

WSTĘP DO KRYPTOGRAFII

ĆWICZENIA IV (własności ciągów binarnych)

Zagadnienia:

- odległość Hamminga i korelacja funkcji boolowskich;
- transformata Walsha-Hadamarda,
- odwrotna transformata Walsha-Hadamarda,
- algorytm kompresji metodą LZ77,
- algorytm dekompresji metodą LZ77,
- rejestr LFSR.

Przypomnienie: odległością Hamminga funkcji boolowskich

$$f : Z_2^n \rightarrow Z_2 \quad \text{ i } \quad g : Z_2^n \rightarrow Z_2$$

określonych tablicami prawdy $X = [x_0 \ x_1 \ \dots \ x_{2^n-1}]$ oraz $Y = [y_0 \ y_1 \ \dots \ y_{2^n-1}]$ nazywamy funkcję $d : Z_2^{2^n} \times Z_2^{2^n} \rightarrow \mathbb{N}$ postaci:

$$d(X, Y) = hwt(X \oplus Y),$$

gdzie $hwt(\alpha)$ jest wagą Hamminga ciągu binarnego α .

Przypomnienie: korelacją funkcji boolowskich

$$f : Z_2^n \rightarrow Z_2 \quad \text{i} \quad g : Z_2^n \rightarrow Z_2$$

określonych tablicami prawdy $X = [x_0 \ x_1 \ \dots \ x_{2^n-1}]$ oraz $Y = [y_0 \ y_1 \ \dots \ y_{2^n-1}]$ nazywamy funkcję $c : Z_2^{2^n} \times Z_2^{2^n} \rightarrow \mathbb{R}$ postaci:

$$c(X, Y) = 1 - \frac{d(X, Y)}{2^{n-1}}.$$

Zadanie 1:

podaj przykład tablic prawdy X oraz Y dla 12-bitowych funkcji

$$f : Z_2^{12} \rightarrow Z_2 \quad \text{ i } \quad g : Z_2^{12} \rightarrow Z_2,$$

zakładając, że wartość wyrażenia

$$d(X, Y) - (c(X, Y))^2$$

ma być maksymalna (oblicz tą wartość).

Rozwiązanie: niech $\alpha = d(X, Y) - (c(X, Y))^2$. Korzystając z definicji funkcji odległości Hamminga i korelacji funkcji binarnych, możemy napisać:

$$\alpha = hwt(X \oplus Y) - \left(1 - \frac{hwt(X \oplus Y)}{\beta}\right)^2,$$

gdzie $\beta = 2^{12-1} = 2^{11}$. Interesuje nas minimum funkcji α zmiennej $hwt(X \oplus Y)$, zatem:

- przekształcamy funkcję α do uproszczonej postaci wielomianowej:

$$\begin{aligned}\alpha &= hwt(X \oplus Y) - \left(1 - \frac{hwt(X \oplus Y)}{\beta}\right)^2 \\ &= hwt(X \oplus Y) - \left(1 - \frac{2}{\beta} \cdot hwt(X \oplus Y) + \frac{1}{\beta^2} \cdot (hwt(X \oplus Y))^2\right) \\ &= -\frac{1}{\beta^2} \cdot (hwt(X \oplus Y))^2 + \frac{\beta + 2}{\beta} \cdot hwt(X \oplus Y) - 1.\end{aligned}$$

- obliczamy pierwszą pochodną po $hwt(X \oplus Y)$ funkcji α :

$$\alpha' = -\frac{2}{\beta^2} \cdot hwt(X \oplus Y) + \frac{\beta + 2}{\beta}.$$

