Математична криптологія

Класичні методи

Шифр Юлія Цезаря (Caesar Cipher)

а б в г ґ д е є ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я
г ґ д е є ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я а б в

Шифр зсуву є звуженням загального шифру заміни на сукупність лише n ключів, у яких нижній рядок є циклічним зсувом верхнього.

Шифрування: $C_i = (M_i + 3) \bmod 33$;

Дешифрування: $M_i = (C_i - 3) \bmod 33$

Частотний аналіз. У досить довгих текстах кожна буква зустрічається із приблизно однаковою частотою, залежною від самої букви і незалежною від конкретного тексту.

Частота символу у тексті дорівнює кількості його входжень у цей текст, поділеній на загальну кількість букв. Розшифрувати наступне повідомлення:

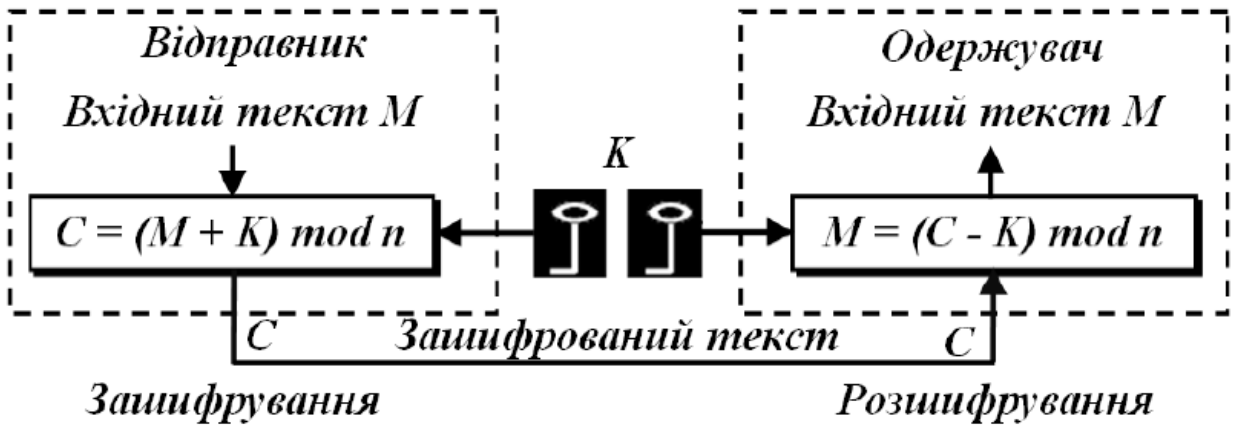
пцпспофнпмпллпибгпефрптмбвмєоп охоронумолокозаводупослаблено

п	0.134	е	0.043	л	0.028	ч	0.011	ї	0.006
о	0.082	р	0.038	у	0.027	б	0.010	є	0.006
н	0.070	і	0.037	п	0.025	х	0.010	ф	0.005
а	0.070	с	0.036	я	0.021	ц	0.009	ш	0.005
и	0.056	к	0.036	з	0.019	ю	0.009	щ	0.003
т	0.051	м	0.033	ь	0.015	ж	0.008	ґ	0.000
в	0.046	д	0.028	г	0.013	й	0.007		

Адитивні моноалфавітні шифри заміни

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
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
00	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

Числове значення
Зашифрований текст
Вхідний текст



Шифрування К=15:

$m_1 = h \rightarrow 07;$ $c_1 = (07 + 15) \bmod 26 = 22 \rightarrow W;$
$m_2 = e \rightarrow 04;$ $c_2 = (04 + 15) \bmod 26 = 19 \rightarrow T;$
$m_3 = l \rightarrow 11;$ $c_3 = (11 + 15) \bmod 26 = 0 \rightarrow A;$
$m_4 = l \rightarrow 11;$ $c_4 = (11 + 15) \bmod 26 = 0 \rightarrow A;$
$m_5 = o \rightarrow 14;$ $c_4 = (14 + 15) \bmod 26 = 3 \rightarrow D,$

Дешифрування К=15:

$c_1 = W \rightarrow 22;$ $m_1 = (22 - 15) \bmod 26 = 07 \rightarrow h;$
$c_2 = T \rightarrow 19;$ $m_2 = (19 - 15) \bmod 26 = 04 \rightarrow e;$
$c_3 = A \rightarrow 00;$ $m_3 = (00 - 15) \bmod 26 = 11 \rightarrow l;$
$c_4 = A \rightarrow 00;$ $m_4 = (00 - 15) \bmod 26 = 11 \rightarrow l;$
$c_5 = D \rightarrow 03;$ $m_5 = (03 - 15) \bmod 26 = 14 \rightarrow o,$

Гомофонний шифр заміни

Був винайдений великим німецьким математиком Карлом Фрідріхом Гаусом. Кожна буква відкритого тексту замінюється не єдиним символом як у шифрі простої заміни, а будь-яким символом із декількох можливих.

