CTF на Физтехе

Занятие 4

Кодировки

ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	A	97	61	a
2	2	*[START OF TEXT]*	34	22	"	66	42	B	98	62	b
3	3	*[END OF TEXT]*	35	23	#	67	43	C	99	63	c
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	D	100	64	d
5	5	*[ENQUIRY]*	37	25	%	69	45	E	101	65	e
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	F	102	66	f
7	7	*[BELL]*	39	27	'	71	47	G	103	67	g
8	8	*[BACKSPACE]*	40	28	(	72	48	H	104	68	h
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	I	105	69	i
10	A	*[LINE FEED]*	42	2A	*	74	4A	J	106	6A	j
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	K	107	6B	k
12	C	*[FORM FEED]*	44	2C	,	76	4C	L	108	6C	l
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	M	109	6D	m
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	N	110	6E	n
15	F	*[SHIFT IN]*	47	2F	/	79	4F	O	111	6F	o
16	10	*[DATA LINK ESCAPE]*	48	30	0	80	50	P	112	70	p
17	11	*[DEVICE CONTROL 1]*	49	31	1	81	51	Q	113	71	q
18	12	*[DEVICE CONTROL 2]*	50	32	2	82	52	R	114	72	r
19	13	*[DEVICE CONTROL 3]*	51	33	3	83	53	S	115	73	s
20	14	*[DEVICE CONTROL 4]*	52	34	4	84	54	T	116	74	t
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	5	85	55	U	117	75	u
22	16	*[SYNCHRONOUS IDLE]*	54	36	6	86	56	V	118	76	v
23	17	*[ENG OF TRANS. BLOCK]*	55	37	7	87	57	W	119	77	w
24	18	*[CANCEL]*	56	38	8	88	58	X	120	78	x
25	19	*[END OF MEDIUM]*	57	39	9	89	59	Y	121	79	y
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	Z	122	7A	z
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*

ANSI Code Pages

Codepage 1251 - Cyrillic Windows

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1-	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2-	0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3-	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4-	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5-	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6-	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7-	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	007F
8-	Ђ 0402	Ѓ 0403	‚ 201A	ѓ 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	‹ 2039	Њ 040A	Ќ 040C	Ћ 040B	Џ 040F
9-	ђ 0452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	0098	™ 2122	љ 0459	› 203A	њ 045A	ќ 045C	ћ 045B	џ 045F
A-	00A0	Ў 040E	ў 045E	Ј 0408	¤ 00A4	Ґ 0490	¦ 00A6	§ 00A7	Ё 0401	© 00A9	Є 0404	« 00AB	¬ 00AC	- 00AD	® 00AE	Ї 0407
B-	° 00B0	± 00B1	І 0406	і 0456	ґ 0491	µ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	є 0454	» 00BB	ј 0458	Ѕ 0405	ѕ 0455	ї 0457
C-	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
D-	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
E-	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
F-	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F

Codepage 1252 - Latin 1 Windows

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1-	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2-	0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3-	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4-	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5-	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6-	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7-	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	007F
8-	€ 20AC	0081	‚ 201A	*f* 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152	008D	Ž 017D	008F
9-	0090	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153	009D	ž 017E	Ÿ 0178
A-	00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B-	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C-	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D-	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E-	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F-	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Unicode

- Каждому символу сопоставлено число от 0 до 0x100000 (code point)

