|  |
| --- |
| Politechnika Wrocławska, Informatyka Stosowana |
| Kryptografia Historyczna |
| Cyberbezpieczeństwo, Laboratorium nr.2 - raport |

|  |
| --- |
| Autor: Aleksander Stepaniuk  Nr. Indeksu: 272644 |

**Zad 1. Algorytmy historyczne – właściwości.**

Teksty których użyłem do analizy kolejnych algorytmów:

**Tekst 1:**

„Jeszcze gdy chodzilem do podstawowki to byl tam taki Pawel i ja jechalem na rowerze i go spotkalem i potem jeszcze pojechalem do biedronki na lody i po drodze do domu wtedy jeszcze juz do domu pojechałem”

**Tekst 2:**

„Cesarz czesal wlosy cesarzowej cesarzowa czesala wlosy cesarza Dzdzystym rankiem gzegzolki i piegze zamiast wziac sie za dzdzownice nazarly sie na czczo miazszu rzezuchy i rzedem rzygaly do rozzarzonej brytfanny Idzie Sasza sucha szosa suszy sobie swoje szorty Gdzie jest kufel pyta brat Moze kufel w kufer wpadł Bracie zawsze ci tlumacze kufel wpadl do kufra raczej Wyjmij z kufra kufel bracie lepiej postaw go na blacie”

**Tekst 3:**

„AAAAAABBBBCCCCCCCCCCCCCCDDDDDDDDD”

Algorytmy które wybrałem do analizy:

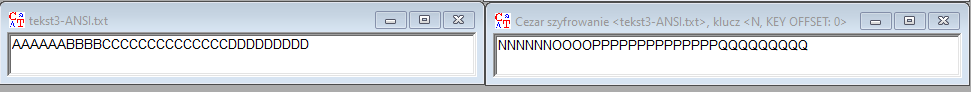
1. Cezar
2. Vigenere
3. Hill
4. Playfair
5. XOR
6. Permutacja / Transpozycja

**Cezar:**

klucz = 13/„N”

Obraz zawierający tekst, Czcionka, oprogramowanie, numer

Opis wygenerowany automatycznie

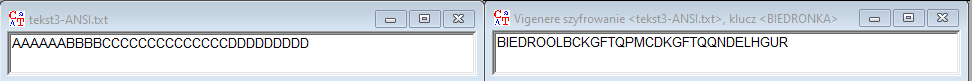


**Vigenere:**

klucz = „BIEDRONKA”

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

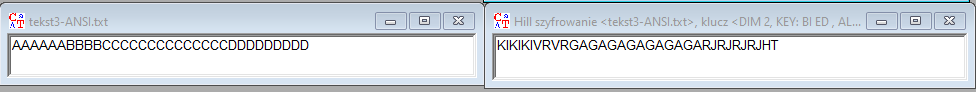


**Hill:**

klucz = „BI ED”, macierz 2x2

**Obraz zawierający tekst, Czcionka, oprogramowanie, numer

Opis wygenerowany automatycznie**

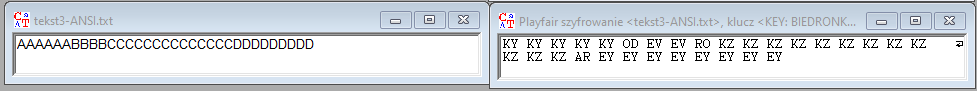
****

**Playfair:**

klucz = „BIEDRONKA”, macierz 5x5, po preformatowaniu

Obraz zawierający tekst, Czcionka, oprogramowanie, numer

Opis wygenerowany automatycznie



**XOR:**

klucz = 08 59

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, numer, linia

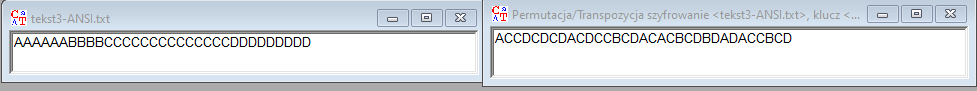
Opis wygenerowany automatycznie

**Permutacja / Transpozycja:**

klucz = 1sza permutacja: „BIEDRONKA”, 2ga perm.: „ROWER” (( 2, 5, 4, 3, 9, 8, 7, 6, 1 ); ( 3, 2, 5, 1, 4 ))

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie



Wnioski:

1. **Algorytm Cezara** nie zmienił kształtu ani długości wejściowego tekstu. W każdym tekście można zauważyć powtarzające się litery na tych samych pozycjach, co w tekście jawnym. Zmiana wartości klucza nie daje w tym przypadku żadnej sensownej zmiany. Wynika to z prostego sposobu w jaki operuje ten algorytm. Szyfrowanie te jest bardzo proste do złamania, chociażby poprzez wykorzystanie analizy częstotliwości występowania liter w tekście zaszyfrowanym i porównanie ich z częstotliwością występowania w alfabecie.
2. **Algorytm Vigenera** nie zmienił kształtu ani długości wejściowego tekstu. Jednak w odróżnieniu od algorytmu Cezara nie jest tak łatwo znaleźć podobieństwa pomiędzy tekstem jawnym, a zaszyfrowanym ponieważ dwa takie same słowa mogą wyglądać inaczej w innym fragmencie tekstu. Dużą wadą tego algorytmu jest to, że może ujawnić klucz, na którym tekst został zaszyfrowany (najlepiej widać to na tekście numer 3, gdzie pierwsze kilka liter układa się w klucz, ponieważ dla litery ‘A’ algorytm dodaje 0 do kolejnych wartości klucza co skutkuje powtarzaniem klucza w tekście wyjściowym). Zmiana wartości klucza nie daje w tym przypadku żadnej sensownej zmiany.
3. **Algorytm Hilla** nie zmienił kształtu ani długości wejściowego tekstu. Jego szyfrowanie opiera się na matematycznej operacji macierzowej, która utrudnia analizę struktury tekstu. Jest bardziej odporny na analizę częstotliwości niż algorytmy Cezara czy Vigenera, jednak

