the chips which did the permutations P-boxes, and those that did the substitution S-boxes.

Feistel had originally started with a “Demonstration Cipher”, with the name truncated to“Demon”, which led logically to the name “Lucifer”, which incidentally suggests “Lu(cipher)”.

P-box Example

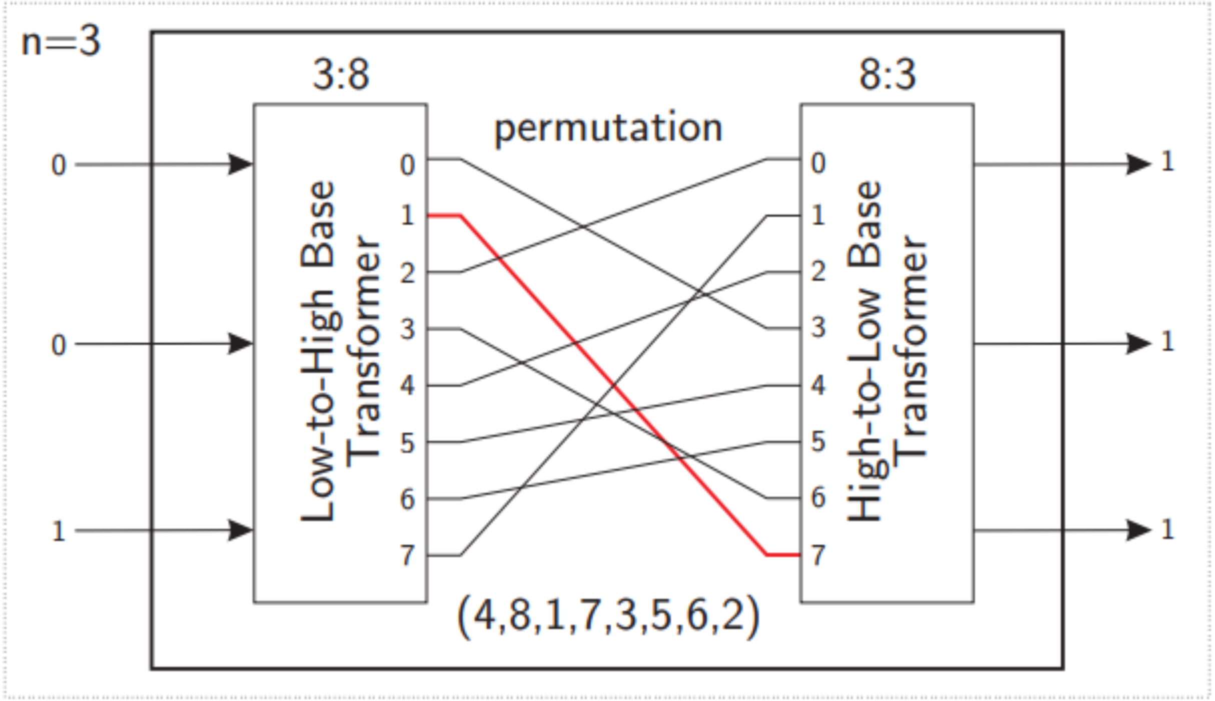
A P-box (Permutation-Box) is a box which permutes the bits of its input.



P-Box

S-box Example

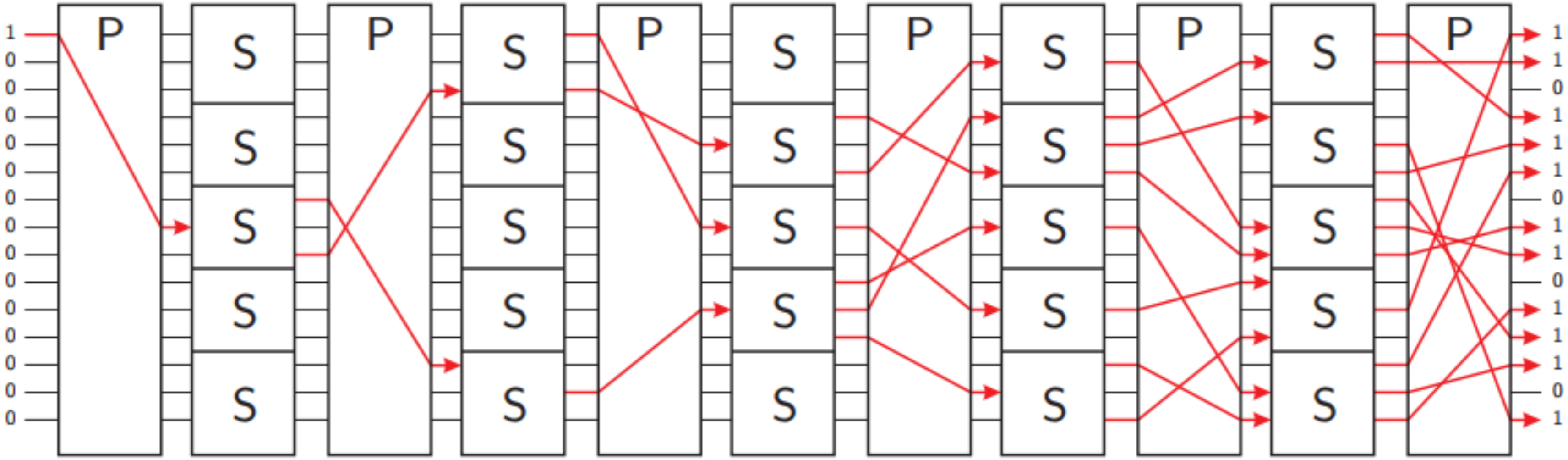
The S-box (Substitution-Box) is a hardware device which encodes n bit numbers to other n bit numbers.



S-Box

Lucifer Illustrative Example

The Lucifer system simply consisted of a number of P & S boxes in alternation:



Lucifer is known as the predecessor for the Data Encryption Standard (DES).

Hayhanen Cipher
(1950's)

Named after the Russian spy caught using it in New York.

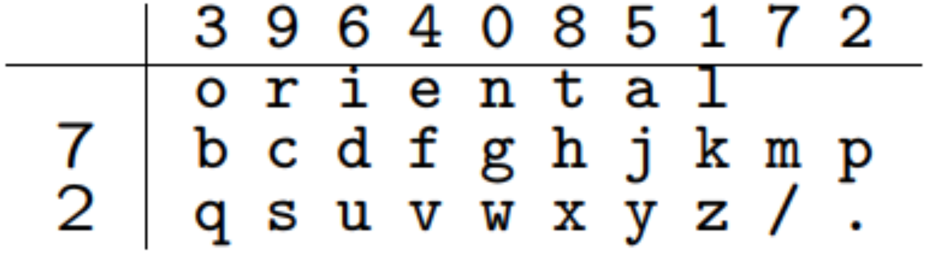
A Hayhanen Cipher is a variation on the VIC/NIHILIST cipher. It's a straddling checkerboard substitution followed by a double columnar transposition with disruptions.

