Szyfr Homofoniczny Podstawieniowy z kluczem

Krzysztof Świderski (272313), Cyberbezpieczeństwo

Opis

Homophonic_substitution_cipher_v1 to program napisany w języku Python. Ściślej mówiąc, jest to implementacja szyfru homofonicznego podstawieniowego z tajnym kluczem. Program pozwala użytkownikowi na zaszyfrowanie prostych tekstowych wiadomości wspomnianym szyfrem, odszyfrowanie ich oraz wygenerowanie klucza, który jest niezbędny do poprawnego zaszyfrowania oraz odczytania tajnej wiadomości. Klucz jest zapisywany w formie pliku, w folderze gdzie znajduje się program.

Projekt składa się z głównego pliku **Homophonic_substitution_cipher_v1.py** oraz pliku uruchamiającego **run.py.**

Pojęcia:

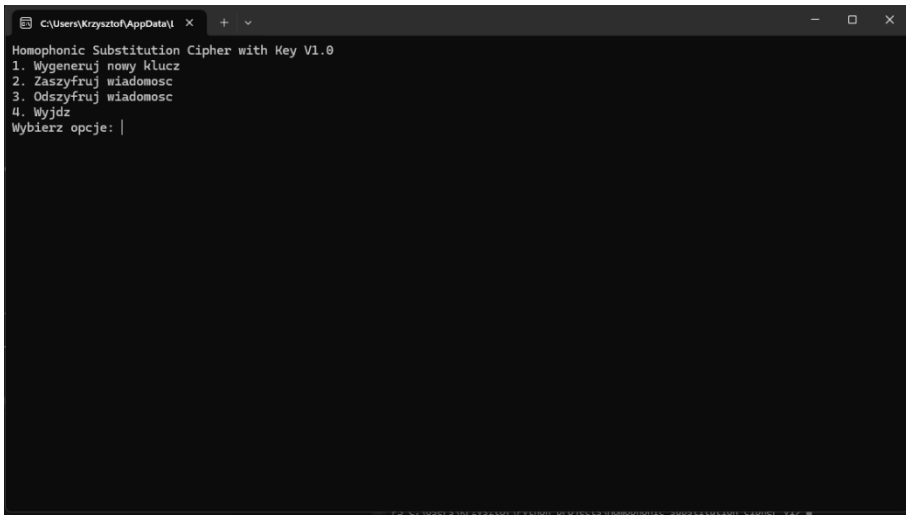
Szyfr podstawieniowy – szyfrowanie polegające na zastąpieniu każdego znaku tekstu jawnego innym znakiem. Np. A – 30, B – 46, C – 89 itd.

Szyfr homofoniczny – rodzaj szyfru podstawieniowego, w którym każdej literze tekstu jawnego odpowiada inny zbiór symboli (homofonów). Np. A = {4, 56, 96} – litera ‘A’ może być zaszyfrowana jako 4, 56 lub 96.

Tajny klucz – w kontekście tego programu kluczem jest lista wszystkich możliwych znaków i przypisane im wartości liczbowe. Na podstawie klucza szyfrowane i odszyfrowywane są wiadomości.

Wymagania:

Zainstalowany Python (preferowana wersja 3.10)



Działanie oraz instrukcja:

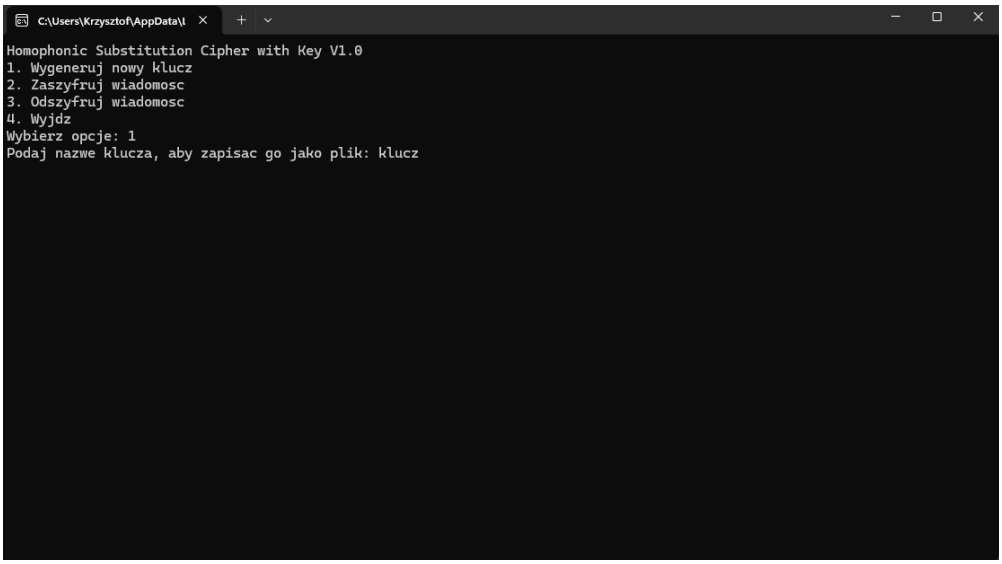
Pliki **Homophonic_substitution_cipher_v1.py** oraz **run.py** należy umieścić w wybranym przez nas folderze.