Hello

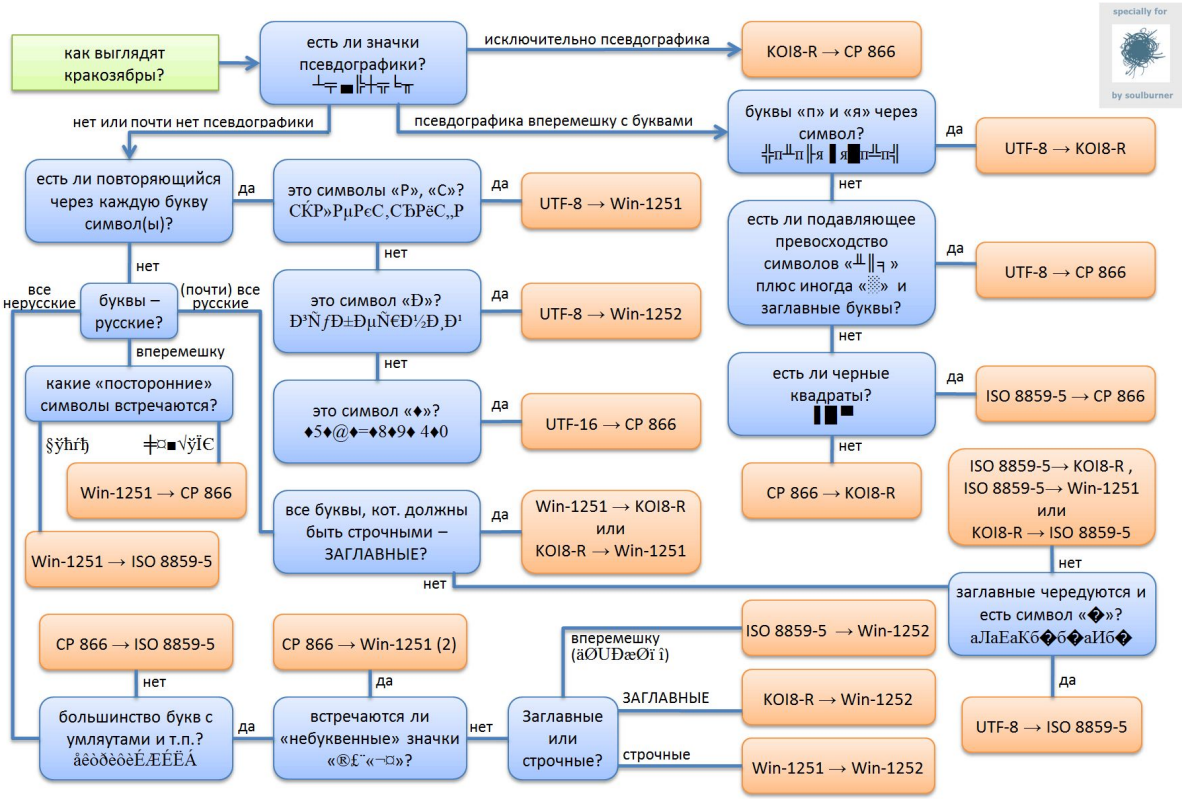
U+0048 U+0065 U+006C U+006C U+006F

UTF-8, UTF-16, UTF-32

- Можно закодировать любой Unicode символ (code point)
- UTF-8: каждый символ 1, 2, 3, 4, 5 или 6 байт
- UTF-16: каждый символ 2 или 4 байта
- UTF-32: каждый символ 4 байта
- **UTF-8 совпадает с ASCII на code point'ах < 128**

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+0000	U+007F	1	0xxxxxxx					
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx				
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx			
21	U+10000	U+1FFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
The patterns below are not part of UTF-8, but were part of the first specification.									
26	U+200000	U+3FFFFFF	5	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+4000000	U+7FFFFFFF	6	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