It resisted cryptanalysis by the FBI and friends (NSA, GCHQ, etc) for four years until Reino Hayhanen defected to the US in 1957 and gave them the key word SNEGOPAD (Russian word for Snowfall) and explains it.

Example: Encrypt "Feb/4/have contacted fred. will consult before investing in coverbusiness.nat" using the key 'oriental'

Feb/4/havecontactedfred.willconsultbeforeinvestingincoverbusiness.n at

First you set up a keysquare, some letters code to 1 digit, some to 2. Note that the numbered rows use the numbers that are not used in the first row. We substitute all the letters according to this keysquare:



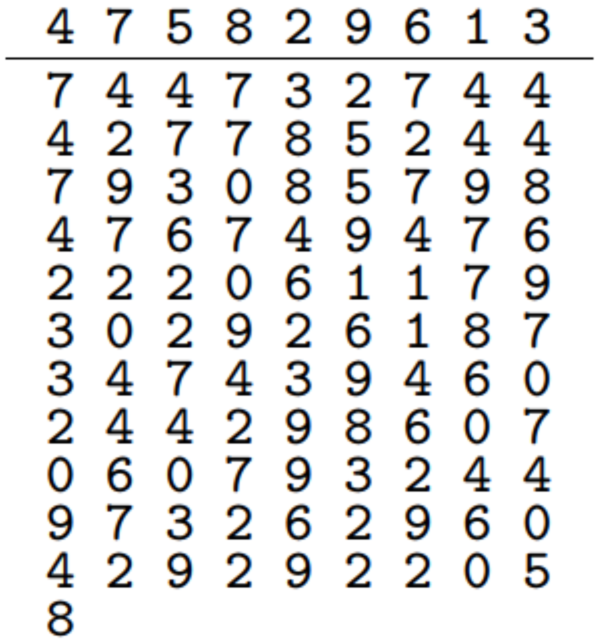
	3	9	6	4	0	8	5	1	7	2
	o	r	i	e	n	t	a	l		
7	b	c	d	f	g	h	j	k	m	p
2	q	s	u	v	w	x	y	z	/	.

F eb / 4 / h av ec ontac ted f red . w illc ons u ltb ef oreinv es 7447327444277852447930857984767494762220611793029261873474394602442 9

ting inc ov erb u s ines s . nat

86070607932449732629604292922058

Next, we write these out in a block for transposition as follows:

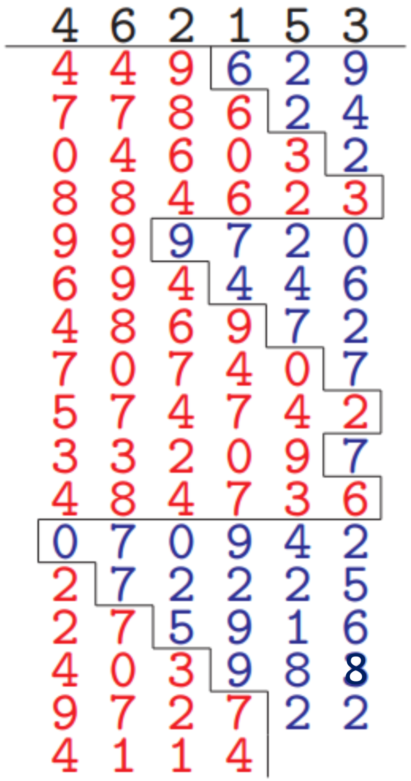


4	7	5	8	2	9	6	1	3
7	4	4	7	3	2	7	4	4
4	2	7	7	8	5	2	4	4
7	9	3	0	8	5	7	9	8
4	7	6	7	4	9	4	7	6
2	2	2	0	6	1	1	7	9
3	0	2	9	2	6	1	8	7
3	4	7	4	3	9	4	6	0
2	4	4	2	9	8	6	0	7
0	6	0	7	9	3	2	4	4
9	7	3	2	6	2	9	6	0
4	2	9	2	9	2	2	0	5
8								

Next, we read down each column in turn, starting with column ‘1’. We copy the numbers into another table, using a route cipher: We fill in the area to the left of the line (in red) first, then the areas to the right.

Note how the line is formed: we start to the left of column 1, and zigzag down to the right. When we reach the edge, we jump to the left of column 2 and zig-zag again, and so on.

4	7	5	8	2	9	6	1	3
7	4	4	7	3	2	7	4	4
4	2	7	7	8	5	2	4	4
7	9	3	0	8	5	7	9	8
4	7	6	7	4	9	4	7	6
2	2	2	0	6	1	1	7	9
3	0	2	9	2	6	1	8	7
3	4	7	4	3	9	4	6	0
2	4	4	2	9	8	6	0	7
0	6	0	7	9	8	2	4	4
9	7	3	2	6	2	9	6	0
4	2	9	2	9	2	2	0	5
8								



4	6	2	1	5	3
4	4	9	6	2	9
7	7	8	6	2	4
0	4	6	0	3	2
8	8	4	6	2	3
9	9	9	7	2	0
6	9	4	4	4	6
4	8	6	9	7	2
7	0	7	4	0	7
5	7	4	7	4	2
3	3	2	0	9	7
4	8	4	7	3	6
0	7	0	9	4	2
2	7	2	2	2	5
2	7	5	9	1	6
4	0	3	9	8	8
9	7	2	7	2	2
4	1	1	4		

Once we have completed this step, we transcribe the numbers by columns in groups of 5:

66067 49470 79299 74986 49467 42402 53219 42306 27276 25632
47089 64753 40224 94223 22470 49342 18247 48998 07387 77071

Modes of Operation

A *block cipher* (such as DES) encrypts plaintext in fixed-size n-bit blocks (often n=64). For messages exceeding n bits we should use the following standard methods (*modes of operation*).

