Uniwersytet Przyrodniczy we Wrocławiu  
Wydział Biologii i Hodowli Zwierząt  
Bioinformatyka



Łukasz Chołodecki

Biblioteka programistyczna wykorzystująca metody analizy statystycznej wobec wybranych klasycznych metod szyfrowania

Praca licencjacka pod kierunkiem  
 dra Jana Jełowickiego

Wrocław, 2021

Contents

[WSTĘP DO SZYFROWANIA 3](#_Toc73735824)

[Zadanie szyfrów 3](#_Toc73735825)

[Historia szyfrów 3](#_Toc73735826)

[Cechy szyfrów 5](#_Toc73735827)

[Bezpieczny szyfr 6](#_Toc73735828)

[METODY KRYPTOANALITYCZNE 7](#_Toc73735829)

[Wstęp do kryptoanalizy 7](#_Toc73735830)

[Podział ze względu na źródło 7](#_Toc73735831)

[Metoda brute-force 8](#_Toc73735832)

[Test chi-kwadrat zgodności 8](#_Toc73735833)

[Kappa-test (index of coincidence) 8](#_Toc73735834)

[Podatności szyfrów 9](#_Toc73735835)

[ANATOMIA JĘZYKA POLSKIEGO W UJĘCIU KRYPTOANALITYCZNYM 9](#_Toc73735836)

[Wstęp do badań anatomii języka 9](#_Toc73735837)

[Korpus języka polskiego 9](#_Toc73735838)

[Częstotliwości liter 10](#_Toc73735839)

[Współczynnik kappa 12](#_Toc73735840)

[Częstotliwości liter na skraju wyrazów 12](#_Toc73735841)

[Częstotliwość liter na początkach słów 13](#_Toc73735842)

[Częstotliwość liter na końcach słów 14](#_Toc73735843)

[Częstotliwość krótkich słów 16](#_Toc73735844)

[Porównanie do języka angielskiego 17](#_Toc73735845)

[OMÓWIENIE BIBLIOTEKI 17](#_Toc73735846)

[Zakres, który obejmuje (opis, teoria) 17](#_Toc73735847)

[Wygląd modułów 17](#_Toc73735848)

[Obróbka tekstu wejściowego 19](#_Toc73735849)

[Szyfry podstawieniowe monoalfabetyczne 19](#_Toc73735850)

[Homofoniczny szyfr podstawieniowy 21](#_Toc73735851)

[Polialfabetyczny szyfr substytucyjny 22](#_Toc73735852)

# WSTĘP DO SZYFROWANIA

## Zadanie szyfrów

Szyfrem jest dowolny algorytm, często również przedstawiany jako funkcja szyfrująca, której głównym zadaniem jest takie zniekształcenie przekazywanej informacji, aby odbiorca postronny nie był w stanie jej odczytać, a jednocześnie możliwe było jej pełne odtworzenie przez osoby posiadające odpowiednią wiedzę. Początkowo szyfry były wykorzystywane nielicznie, głównie w zastosowaniach wojskowych, ale wraz z czasem ich popularność rosła, a w trakcie pierwszej i drugiej wojny światowej były one kluczowym elementem rywalizacji pomiędzy stronami. Obecnie, ze względu na powszechność ich wykorzystania w świecie cyfrowym, kryptografia odgrywa kluczową rolę ze względu na podstawę wszystkich zabezpieczeń.

## Historia szyfrów

Za pierwszy szyfr w historii uznaje się, pochodzący z V wieku przed naszą erą, skytale, opracowany przez starożytnych Spartan. Metoda zapisu polegała na owinięciu długiego paska skóry na lasce o specyficznej grubości i zapisaniu wiadomości w poprzek. Dzięki temu, po rozwinięciu paska otrzymywano zniekształcony szyfrogram. W gruncie rzeczy był to zwykły szyfr przestawieniowy, jednak możliwy do szybkiego szyfrowania i odszyfrowania dzięki specyficznemu, fizycznemu kluczowi.

Drugim szyfrem pochodzącym z tego okresu był hebrajski atbasz, w którym litery były wzajemnie zastępowane przez te oddalone o równą długość od drugiego końca, np. A-Z, B-Y. Stąd też pochodzi nazwa tego szyfru, powstała z zakodowania dwóch pierwszych liter hebrajskiego alfabetu: alef-taw, beth-shin.

Następny szyfr to jeden z opracowanych przez Juliusza Cezara (i od niego później nazwany) — szyfr Cezara, pochodzący z I wieku przed naszą erą. Metoda szyfrowania opiera się na cyklicznym przesunięciu alfabetu o stałą wartość. Oryginalnie szyfr wykorzystywał przesunięcie o 3 znaki, ale był później używany również z innymi wartościami obrotu.

Bezpieczeństwo tych metod szyfrowania opierało się głównie, nie na sile szyfru, a nieznajomości zastosowanej metody szyfrowania, co wraz z powstaniem metody kryptoanalizy opartej na statystyce częstotliwości liter w IX wieku, sprawiało, że konieczne było stosowanie bardziej skomplikowanych algorytmów szyfrujących.

Kolejną próbą, był szyfr Marii Stuart, opracowany na potrzeby królowej Szkocji, rywalizującej z królową Anglii, pod koniec XVI wieku. Szyfr oparty był na zamianie liter na symbole oraz dodatkowej zamianie najczęściej używanych słów na pojedyncze symbole, kilku znakach odpowiadających przerwie między literami i zastąpieniu podwojonych liter specjalnym znakiem. Mimo zastosowania dodatkowych zabezpieczeń oraz ze względu na przekazywanie przez posłańca wszystkich wiadomości do angielskich kryptoanalityków (a więc dużej bazie tekstu), szyfr został złamany, a królowa Maria Stuart skazana na śmierć za zdradę korony.

W wyniku rozwiniętej kryptoanalizy jakiekolwiek szyfry opierające się na monoalfabetycznych przekształceniach i podstawieniach były możliwe do złamania, stąd zaczęto skłaniać się ku szyfrom bardziej skomplikowanym. Pierwszy szyfr, Johannesa Trithemiusa, oparty był o tabula recta — kwadrat złożony z wypisanego alfabetu w pierwszym wierszu, a następnie przepisywanemu do poniższych wierszy, przesuwając go o jedną literę (jak w szyfrze Cezara). Kolejne wariacje, również oparte na tabula recta, wykonali Giambattista della Porta, a następnie Blaise de Vigenère, końcowo tworząc szyfr nazwany szyfrem Vigenère’a. Szyfr ten opiera się o kodowanie z wykorzystaniem wielu alfabetów oraz dodatkowego, unikalnego klucza, który był wielokrotnie powtarzany, równolegle do tekstu, przez co podatny był na kryptoanalizę. Pomimo tego, aż do drugiej połowy XIX wieku nie został on złamany.

Od XIX wieku nastąpił znaczny rozwój zarówno kryptografii, jak i kryptoanalizy, przestano również polegać na utrzymywaniu algorytmu szyfrującego w tajemnicy jako głównego elementu bezpieczeństwa szyfrów, a skupiono się na znaczeniu klucza, który powinien pozwalać na zachowanie wiadomości w tajemnicy, tak długo, jak sam klucz nie zostanie odgadnięty.

