Szyfr Vigenere

2 października 2014

Ten szyfr jest modyfikacją szyfru Cezara. Kluczem jest n-literowe słowo. Tekst zaszyfrowany otrzymujemy poprzez dodanie do wiadomości klucza. Przykład:

tekst: bardzosiecieszezewreszciezaczalsierokakademicki
klucz: klamstwo
wiadomość zaszyfrowana: llrprhowoniqksanohrqksywokaorthgspractgonpmuude
pierwszy blok: b+k = (1+10)%26 = 11 = l, a+l = (0 + 11)%26 = 11 = l,r+a = (17+0)%26=17=r, d+m = (3+12)%26 =15= p, z+s =(25+18)%26 =17=r,o+t= (14+19)%26 = 7 = h, s+w = (18+22)%26=14=o, i+o = (8+14)%26=22=w deszyfrowanie l-k = (11-10)%26 = (11-10+26)%26 =(11+16)%26=1=b itp.

Część 1 zadania: napisać program szyfrujący, tekst do szyfrowania ma pochodzić z pliku, klucz ustawić można w programie.

Kryptoanaliza szyfru Vignere'a

Pierwszym krokiem do wydobycia z zaszyfrowanego tekstu klucza jest ustalenie jego długości. W tym celu wykorzystać można indeks zgodności $I_c(x)$, który określa prawdopodobieństwo tego, że dwa wyrazy w ciągu x są identyczne. Przyjmijmy, że częstość wystąpienia w ciągu x o długości n liter a,b,...,z wynosi odpowiednio $f_0, f_1, ..., f_{25}$, wtedy

$$I_c(x) = \frac{\sum_{i=0}^{25} f_i * (f_i - 1)}{n * (n-1)} \tag{1}$$

dla języka angielskiego $I_c(x^{ang}) \approx 0,065$. Domyślamy się, że klucz ma długość d, dzielimy więc tekst na bloki $x_0x_dx_{2d}...$, $x_1x_{d+1}x_{2d+1}...$ i dla każdego z bloków obliczamy I_c (dla całkowicie losowego ciągu $I_c \approx 0,038$). Jeżeli dla każdego z bloków I_c ma wartość zbliżoną do 0,065 to mamy podstawy sądzić, że klucz ma długość d. Jeżeli nie, należy wykonać obliczenie dla innej długości d' tak długo aż uzyskamy zadowalającą zgodności.

Przykład
klucz: can
tekst: itwasagoodwar...
wiadomość zaszyfrowana: ktjcsniobfwnt...
Zakładana długość klucza 2, mamy dwa bloki:
1: k,j,s,i,b,w,t ...
2: t,c,n,o,f,n ...
Dla każdej z grup obliczamy I_c: 1 - $I_c \approx 0.042$, 2 - $I_c \approx 0.048$. Wartości wskazują na to, że ten podział nie jest prawidłowy. Następnym krokiem jest zmiana długości klucza na 3. Wtedy mamy 3 grupy (w nawiasach podane wartości I_c dla tego podziału):
1: k,c,i,f,t ... ($I_c \approx 0.061$)
2: t,s,o,w, ... ($I_c \approx 0.069$)
3: j,n,b,n, ... ($I_c \approx 0.069$)
Wartości wskazują na to, że jest to prawidłowa długości klucza (można sprawdzić, że dla długości 4 wartość I_c dla każdej grupy spadnie).

Zakładając, że ustaliliśmy długość klucza i wynosi ona d możemy przystąpić do wyznaczenia klucza. W tym celu ponownie korzystamy z bloków postaci $x_0x_dx_{2d}...$, $x_1x_{d+1}x_{2d+1}...$. Załóżmy, że częstość występowania liter a,b,...,z w tych ciągach wynosi odpowiednio $f_0, f_1, ..., f_{25}$ oraz $f'_0, f'_1, ..., f'_{25}$. Wzajemny indeks zgodności definiujemy jako

$$MI_c(x, x') = \frac{\sum_{i=0}^{25} f_i * f'_i}{n * n'} \tag{2}$$

gdzie n,n' to długości odpowiednich ciągów. Wiemy, że klucz składa się w d liter $k = (k_1, k_2, ..., k_d)$. Wzajemny indeks zgodności pozwala określić relację pomiędzy poszczególnymi k_i. Przyjmijmy, że rozważamy elementy zaszyfrowane przez k_1 i k_2. Jeżeli