- obliczmy drugą pochodną po $hwt(X \oplus Y)$ funkcji α :

$$\alpha'' = -\frac{2}{\beta^2}.$$

- zauważamy, że $\beta > 0 \Rightarrow \alpha'' < 0$, czyli jedyne optimum funkcji α stanowi zarazem maksimum tej funkcji. Na tej podstawie $\alpha = d(X, Y) - (c(X, Y))^2$ przyjmuje wartość maksymalną wttw. gdy $\alpha' = 0$, czyli:

$$0 = -\frac{2}{\beta^2} \cdot hwt(X \oplus Y) + \frac{\beta + 2}{\beta}$$

$$\begin{aligned} hwt(X \oplus Y) &= \frac{\beta + 2}{\beta} \cdot \frac{\beta^2}{2} \\ &= \frac{\beta \cdot (\beta + 2)}{2}. \end{aligned}$$

Ponieważ $\frac{\beta \cdot (\beta + 2)}{2} = \frac{2^{11} \cdot (2^{11} + 2)}{2} = 2^{10} \cdot (2^{11} + 2) > 2^{12}$ i α jest wielomianem drugiego stopnia to wartość wyrażenia $d(X, Y) - (c(X, Y))^2$ jest maksymalna dla $hwt(X \oplus Y) = 2^{12}$. Zatem wystarczy aby:

$$X = \begin{bmatrix} 0_0 & 0_1 & \dots & 0_{2^n-1} \end{bmatrix} \quad \text{i} \quad Y = \begin{bmatrix} 1_0 & 1_1 & \dots & 1_{2^n-1} \end{bmatrix}.$$

Wtedy

$$\begin{aligned} d(X, Y) - (c(X, Y))^2 &= 2^{12} - \left(1 - \frac{2^{12}}{2^{11}}\right)^2 \\ &= 2^{12} - (-1)^2 = 2^{12} - 1. \end{aligned}$$



Przypomnienie: niech $f : Z_2^n \rightarrow Z_2$ będzie funkcją boolowską określoną tablicą prawdy $X = \begin{bmatrix} x_0 & x_1 & \dots & x_{2^n-1} \end{bmatrix}$, wtedy transformatą Walsh-Hadamarda funkcji f nazywamy funkcję $F : Z_2^n \rightarrow \mathbb{R}$ reprezentowaną przez tablicę prawdy $Y = \begin{bmatrix} y_0 & y_1 & \dots & y_{2^n-1} \end{bmatrix}$ taką, że:

$$y_j = \sum_{i=0}^{2^n-1} f(i_B) \cdot (-1)^{\langle i_B, j_B \rangle},$$

dla $j \in \{0, 1, \dots, 2^n-1\}$.

Zadanie 2:

niech $f : Z_2^2 \rightarrow Z_2$ będzie funkcją boolowską zgodną z tablicą prawdy

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}.$$

Podaj tablicę prawdy funkcji $F : Z_2^n \rightarrow \mathbb{R}$ będącej transformatą Walsh-Hadamarda funkcji f .

Rozwiązanie: z tablicy prawdy funkcji f odczytujemy, że

$$f(0 = 00) = 1,$$

$$f(1 = 01) = 1,$$

$$f(2 = 10) = 0,$$

$$f(3 = 11) = 0.$$

Zatem:

$$\begin{aligned} F(0 = 00) &= \sum_{i=0}^3 f(i_B) \cdot (-1)^{\langle i_B, 0_B \rangle} \\ &= f(0_B) \cdot (-1)^{\langle 0_B, 0_B \rangle} + f(1_B) \cdot (-1)^{\langle 1_B, 0_B \rangle} + \\ &\quad f(2_B) \cdot (-1)^{\langle 2_B, 0_B \rangle} + f(3_B) \cdot (-1)^{\langle 3_B, 0_B \rangle} \\ &= 1 \cdot (-1)^0 + 1 \cdot (-1)^0 + 0 \cdot (-1)^0 + 0 \cdot (-1)^0 \\ &= 2, \end{aligned}$$