Наприклад:

а → 10, 17, 23, 46, 55

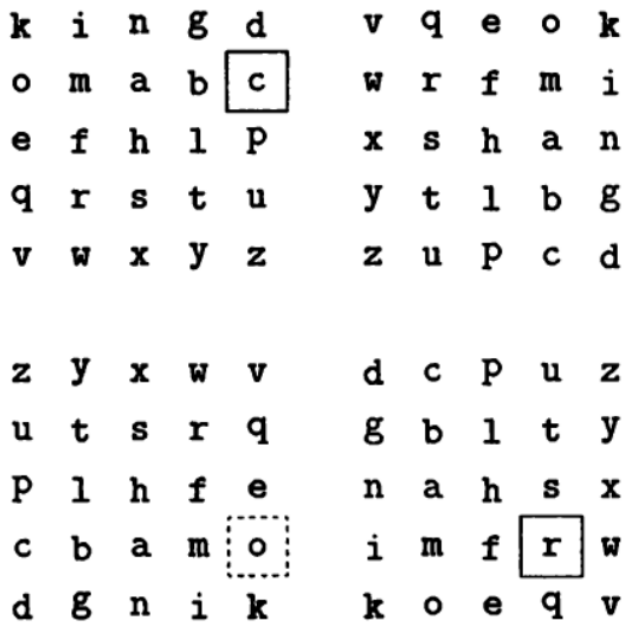
б → 12, 71

в → 34, 45, 7

Вибір одного з можливих варіантів щоразу робиться випадково. Якщо кількість варіантів для кожної букви пропорційна її частоті в мові, то всі символи у досить довгому криптотексті зустрічатимуться з приблизно однаковою частотою. Однак гомофонний шифр піддається ретельнішому та трудомісткішому різновиду частотного аналізу, який окрім частот символів враховує також частоти пар символів.

Поліграмні шифри

Послідовність кількох букв у тексті називається поліграмою. Послідовність із двох букв називається біграмою, із *l* букв – *l*-грамою. Поліграмний шифр заміни полягає у розбитті відкритого тексту на *l*-грами для деякого фіксованого числа *l* і заміні кожної з них на якийсь символ чи групу символів. Ключем є правило за яким відбувається заміна. Якщо загальна кількість символів у тексті не ділиться націло на *l*, то остання група символів доповнюється до *l*-грами довільним наперед обумовленим способом.



Поліалфавітні шифри. Шифр Віженера

Шифр Віженера – поліалфавітний шифр який використовує слово в якості ключа. Отримав назву на честь Блеза де Віженера, хоча насправді його винайшов італійський криптограф Джованні Баттіста Белласо (19 ст.)

Шифрування: $C_i = (M_i + K_i) \bmod 33$;

Дешифрування: $M_i = (C_i - K_i) \bmod 33$

	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
а	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
б	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а
в	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б
г	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в
ґ	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г
д	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ
е	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д
є	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е
ж	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є
з	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж
и	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з
і	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и
ї	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і
й	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї
к	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й
л	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к
м	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л
н	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м
о	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н
п	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о
р	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п
с	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р
т	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с
у	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т
ф	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у
х	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф
ц	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х
ч	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
ш	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
щ	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
ь	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
ю	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
я	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю

```
  БОРОНІТЬКОРОЛІВНУВІДВОРОГІВ
+ КЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮЧКЛЮ
  ЛАОЇЮЦРФШАОЇЩЦАІҐНЗЯМАОЇНЦА
```

```
Б   Н   К   Л   У   В   Г
 О   І   О   І   В   О   І
  Р   Т   Р   В   І   Р   В
   О   Ь   О   Н   Д   О
```

Криптоаналіз: Триграма “аої” зустрічається тричі а біграма “ца” двічі. Це означає, що відстань між ними є кратною довжині ключа. Звідси висновок, що довжина ключа дорівнює 1, або 2, або 4.

Шифр з автоключем грунтується на ідеях Віженера та Кардано:

```
  БОРОНІТЬКОРОЛІВНУВІДВОРОГІВ
+ КЛЮЧБОРОНІТЬКОРОЛІВНУВІДВОР
  ЛАОЇОЩЗЛЮЩЗЛЩЩТВДЙЙТХРЮУДЩТ
```

Шифр Віженера

The ASCII code

American Standard Code for Information Interchange

ASCII control characters

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Extended ASCII characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

Шифр Віженера - дешифрування

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

PRIVATE KEY = SECRET

S E CR E T S E CR E T S E CR E T S

Ciphertext: **LLKJ ML BYUK EG WBCDTEW**

Plaintext: **THIS IS JUST AN EXAMPLE**

$$D_i(x_i) = (x_i - K_i) \bmod 26$$

Криптоаналіз шифру Віженера

Cracking the **Vigenere cipher** is way harder than cracking **Caesar cipher**
~ of course because the complexity of cracking a cipher is
proportional to the size of the keyspace