Po uruchomieniu programu **run.py**, w pierwszej kolejności wyświetla się menu główne. Użytkownik ma do wyboru:

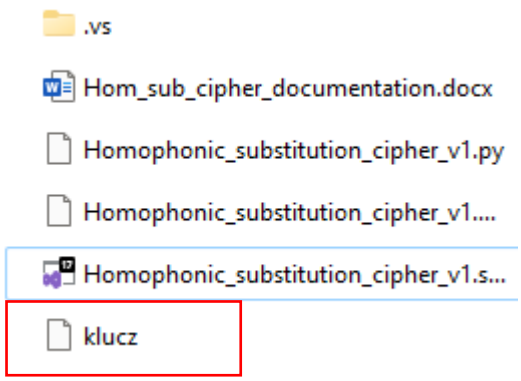
- Wygenerować unikalny klucz i zapisać go w folderze, w którym znajduje się program
- Zaszyfrować wiadomość wygenerowanym kluczem
- Odszyfrować wiadomość kluczem
- Zakończyć działanie programu

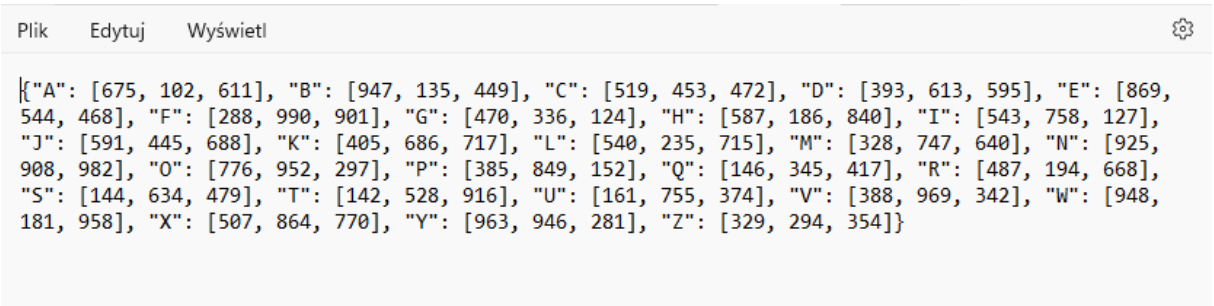
Wybór dokonywany jest poprzez wpisanie odpowiedniej cyfry i naciśnięcie ENTER.

W pierwszej kolejności należy wygenerować klucz, który jest niezbędny do zaszyfrowania i odszyfrowania wiadomości.



W tym momencie klucz został zapisany jako plik.