$$F(1 = 01) = \sum_{i=0}^3 f(i_B) \cdot (-1)^{\langle i_B, 1_B \rangle}$$

$$\begin{aligned}
&= f(0_B) \cdot (-1)^{\langle 0_B, 1_B \rangle} + f(1_B) \cdot (-1)^{\langle 1_B, 1_B \rangle} + \\
&\quad f(2_B) \cdot (-1)^{\langle 2_B, 1_B \rangle} + f(3_B) \cdot (-1)^{\langle 3_B, 1_B \rangle} \\
&= 1 \cdot (-1)^0 + 1 \cdot (-1)^1 + 0 \cdot (-1)^0 + 0 \cdot (-1)^1 \\
&= 0,
\end{aligned}$$

$$\begin{aligned}
F(2 = 10) &= \sum_{i=0}^3 f(i_B) \cdot (-1)^{\langle i_B, 2_B \rangle} \\
&= f(0_B) \cdot (-1)^{\langle 0_B, 2_B \rangle} + f(1_B) \cdot (-1)^{\langle 1_B, 2_B \rangle} + \\
&\quad f(2_B) \cdot (-1)^{\langle 2_B, 2_B \rangle} + f(3_B) \cdot (-1)^{\langle 3_B, 2_B \rangle} \\
&= 1 \cdot (-1)^0 + 1 \cdot (-1)^0 + 0 \cdot (-1)^1 + 0 \cdot (-1)^1 \\
&= 2,
\end{aligned}$$

$$\begin{aligned}
F(3 = 11) &= \sum_{i=0}^3 f(i_B) \cdot (-1)^{\langle i_B, 3_B \rangle} \\
&= f(0_B) \cdot (-1)^{\langle 0_B, 3_B \rangle} + f(1_B) \cdot (-1)^{\langle 1_B, 3_B \rangle} +
\end{aligned}$$

$$\begin{aligned}
& f(2_B) \cdot (-1)^{\langle 2_B, 3_B \rangle} + f(3_B) \cdot (-1)^{\langle 3_B, 3_B \rangle} \\
= & 1 \cdot (-1)^0 + 1 \cdot (-1)^1 + 0 \cdot (-1)^1 + 0 \cdot (-1)^0 \\
= & 0.
\end{aligned}$$

Na tej podstawie tablica prawdy funkcji F ma postać:

$$Y = \begin{bmatrix} 2 & 0 & 2 & 0 \end{bmatrix}.$$



Przypomnienie: niech $F : Z_2^n \rightarrow \mathbb{R}$ będzie funkcją transformaty Wlasha-Hadamarda pewnej funkcji boolowskiej $f : Z_2^n \rightarrow Z_2$, określonej tablicą prawdy $Y = [y_0 \ y_1 \ \dots \ y_{2^n-1}]$. Wtedy odwrotną transformatą Walsha-Hadamarda funkcji F nazywamy funkcję $F^{-1} : Z_2^n \rightarrow Z_2$ równoważną funkcji f i reprezentowaną przez tablicę prawdy $Z = [z_0 \ z_1 \ \dots \ z_{2^n-1}]$ taką, że:

$$z_j = \frac{1}{2^n} \cdot \sum_{i=0}^{2^n-1} F(i_B) \cdot (-1)^{\langle i_B, j_B \rangle},$$

dla $j \in \{0, 1, \dots, 2^n-1\}$ (oczywiście $F^{-1} \circ F$ jest przekształceniem identycznościowym).

Zadanie 3:

niech $F : Z_2^2 \rightarrow \mathbb{R}$ będzie funkcją transformaty Walsh-Hadamarda funkcji boolowskiej $f : Z_2^n \rightarrow Z_2$, zgodną z tablicą prawdy

$$Y = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}.$$

Podaj tablicę prawdy funkcji f będącej odwrotną transformatą Walsh-Hadamarda funkcji F .