1. ECB : Electronic Code Block
2. CBC : Cipher-Block Chaining
3. CFB : Cipher Feedback
4. OFB : Output Feedback
5. CTR : CounTeR

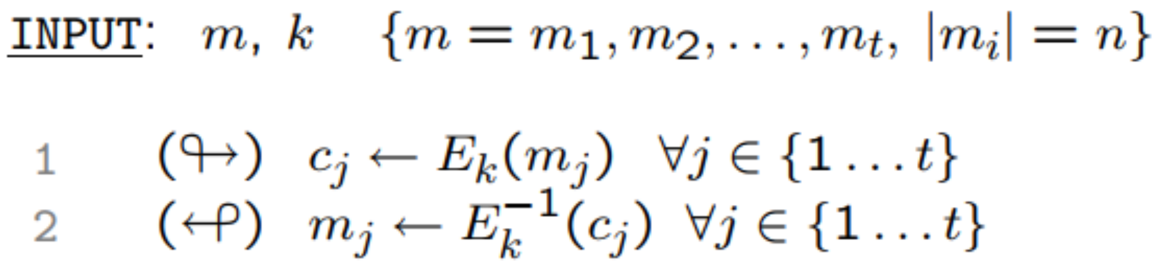
Notation

- E_k : Encryption function *E* using key *k*
- E_k^{-1} : Decryption function E^{-1} using key *k*
- m : Plain message $m = m_1,\ m_2,\dots\ m_t$
- *k* : The key
- *IV* : Initialization vector
- >> : Shift to the right
- (↬) : Encryption
- (↫) : Decryption

ECB : Electronic Code Block

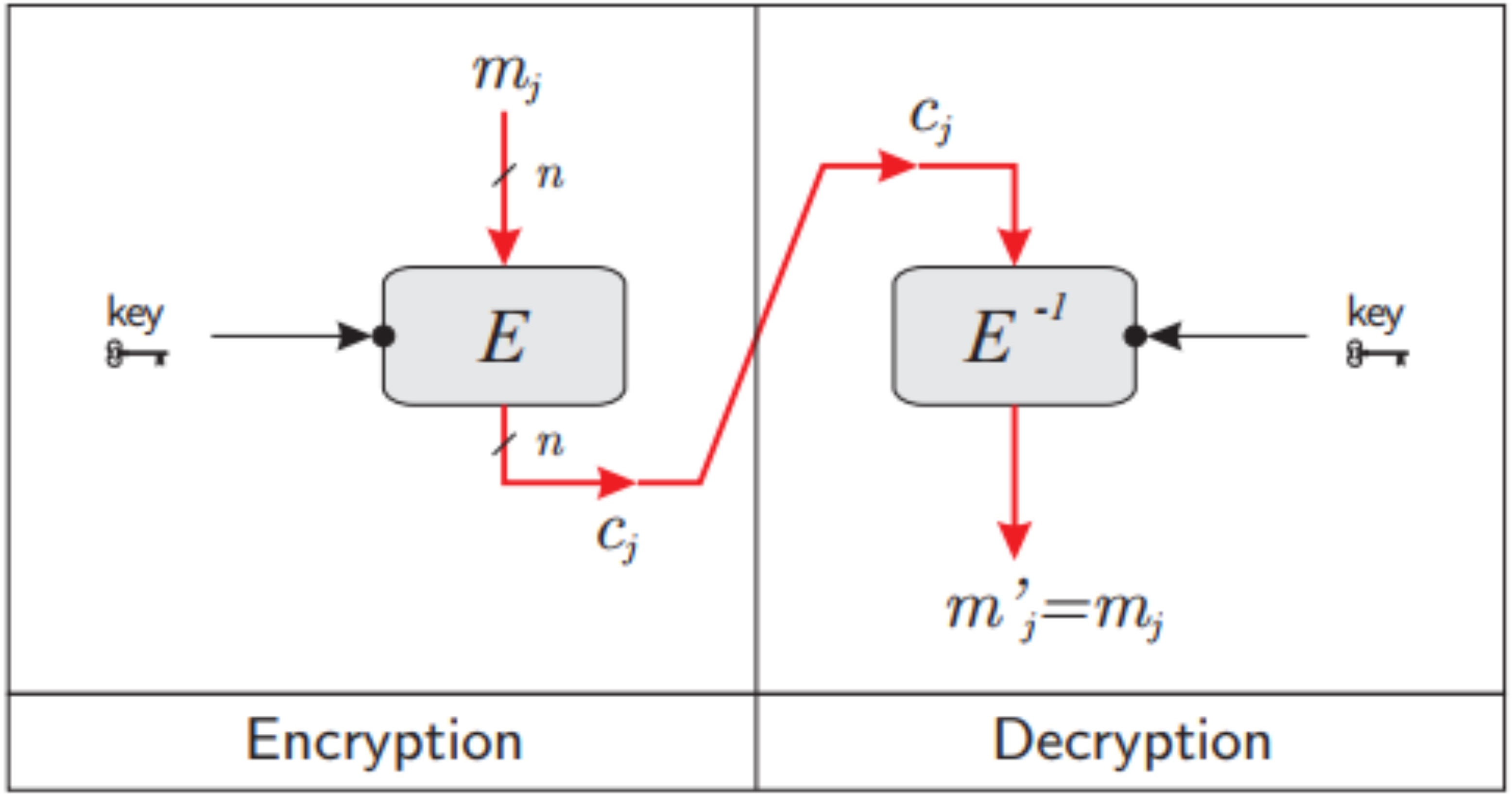
Operates like a *code book*, where every block of plaintext maps to exactly one block of ciphertext. This is usually the highest performance mode, but suffers from the problem that when the same pattern appears, it is always encrypted the same. Other modes, such as CBC and CFB combine cipher text with previous cipher text, causing similar messages to be encoded completely differently.

Pseudo-code:



```
INPUT: $m,\ k \quad \{m = m_1, m_2, \ldots, m_t,\ |m_i| = n\}$

    $(P\!\rightarrow)$  $c_j \leftarrow E_k(m_j) \ \ \forall j \in \{1 \ldots t\}$
    $(\leftarrow\!P)$  $m_j \leftarrow E_k^{-1}(c_j) \ \ \forall j \in \{1 \ldots t\}$
```



Properties:

1. Identical plaintext : YES
2. Chaining dependencies : NO
3. Error propagation : NO

Example

Let E_π be a permutation cipher, and $\mathcal{A}$ = {0, 1}.

E_π= $\{0, 1\}^4 \rightarrow \{0, 1\}^4$, π = $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

Encrypt the plaintext m = 101100010100101.

