Aplikace pro kryptoanalýzu klasických šifer

Matěj Ženčák

Bakalářská práce
2022

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Matěj Ženčák**
Osobní číslo:	**A19159**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**Prezenční**
Téma práce:	**Aplikace pro kryptoanalýzu klasických šifer**
Téma práce anglicky:	**Application for Cryptoanalysis of Classical Cryptography**

Zásady pro vypracování

1. Nastudujte a rozepište problematiku spojenou s klasickou kryptografií a možnosti její kryptoanalýzy.
2. Vyberte vhodné algoritmy pro implementaci.
3. Zvolte vhodné technologie pro implementace desktopové aplikace.
4. Implementujte desktopovou aplikaci s vámi zvolenými algoritmy.
5. Aplikaci otestujte a prezentujte výsledky a vhodně vyhodnoťte.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BAUMSLAG, Gilbert, Benjamin FINE, Martin KREUZER a Gerhard ROSENBERGER. A Course in Mathematical Cryptography. Berlin/Boston: De Gruyter, 2015, 1 online zdroj. De Gruyter Textbook. ISBN 9783110386165. Dostupné také z: https://proxy.k.utb.cz/login?url=https://www.degruyter.com/openurl?genre=book&isbn=9783110372779
2. JAŠEK, Roman. Informační a datová bezpečnost. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta managementu a ekonomiky, 2006, 140 s. ISBN 8073184567.
3. KNUDSEN, Lars a Matthew ROBSHAW. The block cipher companion. Berlin: Springer, c2011, xiv, 267 s. Information security and cryptography. Dostupné z: doi:9783642173424
4. SINGH, Simon. Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii. Praha: Dokořán, 2003, 382 s. Aliter. ISBN 8072034995.
5. FOJTOVÁ, Lucie a Karel BURDA, 2010. Softwarová podpora výuky klasické kryptoanalýzy: Software support of education in classical cryptoanalysis. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií.

Vedoucí bakalářské práce: **Ing. Petr Žáček, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **15. ledna 2022**
Termín odevzdání bakalářské práce: **20. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

Matěj Ženčák v.r.
podpis studenta

ABSTRAKT

Bakalářská práce se zabývá možnostmi kryptoanalýzy v oblasti klasické kryptografie. V bakalářské práci jsou vybrány vhodné šifrovací algoritmy, které jsou v teoretické části popsané a v praktické části i vhodně implementované. Hlavní část práce se zaměřuje na studium, výběr a implementaci vhodných kryptoanalytických metod k analýze a prolamování vybraných šifer klasické kryptografie. Výsledkem práce je souhrnná desktopová aplikace schopná šifrování, dešifrování, analýzy a prolamování vybraných šifer. Práce obsahuje i ukázky a srovnání různých útoků na vybrané šifrovací algoritmy.

Klíčová slova: Kryptologie, Klasická kryptografie, Kryptoanalýza, Desktopová aplikace, Python

ABSTRACT

This Bachelor's thesis focuses on the cryptanalysis of classical ciphers. The goal of this thesis was to choose suitable ciphers and cryptanalytic methods and implement them into a desktop application. The output of the thesis is a desktop application capable of encrypting, decrypting, analyzing, and breaking chosen ciphers using appropriate methods, showcase and comparison of different attack methods against chosen ciphers.

Keywords: Cryptography, Classical Cryptography, Cryptanalysis, Application, Python

Chtěl bych poděkovat Ing. Petru Žáčkovi, Ph.D. za odborné vedení, vstřícnost při konzultacích a cenné rady při zpracování této práce. Dále bych chtěl poděkovat rodině a kamarádům za podporu během studia.

OBSAH

ÚVOD **8**
I TEORETICKÁ ČÁST **9**
1 ZÁKLADNÍ POJMY **10**
2 ROZDĚLENÍ ŠIFER **11**
2.1 SUBSTITUČNÍ ŠIFRY 11
2.1.1 Monoalfabetická substituce 11
2.1.2 Polyalfabetická substituce 12
2.1.3 Polygrafická substituce 12
2.2 TRANSPOZIČNÍ ŠIFRY 12
2.3 BLOKOVÉ ŠIFRY 12
2.4 PROUDOVÉ ŠIFRY 12
2.5 SYMETRICKÉ ŠIFRY 13
2.6 ASYMETRICKÉ ŠIFRY 13
3 VYBRANÉ ŠIFRY KLASICKÉ KRYPTOGRAFIE **14**
3.1 SUBSTITUČNÍ ŠIFRY 14
3.1.1 Afinní šifra 14
3.1.2 Playfair 16
3.1.3 ROT13 18
3.1.4 Vigenèrova šifra 18
3.2 TRANSPOZIČNÍ ŠIFRY 21
3.2.1 ADFGVX 21
3.2.2 Railfence 23
3.2.3 Route 24
3.2.4 Scytale 25
4 MOŽNOSTI KRYPTOANALÝZY **27**
4.1 FREKVENČNÍ ANALÝZA 27
4.2 KASISKÉHO METODA 29
4.3 INDEX KOINCIDENCE 30
4.4 ÚTOK HRUBOU SILOU 31
4.5 SLOVNÍKOVÝ ÚTOK 31
II PRAKTICKÁ ČÁST **32**
5 NÁSTROJE **33**
5.1 PYTHON 33
5.2 VOLBA VÝVOJOVÉHO PROSTŘEDÍ 33
6 ŠIFROVACÍ APLIKACE **34**
6.1 FUNKCIONALITY PROGRAMU 34
6.1.1 Šifrování 34
6.1.2 Analýza 35
6.1.3 Prolamování 37