Rozwiązanie: z tablicy prawdy funkcji f odczytujemy, że

$$\begin{aligned} F(0 = 00) &= 1, \\ F(1 = 01) &= 1, \\ F(2 = 10) &= -1, \\ F(3 = 11) &= -1. \end{aligned}$$

Zatem:

$$\begin{aligned} F^{-1}(0 = 00) &= \frac{1}{4} \cdot \sum_{i=0}^3 F(i_B) \cdot (-1)^{\langle i_B, 0_B \rangle} \\ &= \frac{1}{4} \cdot \left(F(0_B) \cdot (-1)^{\langle 0_B, 0_B \rangle} + F(1_B) \cdot (-1)^{\langle 1_B, 0_B \rangle} + \right. \\ &\quad \left. F(2_B) \cdot (-1)^{\langle 2_B, 0_B \rangle} + F(3_B) \cdot (-1)^{\langle 3_B, 0_B \rangle} \right) \\ &= \frac{1}{4} \cdot \left(1 \cdot (-1)^0 + 1 \cdot (-1)^0 + (-1) \cdot (-1)^0 + (-1) \cdot (-1)^0 \right) \\ &= 0, \end{aligned}$$

$$\begin{aligned}
F^{-1}(1 = 01) &= \frac{1}{4} \cdot \sum_{i=0}^3 F(i_B) \cdot (-1)^{\langle i_B, 1_B \rangle} \\
&= \frac{1}{4} \cdot \left(F(0_B) \cdot (-1)^{\langle 0_B, 1_B \rangle} + F(1_B) \cdot (-1)^{\langle 1_B, 1_B \rangle} + \right. \\
&\quad \left. F(2_B) \cdot (-1)^{\langle 2_B, 1_B \rangle} + F(3_B) \cdot (-1)^{\langle 3_B, 1_B \rangle} \right) \\
&= \frac{1}{4} \cdot \left(1 \cdot (-1)^0 + 1 \cdot (-1)^1 + (-1) \cdot (-1)^0 + (-1) \cdot (-1)^1 \right) \\
&= 0,
\end{aligned}$$

$$\begin{aligned}
F^{-1}(2 = 10) &= \frac{1}{4} \cdot \sum_{i=0}^3 F(i_B) \cdot (-1)^{\langle i_B, 2_B \rangle} \\
&= \frac{1}{4} \cdot \left(F(0_B) \cdot (-1)^{\langle 0_B, 2_B \rangle} + F(1_B) \cdot (-1)^{\langle 1_B, 2_B \rangle} + \right. \\
&\quad \left. F(2_B) \cdot (-1)^{\langle 2_B, 2_B \rangle} + F(3_B) \cdot (-1)^{\langle 3_B, 2_B \rangle} \right) \\
&= \frac{1}{4} \cdot \left(1 \cdot (-1)^0 + 1 \cdot (-1)^0 + (-1) \cdot (-1)^1 + (-1) \cdot (-1)^1 \right) \\
&= 1,
\end{aligned}$$

$$\begin{aligned}
F^{-1}(3 = 11) &= \frac{1}{4} \cdot \sum_{i=0}^3 F(i_B) \cdot (-1)^{\langle i_B, 3_B \rangle} \\
&= \frac{1}{4} \cdot \left(F(0_B) \cdot (-1)^{\langle 0_B, 3_B \rangle} + F(1_B) \cdot (-1)^{\langle 1_B, 3_B \rangle} + \right. \\
&\quad \left. F(2_B) \cdot (-1)^{\langle 2_B, 3_B \rangle} + F(3_B) \cdot (-1)^{\langle 3_B, 3_B \rangle} \right) \\
&= \frac{1}{4} \cdot \left(1 \cdot (-1)^0 + 1 \cdot (-1)^1 + (-1) \cdot (-1)^1 + (-1) \cdot (-1)^0 \right) \\
&= 0.
\end{aligned}$$

