Получаваме криптотекста TRBXHFGUIDYY:

```
c r y p t o g r a p h y
R A D I O R A D I O R A

T R B X H F G U I D Y Y
```

При дешифриране процедурата е следната - търсим реда, който има буквата T в стълба, индексиран с R. Така намираме c и продължаваме по същия начин до пълното дешифриране. Ако откритият текст е по-дълъг от ключовата дума, то ние я повтаряме многократно. Така в горния пример ключовата дума RADIO, приложена към текст от 12 букви приема вида RADIORADIORA.

Описаната процедура може да бъде използвана и с други квадрати, най-известен от които е квадратът на Beaufort. [5]

Формално шифърът на Vigenère може да бъде описан така. Нека $m > 1$ е фиксирано естествено число. Нека по-нататък $\mathcal{P} = \mathcal{C} = \{0, 1, \ldots, 25\}^*$ (не е много удобно да пишем $(\mathbb{Z}_{26}^*)^*$), а $\mathcal{K} = \{0, 1, \ldots, 25\}^m$. При зададен открит текст $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ и избран ключ $\boldsymbol{k}k = (k_0, k_1, \ldots, k_{m-1}$ шифрирането се задава чрез $E_{\boldsymbol{k}}(\boldsymbol{x}) = (y_0, y_1, \ldots, y_{n-1})$, където $y_i = x_i + k_{i \mod m} \pmod{26}$. За дешифрирането очевидно имаме $D_k(\boldsymbol{y}) = (z_0, z_1, \ldots, z_{n-1})$, където $z_i = y_i - k_{i \mod m} \pmod{26}$.

Ясно е, че и при шифъра на Хил и при този на Vigenére става въпрос за афинни трансформации на m-мерни вектори над $\mathbb{Z}_{26}$. Нека $\mathcal{M}_m(\mathbb{Z}_{26})$ е множеството на обратимите $m \times m$ матрици на $\mathbb{Z}_{26}$ и да положим

$$\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}^m, \ \mathcal{K} = \mathcal{M}_m(\mathbb{Z}_{26}) \times \mathbb{Z}_{26}^m.$$

шифрирането и дешифрирането са задаени чрез:

$$E_{(K,\boldsymbol{k})}(\boldsymbol{x}) = \boldsymbol{x}K + \boldsymbol{k}, \quad D_{(K,\boldsymbol{k})}(\boldsymbol{y}) = \boldsymbol{y}K^{-1} - \boldsymbol{k}K^{-1}.$$

Това обобщава по естествен начин шифъра на Хил. В случая $m = 1$ получаваме афинния шифър, а при $\boldsymbol{k} = \boldsymbol{0}$ – класическия шифъна Хил. Шифърът на Vigenére получаваме $K = I_m$, къдети I_m е единичната матрица от ред m.

При практическото използване на шифрите на Vigenére и Хил дължината на ключа е неизвестна за опонента и може да се счита за част от ключа. Това води до трудности при криптанализа, които не са решени по удовлетворителен начин до средата на XIX век.

1.6 Пермутационни шифри

Обща черта на шифрите, разгледани дотук, е замяната на буква или група от букви с друга буква или друга група от букви по определено правило. Друг подход при създаването на шифър е да запазим непроменени символите от откритиян текст и да променим позициите им. [6] а дефиниция на общ пермутационен шифър е следната.

[5] Носи името на адмирал сър Francis Beaufort, създател и на скала за измерване на скоростите на ветровете носеща неговото име.

[6] Пермутационните шифри се споменават за пръв път при Giovanni Porta (ок. 1563 г.)

Нека $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$ и нека $m \geq 2$ е фиксирано естествено число. Множествата на откритите и криптотекстовете са $\mathcal{P} = \mathcal{C} = \mathcal{X}$, а множеството на ключовете е множеството на всички пермутации на елементите на $\mathcal{X}$, т.е. $\mathcal{K} = S_m$ (тук S_m е симетричната група, действаща върху $\{1, \ldots, m\}$). При даден ключ $\pi \in \mathcal{K}$ шифриращата и дешифриращата трансформации се задават чрез

$$E_\pi(x_1, x_2, \ldots, x_m) = (x_{\pi(1)}, x_{\pi_2}, \ldots, x_{\pi(m)})$$

и

$$D_\pi(y_1, y_2, \ldots, y_m) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \ldots, y_{\pi^{-1}(m)}),$$

където π^{-1} е обратната пермутация на π.

