

**Akademia Techniczno-Humanistyczna**

**w Bielsku-Białej**

Wydział Budowy Maszyn i Informatyki

**Bezpieczeństwo technologii informatycznych**

**(ćwiczenia laboratoryjne)**

**Ćwiczenie numer:**

**3**

**Czas realizacji zajęć:** 180 min.

**Temat ćwiczenia:**

**Szyfr afiniczny**

**1. Wstęp teoretyczny.**

**Szyfr afiniczny**

**Szyfr afiniczny** – [szyfr](https://pl.wikipedia.org/wiki/Szyfr) należący do grupy [monoalfabetycznych](https://pl.wikipedia.org/wiki/Szyfr_monoalfabetyczny) [szyfrów podstawieniowych](https://pl.wikipedia.org/wiki/Szyfr_podstawieniowy).

Rodzina szyfrów monoalfabetycznych posiada jedną bardzo ważną cechę, a mianowicie jednej literze alfabetu jawnego odpowiada dokładnie jedna litera alfabetu tajnego. Funkcja szyfrująca wygląda następująco:

f(x) = ax+ b\ \mod\ m, gdzie*x* to szyfrowana litera, (a,b) jest kluczem, a *m* to liczba liter w alfabecie (zwykle korzystamy z m=26 bo tyle liter ma język angielski)

Łatwo zauważyć, że jeśli a = 1, to mamy do czynienia ze zwykłym przesunięciem.

Szyfr afiniczny ma sens tylko wtedy, gdy funkcja afiniczna *f* jest [różnowartościowa](https://pl.wikipedia.org/wiki/Funkcja_r%C3%B3%C5%BCnowarto%C5%9Bciowa) tzn. gdy dla dowolnego y należącego do [zbioru klas reszt](https://pl.wikipedia.org/wiki/Arytmetyka_modularna) {\mathbb Z}_m równanie

ax+b\equiv y \mod\ m

ma co najwyżej jedno rozwiązanie ze względu na zmienną *x*. Zapiszmy nasze równanie w sposób następujący:

ax\equiv y -b \mod\ m.

Zauważmy, że gdy wartości *y* przebiegają cały zbiór {\mathbb Z}_m, to i wartości y-b się wyczerpują, czyli wystarczy jeśli zbadamy rozwiązywalność równań

ax \equiv y \mod\ m

dla y\in {\mathbb Z}_m. Równanie to ma dokładnie jedno rozwiązanie dla każdego y\in {\mathbb Z}_m wtedy i tylko wtedy, gdy {\rm NWD}(a,m)=1 (gdzie NWD oznacza [największy wspólny dzielnik](https://pl.wikipedia.org/wiki/Najwi%C4%99kszy_wsp%C3%B3lny_dzielnik) dwóch liczb).

Na przykład, gdy m=26=2\cdot 13 to wartości *a* należące do {\mathbb Z}_{26} dla których {\rm NWD}(a,26) =1 są następujące:

1 , 3 , 5 , 7 , 9 , 11 , 15 , 17 , 19 , 21 , 23 , 25.

Parametr *b* może być dowolny toteż mamy 12\cdot 26=312 możliwych kluczy. Jest to bardzo mała liczba kluczy, która nie daje odpowiedniego bezpieczeństwa, toteż szyfr ten nie jest w zasadzie stosowany.

Funkcja deszyfrująca dla tego szyfru wygląda tak :

d(y)= a^{-1} * (y - b)  \mod m,

gdzie a^{-1} jest [odwrotnością](https://pl.wikipedia.org/wiki/Liczba_odwrotna) a w [pierścieniu](https://pl.wikipedia.org/wiki/Pier%C5%9Bcie%C5%84_(matematyka)) {\mathbb Z}_{26}.

Wzór wynika z wyliczeń:

\begin{align}
\mbox{D}(\mbox{E}(x)) &= a^{-1}(\mbox{E}(x)-b)\mod{m}\\
  &= a^{-1}(((ax+b)\mod{m})-b)\mod{m} \\
  &= a^{-1}(ax+b-b)\mod{m} \\
  &= a^{-1}ax \mod{m}\\
  &= x\mod{m}.
\end{align}

**Przykład działania**

Przyjmując, że K = (7 ,5) należy zaszyfrować i odszyfrować słowo KOT.

Dla uproszenia korzystamy z mod 26 (alfabet angielski ma 26 znaków). Funkcja szyfrująca ma postać:

e(x)= 7x + 5\ 

Zmieniamy litery wyrazu "kot" na wartości liczbowe: K=10; O=14; T=19;

Szyfrowanie :

10 * 7 + 5 \mod 26 = 75  \mod 26 = 23

14 * 7 + 5 \mod 26 = 103 \mod 26 = 25

19 * 7 + 5 \mod 26 = 138 \mod 26 = 8

Tekst zaszyfrowany odpowiada ciągowi 23, 25, 8 czyli: XZI.

Deszyfrowanie:

7^{-1} mod 26 = 15\ 

Funkcja deszyfrująca ma postać :

d(y)= 15 * (y - 5) \mod 26

15 * (23 - 5) \mod 26 = 270 \mod 26 = 10

15 * (25 - 5) \mod 26 = 300 \mod 26 = 14

15 * (8 - 5) \mod 26 = 45 \mod 26 = 19

Ciąg liczb 10, 14, 19 odpowiada naszemu wyjściowemu KOT.

**2. Plan wykonania ćwiczenia.**

1. Napisz program, który szyfruje tekst zgodnie z algorytmem szyfru afiniczny.
2. Napisz program, który odszyfrowuje tekst zgodnie z algorytmem szyfru afiniczny.
3. Napisz sprawozdanie.