Caesar cipher's keyspace = **26**

Vigenere cipher's keyspace = $\mathbf{26}^{\text{SIZE OF THE KEY}}$

1.) we can use **dictionary attack**: so we have a dictionary (file containing the words)
and we use these words as the possible keys

~ it is a form of **brute force attack**

2.) Kasiski-algorithm: a smarter approach to crack Vigenere cipher

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

→ it was constructed by **Friedrich Kasiski** in **1863** although it was independently discovered by **Charles Babbage** as well

→ if we know the size of the key then we can use **frequency analysis** in order to decrypt a given ciphertext

AGAIN WE TAKE ADVANTAGE OF THE INFORMATION LEAKING !!!

Algorithm:

1.) we have to find the size of the key: we can analyse repeated substrings and their factors to get a good guess

2.) we can construct substrings from the ciphertext that are encrypted by the same letters
3.) we can use frequency analysis to find the letters of the key

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

1.) first we have to find repeated substrings in the ciphertext
(the size of these substrings are at least **3** letters long)

**BY THE WAY THIS IS
WHY TO LEARN ALGORITHMS
AND DATA STRUCTURES
(SUFFIX TREES)**

Plaintext: **CRYPTOGRAPHY IS QUITE IMPORTANT IN CRYPTOCURRENCIES**

Key: **TABLE**

Ciphertext**:**

T A B L E T A B L E TABLETA BLETABL E T A B L E TABLETABLETABLET
CRYPTOGRAPHY IS QUITE IMPORTANT IN CRYPTOCURRENCIES
WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

1.) first we have to find repeated substrings in the ciphertext
(the size of these substrings are at least **3** letters long)

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

→ so here we can find a repeated substring (**WS AY**) because both occurrences of „**CRYPT**” line up with „**TABLE**”

→ note that we can get the same repeated substrings by accident: because the same index can be obtained several ways !!!

→ we can assume that if the repeated string occurs in the plaintext and the distance between corresponding characters is a multiple of the keyword length then the keyword letters will line up in the same way with both occurrences

AGAIN IT IS INFORMATION LEAKING !!!

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

2.) second step is to consider the distances between these repeated substrings and find the factors of these values

REPEATED SUBSTRING	DISTANCE
WS AY	**25 (5x5)**
HHA	**10 (2x5)**
KKLA	**20 (2x2x5)**

Kasiski-algorithm assumes that length of the key is the factor with the highest count !!!

THE LENGTH OF THE KEY IS 5

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

→ if the length of the key is **N** then we know that every **N-th** letter must have been encrypted using the same subkey

→ so we create substrings containing every **N-th** letter
~ there will be **N** substrings after this operation

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL**

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL**

#2 substring: **SHZRASASDO**

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL**

#2 substring: **SHZRASASDO**

#3 substring: **TBWKVK WE**

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL**

#2 substring: **SHZRASASDO**

#3 substring: **TBWKVK WE**

#4 substring: **AMUUYMZACU**

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL**

#2 substring: **SHZRASASDO**

#3 substring: **TBWKVK WE**

#4 substring: **AMUUYMZACU**

#5 substring: **YUXYUSEYWJ**

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

WS AYHHTMUAZBUXTRWUYYAKYUHSVMSMAKZEWS AYHDWCWYOEUJL

#1 substring: **WHATYHMWHYL** ← first letter of the key encrypted this substring

#2 substring: **SHZRASASDO** ← second letter of the key encrypted this substring

#3 substring: **TBWKVK WE** ← third letter of the key encrypted this substring

#4 substring: **AMUUYMZACU** ← fourth letter of the key encrypted this substring

#5 substring: **YUXYUSEYWJ** ← fifth letter of the key encrypted this substring

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

→ we apply all possible **26** subkeys on the ciphertext

→ we know the frequency distribution of the letters in the english alphabet

→ compare the two frequency distributions so we count the
letter frequency matches (decrypted text + english alphabet)

For example: if the most frequent letter in the decrypted text is **E** then
counter+1 because **E** is the most frequent letter in the
english alphabet is well

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis** because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

#1 substring
WHATYHMWHYL

SUBKEY	DECRYPTED #1 SUBSTRING	MATCH
A	VG SXGLVGXK	0
B	UFZRWFKUFWJ	0
C	TEYQVEJTEVI	2
…	…	…

So we have to try with all possible letter (**26** letters so **A-Z**) and consider the matches with highest values

+ we have to do the same operation for the other substrings as well

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

#1 substring possible subkeys: **C**, **T** and **E**

#2 substring possible subkeys: **A** and **H**

#3 substring possible subkeys: **B**

#4 substring possible subkeys: **K** and **L**

#5 substring possible subkeys: **A, E** and **I**

Now we have to use **brute-force method** to get all possible key values
~ there are **3x2x1x2x3=36** possible values which can be done
with brute-force without any issues

WE CONSIDER ALL THESE 36 POSSIBLE KEYS AND CHECK WHETHER THE DECRYPTED TEXT IS VALID (SO ENGLISH) OR NOT !!!

