|  |
| --- |
| 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

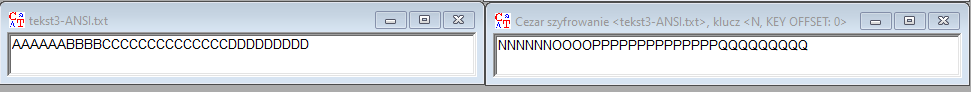
**Zadanie 1.1; 1.2:**

**Cezar:**

klucz = 13/„N”

Obraz zawierający tekst, Czcionka, oprogramowanie, numer

Opis wygenerowany automatycznie

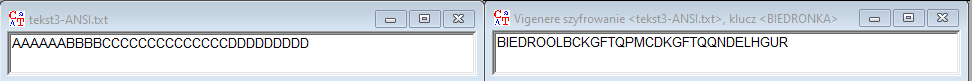


**Vigenere:**

klucz = „BIEDRONKA”

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

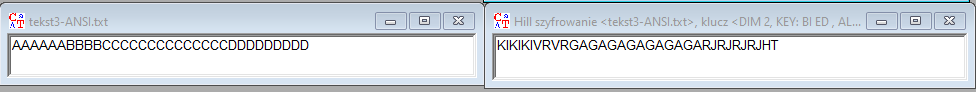


**Hill:**

klucz = „BI ED”, macierz 2x2

**Obraz zawierający tekst, Czcionka, oprogramowanie, numer

Opis wygenerowany automatycznie**

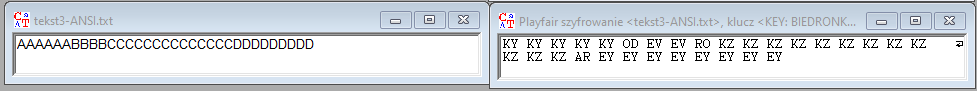
****

**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

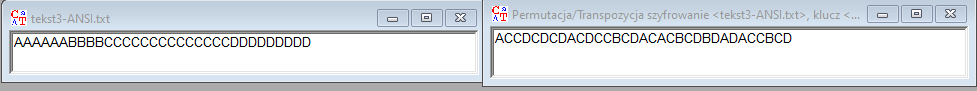
Opis wygenerowany automatycznie

**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 jak i wielokrotne szyfrowanie 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). Dla dłuższych kluczy (np. o długości tekstu jawnego) algorytm wydaje się być trudniejszy do złamania, jeśli klucz ten jest losowy i bezokresowy. Wielokrotne szyfrowanie wydłuża jedynie długość klucza.
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 czy rozmiaru macierzy 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 przypomina swoją dwójkową strukturą algorytm Playfair. Jego siła wynika głównie z losowości klucza, jeśli klucz jest użyty jednorazowo i ma odpowiednią długość (podobnie 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 1.3:**

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 (korzystający z jednego alfabetu z przesunięciem modulo 26) 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 wielokrotne szyfrowanie czasem utrudnia proces rozszyfrowania, bo może się znacznie skomplikować klucz deszyfrujący, lecz czasami szyfrowanie wielokrotne powoduje całkowite rozszyfrowanie tekstu (szczególnie dobrze widać to na przykładzie drugim, gdzie po podwójnym zaszyfrowaniu przy pomocy klucza matrycy 2x2 (AZ MA) tekst wraca do oryginalnego stanu i nie jest wcale zaszyfrowany). Wielokrotne szyfrowanie z użyciem XOR z tym samym kluczem nie ma sensu, ponieważ podwójne szyfrowanie XOR jest równoważne z odszyfrowaniem tekstu.

Podsumowując efekty wielokrotnego szyfrowania mają niewielki (Vigener, Hill) lub zerowy (Cezar, XOR) wpływ na trudność procesu rozszyfrowania tekstu w porównaniu do tekstu zaszyfrowanego jednokrotnie.