obliczając MI_c uzyskamy wartość $MI_c(x, x') \approx 0,065$ oznacza to, że $k_1 = k_2$, jeżeli nie to przesuwamy wyrazy w ciągu x' (a występowało f'_0 razy,po przesunięciu tyle razy występować będzie b a a będzie występować teraz f'_{25} razy) i znowu obliczamy MI_c (formalnie $MI_c(x, x') = \frac{\sum_{i=0}^{25} f_i(f'_{(i-g)})}{n*(n')}$). Ostatecznie, po znalezieniu odpowiedniej wartości g będziemy wiedzieli, że $k_1 - k_2 = g$. Ponieważ postępowanie to opiera się na statystyce tekstu w języku angielskim należy je stosować dla: 1) długich tekstów 2) kluczy krótkich w stosunku do długości tekstu. Niech d=5, wtedy $k = (k_1, k_2, k_3, k_4, k_5)$. Praktycznie jest wyznaczyć indeksy zgodności dla wszystkich par $(k_1, k_2), ..., (k_1, k_5), (k_2, k_3), ..., (k_2, k_5)$ bo w zależności od specyficznego tekstu nie musi być tak, że tylko pary $(k_1, k_2), ..., (k_1, k_5)$ dadzą nam rozwiązanie. Ostatecznie uwzględniając relacje pomiędzy znakami klucza powinniśmy moc napisać, że klucz jest postaci k= $(k_1, k_1 + y_2, k_1 + y_3, k_1 + y_4, k_1 + y_5)$. Następnym krokiem może być podstawienie za k_1 wszystkich możliwych liter i sprawdzenie, użycie którego klucza daje sensowną wiadomość
Przykład

Kontynuując poprzedni przykład (tekst angielski, podział na 3 grupy) mamy następujące częstości występowań liter a...z w grupie 1 oraz 2:
1: $[f^1_0, ..., f^1_{25}]$ = [5, 0, 33, 7, 9, 13, 30, 4, 5, 23, 19, 1, 2, 14, 9, 21, 18, 7, 1, 13, 17, 26, 5, 1, 9, 2]
2: $[f^2_0, ..., f^2_{25}]$ = [27, 1, 4, 15, 39, 5, 5, 22, 21, 0, 1, 9, 7, 15, 20, 8, 0, 16, 26, 29, 6, 2, 9, 1, 5, 0]

Na początku liczymy $\frac{1}{n^1*n^2}(f^1_0 * f^2_0 + ...)$ co daje $MI_c \approx 0.029$, następnie $\frac{1}{n^1*n^2}(f^1_0 * f^2_{25} + f^1_1 * f^2_0...) \approx 0.039$ aż dochodzimy do $\frac{1}{n^1*n^2}(f^1_0 * f^2_{24} + f^1_1 * f^2_{25}...) \approx 0.066$.
Część 2 zadania: napisać program, który wyznacza długość klucza (tekst w języku angielskim) i podaje względne relacje pomiędzy wyrazami klucza (nie trzeba zapisywać klucza w postaci, k= $(k_1, k_1 + y_2, k_1 + y_3, k_1 + y_4, k_1 + y_5)$ wystarczy wypisać pozytywne wyniki analizy wzajemnego indeksu zgodności na ekran np. w postaci k_1, k_2, g itp.)

Punktacja (łącznie 10 punktów):

- 3 punkty - parawidłowo działająca pierwsza część zadania, czyli wczytanie tekstu z pliku i zaszyfrowanie go oraz wpisanie zaszyfrowanego tekstu do pliku. Nazwy plików można ustalić w programie, nie muszą być podawane przez użytkownika.
- 3 punkty - wczytanie zaszyfrowanego tekstu z pliku i obliczenie indeksu zgodności dla zadanego przez użytkownika podziału na grupy, ewentualnie obliczanie indeksu zgodności w pętli dla podanego przez użytkownika górnego zakresu sprawdzania (maksymalnie 10 grup).
- 4 punkty - dla zadanej przez użytkownika ilości grup obliczenie wzajemnego indeksu zgodności między wszystkimi grupami.