Na tej podstawie tablica prawdy funkcji F ma postać:

$$X = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}.$$



Przypomnienie: niech INF będzie ciągiem znaków nad pewnym alfabetem Σ , $l_1 > 0$ i $l_2 > 0$ ustalonymi liczbami naturalnymi, wtedy algorytm kompresji metodą LZ77 ma następującą postać:

```

char [] COMPRESS_LZ77(char INF[],int l1,int l2)

{

    char tmp1 []=∅,tmp2 []=∅,CINF []=∅;

    CINF:=pobierz pierwszy znak z ciągu INF;

    INF:=dodaj do ciągu INF prefiks składający się z l1 powtórzeń pierwszego znaku

        ciągu INF;

    while |INF| > l1 do

    {

        tmp1:=pobierz l1+l2 pierwszych znaków z ciągu INF;

        tmp2:=pobierz z tmp1 najdłuższy podciąg co najwyżej l2 znakowy,

```

zaczynający się na co najwyżej l_1-1 pozycji i będący jednocześnie

prefiksem podciągu zaczynającego się od pozycji l_1 ;

CINF:=dopisz trójkę znaków (p,q,s);

/* p - pozycja podciągu tmp2 w ciągu tmp1,

q - długość podciągu tmp2,

s - znak ciągu tmp1 znajdujący się na pozycji l_1+q */

INF:=usuń q+1 pierwszych znaków z ciągu INF;

}

return CINF;

}

Zadanie 4:

Niech $INF = \text{bbbaaccabbac}$, $l_1 = 5$ oraz $l_2 = 4$. Podaj rezultat procedury $\text{COMPRESS_LZ77}(INF, l_1, l_2)$.

Rozwiązanie:

INF	tmp1	tmp1	CINF
bbbaaccabbac	\emptyset	\emptyset	\emptyset
-	-	-	b
bbbbbbbaaccabbac	-	-	-
-	bbbbbb bbba	-	-
-	-	bbb	-
-	-	-	b23a
bbbbaaccabbac	-	-	-
-	bbbba acca	-	-
-	-	a	-
-	-	-	b23a41c
bbaaccabbac	-	-	-
-	bbac cabb	-	-
-	-	c	-

-	-	-	b23a41c41a
aaccabbac	-	-	-
-	aacca bbac	-	-
-	-	∅	-
-	-	-	b23a41c41a?0b
accabbac	-	-	-
-	accab bac	-	-
-	-	b	-
-	-	-	b23a41c41a?0b41a
cabbac	-	-	-
-	cabba c	-	-
-	-	c	b23a41c41a?0b41a01ε
bbac	-	-	-

Odpowiedź to: b23a41c41a?0b41a01ε.



Przypomnienie: niech CINF będzie ciągiem znaków nad pewnym alfabetem Σ , $l_1 > 0$ ustaloną liczbą naturalną, wtedy algorytm dekompresji metodą LZ77 ma następującą postać:

```

char [] UNCOMPRESS_LZ77(char CINF[],int l1)

{

    char tmp[]=∅,INF[]=∅;

    tmp:=dodaj do tmp prefiks składający się z l1 powtórzeń pierwszego znaku

        ciągu INF;

    CINF:=usuń pierwszy znak z ciągu CINF;

    while |CINF| > 0 do

    {

        (p,q,s):=pobierz trzy pierwsze znaki z ciągu CINF;

        CINF:=usuń trzy pierwsze znaki z ciągu INF;
    }

```

tmp:=dopisz q znaków z ciągu tmp poczynając od pozycji p-tej;

tmp:=dopisz znak s;

INF:=dopisz q+1 ostatnich znaków z ciągu tmp;

tmp:=usuń q+1 pierwszych znaków z ciągu tmp;

}

return INF;

}

Zadanie 5:

Niech $CINF = b23a41c41a?0b41a01\epsilon$, $l_1 = 5$. Podaj rezultat procedury $UNCOMPRESS_LZ77(CINF, l_1, l_2)$.