Кракозябры


как выглядят кракозябры?
есть ли значки псевдографики?
╩╤■╠╪╤└╥
исключительно псевдографика
KOI8-R → CP 866
нет или почти нет псевдографики
псевдографика вперемешку с буквами
буквы «п» и «я» через символ?
╬╥╨╥╟я▐я█п╨╥╣
да
UTF-8 → KOI8-R
нет
есть ли повторяющийся через каждую букву символ(ы)?
да
это символы «Р», «С»?
CЌP»PµPєC‚CЂPёC„P
да
UTF-8 → Win-1251
нет
есть ли подавляющее превосходство символов «╨╟╕ » плюс иногда «▒» и заглавные буквы?
да
UTF-8 → CP 866
нет
все нерусские
буквы – русские?
(почти) все русские
вперемешку
это символ «Ð»?
Ð³Ñƒ Ð±ÐµÑ€Ð½Ð‚Ð¹
да
UTF-8 → Win-1252
нет
есть ли черные квадраты?
▌▐▀
да
ISO 8859-5 → CP 866
нет
какие «посторонние» символы встречаются?
§ÿђґђ
╪¤■√ÿЇЄ
это символ «♦»?
♦5♦@♦=♦8♦9♦ 4♦0
да
UTF-16 → CP 866
нет
CP 866 → KOI8-R
ISO 8859-5→ KOI8-R , ISO 8859-5→ Win-1251 или KOI8-R → ISO 8859-5
Win-1251 → CP 866
Win-1251 → ISO 8859-5
все буквы, кот. должны быть строчными – ЗАГЛАВНЫЕ?
да
Win-1251 → KOI8-R или KOI8-R → Win-1251
нет
заглавные чередуются и есть символ «�»?
аЛаЕаКб�б�аИб�
нет
да
UTF-8 → ISO 8859-5
CP 866 → ISO 8859-5
CP 866 → Win-1251 (2)
вперемешку
(äÖUÐæØî ì)
ISO 8859-5 → Win-1252
ЗАГЛАВНЫЕ
KOI8-R → Win-1252
строчные
Win-1251 → Win-1252
Заглавные или строчные?
нет
да
встречаются ли «небуквенные» значки «®£¨«¬¤»?
да
нет
большинство букв с умляутами и т.п.?
åêòðèôèÉÆÉËÁ

There Ain't No Such Thing As Plain Text

base64

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Text content	M								a								n							
ASCII	77 (0x4d)								97 (0x61)								110 (0x6e)							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19								22								5							
Base64-encoded	T								W								F							

Winter is coming

V2ludGVyIGlzIGNvbWluZwo=

Bash

```
file --mime file.txt
```

```
iconv -f cp1251 -t utf8 old.txt > new.txt
```

```
echo ‘String to encode’ | base64
U3RyaW5nIHRvIGVuY29kZQo=
```

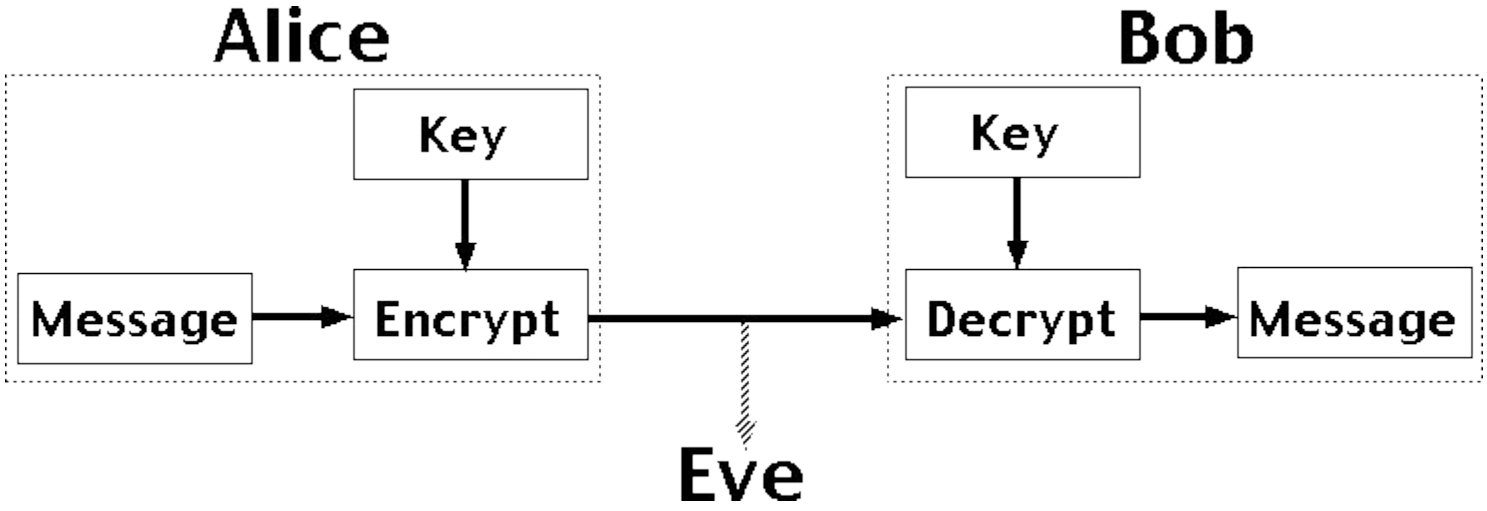
```
echo U3RyaW5nIHRvIGVuY29kZQo= | base64 -d
String to encode
```