**Pytanie 1.4:**

Z przetestowanych wcześniej algorytmów można uznać **Vigenere** oraz **Hill** za silniejsze od reszty, zwłaszcza przy odpowiednich ustawieniach kluczy. Vigenere z długim, losowym kluczem podobnie jak Hill z dobrze dobraną macierzą jest trudniejszy do złamania niż Cezar, który łatwo złamać przeprowadzając prostą analizę frekwencji występowania znaków. **XOR** mimo swojej prostoty jest bezpieczny tylko wtedy, gdy klucz jest równie długi jak tekst, w innym przypadku jest podatny na ataki.

**Zad 2. Analiza własności dostępnych algorytmów.**

**Zadanie 2.1:**

Przetestowane na podstawie fragmentu z artykułu "Algorytm" na Wikipedii.

|  |  |  |
| --- | --- | --- |
| **Entropia tekstów jawnych dla języków: polski, angielski, hiszpański** | | |
| **Polski** | **Angielski** | **Hiszpański** |
| **4,19** | **4,06** | **3,93** |

**Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, Strona internetowa

Opis wygenerowany automatycznie**

**Zadanie 2.2:**

|  |  |
| --- | --- |
| **Entropia tekstu jawnego i tekstu zaszyfrowanego (dla tekstu polskiego)** | |
| Cezar | 4,19/4,7 |
| Adfgvx | 2,57/4,7 |
| Homofony | 6,86/8,0 (znormalizowane: 4,03) |
| Permutacyjny | 4,19/4,7 |
| Vigenere (klucz: AE) | 4,44/4,7 |
| Vigenere (klucz: BIEDRONKA) | 4,62/4,7 |
| Vigenere (klucz: KROLKAROLKUPILKROLOWEJ) | 4,53/4,7 |
| Hilla (klucz: BI ED) | 4,56/4,7 |
| Hilla (klucz: JAC ABA CAK) | 4,62/4,7 |
| Hilla (klucz: SVBQH HLIMN ASMCF XVHML FRYAA) | 4,57/4,7 |

**Zadanie 2.3:**

Histogramy ASCII kolejno dla języków: polski, angielski, hiszpański

**Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie**

**Zadanie 2.4:**

1. Tekst jawny:

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

1. Cezar

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, linia

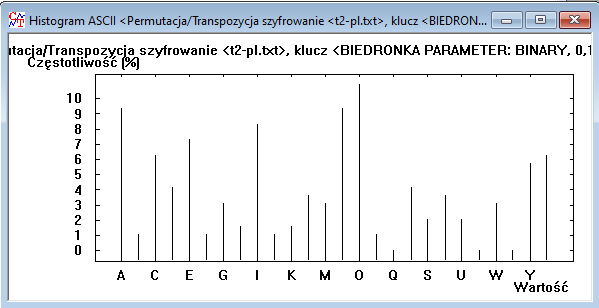
Opis wygenerowany automatycznie

1. Adfgvx:

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, numer

Opis wygenerowany automatycznie

1. Homofony: Histogramu dla homofonów nie udało się wygenerować z uwagi na błąd w programie Cryptool.
2. Permutacyjny

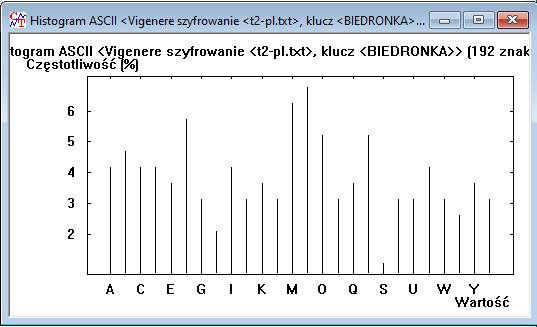


1. Vigenere (klucz: AE)

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

1. Vigenere (klucz: BIEDRONKA)



1. Vigenere (klucz: KROLKAROLKUPILKROLOWEJ)

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, numer

Opis wygenerowany automatycznie

1. Hilla (klucz: BI ED)

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

1. Hilla (klucz: JAC ABA CAK)

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, numer

Opis wygenerowany automatycznie

1. Hilla (klucz: SVBQH HLIMN ASMCF XVHML FRYAA)

Obraz zawierający tekst, zrzut ekranu, numer, wyświetlacz

Opis wygenerowany automatycznie

**Zadanie 2.5:**

Najczęstsze Digramy, Trigramy i N-gramy (dla tekstu numer 2 przetłumaczonego na inne języki) kolejno w języku:

1. Polskim:

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie**

1. Angielskim:

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

1. Hiszpańskim:

**Obraz zawierający tekst, zrzut ekranu, paragon, Czcionka

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

**Zadanie 2.6:**

Najczęstsze Digramy, Trigramy i N-gramy dla tekstu numer 2 zaszyfrowanego przez:

1. Nic

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie**

1. Cezara

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, paragon, Czcionka

Opis wygenerowany automatycznie

1. Adfgvx

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

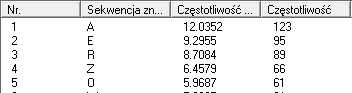
Opis wygenerowany automatycznie  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

1. Homofony – nie wygenerowało
2. Permutacyjny



Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Vigenere (klucz: AE)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Vigenere (klucz: BIEDRONKA)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Vigenere (klucz: KROLKAROLKUPILKROLOWEJ)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Hilla (klucz: BI ED)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, paragon, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Hilla (klucz: JAC ABA CAK)

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

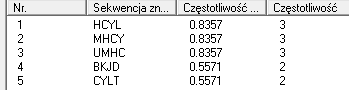
Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie



1. Hilla (klucz: SVBQH HLIMN ASMCF XVHML FRYAA)

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Zadanie 2.7:**

Autokorelacja dla algorytmu szyfrującego XOR oraz rosnącą długością klucza:

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Autokorelacja dla algorytmu szyfrującego Vigenera oraz rosnącą długością klucza:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

**Pytanie 2.8:**

1. Największą entropię spośród analizowanych języków posiada język polski (4,19), natomiast najmniejszą język hiszpański (3,93)
2. Dla szyfru Cezara i permutacyjnego entropia nie ulega zmianie, ponieważ nie tracimy żadnej informacji, jedynie przesuwamy każdy znak o ten sam klucz w przypadku cezara lub zamieniamy znaki miejscami w przypadku permutacji. Dla szyfru ADFGVX entropia spada znacznie, ponieważ używamy tylko 6 znaków. Dla homofonów entropia rośnie, bo pojawiają się nowe znaki. Dla Vigenere i Hilla entropia wzrasta, wraz z wzrostem długości klucza / wzrostem rozmiaru macierzy.
3. We wszystkich tekstach najczęściej występowały samogłoski, z drobnymi różnicami w częstotliwości np. w języku angielskim najczęstszą samogłoską było „E” natomiast w hiszpańskim oraz polskim było to „A”. Z kolei w języku polskim jako jedynym tak często padały litery „Y” oraz „Z”
4. Dla szyfru cezara nie zmienił kształtu, tylko przesunął się na osi X. Dla szyfru adgfvx widać tylko 6 liter, które się powtarzają. Permutacyjny jest bez zmian z tekstem jawnym. Vigener oraz Hill posiadają coraz bardziej płaskie histogramy wraz ze wzrostem rozmiaru klucza (długości klucza lub rozmiaru matrycy)
5. Można z łatwością rozpoznać pewne konstrukcje językowe dostępne tylko w konkretnych językach, które dużo rzadziej pojawiają się w innych językach (np. „PCH” „RZY” które są konstrukcjami charakteryzującymi język polski)
6. Dla cezara są te same konstrukcje co w tekście jawnym, ale przesunięte. Dla permutacyjnego konstrukcje jednoznakowe zostają takie same, ale wszystkie dłuższe zostają zniszczone i zostaje na ich miejscu „szum”.
7. Dla XORa liczba pasujących znaków maleje wraz z wzrostem długości klucza. Okresowości dla tego szyfru są raczej niewidoczne. Vigener dla krótkich kluczy ujawnia swoją okresowość, wraz ze zwiększaniem długości klucza problem ten zanika i staje się mniej dostrzegalny.