Jednym z pomysłów był szyfr Playfair, opierający się o kodowanie tekstu podzielonego na dwuznaki, zamiast pojedyncze litery, co znacząco utrudniało analizę częstotliwości, ze względu na o wiele większą liczbę możliwych kombinacji. Oprócz tego używany był klucz, zapewniający tajność. Jednak ze względu na liczne, często mniej oczywiste podatności, został on stosunkowo szybko złamany.

Pod koniec pierwszej wojny światowej powstał niemiecki szyfr ADFGVX, który stosował podwójne szyfrowanie (oraz dwa klucze) — najpierw podstawienie każdej litery i cyfry według wzoru (wykorzystując wersję szachownicy Polybiusza) oraz szyfr przestawieniowy, rozbijający powstałe we wcześniejszym etapie dwuznaki. Oprócz tego, dzięki zapisie całej wiadomości za pomocą tylko sześciu liter, znacząco ułatwiło to jego nadawanie za pomocą kodu Morse’a. Pomimo stosunkowej złożoności metod i użycia dwóch kluczy (co miało według Niemców zapewnić zupełną odporność szyfru na łamanie) został złamany w ciągu trzech miesięcy.

W kolejnych latach, wraz z rozwojem technologii i elektroniki, zaczęły powstawać znacznie bardziej skomplikowane maszyny szyfrujące, pozwalające w łatwy sposób zakodować i odczytać wiadomość, przeprowadzając serię przekształceń, która w wykonaniu człowieka zajmowałaby znacznie więcej czasu. Jednym z najbardziej znanych przykładów takich elektromechanicznych była Enigma, powstała na początku lat 20., na początku w komercyjnych zastosowaniach, a później w trakcie drugiej wojny światowej przez niemieckie wojsko. Zbudowana była z wielu niezależnych od siebie elementów, które mogły być łączone w różnych konfiguracjach, m.in. wirniki, łącznica kablowa i uzwojenia. Dzięki pracom polskich matematyków pierwszy raz udało się złamać szyfr w 1932, jednak wraz z upływem czasu poziom skomplikowania maszyn rósł, przez co w momencie wybuchu wojny wersja używana przez wojsko była dalej niezłamana. Polacy następnie przekazali wyniki swoich badań sojuszowi Aliantów, co znacząco przyspieszyło ich prace. Ostatecznie złamana została w pierwszej połowie 1940 przez Alana Turinga, dzięki konstrukcji zaawansowanego narzędzia kryptoanalitycznego zwanego bombą.

W wyniku co raz szybciej postępującej automatyzacji, powstaniu pierwszych komputerów przekraczających dotychczasowe możliwości obliczeniowe, zwykłe szyfry nie zapewniały jakiejkolwiek ochrony ze względu na możliwość złamania takiego szyfru w ekstremalnie krótkim czasie. Nowe metody musiały być na tyle skomplikowane obliczeniowo, aby ewentualna próba złamania zabezpieczenia, nawet z użyciem komputera zajęła niewspółmiernie dużo czasu.

W teraźniejszych czasach znaczenie szyfrów jest niezwykle ważne, ponieważ to one odpowiadają za wszystkie zabezpieczenia obecne w urządzeniach elektronicznych — od telefonów po serwery sieciowe. Jednym z najpopularniejszych szyfrów, używanych od początku XXI wieku jest algorytm AES szyfrujący 128-bitowe bloki danych przy użyciu klucza 128, 192 lub 256-bitowego. Pomimo szerokiego zastosowania w bezpieczeństwie od ponad 20 lat nie znaleziono żadnego sposobu na bezpośrednie złamanie tego szyfru, a wynalezione sposoby osłabiania, nawet dla klucza 128-bitowego, wymagałyby milionów lat pracy na komputerach z obecnym rzędem wydajności.

## Cechy szyfrów

Pierwszy podział szyfrów dzieli je na sposób działania - najpopularniejsze są szyfry substytucyjne, które według różnych zasad zamieniają litery tekstu na inne ustalone znaki, litery, bądź symbole. Te szyfry jednak oferują bardzo słabą ochronę w pierwotnej wersji, jeśli cały tekst przekształcany jest jednakowo (szyfry monoalfabetyczne). Szyfry polialfabetyczne dokonują zmian liter według dodatkowych reguł, dzieląc tekst na kilka grup, z których każda szyfrowana jest w inny sposób (najczęściej z wykorzystaniem innego klucza). Drugim podejściem są szyfry homofoniczne, w których jednemu znakowi, odpowiada kilka różnych znaków (bądź częściej zbitek znaków jak liczby czy słowa). Przy odpowiednim dopasowaniu ilości różnych znaków dla jednej litery na podstawie oczekiwanej częstotliwości wystąpienia tej litery, możliwe jest całkowite uodpornienie szyfru na prostą kryptoanalizę statystyczną. Możliwe jest też szyfrowanie, nie pojedynczych znaków, a ich grup — na przykład dwuznaków. Takie rozwiązanie również znacznie utrudnia analizę statystyczną, ze względu na znacznie większe rozdrobnienie analizowanych elementów.

Szyfry przestawieniowe nie zmieniają samych znaków wiadomości, a jedynie ich kolejność. Ze względu na zwyczajowe korzystanie z prostych przekształceń, liczba możliwości jest mała i powtarzalna między różnymi szyfrowaniami, stąd też małe bezpieczeństwo takiego szyfru; można natomiast stosować go w kombinacji z szyframi substytucyjnymi.

Współczesne szyfry komputerowe można podzielić na kilka grup, według różnych kryteriów. Podstawowa dzieli szyfry ze względu na rodzaje klucza. Pierwsza, obejmujące powyższe szyfry historyczne, jest to grupa szyfrów symetrycznych (szyfrów z kluczem prywatnym). W założeniu obie strony znają ten sam klucz i za jego pomocą szyfrują wzajemnie wysyłane wiadomości, które następnie rozszyfrowują z wykorzystaniem tego samego klucza. Drugi model — asymetryczny (szyfr z kluczem publicznym) posiada dwa klucze — publiczny, pozwalający dowolnej osobie na zaszyfrowanie wiadomości, oraz klucz prywatny, który posiada tylko odbiorca i tylko za jego pomocą można wiadomości odszyfrowywać.

## Bezpieczny szyfr

Pod koniec XIX wieku wymyślono termin szyfru z kluczem jednorazowym (one-time pad), który pozwalał przy spełnieniu pewnych założeń, na szyfrowanie w pełni odporne na dowolną kryptoanalizę. Pierwszym historycznym szyfrem, który można wykorzystać z kluczem jednorazowym był znacznie starszy szyfr Vigenère’a, później popularność zdobyło szyfrowanie bitowe za pomocą operatora XOR (z racji równego prawdopodobieństwa zwrócenia wartości 0 i 1, powodując, że każda wartość z równą szansą została, bądź nie została zmieniona). Jeśli klucz, którego użyjemy, spełni poniższe warunki, niemożliwe będzie odtworzenie wiadomości. Po pierwsze klucz musi być w pełni losowy – oznacza to, że nie istnieje żadna metoda, która pozwoliłaby celowo ponownie wygenerować ten sam klucz (niezależnie od formy, jaką przyjmuje). Po drugie klucz, musi być co najmniej długości szyfrowanego tekstu – dzięki temu, nie będzie on wykorzystany drugi raz do zaszyfrowania innej części tekstu, co pozwoliłoby na porównywanie ze sobą dwóch elementów zaszyfrowanego tekstu. Z tego samego powodu klucz nie może być ponownie wykorzystany, do szyfrowania innej wiadomości.