Пример 1.3. Нека $m = 10$ и ключ е пермутацията

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 3 & 6 & 9 & 2 & 5 & 8 & 1 & 4 & 10 & 7 \end{pmatrix} = (1\ 3\ 9\ 10\ 7)(2\ 6\ 8\ 4\ )(5).$$

Очевидно

$$\pi^{-1} = (1\ 7\ 10\ 9\ 3)(2\ 4\ 8\ 6)(5).$$

Откритият текст

`T H I S I S T H E W   I N T E R O F O U R   D I S C O N T E N T`

се шифрира в

`I S E H I H T S W T   T O U N R O I E R F   S N N I O E D C T T`

И тук параметърът m остава скрит и може да се счита за част от ключа.

Не е трудно да се забележи, че общият пермутационен шифър може да се разглежда като частен случай на шифъра на Hill. Да заменим множеството $\mathcal{X}$ с $\mathbb{Z}_{26}$. С всяка пермутация $\pi \in S_m$ може да се свърже пермутационна матрица[7] $K_\pi = (k_{i,j})_{m \times m}$, зададена чрез:

$$k_{i,j} = \begin{cases} 1 & \text{ако } i = \pi(j), \\ 0 & \text{в противен случай.} \end{cases}$$

есно се пролерява, че ако $G \leq S_m$ е група от пермутации, множеството от матрици

$$H = \{K_\pi \mid \pi \in G\}$$

група от матрици изоморфна на G. В частност

$$K_\pi^{-1} = K_{\pi^{-1}} = K_\pi^T.$$

[7] Пермутационна матрица от ред n е всяка матрица, която се получава от единичната матрица от ред n чрез елементарни преобразувания по редове стълбове. С други думи, пермутационна матрица е всяка $(0, 10$-матрица с точно една единица във всеки ред и всеки стълб.

Така пермутацията, коыото използвахме в горния пример свързваме матрицата

$$K_\pi = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Сега очевидно

$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})K = (x_3, x_6, x_9, x_2, x_5, x_8, x_1, x_4, x_{10}, x_7),$$

т.е. разместването на символите в рамките на един блок може да се реализира чрез умножние отдясно с подходяща пермутационна матрица.

Както и при простата субституция и тук може да се използват различни евристики за запомняне на пермутацията π. Един начин е да използваме таблица с m реда и n стълба. Откритият текст се записва в таблицата по редове, а криптотекстът се получава след прочитане на буквите от таблицата по стълбове. Така с открития текст от примера по-горе и таблица 5×6 получваме

T	H	I	S	I	S
T	H	E	W	I	N
T	E	R	O	F	O
U	R	D	I	S	C
O	N	T	E	N	T

и криптотекстът е `TTTUO HHERN IERDT SWOIE IIFSN SNOCT`. Сега ключът може да се мисли като двойката числа, задаващи размера на таблицата, в случая $k = (6, 5)$. Описаното преобразувание се реализира от общ пермутационен шифър с ключ

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & 29 & 30 \\ 1 & 7 & 13 & 19 & \ldots & 24 & 30 \end{pmatrix}.$$

Разбира се, описаната процедура намалява броя на възможните ключове и отслабва системата. Едно възможно усилване е да поставим забранени полета в таблицата.

T		H	I	S	I
S	T	H		E	W
I	N	T	E	R	
O	F		O	U	R
D	I	S	C		O
	N	T	E	N	T

Тук получаваме криптотекста `TSIOD TNFIN HHTST IEOCE SERUN IWROT`. Друга възможност да запишем текста вфиксиран брой стълбове, които да разместим в съответствие с някава ключова дума. да изберем, например, ключовата дума `TABLE`. записваме откриития текст в пет стълба (колкото е дължината на ключовата дума) като записъсе извършва по редове. Всеки стълб отговаря на буква от ключовата дума. След това записваме стълбовете в реда, определен от естествения ред на буквите [8] в ключовата дума. В нашия пример получаваме

T	A	B	L	E
1	2	3	4	5
T	H	I	S	I
S	T	H	E	W
I	N	T	E	R
O	F	O	U	R
D	I	S	C	O
N	T	E	N	T

и криптотекстът е `HTNFI TIHTO SEIWR ROTSE EUCNT SIODN`.