Криптоаналіз шифру Віженера

KASISKI-ALGORITHM

3.) if we know the size of the key then we can use **frequency analysis**
because **Vigenere cipher** is the same as **Caesar cipher**
~ of course it uses multiple subkeys

#1 substring possible subkeys: **C**, **T** and **E**

#2 substring possible subkeys: **A** and **H**

#3 substring possible subkeys: **B**

#4 substring possible subkeys: **K** and **L**

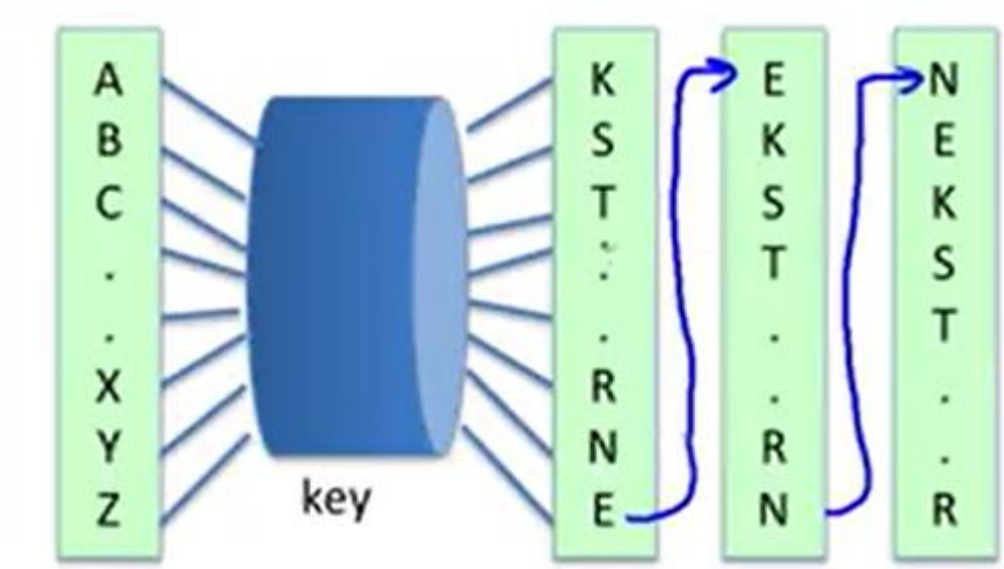
#5 substring possible subkeys: **A, E** and **I**

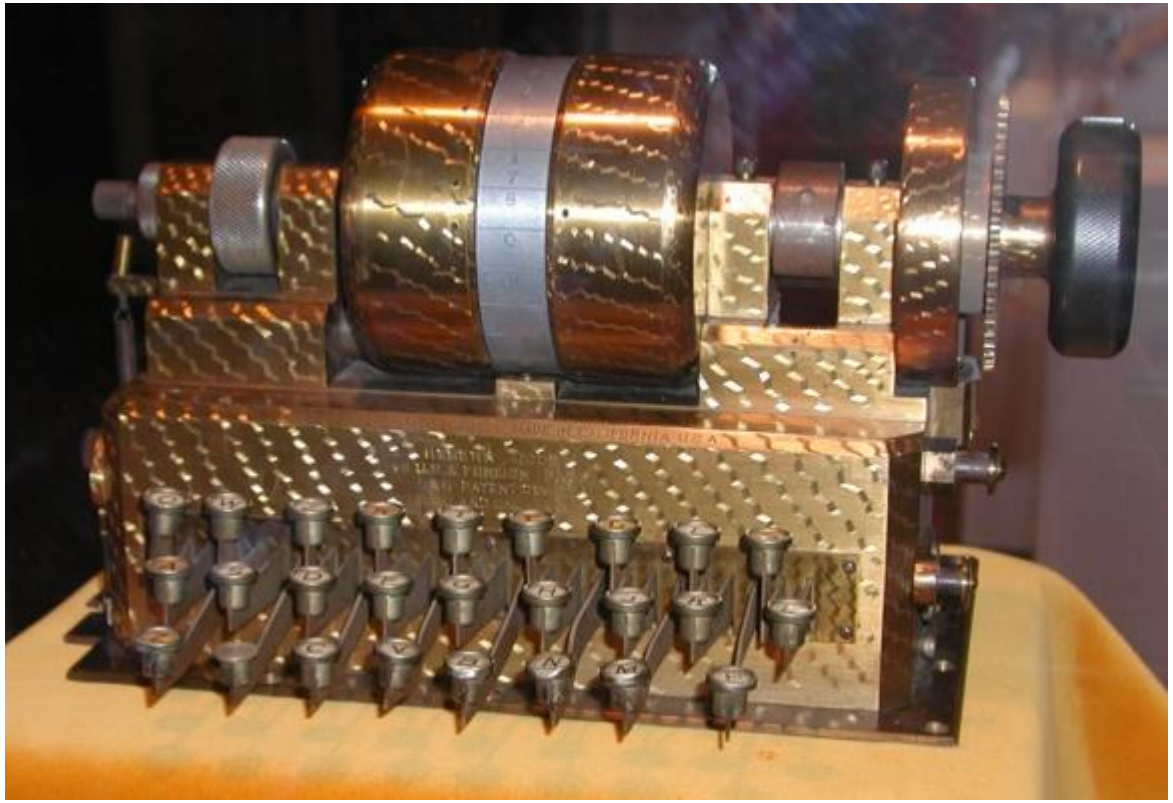
Now we have to use **brute-force method** to get all possible key values
~ there are **3x2x1x2x3=36** possible values which can be done
with brute-force without any issues