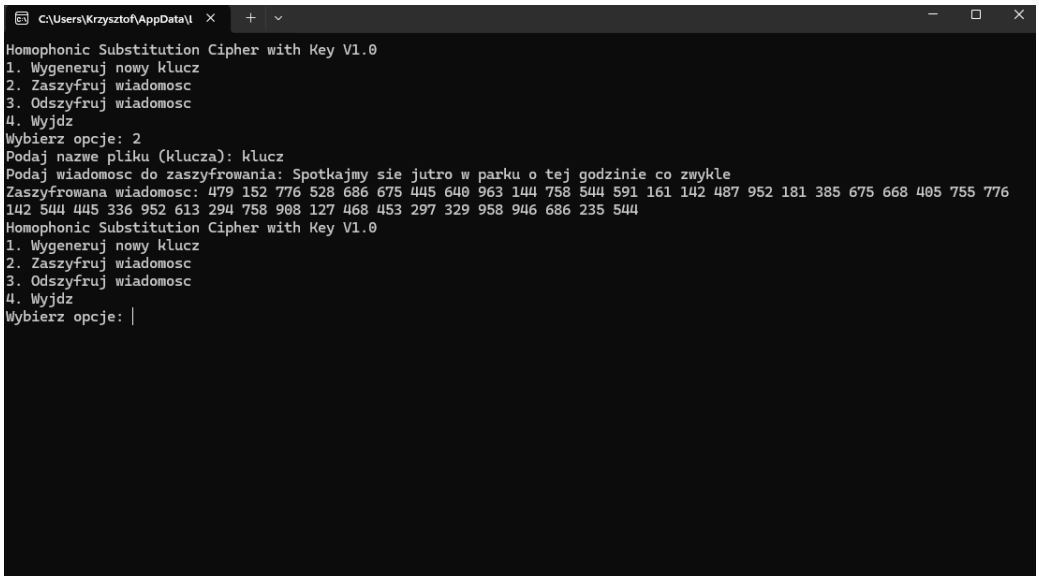


Szyfrowanie

Teraz zaszyfrujemy prostą wiadomość: Spotkajmy się jutro w parku, o tej godzinie co zwykle.

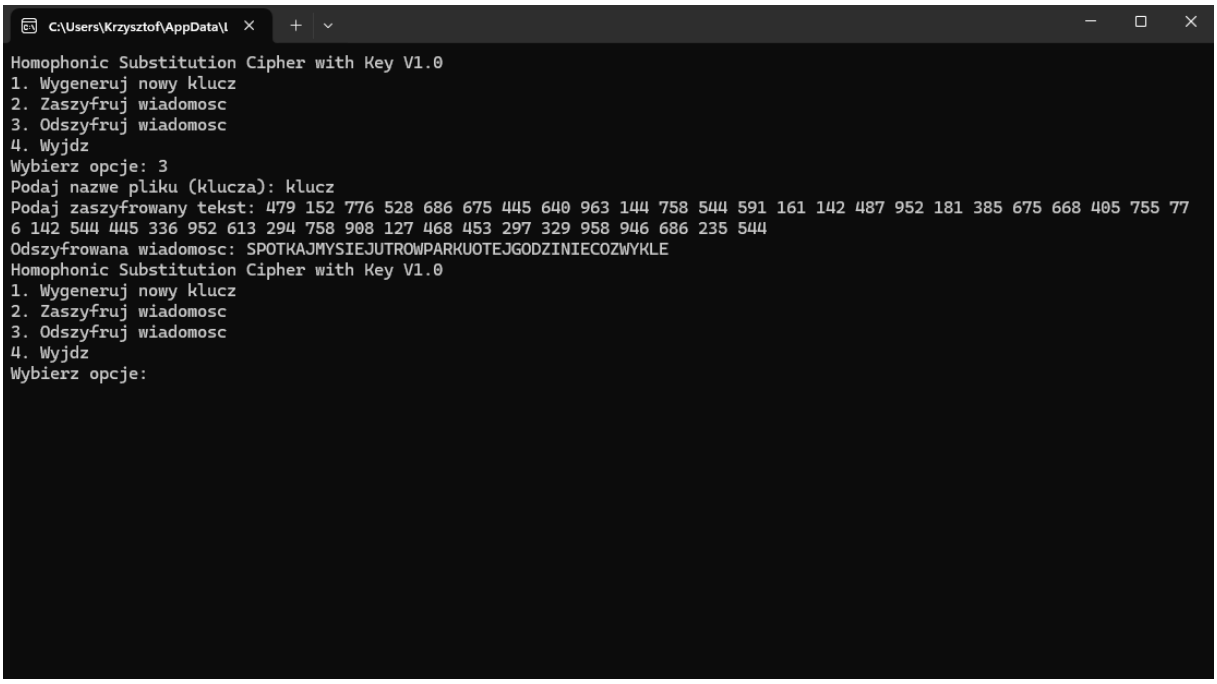
Dla poprawnego działania usuwamy najpierw polskie znaki oraz znaki interpunkcyjne: Spotkajmy sie jutro w parku o tej godzinie co zwykle

Wybieramy odpowiednią opcję i podajemy nazwę klucza.



Otrzymujemy ciąg liczb – szyfrogram. Jak można zauważyć, białe znaki są pomijane podczas szyfrowania i dzięki temu nie możemy stwierdzić gdzie kończą się słowa oraz z ilu składa się wiadomość. Każdej literze przyporządkowana jest jedna z trzech losowych liczb 3 cyfrowych.

Odszyfrowanie



Proces odszyfrowywania wygląda bardzo podobnie – musimy podać nazwę klucza oraz zaszyfrowaną wiadomość. Ważne jest żeby podać szyfrogram **z białymi znakami**, w takiej formie jak po zaszyfrowaniu. Nie należy łączyć liczb w jeden długi ciąg.

Program umożliwia wygenerowanie wielu kluczy, więc jeśli jeden zostanie skompromitowany, to informacje mogą zostać szybko zaszyfrowane drugim.