1.7 Поточни шифри

В шифрите, разгледани дотук, преобразувахме последователни блокове открит текст в блокове криптотекст по правило, зависещо от избрания ключ k. Формално, ако откритият текст x е разбит на блокове $x_1, x_2, x_3, \ldots$, то криптотекстът y е

$$y = y_1y_2y_3 \ldots = E_k(x_1)E_k(x_2)E_k(x_3) \ldots .$$

Друга идея е да използваме ключа k за генериране на редица $z-1, z_2, z_3, \ldots$, наречена *ключова редица*, с чиято помощ да шифрираме открития текст:

$$y = y_1y_2y_3 \ldots = E_{z_1}(x_1)E_{z_2}(x_2)E_{z_3}(x_3) \ldots .$$

При това елементът z_i се получава като стойност на някаква функция f_i, която зависи от ключа и от първите $i-1$ блока от открития текст, т.е. имаме

$$z_i = f_i(k, x_1, \ldots, x_{i-1}.$$

Тук шифриращите трансформации се индексират не от елементитена $\mathcal{K}$, а от елементите z_i на ключовата редица.

Както и при общата дефиниция на криптосистема и тук можем да дадем формална дефиниция на понятието *поточен шифър*.

Дефиниция 1.4. Поточен шифър наричаме наредената седморка $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{F}, \mathcal{E}, \mathcal{D})$, където $\mathcal{P}$ и $\mathcal{C}$ са, сътветно, множествата на откритите текстове и криптотекстовете, а $\mathcal{K}$ е множеството на ключовете. Множеството $\mathcal{L}$ е азбуката на ключовия поток, а $\mathcal{F}$ е редица от функции $F_1 f_2, f_3, \ldots$, където

$$f_i : \mathcal{K} \times \mathcal{P}^{i-1} \to \mathcal{L}.$$

[8] Тук под естествен ред разбираме реда на буквите в азбуката.

За всяко $z \in \mathcal{L}$ съществуват правило за шириране $E_z \in \mathcal{E}$ и правило за дешифриране $D_z \in \mathcal{D}$,

$$E_z : \mathcal{P} \to \mathcal{C}, \quad D_z : \mathcal{C} \to \mathcal{P},$$

за които е изпълнено $D_z(E_z(x)) = x$ за всяко $x \in \mathcal{P}$.

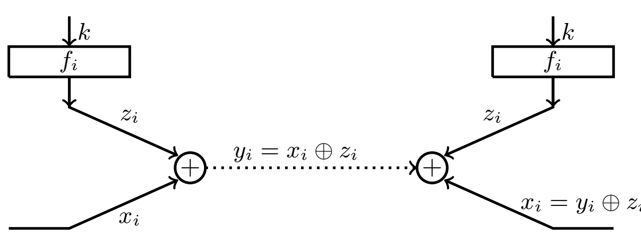
Разликата от общата дефиниция за криптосистема се състои в това, че шифриращата и дешифриращата трансформации не зависят пряко от ключа k, а от ключовата редица, която се генерира от него. Блоковите шифри могат да се разглеждат като поточни шифри от специален вид, т.е. такива, при които $z_i = k$ за всяко $i \geq 1$.

Един поточен шифър наричаме *синхронен*, ако ключовата редица $z_1, z_2, z_3, \ldots$ не зависи от открития текст, т.е. $f_i : \mathcal{K} \to \mathcal{L}$. В този случай ключът генерира ключовия поток, независимо от открития текст. Поточен шифър, за който ключовата редица зависи и от открития текст, наричаме *несинхронен*. Един поточен шифър наричаме *периодичен* с период d, ако $z_{i+d} = z_i$ за всяко $i \geq 1$. [9] В случая на шифър на Vigenère с ключова дума $k = (k_1, \ldots, k_m)$ имаме $z_{jm+i} = k_i, i = 1, \ldots, m, \ j = 1, 2, \ldots$. Сега шифриращата и деширираща функции, индексирани със z, съвпадат с тези на транслционния шифър $E_z(x) = x + z$ и $D_z(y) = y - z$, където събирането и изваждането са в $\mathbb{Z}_{26}$.

Много често при поточните шифри се използват двоични азбуки, т.е. $\mathcal{P} = \mathcal{C} = \mathcal{L} = \mathbb{Z}_2$. В този случай шифрирането и дешифрирането са идентични и се състоят в събиране по модул 2:

$$E_z(x) = x \oplus z, \;\; D_z(y) = y \oplus z.$$

На схемата по-долу е представен такъв шифър.



Една възможност за създаване на (синхронен) ключов поток е използванто на линейна рекурентна редица над $\mathbb{Z}_2$ Нека първите m члена на редицата са фиксирани:

$$z_0 = b_0, z_1 = b_1, \ldots, z_{m-1} = b_{m-1}$$

и нека тя да удовлетворява рекурентното уравнение

$$z_{i+m} = c_0 z_i \oplus c_1 z_{i+1} \oplus \ldots \oplus c_{m-i} z_{i+m-1}, \tag{1.1}$$

където $c_, c_1, \ldots, c_{m-1}$ са подходящо избрани константи от $\mathbb{Z}_2$. Без ограничение на общността ще приемем, че $c_0 = 1$; в противен случай рекурентното уравнение е от

[9] На практика при периодичните шифри допускаме и наличието на *предпериод*, т.е. $z_{i+d} = z_i$ от някакво място нататък.