Python

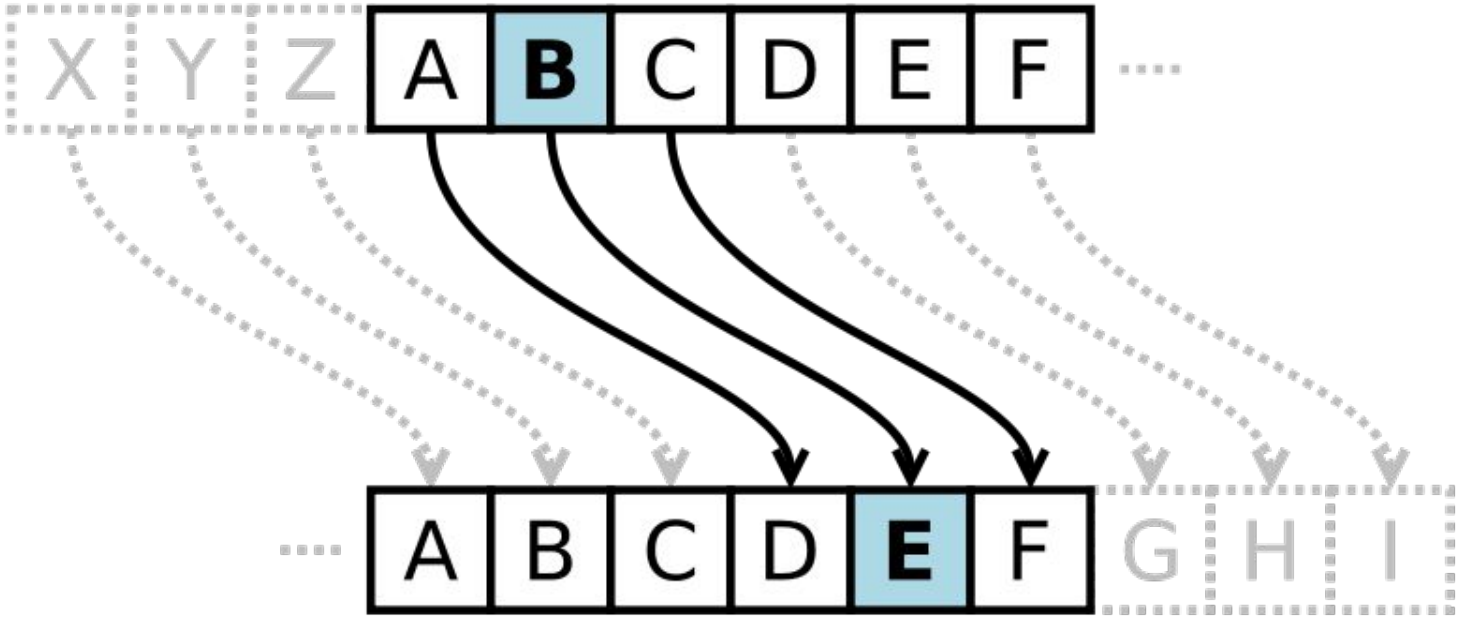
```
>>> import base64
>>> base64.b64encode('String to encode')
'U3RyaW5nIHRvIGVuY29kZQ=='
>>> base64.b64decode('U3RyaW5nIHRvIGVuY29kZQ==')
'String to encode'
```