We first decompose m into blocks of length 4. Last block is padded with zeros.

m_1 = 1011, m_2 = 0001, m_3 = 0100, m_4 = 1010

c_1 = $E_\pi(m_1)$ = 0111, c_2= E_π (m_2) = 0010

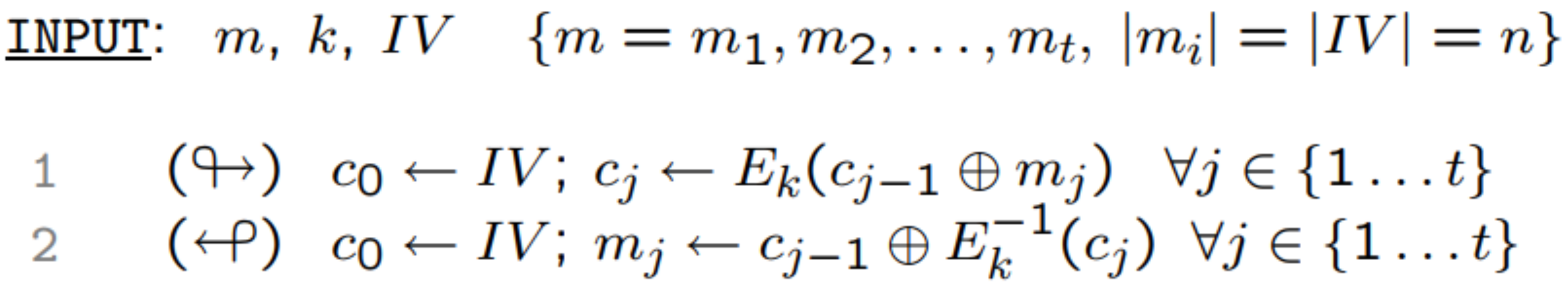
c_3 = E_π (m_3) = 1000, c_4 = E_π (m_4) = 0101

c = 0111001010000101

CBC : Cipher-Block Chaining

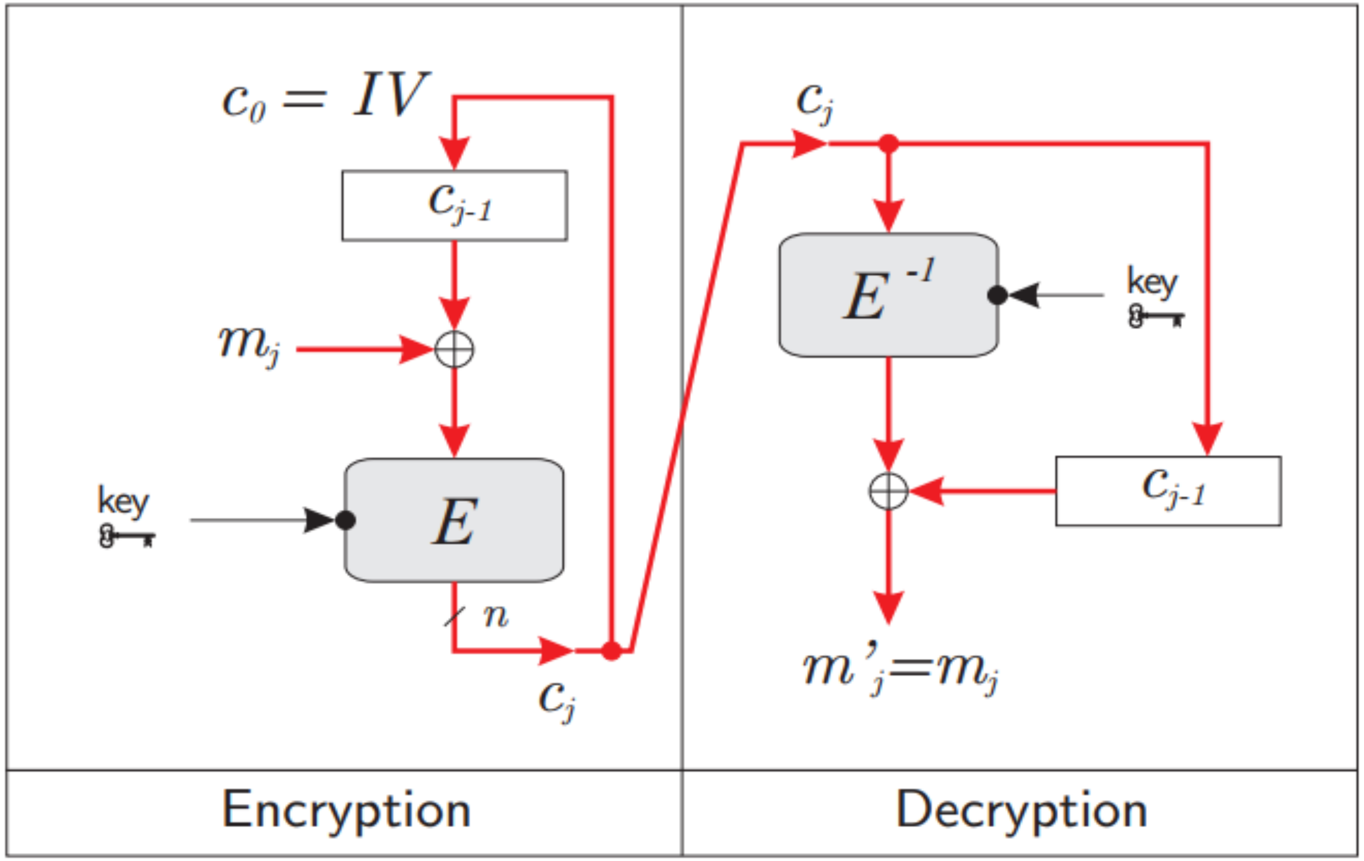
CBC is similar to ECB, except that each plaintext block is XORed with the previous block of ciphertext. Thus, identical patterns in different messages will be encrypted differently, depending upon the difference in previous data.

Pseudo-code:



```
INPUT: $m,\ k,\ IV$    $\{m = m_1, m_2, \ldots, m_t,\ |m_i| = |IV| = n\}$

$(\mapsto)$  $c_0 \leftarrow IV;\ c_j \leftarrow E_k(c_{j-1} \oplus m_j)\ \ \forall j \in \{1 \ldots t\}$
$(\mapsfrom)$  $c_0 \leftarrow IV;\ m_j \leftarrow c_{j-1} \oplus E_k^{-1}(c_j)\ \ \forall j \in \{1 \ldots t\}$
```



Properties:

1. Identical plaintext : NO
2. Chaining dependencies : YES
3. Error propagation : YES
4. Error recovery : YES

Example

E_π= {0, 1}^{4} → {0, 1}^{4}, π = $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