Choć taka metoda szyfrowania jest w pełni bezpieczna, wymagania potrzebne do jej użycia stanowią problem – ponieważ w bezpieczny sposób musimy przekazać hasło długości wiadomości, a więc zazwyczaj możemy po prostu przekazać w taki sposób wiadomość. Z tego powodu zastosowanie ma to głównie dla niezwykle ważnych kanałów, dla których w przeszłości w bezpieczny sposób obie strony ustaliły hasła. Jednak, im klucz lepiej będzie spełniać to zadanie, tym trudniejsze będzie jego złamanie.

# METODY KRYPTOANALITYCZNE

## Wstęp do kryptoanalizy

Kryptoanaliza rozwijała się wraz z postępem w tworzeniu nowych metod szyfrowania. O ile pierwsze szyfry nie wymagały skomplikowanych metod do ich łamania, to kiedy zaczęto stosować szyfry substytucyjne, potrzebne były bardziej skonkretyzowane narzędzia. W IX wieku arabski matematyk Al-Kindi opublikował pierwszą pracę poświęconą temu tematowi, gdzie przedstawiał metody analizy częstotliwości, opisy częstotliwości znaków w języku arabskim i sposoby łamania różnych szyfrów, w tym polialfabetycznym. W Europie dopiero w drugiej połowie XVI wieku pojawiła się pierwsza praca dotycząca tego tematu, napisana przez włoskiego uczonego Giambattista della Porta, choć już wcześniej zajmowano się tą tematyką.

Sama kryptoanaliza dzieli się na dwie zasadnicze części. Pierwszą jest odkrycie działania szyfru wykorzystanego do zaszyfrowania danej wiadomości. Im dokładniejsza jest znajomość algorytmu szyfrowania, tym łatwiej później złamać szyfr, chociaż nie jest to warunkiem koniecznym – jeśli wiemy tylko, że użyty szyfr korzysta z jakiejś monoalfabetycznej metody substytucyjnej, możemy go złamać, stosując najszersze podejście – jednak jeśli wiedzielibyśmy, że użyto konkretnie szyfru Cezara, złamanie byłoby znacznie prostsze przy wykorzystaniu odpowiedniej metody.

Drugą częścią jest próba złamania bezpośrednio klucza (jeśli szyfr nie posiada klucza, sama znajomość algorytmu, pozwala na rozwiązanie szyfru). W przypadku kiedy szyfr jest unikatowy, należy najpierw opracować, lub dostosować metody łamania, wyszukać podatności i dopiero wtedy przystępować do łamania. Dla bardziej skomplikowanych szyfrów potrzebne są wielostopniowe działania, np. próby osłabiania szyfru, eliminowania możliwych kluczy etc.

## Podział ze względu na źródło

Największe zróżnicowanie w kryptoanalizie występuje ze względu na rodzaj dostępnych materiałów i wiedzy rozszyfrującego. Najczęstszym przypadkiem, kiedy chcemy poznać algorytm szyfrowania jest sytuacja, w której mamy do dyspozycji tylko zaszyfrowany tekst (bądź teksty), zazwyczaj z użyciem różnych kluczy, ale możliwe są także sytuacje, w których mamy również dostęp do czystego tekstu (przed zaszyfrowaniem) i odpowiadającemu mu zaszyfrowanego. W projektowaniu biblioteki założyłem jedynie wariant, w którym mamy do dyspozycji jedynie jeden szyfrogram i ewentualną wiedzę na temat użytego szyfru

## Metoda brute-force

Atak brute-force jest najprostszym możliwym sposobem łamania szyfrów, a mimo to, jest dalej szeroko stosowany (zazwyczaj jako jeden z elementów). Polega na prostym, siłowym próbowaniu każdej możliwej kombinacji szyfru, aż odnajdzie się prawidłowy klucz. Rozwinięciem tej metody jest, np. metoda słownikowa, która zakłada, że prawdopodobne jest użycie jako klucz istniejącego, popularnego słowa i zaczyna atak od zbadania słów z zebranego zasobu — słownika. Głównym ograniczeniem tej metody jest ilość możliwych kluczy do zbadania – o ile w przypadku szyfru Cezara istnieje tylko 26 wariantów w postaci liczby z tego zakresu, to wraz z długością hasła i możliwymi znakami do użycia, poziom skomplikowania rośnie wykładniczo – dla klucza składającego się z 4 dowolnych liter (bez rozróżnia wielkości), możliwe jest prawie pół miliona kombinacji, a dla odpowiednika o długości 8 – już ponad 200 miliardów. Z tego powodu próba złamania nowożytnych szyfrów tylko w ten sposób, przy użyciu obecnej technologii, jest bezcelowa, ponieważ niemożliwe jest uzyskanie wyników w rzeczywistym czasie.

## Test chi-kwadrat zgodności

Kiedy zaczęto odchodzić od prostych szyfrów i zaczęto opierać się na szyfrach substytucyjnych, metoda brute-force przestała mieć zastosowanie, ze względu na ogromną ilość możliwości, którą trzeba by manualnie sprawdzić, skierowano się wówczas do bardziej skomplikowanych metod. Odkryto, że każdy język naturalny posiada swoje unikalne cechy związane m.in. z częstotliwościami występowania różnych liter, a w przypadku szyfrowania monoalfabetycznego pozostają one niezmienione – przekształceniu ulegają tylko symbole reprezentujące poszczególne litery. Obecnie do porównywania częstotliwości występowania wykorzystuje się test chi-kwadrat zgodności (Pearsona), w którym statystyka testowa dana jest wzorem:  
Gdzie dla każdej litery Oi to ilość obserwowana, a Ei to odpowiadająca ilość oczekiwana. Wartości testowe przyjmują wartości dodatnie, a im lepsze dopasowanie, tym niższa wartość.

## Kappa-test (index of coincidence)

Współczynnik kappa określa z jakim prawdopodobieństwem analizując, litera po literze, dwa teksty napisane w tym samym języku mamy szansę trafić na tę samą literę w obu tekstach– jego wielkość wynika ściśle z rozłożenia częstotliwości liter w danym języku, bez znaczenia na ich kolejność. Oznacza to, że tekst w danym języku naturalnym będzie zawsze miał podobną wartość współczynnika kappa – (nawet po zaszyfrowaniu dowolnym monoalfabetycznym szyfrem substytucyjnym, jako że kolejność nie ma wpływu na kappa). Test kappa dany jest wzorem:

Gdzie ni to ilość wystąpień poszczególnych liter w tekście, a N jest sumą wszystkich liter. Porównując między sobą teksty o znacznej różnicy w długości alfabetu, warto znormalizować współczynnik poprzez pomnożenie kappa o długość alfabetu

Współczynnik ten jest najczęściej wykorzystywany jako wstępna analiza szyfrogramu, aby stwierdzić z jakiego rodzaju szyfrem mamy do czynienia. Analizując tekst

## Podatności szyfrów

Oprócz różnych metod ogólnych służących do rozszyfrowania ważnym elementem są podatności szyfrów. Większość szyfrów w wyniku zróżnicowanej i oryginalnej budowy posiada różne cechy, nieprzewidziane przez twórcę, które mogą znacząco pomóc w kryptoanalizie. Podatności mogą również dotyczyć przesyłanego tekstu i powtarzających się fraz szyfrowanych.