Классические шифры

Alice, Bob & Eve



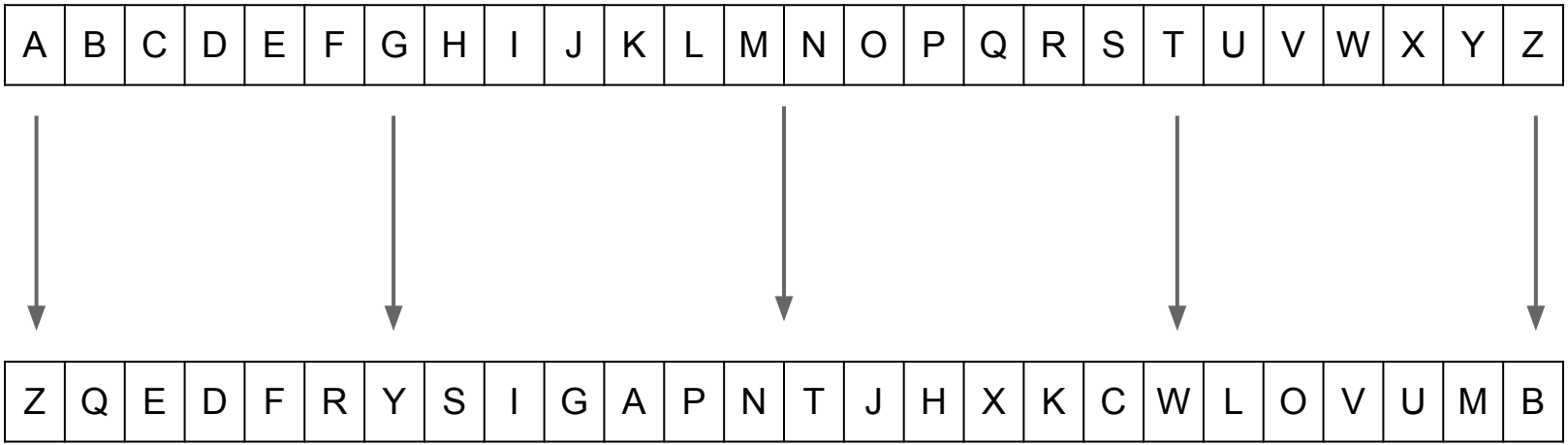
Шифр Цезаря



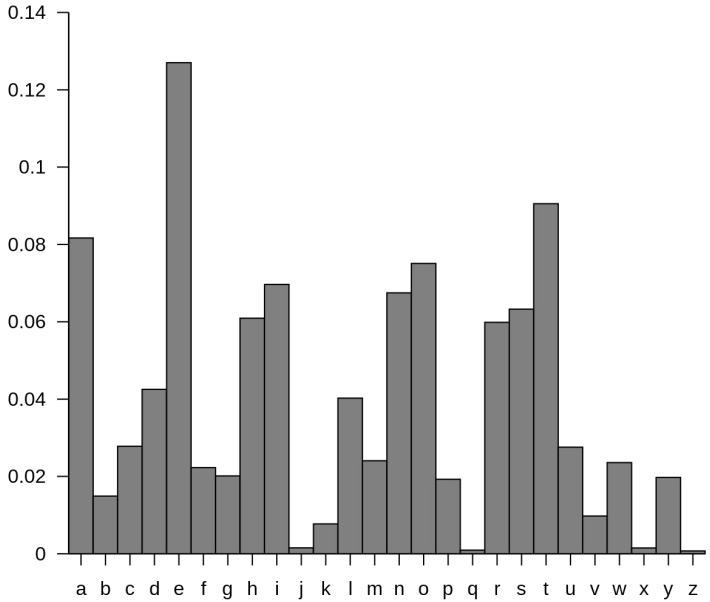
Шифр Цезаря

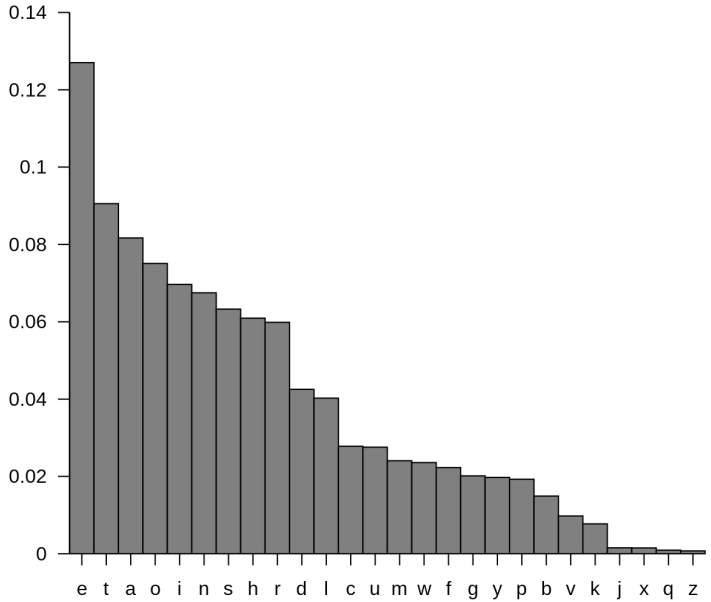
```
00 JVAGREVFPBZVAT
01 KWBHSFWGQCAWBU
02 LXCITGXHRDBXCV
03 MYDJUHYISECYDW
04 NZEKVIZJTFDZEX
05 OAFLWJAKUGEAFY
06 PBGMXKBLVHFBGZ
07 QCHNYLCMWIGCHA
08 RDIOZMDNXJHDIB
09 SEJPANEOYKIEJC
10 TFKQBOFPZLJFKD
11 UGLRCPGQAMKGLE
12 VHMSDQHRBNLHMF
13 WINTERISCOMING
14 XJOUFSJTDPNJOH
15 YKPVGTKUEQOKPI
16 ZLQWHULVFRPLQJ
17 AMRXIVMWGSQMRK
18 BNSYJWNXHTRNSL
19 COTZKXOYIUSOTM
20 DPUALYPZJVTPUN
21 EQVBMZQAKWUQVO
22 FRWCNARBLXVRWP
23 GSXDOBSCMYWSXQ