Encrytion:

m_1 = 1011, m_2 = 0001, m_3 = 0100, m_4 = 1010

Let IV = 1010

C_0 = IV = 1010

C_1 = $E_\pi(C_0 \otimes m_1)$ = E_π (0001) = 0010

C_2 = E_π $(C_1 \otimes m_2)$ = E_π (0011) = 0110

C_3 = E_π $(C_2 \otimes m_3)$ = E_π (0010) = 0100

C_4 = E_π $(C_3 \otimes m_4)$ = E_π (1110) = 1101

c = 0010011001001101

Decrytion:

c = 0010011001001101

C_1 = 0010, C_2 = 0110, C_3 = 0100, C_4 = 1101

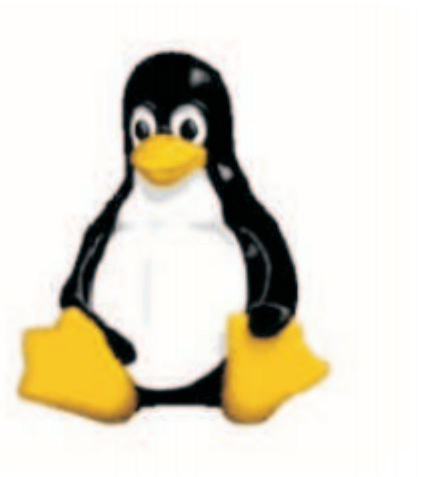
m_1 = $C_0 \otimes E_{\pi}^{-1}$ (C_1) = 1010 $\otimes$ 0001 = 1011

m_2 = $C_1 \otimes E_{\pi}^{-1}$ (C_2) = 0010 $\otimes$ 0011 = 0001

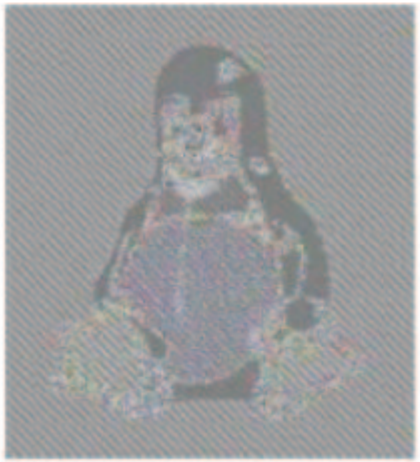
m_3 = $C_2 \otimes E_{\pi}^{-1}$ (C_3) = 0110 $\otimes$ 0010 = 0100