**Pytanie 2.9:**

1. Pomiar entropi – wysoki jej pomiar może oznaczać algorytm Hilla z macierzą o dużych rozmiarach lub Vigenera o długim kluczu, mała entropia natomiast może oznaczać adfgvx, natomiast entropia zbliżona do średniego poziomu w danym języku może oznaczać Cezara lub permutacyjny.
2. **Analiza histogramu** – jeśli można połączyć często występujące litery z ich częstotliwością występowania w danym języku może to oznaczać algorytm Cezara lub algorytm permutacyjny. Jeśli liter występujących jest mało, a pojawiają się często możemy mieć do czynienia z szyfrem adfgvx. Jeśli liter powtarzających się jest mało to możemy mierzyć się z Hillem lub Vigenerem.
3. **Analiza n-gramów** – jeśli niektóre fragmenty pojawiają się wyraźnie częściej niż pozostałe to najprawdopodobniej jest to szyfr cezara, adfgvx lub permutacyjny. Jeśli dużo różnych cząstek 3-4 znakowych możemy mieć do czynienia z Vigenerem lub Hillem.
4. **Analiza wykresu autokorelacji** – jeśli widać na wykresie widać powtarzalne w sposób okresowy fragmenty, to możemy mieć do czynienia z Vigenerem. Jeśli takiej powtarzalności brak – to może to być XOR.

**Pytanie 2.10:**

**Analiza histogramu** – jeśli mamy do czynienia z szyfrem Cezara lub Vigenera o krótkim kluczu, najczęściej występujące litery mogą odpowiadać literom o największej frekwencji w języku. Można na tej podstawie odtworzyć przesunięcia klucza.

**Analiza n-gramów** – można rozpoznać specyficzne wzorce, które mogły powstać w wyniku szyfrowania kluczami o określonej długości.

**Analiza wykresu autokorelacji** – pozwala wykryć powtarzające się wzorce w zaszyfrowanym tekście, co jest szczególnie przydatne dla szyfru Vinegera. Okresowe powtarzanie wzorców w tekście może wskazać klucz albo przynajmniej jego długość, co jest pierwszym krokiem do jego ustalenia.

**Zad 3. Analiza dostarczonych plików.**

**Zadanie 3.1:**

**1\_2.txt** – Analiza histogramu, wskazuje na przesunięcie liter charakterystyczne dla szyfru Cezara. Najczęściej występującą literą była litera K, co sugerowało niewielkie przesunięcie względem standardowej częstotliwości liter w popularnych językach (gdzie najczęściej jest to A lub E). Niska entropia (4,10), która zbliżona jest do entropii popularnych języków również potwierdza teorię o wykorzystaniu szyfru Cezara. Korzystając z programu brute-force napisanego w python, i analizując przesunięcia na wszystkich 26 kombinacjach szyfru Cezara ustalono, że klucz to litera G (przesunięcie o 6).

Obraz zawierający zrzut ekranu, wzór, tekst, sztuka

Opis wygenerowany automatycznie

**1\_1.txt** – Narzędzia autokorelacji w Cryptool wskazuje na to, że tekst został zaszyfrowany algorytmem powtarzającym klucz co stałą ilość znaków. Powtarzalne odległości luk między znakami wskazywały na długość klucza na około 6-8 znaków. Jest to cecha charakterystyczna między innymi szyfru Vinegera. Następnie wykorzystując tekst jawny z poprzedniego punktu, udało się przyporządkować wzrost liczbowy każdej z kolejnych liter (THEENIG… -> USYIFSE…), co pozwoliło ustalić, że klucz użyty do szyfrowania to „BLUESKY”.