ред по-малък от m. Приемаме, че ключът е с дължина $2m$ и се състои от елементите (битовете) $b_0, b_1, \dots, b_{m-1}$ и $c_0, c_1, \dots, c_{m-1}$. Разбира се, трябва да изберем $(b_0, \dots, b_{m-1}) \neq (0, \dots, 0)$, както и $(c_0, \dots, c_{m-1}) \neq (0, \dots, 0)$, тъкато в противен случай (z_i) е нулевата редица[10] и криптотекстът е идентичен с открития текст. По-нататък ще докажем, че при подходящ избор на $c_0, \dots, c_{m-1}$, редицата (z_i) е с период $2^m - 1$, който е и максималният възможен. така с относително къс ключ с дължина $2m$ генерираме редица с голчм период – $2^m - 1$.

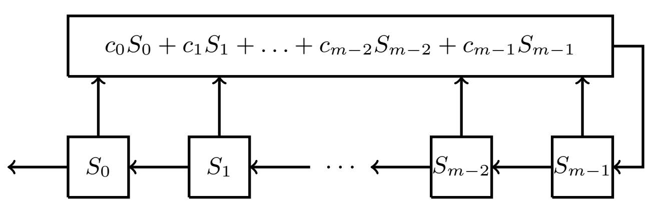
Да отблежим, че линейните рекурентни редици могат да бъдат генерирани ефективно с устройства, наречени линейни регистри с обратна връзка (linear feedback shift registers, LFSR). Редицата, получена от (1.1) се получава от линеен регистър с дължина m.

Един ценен аспект на линейните рекурентни редици е възможността те да бъдат ефективно имплементирани в хардуер чрез т.нар. *линейни регистри с обратна врвзјка* (linear feedback shift register или LFSR). Един линеен регистър се състои от m клетки $S_0, S_1, \dots, S_{m-1}$, във всяка от които може да бъде записан символ от предварително зададена азбука (в нашия случай $\{0, 1\}$). Той работи в дискретни моменти от време $t, t{+}1, t{+}2, \dots$. В началния момент клетките $S_0, \dots, S_{m-1}$ съдържат m-орката $(b_0, \dots, b_{m-1})$. На всеки такт регистърът извършва следните операции:

1) съдържанието на S_0 се извежда като изходен бит;

2) съдържанието на всяка от клетките $S_1, \dots, S_{m-1}$ се прехвъля в клетката вляво, т.е. $S_{i-1} \leftarrow S_i$, $i = 1, \dots, m-1$;

3) новата стойност на S_{m-1} се задава чрез $S_{m-1} \leftarrow \sum_{j=0}^{m-1} c_j S_j$.

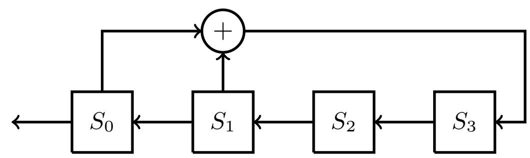


Ако означим със $S_i(t)$ съдържанието на клетката S_i в момент t, то работата на линейния регистър се описва с равенствата

$$\left|\begin{array}{lcl} S_i(t+1) & = & S_i(t), \;\; i = 0, \dots, m-2, \\ S_{m-1}(t+1) & = & c_0S_0(t) + c_1S_1(t) + \dots + c_{m-1]S_m-1}(t). \end{array}\right.$$

Да разгледаме линеен регистър с дължина $m = 4$, който генерира редица, удовлетворяваща рекурентното уравнение $z_{i+4} = z_i \oplus z_{i+1}$.

[10]Ако $(c_0, \dots, c_{m-1}) = (0, \dots, 0)$ имаме $z_{m+i} = 0$ за $i \geq 0$.