m_4 = $C_3 \otimes E_{\pi}^{-1}$ (C_4) = 0100 $\otimes$ 1110 = 1010

ECB Demo



Original



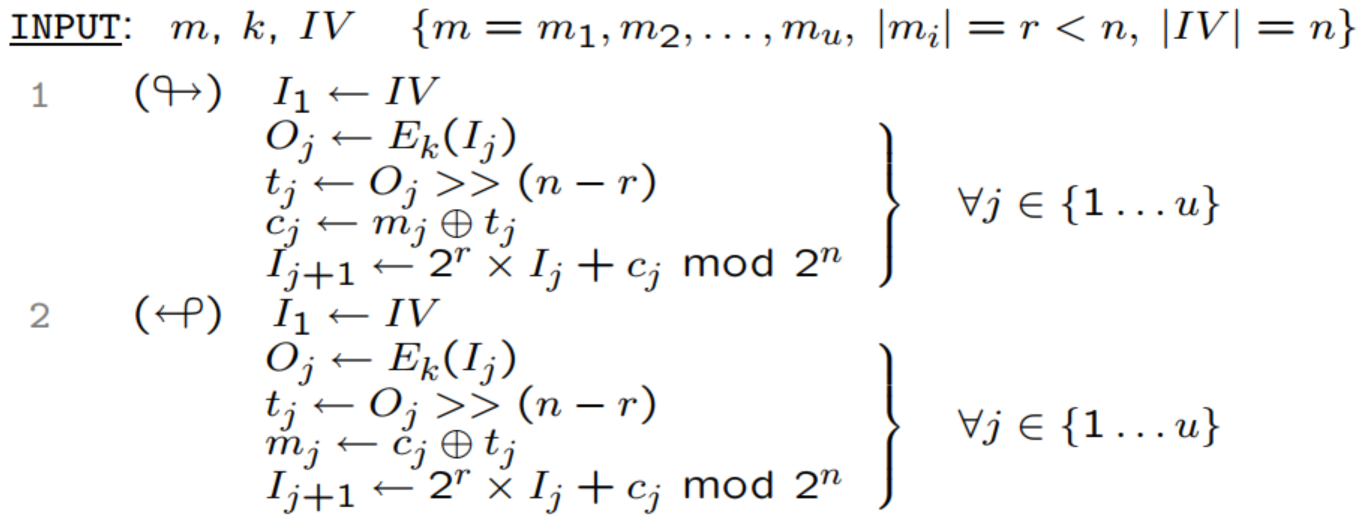
ECB

CBC

CFB : Cipher Feedback

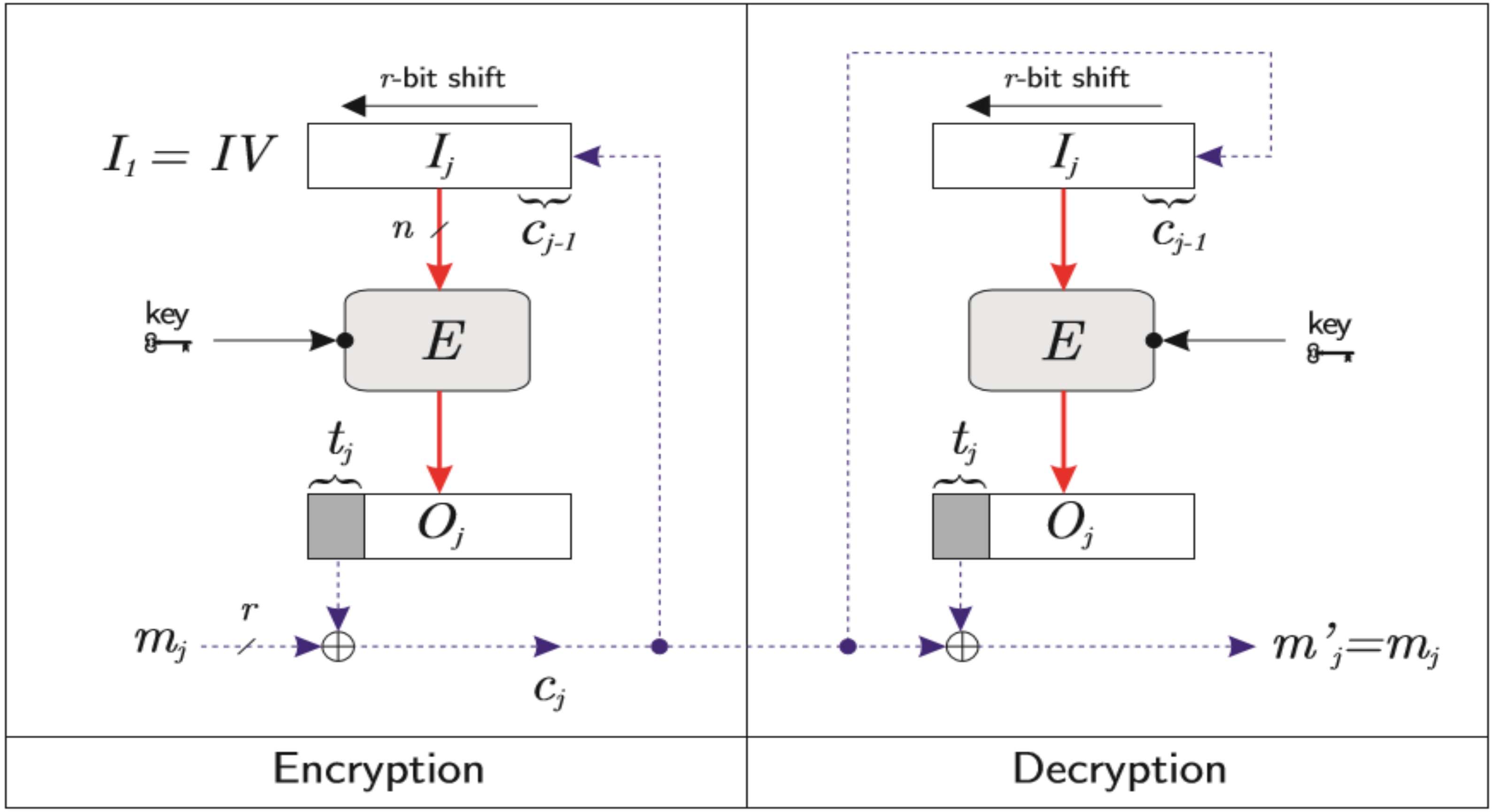
CFB is similar to CBC in that following data is combined with previous data so that identical patterns in the plain text result in different patterns in the cipher text. The difference is that data is encrypted a byte (or a bit) at a time. This mode is useful for data streams, such as terminal sessions, where data arrives a byte at a time.

Pseudo-code:



```
INPUT: $m,\ k,\ IV \quad \{m = m_1, m_2, \ldots, m_u,\ |m_i| = r < n,\ |IV| = n\}$
1    (ꟼ→)    $I_1 \leftarrow IV$
             $O_j \leftarrow E_k(I_j)$                          ⎫
             $t_j \leftarrow O_j >> (n - r)$                     ⎪
             $c_j \leftarrow m_j \oplus t_j$                      ⎬  $\forall j \in \{1 \ldots u\}$
             $I_{j+1} \leftarrow 2^r \times I_j + c_j \bmod 2^n$   ⎭
2    (←P)    $I_1 \leftarrow IV$
             $O_j \leftarrow E_k(I_j)$                          ⎫
             $t_j \leftarrow O_j >> (n - r)$                     ⎪
             $m_j \leftarrow c_j \oplus t_j$                      ⎬  $\forall j \in \{1 \ldots u\}$
             $I_{j+1} \leftarrow 2^r \times I_j + c_j \bmod 2^n$   ⎭
```

$I_{j+1} = (2^r \text{ x } I_j + c_j)\text{mod } 2^n$



Properties:

1. Identical plaintext :NO
2. Chaining dependencies :YES
3. Error propagation :YES
4. Error recovery :YES
5. Throughput :[n/r]

Example

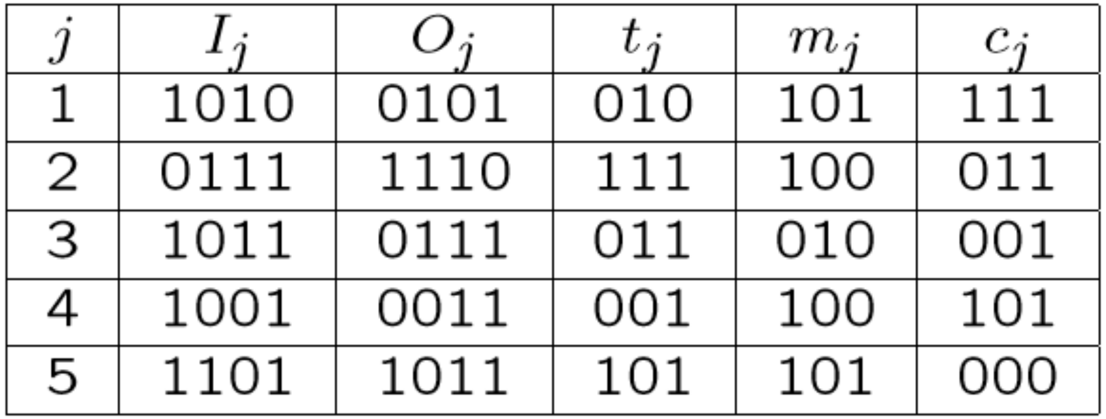
$E_\pi = \{0,1\}^4 \rightarrow \{0,1\}^4$, $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

Encryption:

r=3

$m_1 = 101$, $m_2 = 100$, $m_3 = 010$, $m_4 = 100$, $m_5 = 101$

Let *IV* = 1010



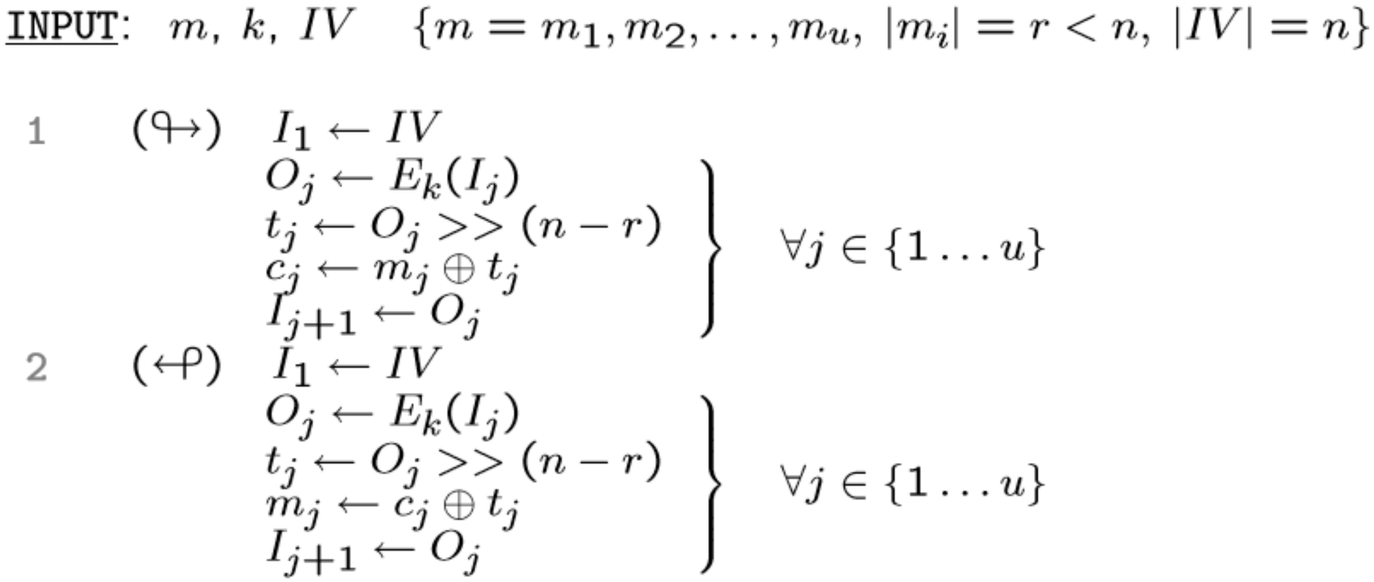
j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0111	1110	111	100	011
3	1011	0111	011	010	001
4	1001	0011	001	100	101
5	1101	1011	101	101	000

c=111011001101000

OFB: OutputFeedback

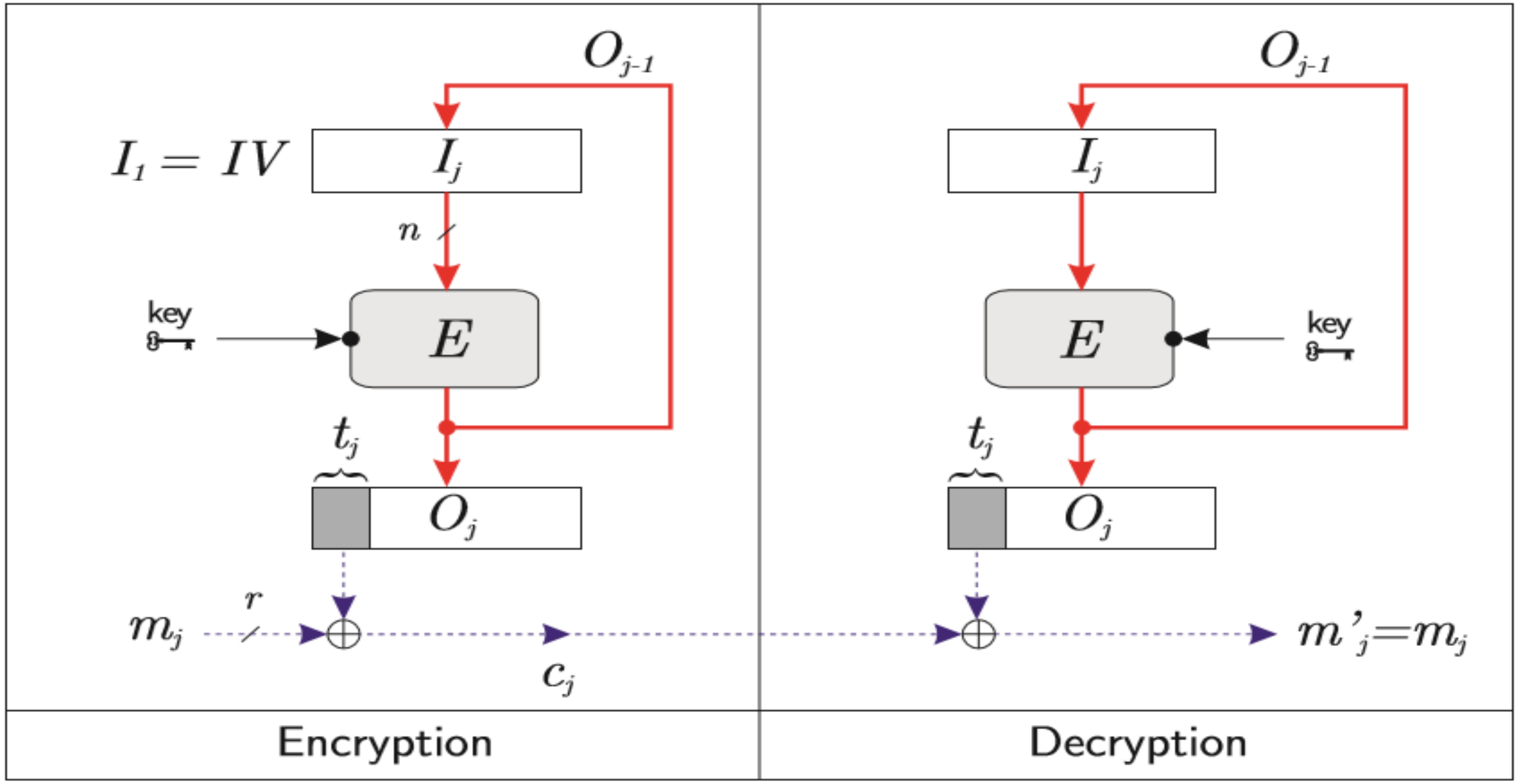
Almost identical to CFB, except that 1-bit errors in the cipher text cause only 1-bit errors in the plain text (instead of ruining the remainder of the stream). However, it is not as secure as CFB.

Pseudo-code:



```
INPUT:  $m,\ k,\ IV \quad \{m = m_1, m_2, \ldots, m_u,\ |m_i| = r < n,\ |IV| = n\}$

1   $(\text{P}\mapsto)$   $I_1 \leftarrow IV$
                  $O_j \leftarrow E_k(I_j)$
                  $t_j \leftarrow O_j >> (n-r)$
                  $c_j \leftarrow m_j \oplus t_j$            $\forall j \in \{1 \ldots u\}$
                  $I_{j+1} \leftarrow O_j$
2   $(\leftarrow\text{P})$   $I_1 \leftarrow IV$
                  $O_j \leftarrow E_k(I_j)$
                  $t_j \leftarrow O_j >> (n-r)$
                  $m_j \leftarrow c_j \oplus t_j$            $\forall j \in \{1 \ldots u\}$
                  $I_{j+1} \leftarrow O_j$
```



Properties:

1. Identical plaintext :NO
2. Chaining dependencies :NO
3. Error propagation :NO
4. Error recovery :YES
5. Throughput :[n/r]