WE CONSIDER ALL THESE 36 POSSIBLE KEYS AND CHECK WHETHER THE DECRYPTED TEXT IS VALID (SO ENGLISH) OR NOT !!!

Роторні криптосистеми (1870-1943)

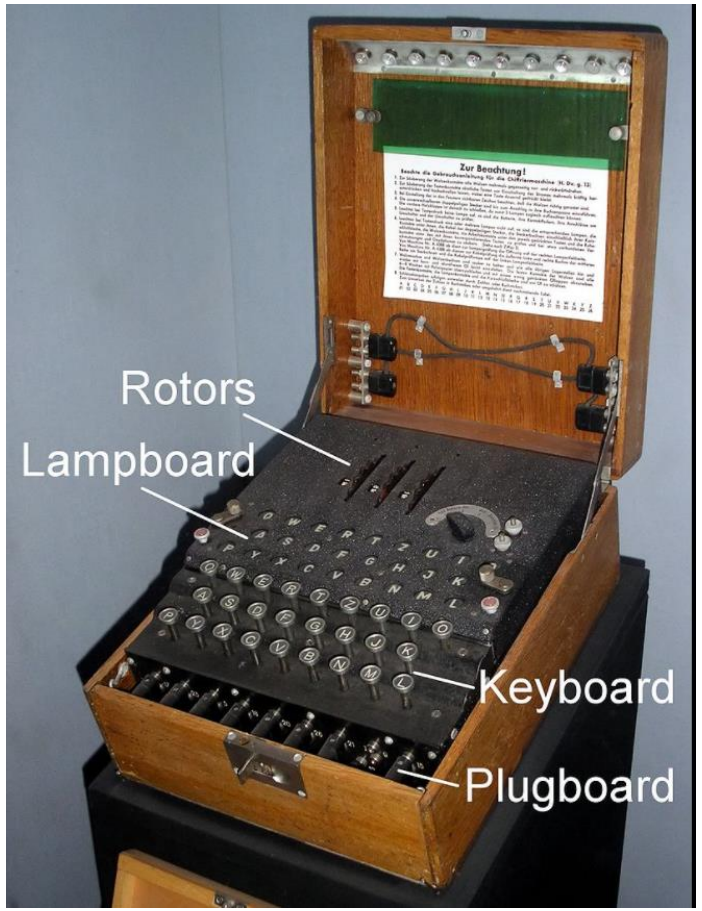
Hebern rotor machine

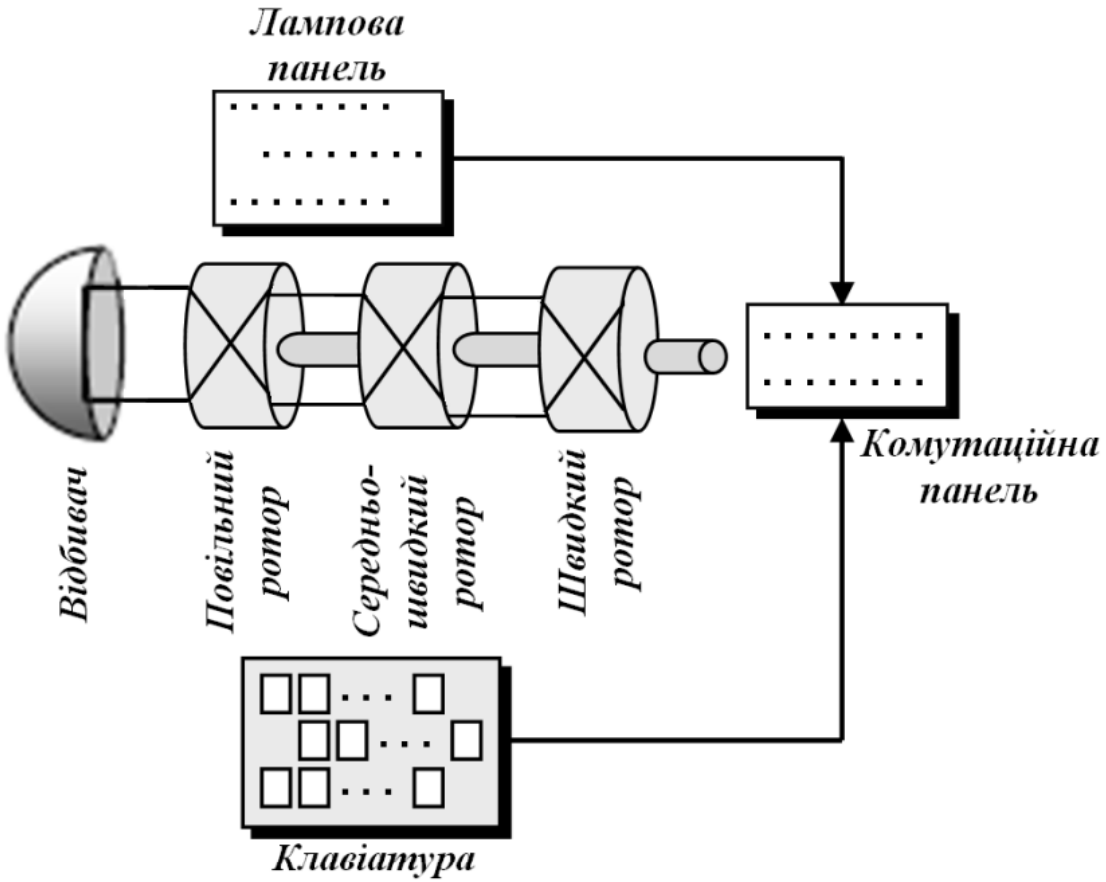




Шифр Enigma

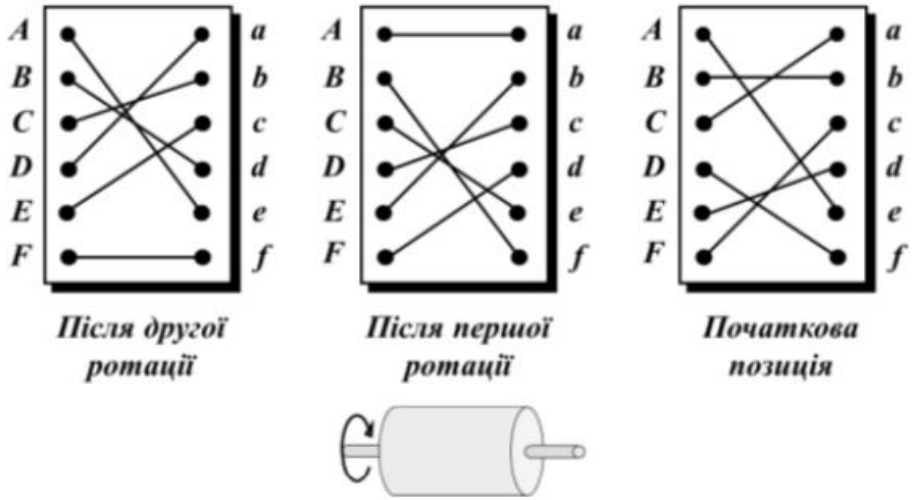
Знаменитий шифр час ІІ-ої світової війни Enigma можна трактувати як вдалу на тоді технічну реалізацію шифру Блеза де Віженера. Перша електронно-обчислювальна машина була сконструйована задля зламу шифру Enigma за участю видатного англійського математики Алана Тьюрінга. Шифр був винайдений німецьким інженером Arthur Scherbius на прикінці І-ї світової війни.