# ANATOMIA JĘZYKA POLSKIEGO W UJĘCIU KRYPTOANALITYCZNYM

## Wstęp do badań anatomii języka

Chcąc korzystać z analizy statystycznej jako metody kryptoanalizy, potrzebny jest punkt odniesienia, specyficzny dla danego języka naturalnego. Celem było opracowanie podstawowych statystyk, które mogą być wykorzystane zarówno do bezpośredniej kryptoanalizy, jak i służyć do sprawdzania wyników.

Przy użyciu skryptów zawartych w bibliotece, omówionych w kolejnej części pracy, zbadano częstotliwości różnych elementów językowych. Ponieważ wykonując szyfrowanie tekstu w języku polskim, w trakcie unormalnienia, przed większością szyfrowań, litery diakrytyczne (np. ą, ę, ć) ulegają zamianie na odpowiedniki bez znaków diakrytycznych, takiemu samemu procesowi poddano badany tekst. Ujednolicono również wielkość liter i usunięto znaki interpunkcyjne, cyfry i dodatkowe znaki.

Badane były cztery najbardziej uniwersalne częstotliwości, możliwe do wykorzystania w prostej kryptoanalizie szyfrów substytucyjnych.

## Korpus języka polskiego

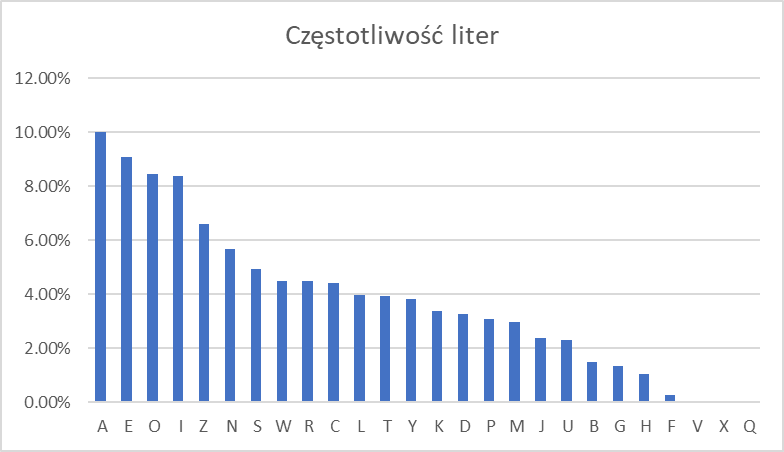
Do badania anatomii języka potrzebna jest duża, zróżnicowana baza tekstu w tym języku, tak, aby, jak najwierniej go odwzorować. Posłużyłem się do tego celu milionowym podkorpusem Narodowego Korpusu Języka Polskiego zbudowanym przez IPI PAN oraz PWN. Jest to największy publiczny zbiór tego rodzaju, zawierający łącznie tekst składający się z ponad 965 tysięcy wyrazów. Teksty zawarte w korpusie pochodzą z wielu różnych źródeł – artykułów naukowych i prasowych, wywiadów, prozy czy wpisów na forach internetowych.

Teksty przechowywane były w plikach formatu XML (zawierających oprócz czystego tekstu wiele dodatkowych informacji lingwistycznych, które nie były wykorzystane w tej pracy). Aby wyodrębnić czysty tekst, napisano konwerter do pliku tekstowego z wykorzystaniem biblioteki XML DOM minidom.

## Częstotliwości liter

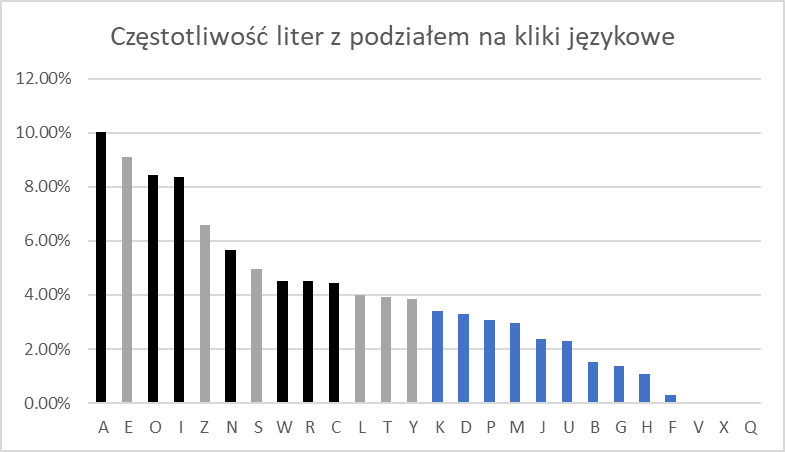
Pierwszą, najczęściej wykorzystywaną statystyką jest częstotliwość liter, oraz dwu- i trzyliterowców. Ma ona znaczenie o tyle szczególne, że może być wykorzystywana w każdej kryptoanalizie tekstowej, nawet jeśli tekst został zniekształcony poprzez pozbawienie przerw między wyrazami i znaków interpunkcyjnych. Statystyka pojedynczych liter może być wykorzystywana dodatkowo przy wyznaczaniu współczynnika kappa.

Oprócz liter używanych tradycyjnie w języku polskim teksty zawierały również litery Q, V, X, pochodzące głównie z obcojęzycznych wtrąceń lub zapożyczeń językowych. Ogólna ich częstotliwość w sumie nie przekraczała jednak 0.05% (dla porównania, najrzadsza polska litera – F, występuję z częstotliwością 0.29%), możemy więc uznać, że ich obecność, bądź jej brak w napotkanym tekście podczas kryptoanalizy, nie będzie znacząco zaburzać otrzymywanych wyników.



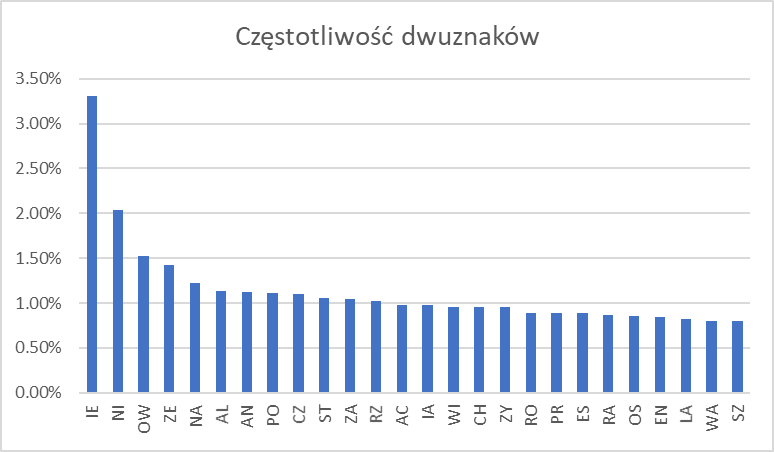
Wykres 1

Oprócz tego wyznaczono kliki językowe – grupy liter o wydzielonej od innych, podobnej częstotliwości. Taki podział ma zastosowanie w rozwiązywaniu szyfrów substytucyjnych, gdzie z dużym prawdopodobieństwem pozwala przypisać to grupie znaków o podobnej częstotliwości najpierw klikę, a następnie z niej, odpowiadający znak, znacząco usprawniając całą metodę.