Rozwiązanie:

CINF	tmp	(p,q,s)	INF
b23a41c41a?0b41a01ε	∅	∅	∅
-	bbbbbb	-	-
23a41c41a?0b41a01ε	-	-	-
-	-	23a	-
41c41a?0b41a01ε	-	-	-
-	bbbbbbbbb	-	-
-	bbbbbbbbbba	-	-
-	-	-	bbba
-	bbba	-	-
-	-	41c	-
41a?0b41a01ε	-	-	-
-	bbbbaa	-	-
-	bbbbaac	-	-

-	-	-	bbbaac
-	bbaac	-	-
-	-	41a	-
?0b41a01ε	-	-	-
-	bbaacc	-	-
-	bbaacca	-	-
-	-	-	bbbaacca
-	aacca	-	-
-	-	?0b	-
41a01ε	-	-	-
-	aacca	-	-
-	aaccab	-	-
-	-	-	bbbaaccab
-	accab	-	-
-	-	41a	-
01ε	-	-	-
-	accabb	-	-

-	accabba	-	-
-	-	-	bbbaaccabba
-	cabba	-	-
-	-	01 ϵ	-
\emptyset	-	-	-
-	cabbac	-	-
-	cabbac ϵ	-	-
-	-	-	bbbaaccabbac ϵ

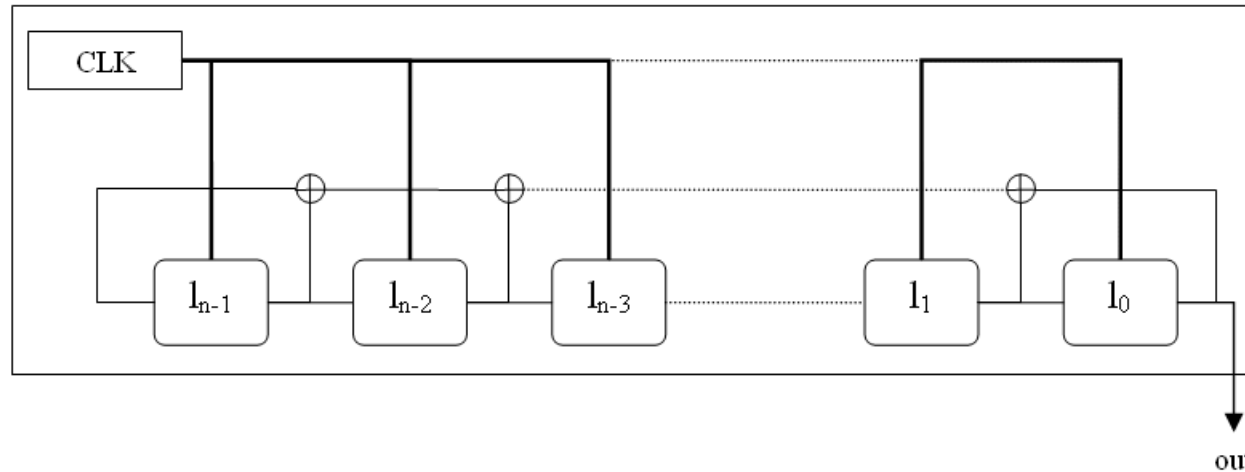
Odpowiedź to: bbbaaccabbac ϵ =bbbaaccabbac.



Przypomnienie: rejestr LFSR (ang. *linear feedback shift register*) długości n to układ o następującej charakterystyce:

- LFSR zbudowany jest z n szeregowo połączonych komórek bitowych l_i (indeksowanych od 0 do $n - 1$),
- wszystkie komórki są zsynchronizowane zegarem CLK układu LFSR,
- każda komórka l_i posiada niezależne wejście i wyjście bitowe oraz wejście sygnału zegara CLK,
- $0 \leq m \leq n$ wybranych komórek rejestru LFSR jest sprzężone zwrotnie (szeregowo z użyciem operacji XOR w kierunku przeciwnym połączeń komórek rejestru LFSR) z wejściem $n - 1$ komórki układu,

- wyjście komórki o indeksie 0 odpowiada wyjściu rejestru LFSR,
- początkowe wartości bitowe komórek są inicjowane n -bitowym kluczem $K = [k_{n-1} \ k_{n-2} \ \dots \ k_0]$ (nazywanym także ziarnem od ang. *seed*).