**1\_3.txt** – Analiza entropii i histogramu wykazała, że rozkład liter w zaszyfrowanym tekście jest identyczny w porównaniu do tekstu jawnego, jednak litery znajdują się na innych pozycjach. Litery bliżej końca alfabetu zdawały się pozostać bez zmian (przynajmniej niektóre). Sugerowało to użycie szyfru podstawieniowego. Skrypt w pythonie pozwolił porównać poszczególne litery z tekstu jawnego z literami w tekście zaszyfrowanym na tych samych pozycjach, co poskutkowało alfabetem: „QWERTYABCDFGHIJKLMNOPSUVXZ” -> co daje klucz „QWERTY”

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

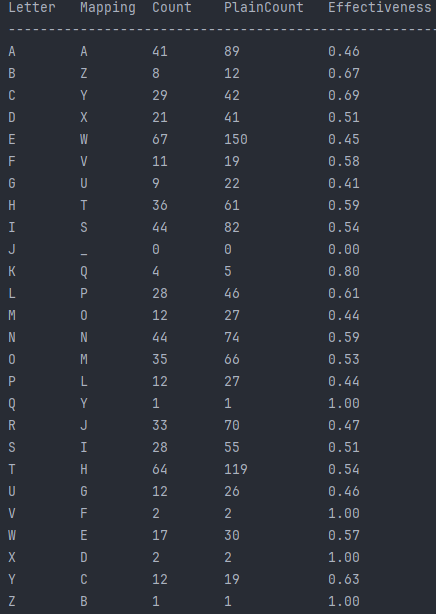
Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

**1\_4.txt** – Analiza zaszyfrowanego tekstu wskazuje na użycie algorytmu, który wyrównuje częstotliwości liter w histogramie. Wartość entropii na wysokim poziomie (4,52) oraz rozkład słupków na histogramie sugerują, że algorytm nieznacznie zmniejsza częstotliwość występowania najczęstszych liter i podnosi częstotliwość rzadszych. Pomimo tego nadal można zaobserwować obecność dwóch wyraźnych pików, prawdopodobnie odpowiadających literom W i H w szyfrogramie, które w tekście jawnym mogą odpowiadać literom A, E lub T.

Autokorelacja nie wskazała na istnienie regularnych pików, co mogłoby sugerować prostą cykliczność w szyfrowaniu. Analiza ngramów wykazała spłaszczenie rozkładu – zmniejszyła się częstotliwość występowania charakterystycznych dla języka angielskiego ngramów, a w zamian pojawiło się więcej różnorodnych ngramów o niższej częstotliwości.

Chociaż nie udało mi się zidentyfikować klucza szyfrowania, zauważyłem pewne regularności w postaci liter, które często pozostają bez zmian w szyfrogramie (np. A->A, N->N) na tych samych pozycjach). Na podstawie tych obserwacji napisałem skrypt w Pythonie, który wygenerował potencjalny alfabet-klucz: „AZYXWVUTS\_QPONMLYJIHGFEDCB”. Niestety, ten klucz nie pasuje do żadnego znanego algorytmu szyfrującego i nie rozwiązuje problemu.



**Zadanie 3.2:**

**2\_1.txt** – autokorelacja powtarzająca się co określony okres czasu, więc podejrzewam vinegera o długości klucza około 11. Używając dostępnych opcji automatycznego łamania szyfru dla szyfru vinegera analiza automatyczna wykryła klucz pośredni: „BLFCKWINDOW” co intuicyjnie poprawiłem na „BLACKWINDOW”.  
**Obraz zawierający tekst, zrzut ekranu, design

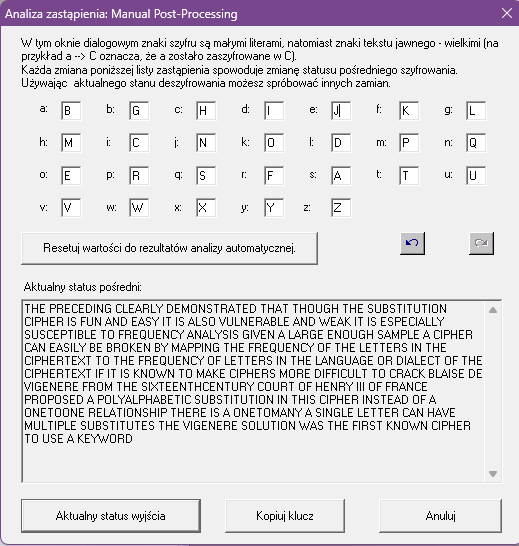
Opis wygenerowany automatycznie**

**2\_2.txt** – Niska entropia (4,15) może sugerować szyfr atbash lub permutacyjny. Końcowa część histogramu pokrywa się z histogramem tekstu jawnego, co wskazuje na szyfr atbash. Wysokości słupków na histogramie mają te same piki i dołki.