wymaga odpowiedniego doboru i odwracalności klucza macierzy, co może być wadą przy niewłaściwej implementacji. Dla przypadku tekstu numer 3 szyfr nie działa najlepiej, ponieważ wciąż widać powtarzalne fragmenty tekstu. Powiększenie macierzy kluczy nie dało lepszego rezultatu.

1. **Algorytm Playfair** grupuje tekst w pary liter, więc długość tekstu jawnego może się zmienić, ponieważ w przypadku wystąpienia podwójnych liter w parze (np. „ee”) wstawiana jest litera zapasowa (np. „x”). Dla przypadku tekstu numer 3 szyfr nie działa najlepiej, ponieważ wciąż widać powtarzalne fragmenty tekstu. Zmiana wartości klucza nie daje w tym przypadku żadnej sensownej zmiany.
2. **Algorytm XOR** zmienia długość tekstu, zależnie od zastosowanego klucza, który może mieć dowolną długość. Algorytm nie zachowuje struktury tekstu jawnego i może wytworzyć zaszyfrowany tekst o zupełnie losowym wyglądzie. Jego siła wynika głównie z losowości klucza – jeśli klucz jest użyty jednorazowo i ma odpowiednią długość (jak w szyfrze Vernama), algorytm XOR może być teoretycznie nie do złamania. Dla przypadku tekstu numer 3 szyfr nie działa najlepiej, ponieważ wciąż widać powtarzalne fragmenty tekstu. Zmiana wartości klucza nie daje w tym przypadku żadnej sensownej zmiany, oprócz zwiększenia się okresu powtarzania znaków.
3. **Algorytm permutacji/transpozycji** – Algorytm ten zmienia jedynie kolejność liter lub bloków liter w tekście, co oznacza, że długość tekstu pozostaje niezmieniona. Jednak zmienia kształt tekstu w sposób znaczący, ponieważ układ liter jest mocno zmieniany. Permutacje mogą sprawić, że odnalezienie tekstu jawnego jest trudniejsze niż w przypadku prostych algorytmów podstawienia, takich jak Cezar czy Vigener. Dla przypadku tekstu numer 3 szyfr działa dużo lepiej niż pozostałe algorytmy, ponieważ pozbywa się jednorodności tekstu i sprawia wrażenie losowego (chociaż wciąż nie używa całego alfabetu znaków). Zmiana wartości klucza nie daje w tym przypadku żadnej sensownej zmiany.

**Pytanie 1:**

Szyfrowanie wielokrotne dla tych samych algorytmów co wcześniej.

**Cezar:**

Klucz1 = 13/„N”, Klucz2 = 4/”D”, klucz do deszyfrowania = 17/”Q”

Obraz zawierający tekst, oprogramowanie, Czcionka, linia

Opis wygenerowany automatycznie

**Vigenere:**

Klucz1 = „BIED”, Klucz2 = „RONKA”, klucz do deszyfrowania = „SWRNBZSQLIVROSEUPVOD”

Obraz zawierający tekst, oprogramowanie, Czcionka, linia

Opis wygenerowany automatycznie

**Hill:**

Klucze 1 i 2 = „BI ED”, klucz deszyfrujący = „WD BI” macierz 2x2

Obraz zawierający tekst, oprogramowanie, linia, Czcionka

Opis wygenerowany automatycznie

Klucze 1 i 2 = „AZ MA” macierz 2x2

Obraz zawierający tekst, elektronika, zrzut ekranu, wyświetlacz

Opis wygenerowany automatycznie

**XOR:**

Klucz1 = „01 02”, klucz2 = „03 04”, klucz deszyfrujący: „02 06”

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Wnioski:

Szyfrowanie wielokrotne **nie utrudnia zbytnio procesu rozszyfrowania tekstu**. W przypadku szyfrów takich jak szyfr Cezara (korzystających z jednego alfabetu) wielokrotne szyfrowanie będzie mieć takie same skutki co szyfrowanie jednokrotnie o inną długość klucza. W przypadku pozostałych algorytmów takich jak szyfr Vigenera (korzystających z wielu alfabetów) wielokrotne szyfrowanie może mieć niewielkie znaczenie, bo sprawia jedynie że komplikuje się klucz służący do odszyfrowania wiadomości.

W przypadku szyfru Hilla oraz XOR wielokrotne szyfrowanie czasem utrudnia proces rozszyfrowania, bo może się znacznie skomplikować klucz deszyfrujący, lecz czasami szyfrowanie wielokrotne powoduje całkowite rozszyfrowanie tekstu.

**Pytanie 2:**