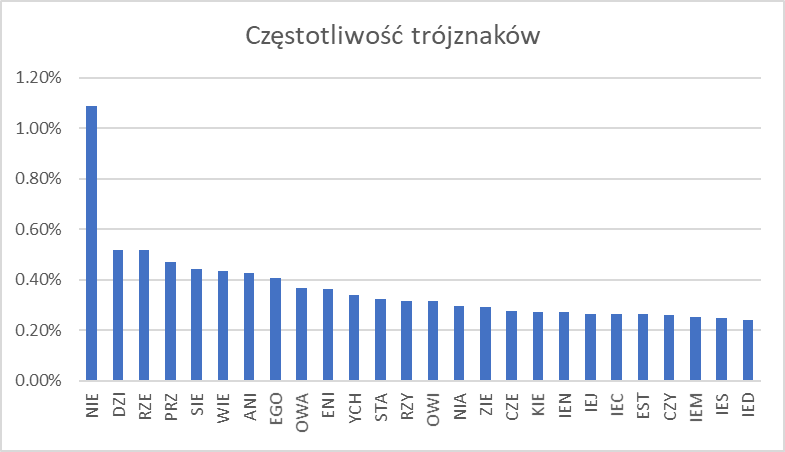
Wykres 2

Taką samą operację powtórzono dla dwuznaków i trójznaków, pominięto jednak występujące w tekście przerwy między słowami. Ze względu na bardzo duże liczebności obu tych grup, przedstawiono tylko najwyższą część (26 wyników).



Wykres 3

Dla dwuznaków widać dwa unikalne piki, bardzo wyraźnie oddzielony pik dla IE, oraz mniejszy dla NI. Kolejne trzy wartości (OW, ZE, NA) są również mocniej odcięte, natomiast między pozostałymi nie ma wyraźnego oddzielenia (różnice pomiędzy kolejnymi wartościami maksymalnie na poziomie 0.05 p.p.).

Wykres 4

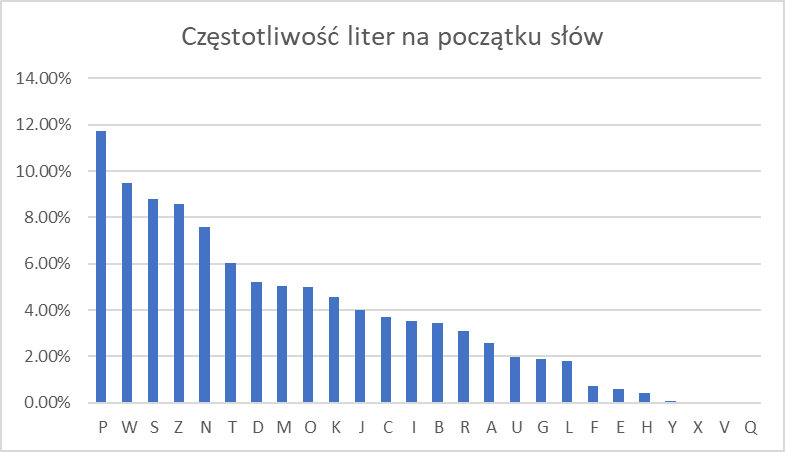
Dla trójznaków występuje jeden znaczący pik, dla NIE, oraz dwie wartości bardziej oddzielone od reszty (DZI, RZE). Pozostałe wartości nie różnią się znacząco od siebie (poniżej 0.05 p.p.).

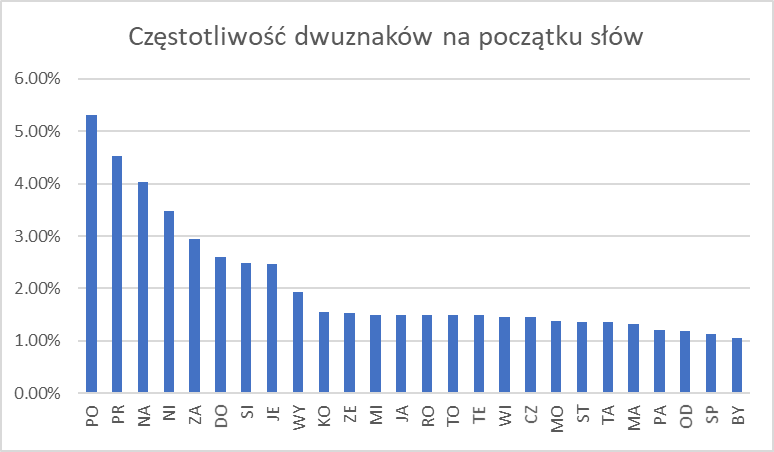
## Współczynnik kappa

## Częstotliwości liter na skraju wyrazów

Jeśli mamy do czynienia z zaszyfrowanym tekstem, w którym pozostawiono przerwy między słowami, możemy użyć dodatkowych statystyk, z tym powiązanych. Zarówno na początkach, jak i końcach wyrazów rozkład liter nie odpowiada tym ogólnym, stanowi więc to kolejny element pozwalający na lepsze dopasowanie rozszyfrowań. Oprócz tego, można wykorzystać krótkie słowa, których częstotliwość jest również bardzo nierównomierna.

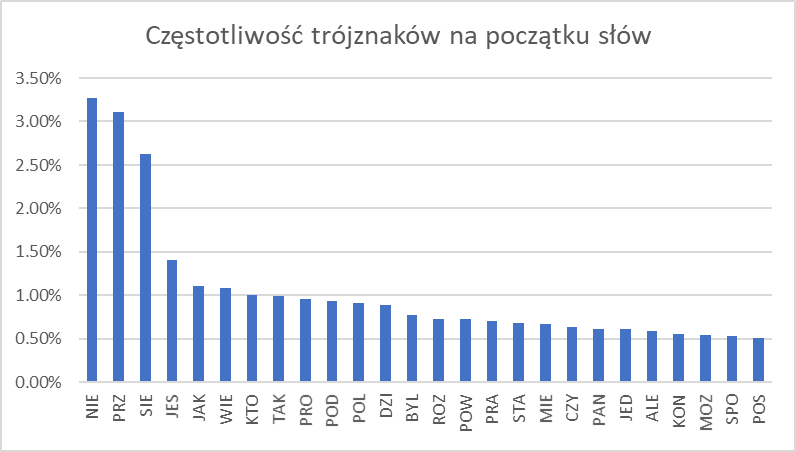
## Częstotliwość liter na początkach słów