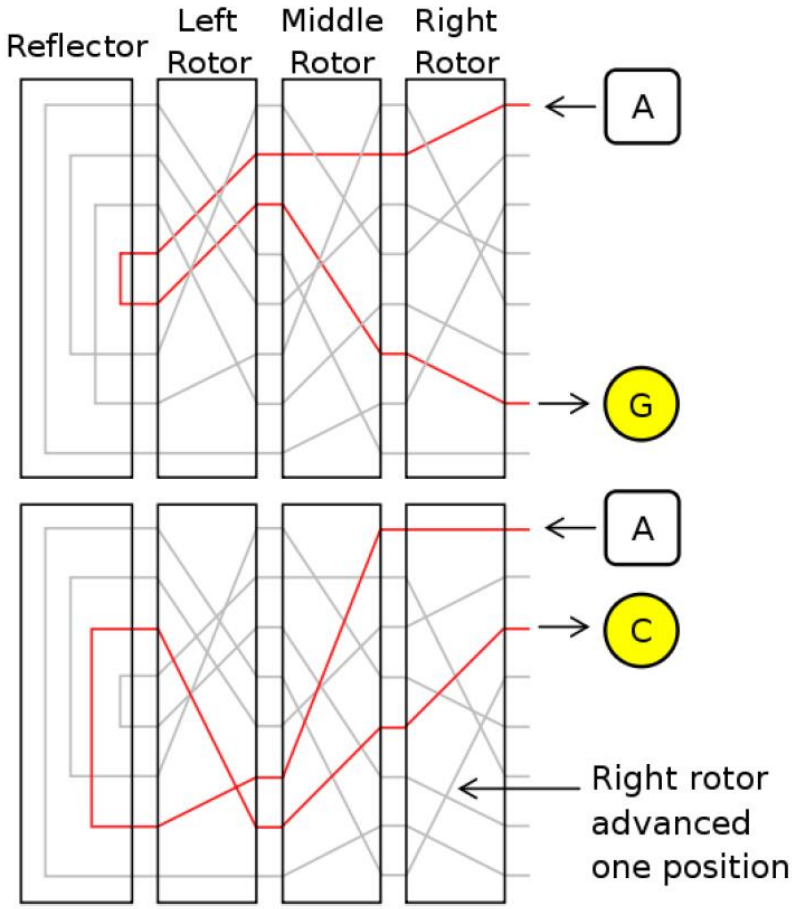
Роторний шифр Enigma

Початкова установка (позиція) ротора — ключ засекречування між відправником і одержувачем — це зашифрований перший символ вхідного тексту. Використовуючи початкову установку, другий символ зашифрований після того, як проведено перше обертання (на рис. — це поворот на 1/6 кола, на реальній установці — поворот на 1/26), і так далі.



Слово з трьома літерами, такими як “bee”, зашифровано як “BCA”. Це показує, що роторний шифр — поліалфавітний шифр, тому що дві появи того самого символу вхідного тексту зашифровані як різні символи.

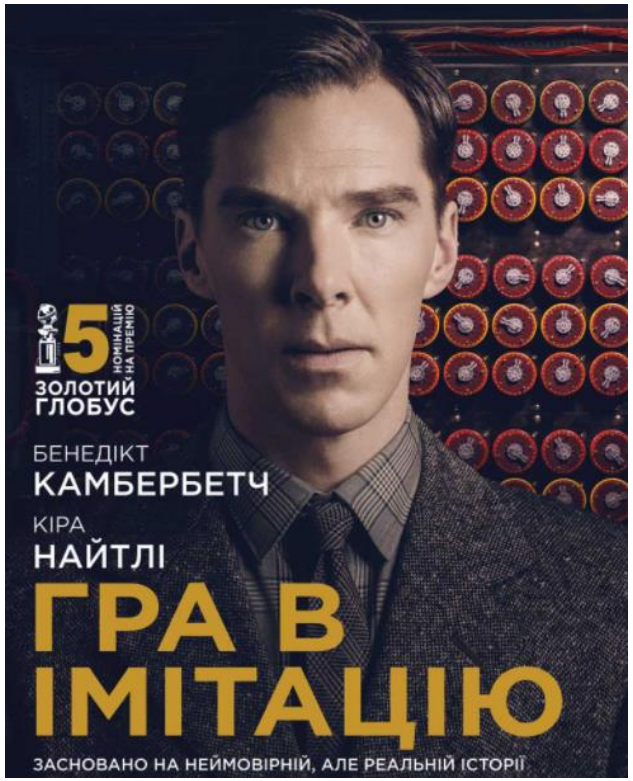
Роторний шифр Enigma



https://en.wikipedia.org/wiki/Enigma_machine

Роторний шифр Enigma

Щоб використовувати машину “Енігму”, було видано кодову книгу, яка протягом кожного дня дає кілька параметрів налаштування, включаючи: а) три ротори, які повинні бути вибрані з п'яти доступних; б) порядок, в якому ці ротори повинні бути встановлені; в) параметри установок для комутаційної панелі; г) код з трьома літерами дня.



Щоб зашифрувати повідомлення, оператор повинен послідовно зробити кроки, перераховані нижче:

1. Установити стартову позицію роторів згідно з кодом дня. На приклад, якщо код був “HUA”, ротори мають бути ініційовані на “H”, “U” і “A” відповідно.
2. Вибрати випадковий код із трьома літерами, наприклад “ACF”. Зашифрувати текст “ACFACF” (повторний код), використовуючи початкову установку роторів кроку 1. Наприклад, припустимо, що зашифрований код — “OPNABT”.
3. Установити стартові позиції роторів до “OPN” (половина зашифрованого коду).
4. Додати зашифрованих шість літер, отриманих на кроці 2 (“OPNABT”), до початкового повідомлення.
5. Зашифрувати повідомлення, включаючи код із шістьма літерами. Передати зашифроване повідомлення.