24 HTYEPCTDNZXTYR
25 IUZFQDUEOAYUZS
```

Шифр простой замены



Частотный анализ



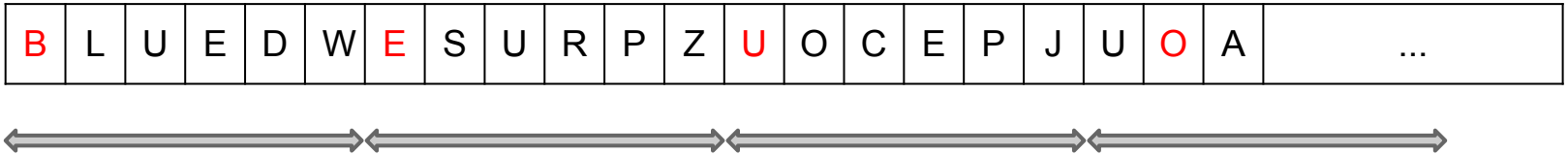


Шифр Виженера

	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

```
Plaintext:  WINTERISCOMING
Key:        THRONETHRONETH
Ciphertext: PPEHRVBZRCZMGN
```

Шифр Виженера



Длина ключа L = 6

- Пусть известна длина ключа L
- Каждый L’тый символ исходной строки зашифрован с помощью одного и того же символа строки-ключа (шифр Цезаря)
- Используем частотный криптоанализ

Индекс совпадений

- Как определить длину ключа L?
- Индекс совпадений - вероятность совпадения двух произвольных букв в строке:

$$I = \sum_i \frac{f_i(f_i - 1)}{n(n-1)}$$

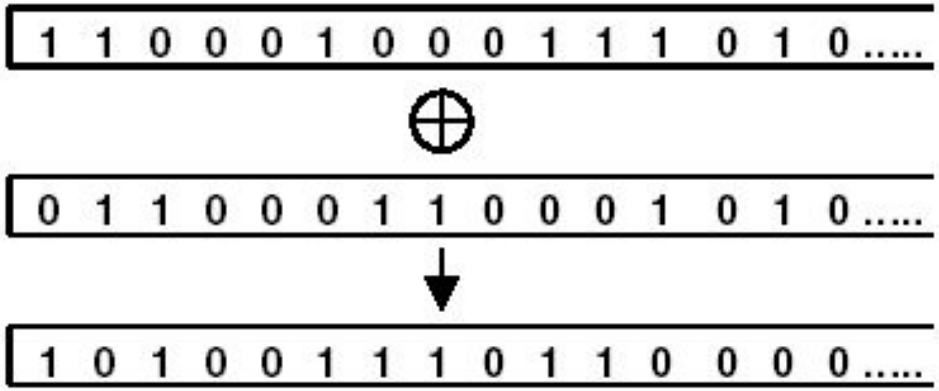
f_i - количество i'ой буквы в строке

n - длина строки

- Для текста на английском индекс совпадений ~ 0.065

L	1	2	3	4	5	6	7	8	9	...
I	0.038	0.043	0.047	0.043	0.038	0.060	0.037	0.042	0.049	...

Одноразовый блокнот



- Подсказка: `chr(ord('a') ^ ord(' ')) == 'A'`

Делайте то, что нетривиально

- Умеет ли это ваш текстовый редактор?
- Есть ли консольная утилита, которая это делает?
- Можно ли нагуглить веб-инструмент для решения этой задачи?
- Можно ли найти исходный код программы, которая это делает?
- Пишите утилиту сами, только если по всем прочим пунктам - “нет”.

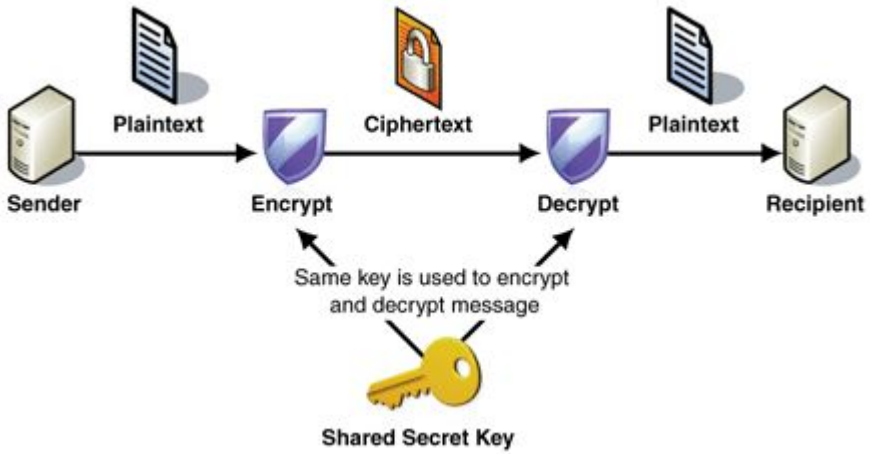
Инструменты

- Декодер (http://www.artlebedev.ru/tools/decoder/)
- quipquip (http://www.quipqiup.com/)
- ViGENER (https://f00l.de/hacking/vigenere.php)
- xortool (https://github.com/hellman/xortool)
- CrypTool (https://www.cryptool.org/en/)