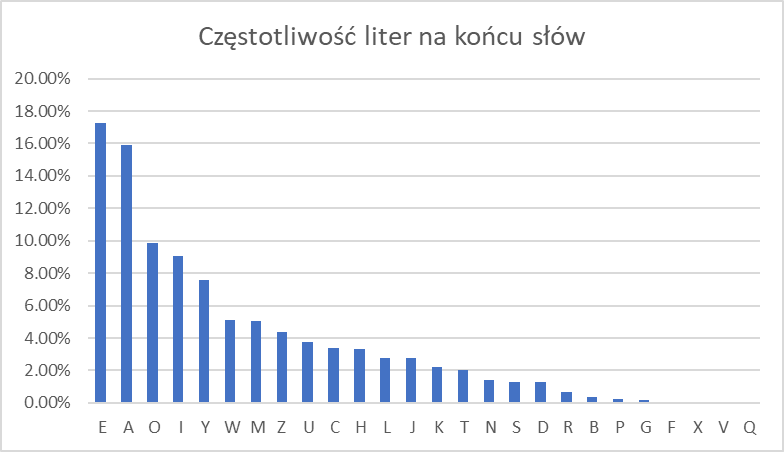


Wykres

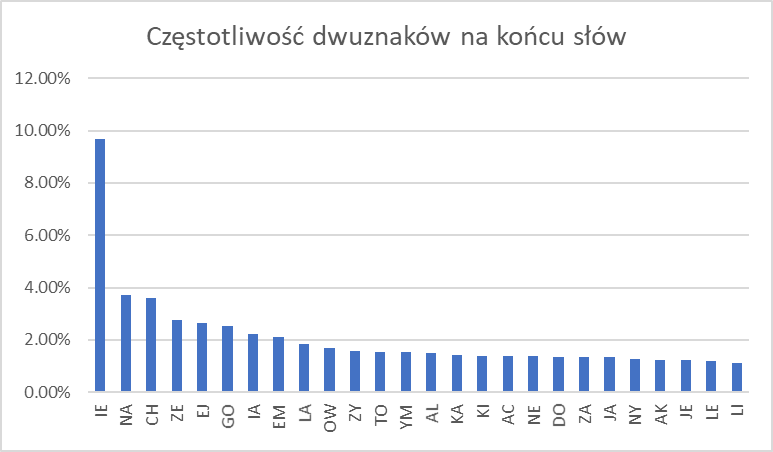
Wykres



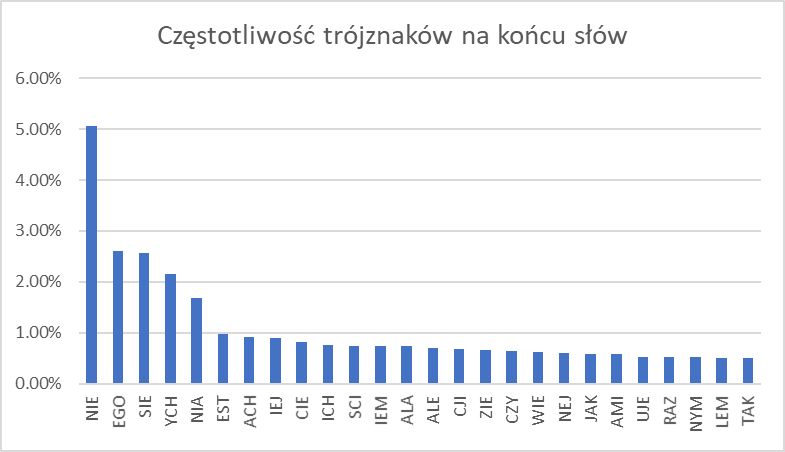
Wykres 7

Częstotliwość liter na końcach słów

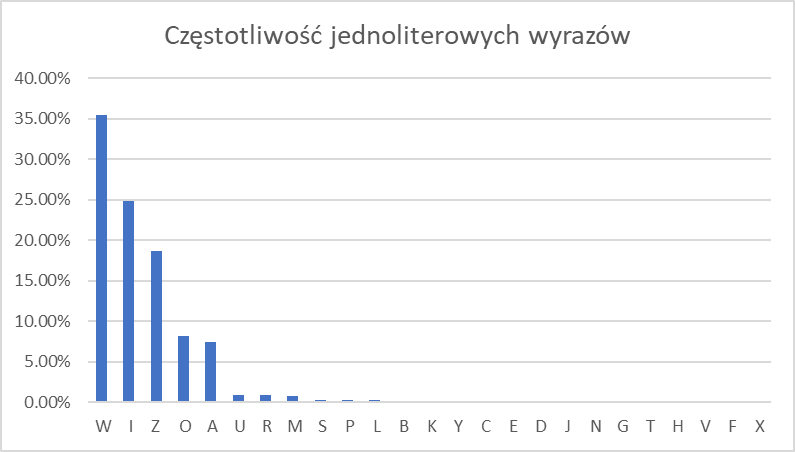
Wykres



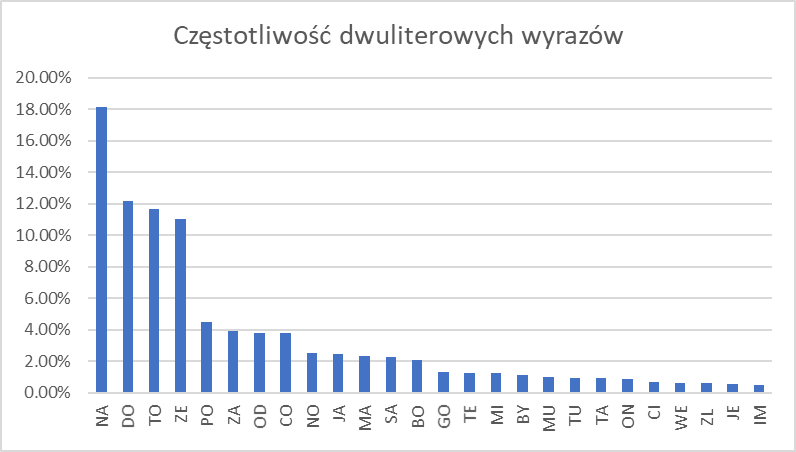
Wykres



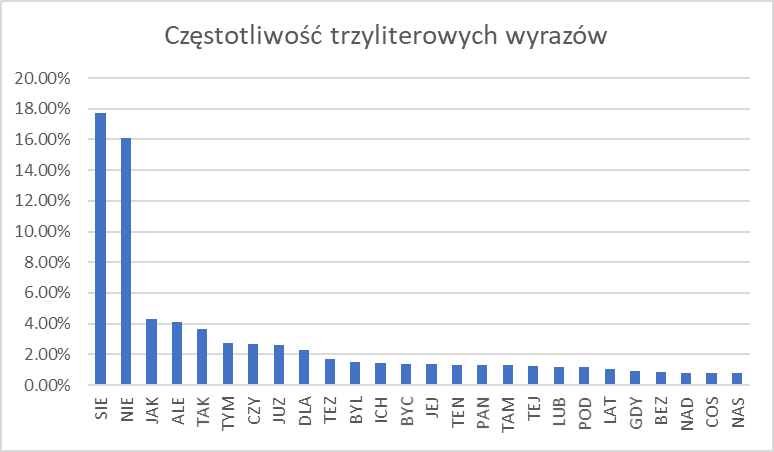
Wykres 10

Częstotliwość krótkich słów

Wykres 11



Wykres



Wykres 13

## Porównanie do języka angielskiego

# OMÓWIENIE BIBLIOTEKI

## Zakres, który obejmuje (opis, teoria)

Stworzona biblioteka ma trzy główne cele. Po pierwsze oferować pozwalające na szyfrowanie i odszyfrowywanie tekstu wybranymi szyframi. Wybrano tutaj szyfry: monoalfabetyczne substytucyjne (w kilku różnych formatach, ze względu na różny wygląd kluczy – szyfr Cezara, podstawienia w parach, czy pełna substytucja), podstawieniowy szyfr homofoniczny, polialfabetyczny szyfr substytucyjny Vigenère’a oraz szyfr poligramowy Playfair. Wszystkie szyfry zgromadzone są w katalogu ciphers