Роторний шифр Enigma

Щоб розшифрувати повідомлення, оператор повинен зробити такі кроки:

1. Отримати повідомлення й відокремити перші шість літер.
2. Встановити стартову позицію роторів згідно з кодом дня.
3. Розшифрувати перших шість літер, використовуючи початко-ву установку кроку 2.
4. Установити позиції роторів на першу половину розшифрова-ного коду.
5. Розшифрувати повідомлення (без перших шести літер).

Відомо, що “*Еніґму*” під час війни було зламано, хоча німецька армія й інша частина світу не знала про цей факт ще кілька десятиліть після того. Хоча німецька мова є надто складною, союзники, так чи інакше, отримали деякі копії машин. Наступним кроком був пошук параметрів установки для кожного дня й коду, переданого для ініціалізації роторів для кожного повідомлення. Винахід першого комп'ютера допоміг союзникам подолати й ці труднощі.

Шифри перестановки

Матричний шифр обходу. Повідомлення записується рядками у вигляді прямокутної матриці. Криптотекст формується зчитуванням букв із матриці у зміненому порядку, а саме, стовпчиками. При цьому послідовніст, у якій зчитуються стовпчики, визначається ключем.

```
GARDEN      Повідомлення:
416235        DON'T PUT IT OFF TILL TOMORROW.
------
DONTPU      Ключове слово:
TITOFF        GARDEN
TILLTO      Криптотекст:
MORROW        OIIOTOLRPFTODTTMUFOWNTLR
```

Шифри перестановки. Криптоаналіз.

Шифр перестановки розкривається спеціальним чином організованим аналізом частот біграм. Щоб визначити, що використовується шифр перестановки – досить пересвідчитись, що кожна буква зустрічається в повідомленні та криптотексті з однакової частотою. Нехай маємо криптотекс, який отриманий шифром перестановки з періодом 5. Розбиваємо криптотекст на блоки по 5 букв і запишемо їх один під другим.

ІЦКАЗИВИМЯИЛКИНОЙРЕСРПІНЗМЕЛБОПИРПИИТИНИИВІСВИТАЛП

ІЦКАЗ		**ЗАКЦІ**
ИВИМЯ	← Криптотекст	**ЯМИВИ**
ИЛКИН	Відкритий текст →	**НИКЛИ**
ОЙРЕС		**СЕРЙО**
РПІНЗ		**ЗНІПР**
МЕЛБО		**ОБЛЕМ**
ПИРПИ		**ИПРИП**
ИТИНИ		**ИНИТИ**
ИВІСВ		**ВСІВИ**
ИТАЛП		**ПЛАТИ**
12345	← Номери стовпчиків →	**54321**

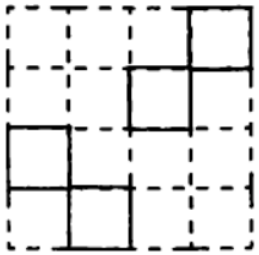
Дешифрування полягає у переставленні стовпчиків у належному порядку. Беручи до уваги позиції букви “**и**” приходимо до висновку, що не можутьзнаходитись поруч 1-й та 4-й, 2-й та 5-й стовпчики, а також будь-які два із 1-го, 3-го та 5-го стовпчиків.

Щоб пересвідчитись чи повідомлення зашифроване шифром перестановки достатньо перевірити, що кожна буква зустрічається у повідомленні та криптотексті з однаковою частотою.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

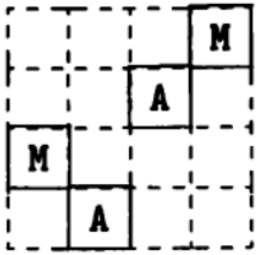
Шифр Кардано

Це блоковий шифр з періодом $l = k^2$оде k парне число. Ключем є вирізаний з паперу в клітинку квадрат розміру k на k, що складається з k^2 клітинок, четверту частину яких $\frac{k^2}{4}$ прорізують. Щоразу ключ повертають на 90 градусів і у нові позиції прорізаних клітинок вписують чергові літер повідомлення.



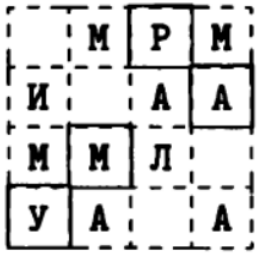
Ключ

Повідомлення



1-й крок

2-й крок



3-й крок

4-й крок: криптотекст

Загальний шифр перестановки з періодом l. Переставляє l букв у довільному порядку, який визначається ключем:

$$\begin{pmatrix} 1 & 2 & & l \\ i_1 & i_2 & & i_l \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 4 & 7 & 9 & 14 & 2 & 5 & 11 & 16 & 3 & 8 & 10 & 13 & 1 & 6 & 12 & 15 \end{pmatrix}$$