Obraz zawierający tekst, zrzut ekranu, linia, numer

Opis wygenerowany automatycznie

Po zabawie w analizie ręcznej szyfru zastąpienia, udało się odnaleźć klucz -> „SAILOR”



**2\_3.txt** – Format zaszyfrowanego tekstu sugeruje użycie algorytmu Playfair. Entropia (4,46). Używając ręcznej analizy playfair doszedłem tylko do tego momentu:   
Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Jednak klucz: „KMNHIRSTPQXYZVWOUDCLEFGAB” nic mi nie mówi, więc nie udało mi się dokładnie złamać tego kodu. Jestem jednak pewien że byłem blisko, bo przy wpisywaniu tekstu jawnego, algorytm podpowiadał mi dalszą jego część poprawnie bez jego znajomości, więc sam klucz gdzieś musiał być tylko nie umiałem skorzystać poprawnie z narzędzia.

**Zadanie 3.3:**

**3\_1.txt** – Hexadecymalne kodowanie sugeruje wykorzystanie algorytmu XOR lub Homofonów. Autokorelacja ma trend spadkowy. Nic więcej nie udało mi się ustalić. Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**3\_2.txt** – Entropia (4,20) może wskazywać że nie jest to ani szyfr Hilla czy Vinegera, gdyż byłaby ona wyższa. Na wykresie autokorelacji nie widać znaczącej cykliczności pików co wyklucza wykorzystanie Vinegera. Histogram sugeruje, że litery ‘U’ oraz ‘R’ mogą odpowiadać popularnym literom w języku polskim lub angielskim (np. ‘A’ lub ‘E’). Na pewno nie jest to szyfr Cezara.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Wykres

Opis wygenerowany automatycznie

**3\_3.txt** – Bardzo wysoka entropia (4,63) wskazuje na to, że jest to Hill lub Vineger. Cykliczna autokorelacja pozwala zauważyć że jest to Vineger, o długości klucza około 10 znaków. Wykorzystując automatyczne narzędzia do analizy szyfru Vinegera, udało się odnaleźć klucz „WHITESTONE”.Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Czcionka

Opis wygenerowany automatycznie

**Pytanie 3.4:**

Siła algorytmu szyfrującego zależy od kilku czynników:

* Długość i złożoność klucza, im dłuższy i bardziej złożony klucz, tym trudniej go złamać. Krótkie klucze są podatne na ataki brute-force, a klucze mniej złożone możemy łatwiej rozpoznać kiedy się wyłonią (jeśli są cokolwiek znaczącym ciągiem znaków). Gdy klucz jest długi i zawiera znaki trudniejsze do zgadnięcia, staje się on bezpieczniejszy.
* Wysokość entropii, im większa entropia tym ciężej odnaleźć wzorce np. na histogramie, ponieważ tekst jest mniej przewidywalny i trudniejszy do odgadnięcia. Szyfr nie powinien ujawniać żadnych przewidywalnych wzorców.
* Istotne jest też, aby algorytm ukrywał naturę tekstu i ewentualną cykliczność na autokorelacji, nie powinien też ujawniać czy tekst zawiera dużo powtarzających się znaków, czy nie.

**Pytanie 3.5:**

Dla niektórych algorytmów wielokrotne szyfrowanie może zwiększyć siłę szyfrowania (np. szyfr Hilla). Jednak na ogół lepiej zwiększyć złożoność i długość klucza, uniknąć w nim powtórzeń i cykliczności samego klucza. Warto też korzystać z elementu losowości, tak aby nie dało się znaleźć w naszym zaszyfrowanym tekście powtarzalnych wzorców np. przez wybór dość losowego algorytmu (przykładowo w algorytmie homofonicznym, homofony są generowane w sposób losowy, co utrudnia jego złamanie), albo losowy dobór klucza.