Kolejnym elementem są podstawowe narzędzia kryptologiczne służące do badania tekstu. Oprócz narzędzi służących do badania częstotliwości liter i wieloznaków w tekście (zarówno w ciągu, jak i na początku, i końcu wyrazów), a także krótkich słów. Dodatkowo istnieje funkcja pozwalająca na wyliczenie współczynnika kappa.

Ostatnia część to moduł kryptoanalityczny, który w prosty sposób pozwala na łamanie wybranych szyfrów, bez ingerencji użytkownika. W bibliotece zawarto narzędzia skonstruowane przeciwko szyfrom Cezara i Vigenère’a.

## Wygląd modułów

Aby utrzymać jednolity wygląd modułów szyfrujących wykorzystano klasę abstrakcyjną (*abstract base class, ABC*), na podstawie której tworzone były poszczególne klasy odpowiadające szyfrom. Dzięki takiemu rozwiązaniu utrzymana jest spójność nazw metod we wszystkich modułach oraz sposób działania.

Fragment kodu 1 – budowa klasy abstrakcyjnej obiektów szyfrujących

**class** MasterCipher(abc.ABC):@abc.abstractmethod  
 **def** \_\_init\_\_(self, \*args, \*\*kwargs):  
 self.cipher\_key = **None** self.decipher\_key = **None**@abc.abstractmethod  
 **def** cipher(self, plain\_text):  
 **return** plain\_text.translate(self.cipher\_key)  
@abc.abstractmethod  
 **def** decipher(self, ciphered\_text):  
 **return** ciphered\_text.translate(self.decipher\_key)

Każda klasa utworzona jako pochodna tej klasy abstrakcyjnej powinna w trakcie inicjowania obiektu (obiekty są specyficzne dla danego klucza), tworzyć dwa atrybuty – odpowiednio sformatowany klucz szyfrujący i deszyfrujący, a w ciele zawierać metody cipher, służącą do szyfrowania przekazanego tekstu, oraz decipher, o analogicznym działaniu w przeciwną stronę. Dzięki takiej budowie klas możliwe jest znaczne ograniczenie wykonywanego kodu, kiedy szyfrujemy wielokrotnie z użyciem tego samego klucza, jako że większość pracy zostanie wykonana tylko raz, w trakcie tworzenia instancji klasy.

Domyślnie obie metody zapisane są w klasie abstrakcyjnej jako wywołanie metody str.translate()na przekazanym tekście korzystając z klucza jako argumentu, ze względu na wykorzystywanie tego mechanizmu we wszystkich monoalfabetycznych szyfrach podstawieniowych (ze względu na bycie najwydajniejszą formą przekształcania tekstu w taki sposób), jednak w trakcie definiowania konkretnych klas obie te metody mogą być nadpisane własnymi. Dodatkowo każdy moduł powinien zawierać słownik nazwany library w którym znajdują się zapisane klucze i ewentualne, niezbędne argumenty do utworzenia obiektów dla przykładowych, najpopularniejszych kluczy.

Funkcje kryptoanalityczne nie mają narzuconych ścisłych ram, jednak w założeniu powinny przyjmować tylko i wyłącznie zaszyfrowany tekst, a wynik zwracać w postaci listy dwóch elementów: wartości bool mówiącej o wiarygodności rozwiązania (True dla miarodajnego) oraz listy złożonej z list wynikowych posortowanych w kolejności od najbardziej prawdopodobnego wyniku do najmniej. Same listy wynikowe powinny składać się z dwóch elementów: klucza oraz odszyfrowanego tekstu.

Ze względu na bardzo dużą różnorodność funkcji statystycznych, nie mają one podyktowanego żadnego sprecyzowanego modelu.

## Obróbka tekstu wejściowego

Przed zaszyfrowaniem tekstu wybraną metodą, najczęściej tekst powinien być ujednolicony – ze względu na to, że większość metod zakłada, że podany tekst wejściowy będzie w odpowiedniej formie. Najpopularniejsza forma polega na zamianie wszystkich liter diakrytycznych na podstawowe litery ASCII (przy zachowaniu wielkości liter). Chociaż biblioteka standardowa pythona zawiera moduł unicodedata z funkcją normalise, jego działanie nie odpowiada w pełni potrzebom biblioteki. Dlatego do tego celu wykorzystano zewnętrzną bibliotekę Unidecode, a funkcję zapisano w module collections jako normalise

Znaki specjalne oraz cyfry są zazwyczaj w trakcie szyfrowania przepisywane bez zmiany, a wielkość liter pozostaje bez zmian (choć istnieją szyfry, jak ROT47, które do szyfrowania unikalnie wykorzystują wszystkie widzialne znaki ASCII, a więc każdy znak ulegnie przekształceniu).

Oprócz tego popularną formą przekształcenia jest zamiana tekstu wejściowego na czysty ciąg dużych liter ASCII (z pominięciem znaków specjalnych, cyfr i przerw). Znacząco utrudnia to kryptoanalizę opartą na badaniu częstotliwości, ponieważ nie można wykorzystać w kryptoanalizie statystyk dotyczących podziałów na słowa. Taka funkcja również znajduje się w module collections, jako supress.

**SZYFROWANIA**

## Szyfry podstawieniowe monoalfabetyczne

Jest to najpopularniejsza historyczna grupa szyfrów, w której według zasady wybrane znaki ulegają wymianie na inne, jednak z zachowaniem zasady, że wszystkie takie same znaki tekstu czystego ulega wymianie na inne, ale takie same znaki tekstu zaszyfrowanego. Sam znak nie musi być literą czy cyfrą, używane były szyfry, w których korzystano z wymyślonych symboli, jak, np. szyfr masoński (*pigpen*), jednak taki zapis w żaden sposób nie wpływa na kryptoanalizę szyfru, chociaż obecnie stanowi pewne utrudnienie, ze względu na brak bezpośredniej kompatybilności takich symboli z komputerami. Szyfr substytucyjny w ogólnej postaci dostępny jest w module mono\_substitution. Do stworzenia obiektu szyfrującego potrzeba wprowadzić dwa argumenty – klucz w postaci listy znaków zaszyfrowanych i odpowiadających im znaków czystych. Nie ma szczegółowych wymagań dotyczących wyglądu podawanych argumentów, jedyne warunki to możliwość iteracji i równa długość, a więc można korzystać przykładowo z list, krotek, stringów, ale również z własnych klas, tak długo jak zaimplementowana jest możliwość iteracji w podanym obiekcie.