6.2 STRUKTURA PROGRAMU 38
6.3 ALGORITMY PRO ANALÝZU 39
6.4 ALGORITMY PRO PROLAMOVÁNÍ 40
6.4.1 Útok hrubou silou 40
6.4.2 Útok frekvenční analýzou 42
6.4.3 Slovníkový útok 43
7 VÝSLEDKY ANALÝZY 45
ZÁVĚR 48
SEZNAM POUŽITÉ LITERATURY 49
SEZNAM PŘEVZATÉHO KÓDU 52
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK 53
SEZNAM OBRÁZKŮ 54
SEZNAM TABULEK 55
SEZNAM PŘÍLOH 56

ÚVOD

Kryptologie při svém vývoji prošla výraznými změnami. Od vědní disciplíny založené na matematice a algoritmizaci využívané pro utajení obsahu zpráv pomocí tužky a papíru, po disciplínu, která je v době moderních výpočetních technologií neodlučitelně spojována s informatikou. Samotné šifrování hrálo vždy důležitou roli od utajování vojenských zpráv po zajištění bezpečnosti prvních počítačových sítí. Společně s kryptologií se tak vyvíjely její podskupiny – kryptografie, kryptoanalýza a steganografie.

Šifrovací algoritmy navržené pro šifrování pomocí tužky a papíru (klasická kryptografie) s příchodem výpočetní techniky přestávaly být bezpečnou alternativou. Kryptografii tak lze rozdělit na klasickou kryptografii využívající pro výpočetní techniku snadno prolomitelné metody, a moderní kryptografii využívající schopnosti moderní výpočetní techniky pro návrh matematicky složitějších šifrovacích algoritmů. V porovnání lze říci, že klasická kryptografie spoléhá na utajení algoritmu, kdežto moderní staví na matematických základech.

Cílem této bakalářské práce bylo popsat vybrané šifrovací algoritmy klasické kryptografie, možnosti jejich kryptoanalýzy a prolamování, implementace vybraných šifrovacích algoritmů do desktopové aplikace s možností jejich kryptoanalýzy a pokusit se prolomit vybrané algoritmy. Aplikace umožňuje využívat principů klasické kryptografie a kryptoanalýzy, shrnutých do jedné univerzální aplikace s vysokou mírou automatizace.

Teoretická část práce obsahuje rozdělení klasické kryptografie a popis jejích vlastností, principy fungování šifer vybraných k implementaci do desktopové aplikace, a metody klasické kryptoanalýzy využívané pro praktickou část práce. Praktická část popisuje nástroje zvolené pro vytvoření aplikace, popis funkcí této aplikace, strukturu aplikace a konkrétní využití jednotlivých kryptoanalytických metod. V této části jsou také zobrazeny výsledky kryptoanalýzy pro jednotlivé šifry a jejich zhodnocení.

