Politechnika Wrocławska

Kryptografia 2

Osoby wykonujące projekt:
Szymon Bęczkowski

Tytuł Projektu:
Opracowanie własnego algorytmu kryptograficznego

Termin zajęć projektowych:
Piątek 9:15-11:00

Data oddania sprawozdania:

Ocena:

Spis treści:

1. Kodowanie ASCII

ASCII – siedmiobitowy system kodowania znaków, używany we współczesnych komputerach oraz sieciach komputerowych, a także innych urządzeniach wyposażonych w mikroprocesor.

Znak	Dziesiętnie	Szesnastkowo	Binarnie
a	97	61	01100001
A	65	41	01000001
5	53	35	00110101

2. Opis algorytmu

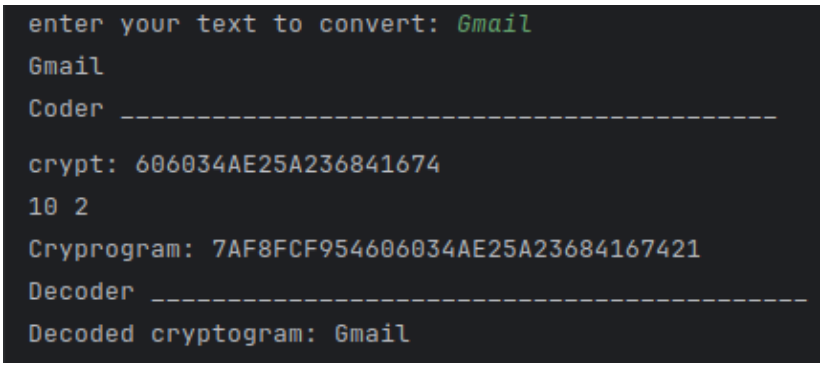
Szyfrowanie wiadomości odbywa się przy pomocy systemu ASCII. Kolejne znaki naszej wiadomości są zamieniane na odpowiednie ciągi w różnych systemach liczbowych – system dziesiętny, szesnastkowy oraz binarny. W algorytmie zastosowane są także operacje odwrócenia ciągów czy negacji losowo wybranych bitów. Do tak powstałego szyfrogramu są doklejane ciągi o losowej długości w celu przesunięcia użytecznej części szyfrogramu z samego środka ciągu. Algorytm został zrealizowany w języku Python. Program składa się z trzech plików:

Main.py – plik główny, w którym wywołuje potrzebne funkcje. Prosi użytkownika o podanie wiadomości, którą szyfruje oraz deszyfruje.

Coder.py – zawiera potrzebne funkcje umożliwiające zaszyfrowania wiadomości

Decoder.py - zawiera potrzebne funkcje umożliwiające odszyfrowania wiadomości

3. Interfejs programu – szyfrowanie i deszyfrowanie wiadomości „GMAIL”



4. Przebieg procesu szyfrowania

- Zamiana każdego znaku na postać decymalną i binarną w kodzie ASCII

Znak	dziesiętnie	binarnie
G	71	01000111
m	109	01101101
a	97	01100001
i	105	01101001
l	108	01101100

- Odwrócenie ciągów binarnych

Znak	Binarnie	Odwrócenie ciągu
G	01000111	11100010
m	01101101	10110110
a	01100001	10000110
i	01101001	10010110
l	01101100	00110110

- Losowanie dwóch różnych pozycji do negacji bitów z zakresu od 0 do 7

Ciąg odwrócony	Pozycje
11100010	6 i 0
10110110	3 i 4
10000110	2 i 5
10010110	3 i 6
00110110	1 i 6

- Negacja bitów

Ciąg odwrócony	Pozycje	Ciąg po negacji
11100010	6 i 0	01100000
10110110	3 i 4	10101110
10000110	2 i 5	10100010
10010110	3 i 6	10000100
00110110	1 i 6	01110100

- Zamiana otrzymanych ciągów na postać szesnastkową

Ciąg po negacji	Ciąg szesnastkowy
01100000	60
10101110	AE
10100010	A2
10000100	84
01110100	74

- Połączenie pozycji do negacji oraz ciągu szesnastkowego

Pozycje	Ciąg szesnastkowy	Połączenie ciągów
6 i 0	60	6060
3 i 4	AE	34AE
2 i 5	A2	25A2
3 i 6	84	3684
1 i 6	74	1674

Wynik połączenia: 606034AE25A236841674

- Dodanie losowego ciągi znaków z przodu i na końcu wiadomości

Losowane są dwie liczby z zakresu od 2 do 15. Liczby te oznaczają długość dodawanych ciągów do szyfrogramu. Następnie są one zamieniane na postać szesnastkową i dodawane na drugiej i przedostatniej pozycji. Wygenerowane ciągi też są generowane z liczb systemu szesnastkowego.

Wylosowane liczby:

10 (dec) = A (hex), Ciąg wygenerowany: 7AF8FCF954

2 (dec) = 2 (hex), Ciąg wygenerowany: 21

Wynik szyfrowania: 7AF8FCF954606034AE25A2368416742 1

5. Przebieg procesu deszyfrowania

- Usunięcie dodatkowych znaków

  To ile należy usunąć znaków z początku i końca szyfrogramu mówi nam druga i przedostatnia pozycja w szyfrogramie:

  7AF8FCF954606034AE25A2368416742 1

  A oznacza 10 od początku szyfrogramu
  2 oznacza 2 znaki od końca szyfrogramu

  W wyniku otrzymamy ciąg: 606034AE25A236841674

- Podzielenie ciągu w grupy po cztery znaki

  6060 34AE 25A2 3684 1674

- Zamiana dwóch ostatnich znaków z każdej grupy na binarne:

Ciąg szesnastkowy	Ciąg binarny
60	01100000
AE	10101110
A2	10100010
84	10000100
74	01110100

- Negacja bitów na podanych pozycjach

Ciąg szesnastkowy	Ciąg binarny	Pozycje do negacji	Ciąg wynikowy
60	01100000	6 i 0	11100010
AE	10101110	3 i 4	10110110
A2	10100010	2 i 5	10000110
84	10000100	3 i 6	10010110
74	01110100	1 i 6	00110110

- Odwrócenie ciągu

Ciąg po negacji	Odwrócenie ciągu
11100010	01000111
10110110	01101101
10000110	01100001
10010110	01101001
00110110	01101100

- Zamiana na kod ASCII

Odwrócony ciąg	Decymalny	Znak ASCII
00100111	71	G
01101101	109	m
01100001	97	a
01101001	105	i
01101100	108	l

W wyniku deszyfrowania otrzymaliśmy wiadomość, którą chcieliśmy zaszyfrować: Gmail

Uwaga: W przypadku polskich znaków takich jak np. ą, ę lub Ż w procesie szyfrowania zamieniamy ją na postać dziesiętną poprzedzoną znakiem X np. dla Ą otrzymamy blok X260. Cały proces jest pomijany.

```
Enter text to encrypt: żółty123$
Coder:   ----------------------------------------
Length of strings, front/back: 2/14
Cryptogram: 12X38056C9X322762D7497739D212C24E472058284944AF816E2
Decoder: ----------------------------------------
Decoded cryptogram: żółty123$
Algorithm work well

Process finished with exit code 0
```