Wszystkie moduły tego rodzaju szyfrowań zbudowane są podobnie – większość kodu wykonywana jest jednokrotnie przy inicjowaniu instancji. W trakcie inicjacji obiektu zachodzi wspólna iteracja, po alfabecie zaszyfrowanym i czystym, referencyjnym, w trakcie której budowane są dwa słowniki, do szyfrowania i odszyfrowywania. Jeśli cała lista referencyjna składa się z liter o takiej samej wielkości, słownik tworzony jest w taki sposób, aby podczas szyfrowania zachować wielkość liter. Na końcu oba słowniki przekształcane są z wykorzystaniem metody str.maketrans i zapisywane jako atrybuty metody i ponownie wykorzystywane podczas wywoływania metod cipher oraz decipher. Obie te metody odwołują się bezpośrednio do odpowiadających im metod, swojej klasy abstrakcyjnej MasterCipher, gdzie wywoływana jest z kolei metoda str.translate, wykonująca całe szyfrowanie.

**from** collection **import** alph\_EN  
**from** pattern **import** MasterCipher  
  
  
**class** MonoSubstitution(MasterCipher):**def** \_\_init\_\_(self, enc\_key, reference=alph\_EN):  
 **assert** len(enc\_key) == len(reference)  
 cipher\_dict, decipher\_dict = {}, {}  
 **if** reference.islower() **or** reference.isupper():  
 **for** x, y **in** zip(enc\_key, reference):  
 **if** x != y:  
 cipher\_dict[y.lower()] = x.lower()  
 cipher\_dict[y.upper()] = x.upper()  
 decipher\_dict[x.lower()] = y.lower()  
 decipher\_dict[x.upper()] = y.upper()  
 **else**:  
 **for** x, y **in** zip(enc\_key, reference):  
 cipher\_dict[y] = x  
 decipher\_dict[x] = y  
 self.cipher\_key = str.maketrans(cipher\_dict)  
 self.decipher\_key = str.maketrans(decipher\_dict)

**def** cipher(self, plain\_text):  
 **return** super().cipher(plain\_text)  
  
 **def** decipher(self, ciphered\_text):  
 **return** super().decipher(ciphered\_text)

Fragment kodu 2.1 – Przykład szyfru substytucyjnego – klasa monosubstitution

Istnieją jednak różne metody zapisywania klucza i jego interpretacji, choć końcowo każdą z tych metod można sprowadzić do powyższej postaci ogólnej. Najprostszym przykładem jest szyfr Cezara, w którym klucz jest pojedynczą liczbą, odpowiadającą odległości przesunięcia alfabetu szyfrowego względem alfabetu czystego (z uwzględnieniem modulo długości klucza – co oznacza, że również ujemne wartości zostaną przekonwertowane jako obrót w przeciwną stronę). Istnieje również możliwość przekazania własnego alfabetu referencyjnego, co pozwala na, np. użycie szyfru ROT47, w którym alfabet składa się z 94 widzialnych znaków ASCII. Obiekt ten znajduje się w module Caesar.

Inny rodzaj szyfrowania, to szyfrowanie w parach, gdzie klucz podany jest jako ciąg par unikalnych liter, a w trakcie szyfrowania litery w parze są wzajemnie wymieniane, np. GA-DE-RY-PO-LU-KI. Taki sposób zapisu szyfru ma przewagę pod względem łatwości zapamiętywania używanego klucza, oraz jest samo odwracalny, to znaczy, że procesy szyfrowania i odszyfrowywania są identyczne (a więc szyfrując zaszyfrowany tekst ponownie z użyciem tego samego klucza, otrzymujemy tekst czysty).

Oprócz tego istnieją szyfry wykorzystujące kombinacje symboli, aby przedstawiać znaki, tak jak, np. w kodzie Morse’a, gdzie wszystkie znaki zapisywane są za pomocą kombinacji dwóch znaków: krótkich (reprezentowanych graficznie przez kropki •) i długich (przedstawianych jako kreski ―), oraz przerw (zapisywanych w postaci ukośnika /). Ponieważ metoda str.translate() obsługuje tylko operacje na pojedynczych znakach. Pojedyncze zbitki znaków, odpowiadające jednej literze w zaszyfrowanym tekście, muszą być wtedy od siebie wyraźnie oddzielone w jednolity sposób, aby po zaszyfrowaniu wiadome było, gdzie kończy się jedna litera, a zaczyna druga (inna możliwość zakłada, że wszystkie zbitki są równej długości, wtedy nie ma potrzeby ich rozdzielania).

## Homofoniczny szyfr podstawieniowy

Wszystkie z powyższych szyfrów są podatne na kryptoanalizę w oparciu o analizę częstotliwości (szczególnie jeśli wiadomo, że użyty został któryś z prostszych sposobów szyfrowania), stąd aby utrudnić to zadanie zaczęto dodatkowo modyfikować podstawową metodę. Jedną z takich zmian było zastosowanie większych słowników, w których jedną literę reprezentuje kilka różnych symboli (lub zbitek symboli, liter -unikalnych między poszczególnymi literami), spośród których każdorazowo wybierany jest jeden losowy podczas napotkania na daną literę w trakcie szyfrowania.

Przykładowo, jeśli w utworzonym słowniku, każda z liter będzie miała proporcjonalną ilość swoich odpowiedników do jej częstotliwości występowania w języku, to analiza częstotliwości będzie bezużyteczna, ponieważ wszystkie zbitki występować będą z jednakową częstotliwością. Jeśli dodatkowo wszystkie zbitki będą o jednakowej długości i nie będą od siebie oddzielane separatorem to kolejnym problemem przy próbie kryptoanalizy będzie ustalenie długości tych zbitek.

Minusem tego podejścia jest konieczność utrzymania względnie dużego słownika, oraz wydłużanie tekstu wiadomości (jeśli każda litera zostanie zastąpiona trzyliterowcem, proporcjonalnie długość wiadomości wzrośnie trzykrotnie).

## Polialfabetyczny szyfr substytucyjny

**Vigenere**

**Playfair**

**METODY KRYPTOANALITYCZNE**

**PRZYKŁADY ZASTOSOWAŃ**

**BIBLIOGRAFIA**

Friedrich L. Bauer - Decrypted Secrets Methods and Maxims of Cryptology (2007)

Hananto, April & Solehudin, Arip & Susilo Yuda Irawan, Agung & Priyatna, Bayu. (2019). Analyzing the Kasiski Method Against Vigenere Cipher. 6. 1-9. 10.29126/23942231/IJCT-V6I6P2.

Dhavare, Amrapali, "EFFICIENT ATTACKS ON HOMOPHONIC SUBSTITUTION CIPHERS" (2011). *Master's Projects*. 207. DOI: https://doi.org/10.31979/etd.u44a-fh82 https://scholarworks.sjsu.edu/etd\_projects/207