Rysunek 1. Przykład n -bitowego rejestru LFSR dla $m = n$.

Zadanie 6:

Podaj osiem pierwszych bitów wyjścia dla 4-bitowego rejestru LFSR o parametrze $m = 4$, inicjowanego kluczem $K = 13$.

Rozwiązanie: liczba dziesiętna 13 odpowiada czterobitowemu ciągowi binarnemu postaci 1101, czyli $K = \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix}$.

czas	l_3	l_2	l_1	l_0	sprzężenie	out
0	1	1	0	1	-	-
1	1	1	1	0	1	1
2	1	1	1	1	1	0
3	0	1	1	1	0	1
4	1	0	1	1	1	1
5	1	1	0	1	1	1
6	1	1	1	0	1	1
7	1	1	1	1	1	0
8	0	1	1	1	0	1



Zadanie 7:

Podaj osiem pierwszych bitów wyjścia dla 4-bitowego rejestru LFSR o parametrze $m = 2$ (l_2 i l_0 są sprzężone), inicjowanego kluczem $K = 9$.

Rozwiązanie: liczba dziesiętna 9 odpowiada czterobitowemu ciągowi binarnemu postaci 1011, czyli $K = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$.

czas	l_3	l_2	l_1	l_0	sprzężenie	out
0	1	0	0	1	-	-
1	1	1	0	0	1	1
2	1	1	1	0	1	0
3	1	1	1	1	1	0
4	0	1	1	1	0	1
5	0	0	1	1	0	1
6	1	0	0	1	1	1
7	1	1	0	0	1	1
8	1	1	1	0	1	0



Zadanie 8:

Zaszyfruj używając szyfru Vernama napis "PJWSTK" zapisany binarnie względem ośmiobitowych wartości znaków w standardzie ASCII przy założeniu, że generator klucza szyfru jest układem LFSR przedstawionym w poprzednim zadaniu (tj. zadanie 7). Pamiętaj, że:

ASCII(P) = 80,

ASCII(J) = 74,

ASCII(W) = 87,

ASCII(S) = 83,

ASCII(T) = 84,

ASCII(K) = 75.

Rozwiązanie: ustalamy binarne (ośmiobitowe) reprezentacje znaków napisu "PJWSTK":

- $\text{ASCII}(\text{P}) = 80 = 01010000,$
- $\text{ASCII}(\text{J}) = 74 = 01001010,$
- $\text{ASCII}(\text{W}) = 87 = 01010111,$
- $\text{ASCII}(\text{S}) = 83 = 01010011,$
- $\text{ASCII}(\text{T}) = 84 = 01010100,$

- $\text{ASCII}(K) = 75 = 01001011.$

Zatem ciąg binarny jaki mamy zamiar szyfrować to:

01010000|01001010|01010111|01010011|01010100|01001011.

Generator klucza szyfru jest układem okresowym o okresie rozmiaru 6 postaci 100111. Na tej podstawie dokonujemy szyfrowania metodą Vernama:

$$\begin{array}{l} 01010000|01001010|01010111|01010011|01010100|01001011 \oplus \\ 10011110|01111001|11100111|10011110|01111001|11100111 = \\ 11001110|00110010|10110000|11001101|00101010|10101100. \end{array}$$

Teraz ciąg wynikowy

11001110|00110010|10110000|11001101|00101010|10101100

przekształcamy do postaci znaków w standardzie ASCII, otrzymujemy:

ASCII(11001110) = ASCII(206),

ASCII(00110010) = ASCII(050),

ASCII(10110000) = ASCII(176),

ASCII(11001101) = ASCII(205),

ASCII(00101010) = ASCII(042),

ASCII(10101100) = ASCII(172).