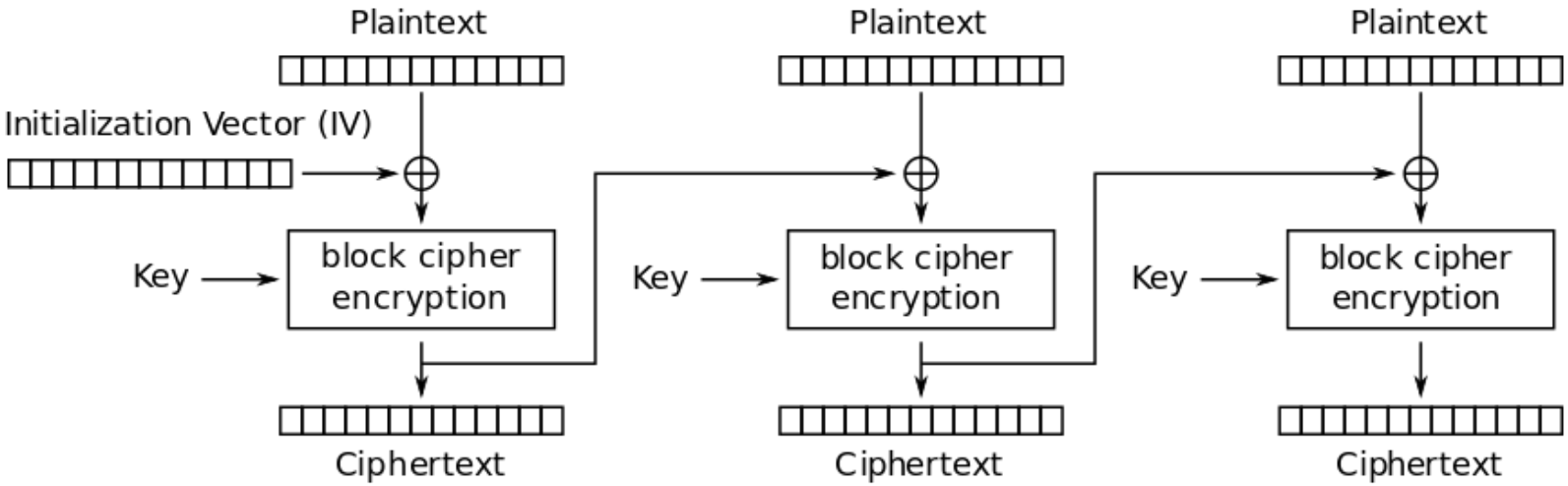
Современные шифры

Симметричное шифрование

- Для шифрования и расшифровки используется один и тот же ключ



Блочные шифры



Cipher Block Chaining (CBC) mode encryption

DES

- Data Encryption Standard
- Разработан IBM в 1970-ых
- Блочный шифр
- Размер блока - 64 бита
- Длина ключа - 56 бит (+ 8 бит для проверки целостности)
- Сейчас считается небезопасным

Атака на DES

- Полный перебор ключей (brute force)
- В 1997 RSA Laboratories запустили DES Challenge
- DES Challenge 1 = 140 days (1997)
- DES Challenge 2 = 41 days (1998)
- DES Challenge 3 = 23 hours (1999)

3DES

- Triple DES - DES примененный три раза с разными ключами
- C1 = Encrypt(P, K1)
- C2 = Decrypt(C1, K2)
- C3 = Encrypt(C2, K3)
- Длина ключа 168 бит
- Все еще используется

Атака Meet-In-The-Middle

- Meet-In-The-Middle (метод встречи посередине)
  - Пусть известен текст P и соответствующий шифротекст C
  - Пусть C = Encrypt(Encrypt(P, K1), K2)
  - Тогда Encrypt(P, K1) = Decrypt(C, K2)
  - Считаем Encrypt(P, K1) для всех K1
  - Считаем Decrypt(C, K2) для всех K2
  - Ищем совпадения Encrypt(P, K1) = Decrypt(C, K2)

Атаки на 3DES

- 3DES: K1 != K2 != K3 != K1
  - Атака Meet-In-The-Middle
  - Эффективная длина ключа 112 бит
- 3DES: K1 = K3
  - Атаки Chosen-Plaintext и Known-Plaintext
  - Эффективная длина ключа 80 бит

AES

- Advanced Encryption Standard (aka Rijndael)
- Принят новым стандартом по результатам конкурса NIST в 2001
- Блочный шифр
- Размер блока - 128 бит
- Длина ключа - 128, 192 или 256 бит
- Активно используется

Атаки на AES

- Ничего значительно лучше перебора пока не придумали
- Side-Channel атаки
  - Успешная Cache-Timing атака на реализацию AES в OpenSSL
  - Требует возможности запустить программу на том же компьютере, где происходит шифрование
  - Восстанавливает AES ключ “почти в реальном времени”

Вопросы?