Example

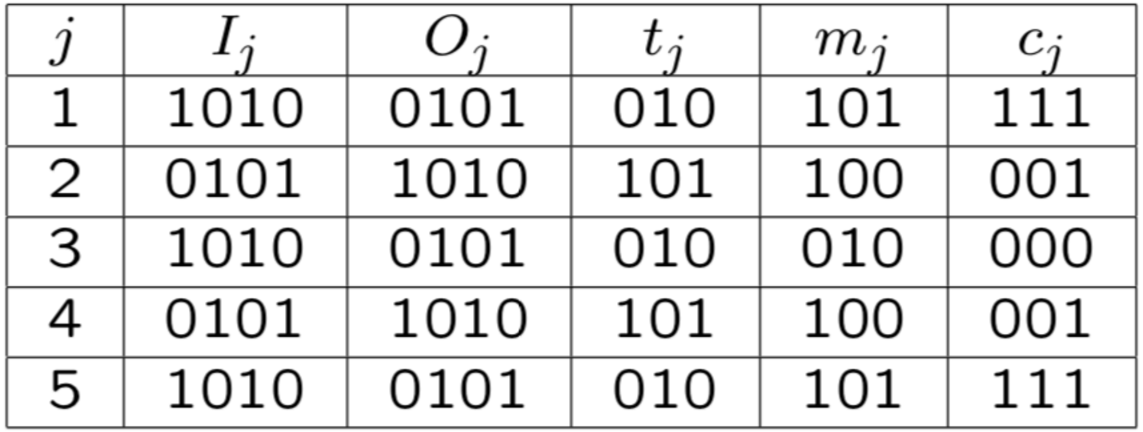
$E_\pi = \{0, 1\}^4 \rightarrow \{0, 1\}^4$, $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

Encryption:

r=3

m_1= 101, m_2 = 100, m_3 = 010, m_4 = 100, m_5 = 101

Let *IV* = 1010



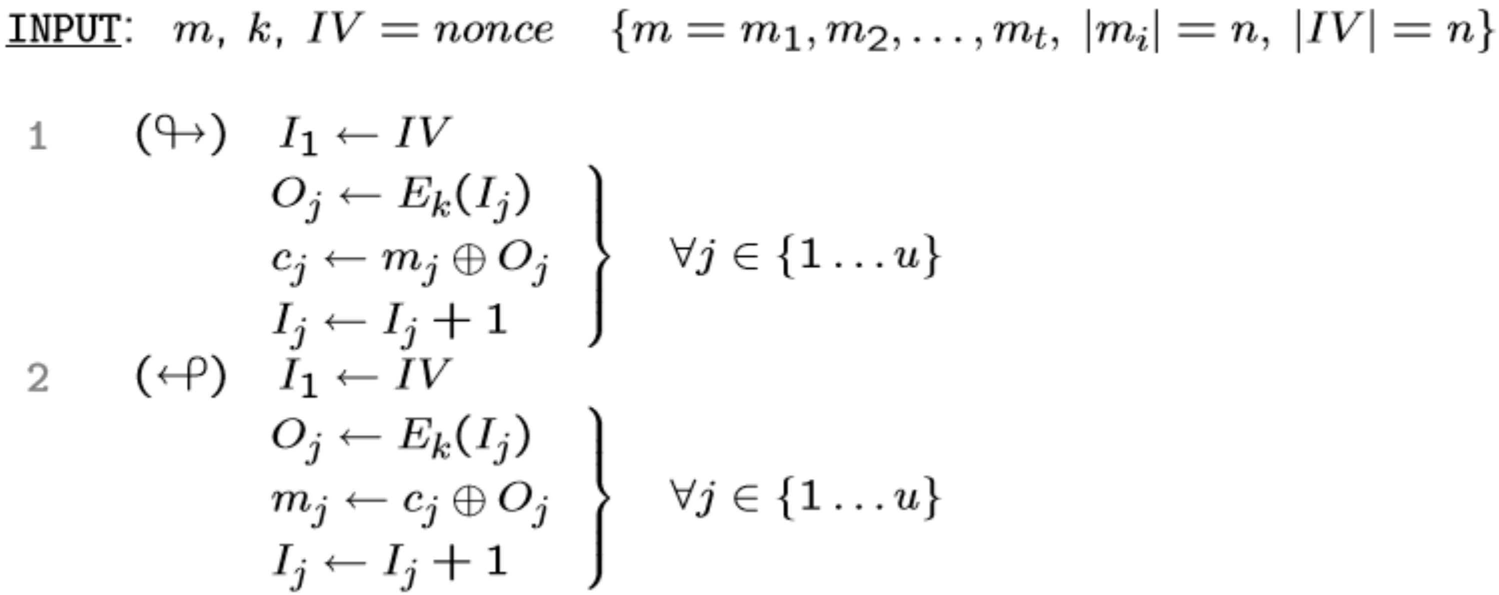
j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0101	1010	101	100	001
3	1010	0101	010	010	000
4	0101	1010	101	100	001
5	1010	0101	010	101	111

c=111001000001111

CTR:CounTeR

In the CTR mode there is no feedback. The pseudorandomness in the key stream is achieved using a counter. The counter is incremented for each block.

Pseudo-code:



```
INPUT: $m,\ k,\ IV = nonce \quad \{m = m_1, m_2, \ldots, m_t,\ |m_i| = n,\ |IV| = n\}$

1   (⊸)  $I_1 \leftarrow IV$
         $O_j \leftarrow E_k(I_j)$
         $c_j \leftarrow m_j \oplus O_j$      $\forall j \in \{1 \ldots u\}$
         $I_j \leftarrow I_j + 1$
2   (⟜)  $I_1 \leftarrow IV$
         $O_j \leftarrow E_k(I_j)$
         $m_j \leftarrow c_j \oplus O_j$      $\forall j \in \{1 \ldots u\}$
         $I_j \leftarrow I_j + 1$
```

nonce is a number used once

References

- Pinzon, Yoan. “Introducción a la criptografía y a la seguridad de la información”, 2013.
- Menezes A., Handbook of applied Cryptografy, 5th Edition. CRC Press, 2001.
- A Graduate Course in Applied Cryptography by D. Boneh and V. Shoup
- H. Delfs and H. Knebl, Introduction to Cryptography, 3rd ed. 2015.
- J. A. Buchmann, Introduction to Cryptography. 2004.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Ch. 31 in Introduction to Algorithms, 3rd ed. 2009.
- B. Lomas de Zamora: Gradi. Hacking desde cero, Fox Andina, 2011.
- Secure Coding Working Group, Japan Smartphone Security Association (JSSEC), Android Application Secure Design/Secure Coding Guidebook, 2017.