Ако в началния момент регистърът съдържа $(b_0, b_1, b_2, b_3) = (1, 0, 0, 0)$, то работата му се описва от таблицата по-долу.

t	S_0	S_1	S_2	S_3
0	1	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	1
5	0	0	1	1
6	0	1	1	0
7	1	1	0	1
8	1	0	1	0
9	0	1	0	1
10	1	0	1	1
11	0	1	1	1
12	1	1	1	1
13	1	1	1	0
14	1	1	0	0
15	1	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Първият стълб съдържа редицата, която се генерира от регистъра. Тые е с период 15 и това е максималният възможен период за редица генерирана от линеен регистър с дължина 4:

$$(1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1), 1, 0, 0, 0, 1, 0, 0, \ldots.$$

Сега ще представим един асинхронен поточен шифър, известен като *автоключ*. Преполага се, че негов автор е Vigenère. Идеята на този шифър е да се използва самият открит текст като ключов поток. Нека $m \geq 1$ е цяло чсило. Полагаме $\mathcal{P} = \mathcal{C} = \mathcal{L} = \mathbb{Z}_{26}$ и $\mathcal{K} = \mathbb{Z}_{26}^m$. Ако откритият текст е $x_0 x_1 x_2 \ldots$, а $K = k_0 k_1 \ldots k_{m-1}$, то полагаме $z_i = k_i$ за $i = 0, 1, \ldots, m-1$ и $z_i = x_{i-m}$ за $i \geq m$. Шифриращата и дешифриращата трансформации се задават съответно чрез $E_z(x) = x + z \pmod{26}$ и $D_z(y) = y - z \pmod{26}$.

Пример 1.5. Да разгледаме автоключ с $m = 3$, ключова дума `MAY`$= (14, 0, 24)$ и съобщението `thepathoftherighteous`. Използвайки стандартното кодиране на английската азбука, получаваме криптотекста `FHCIHXWOYAVJKPKYBKVNW`:

t	h	e	p	a	t	h	o	f	t	h	e	r	i	g	h	t	e	o	u	s
m	a	y	t	h	e	p	a	t	h	o	f	t	h	e	r	i	g	h	t	e
19	7	4	15	0	19	7	14	5	19	7	4	17	8	6	7	19	4	14	20	18
12	0	24	19	7	4	15	0	19	7	14	5	19	7	4	17	8	6	7	19	4
5	7	2	18	7	23	22	14	24	0	21	9	10	15	10	24	1	10	21	13	22
F	H	C	I	H	X	W	O	Y	A	V	J	K	P	K	Y	B	K	V	N	W

Дешифрирането е очевидно. при намирането на първите $m = 3$ букви от открития текст получателят използва ключа k, а за следващите символи - вече възстановените букви от открития текст. Първите няколко стъпки при дешифрирането на горния криптотекст изглеждат така:

криптотекст	ключова редица	дешифриране	otkrit tekst
F	M	$5 - 12 = 19$	t
H	A	$7 - 0 = 7$	h
C	Y	$2 - 24 = 4$	e
I	T	$8 - 19 = 15$	p
H	H	$7 - 7 = 0$	a
X	E	$23 - 4 = 19$	t
W	P	$22 - 15 = 7$	h

1.8 Шифър на **Vernam**

Естествено продължение на шифъра на Vigenére е шифърът на Vernam.[11] Разликите с шифъра на Vigenér са че (1) използваната азбука е двоична и (2) ключът е с дължина, равна на дължината на открития текст. Формално имаме $\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$ и

$$E_k(x) = x \oplus k, \;\; D_k(y) = y \oplus k,$$

като събирането е побитово. Този шифъе известен и под името *еднократен ключ*, тъй като всеки бит от ключа се използва само веднъжпри шифриране.

Vernam създава и патентова[12] устройство, при което ключът се съхранява върху перфолента с достатъчно голяма дължина. Идеята няма финансов успех, макар да е с изключителна теоретична важност.[13] Основната трудност при този шифър лежи в създаването и съхраняването на огромния по обем ключов материал, необходим при интензивен обмен на данни. [14] Тези трудности могат да бъдат овладяни само при чисто двустранни комуникации като, например, между дипломатически представителства, шпиони или висшите нива на военните щабове.[15]

[11] Gilbert Vernam (1890-1960) Служител на AT&T.

[12] патент No1,310,719 от 1918 г.

[13] Още през 1917 г. Parker Hill отбелязва: No message is safe in cipher unless the key phrase is comparable in length with the message itself.

[14] Например при нестабилни ситуации (на бойното поле). Еднократният ключ по дефиниция трябва да бъде унищожаван след използване. При уредите на Vernam това може да става механично.

[15] В тайните служби на СССР към 1926 г. се преминава към използване на индивидуални еднократни ключове. За създаването и използването на ключове вж. F. L. Bauer, Entzifferte Geheimnisse, S. 148.

Забележително е, че шифърът на Vernam притежава свойството “съвършена сигурност” (perfect secrecy), за което ще говорим по-късно. С това той е първият доказано сигурен шифър.