I. TEORETICKÁ ČÁST

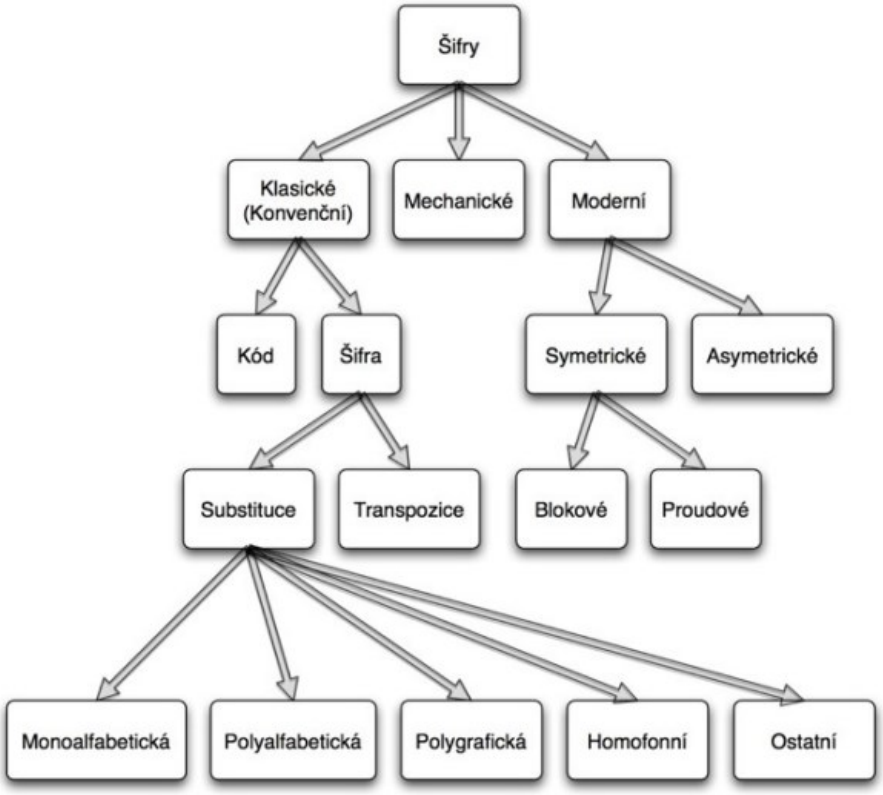
1 ZÁKLADNÍ POJMY

- *Kryptologie* je věda o šifrách, dělí se na části – kryptografie, kryptoanalýza, luštění šifer.
- *Kryptografie* je věda o vytváření šifrovacích algoritmů a nástrojů pro zašifrování otevřeného textu – šifrovací klíče.
- *Kryptoanalýza* je věda zkoumající vlastnosti šifrovacích klíčů, šifrovaného textu a otevřeného textu.
- *Šifrování* je proces transformace čitelného otevřeného textu na často zdánlivě nesmyslný šifrovaný text za použití šifrovacího klíče a šifrovacího algoritmu.
- *Dešifrování* je proces transformace šifrovaného textu na otevřený text za použití správného šifrovacího klíče a šifrovacího algoritmu.
- *Luštění šifer* je proces získání otevřeného textu ze zašifrovaného textu. Při úspěšném luštění šifry je možné odhalit šifrovací algoritmus a šifrovací klíč pro snadné dešifrování dalších, stejně zašifrovaných zpráv.
- *Otevřený text* je nezašifrovaný text. Jedná se o vstup do šifrovacího a výstup z dešifrovacího algoritmu.
- *Šifrovaný text* je text zašifrovaný šifrovacím algoritmem.
- *Šifrovací algoritmus* také označován jako šifra, je postup transformace otevřeného textu na šifrovaný text za použití šifrovacího klíče.
- *Šifrovací klíč* je znak, číslo nebo jejich posloupnost či kombinace. Je potřebný k určení pořadí úkonů vedoucí k zašifrování nebo dešifrování textu.
- *Abeceda* je množina znaků povolených pro určitý šifrovací algoritmus. Otevřený text musí být složen pouze z těchto znaků. Zašifrovaný text může mít odlišnou abecedu v závislosti na šifrovacím algoritmu.

[1][2][3]

2 ROZDĚLENÍ ŠIFER

Tato práce se zabývá šiframi, které jsou označovány jako klasické. V této kapitole budou popsány principy jednotlivých typů šifer.



Obrázek 1.: Členění kryptografie [1]

2.1 Substituční šifry

Při šifrování substituční šifrou je určitý znak nebo skupina znaků nahrazena jiným znakem nebo skupinou znaků v přípustné abecedě, tento proces se nazývá substituce. Vlastnosti substituce udává šifrovací klíč. Nevýhodou substituční šifry je neměnné pořadí znaků v textu, v mnoha případech není pozměněn ani počet znaků, což činní substituční šifry náchylné k prolomení útoky hrubou silou nebo frekvenční analýzou.

2.1.1 Monoalfabetická substituce

Při šifrování monoalfabetickou substituční šifrou platí předpis pro substituci daný klíčem pro celou zprávu. Tohoto principu využívají šifry jako ATBASH, Caesarova šifra, Afinní šifra nebo ROT13. [4]

2.1.2 Polyalfabetická substituce

Polyalfabetické substituční šifry užívají několika abeced pro substituci, pro zašifrování jedné zprávy. V ideálním případě je změna substitučních abeced náhodná, pro znesnadnění rozluštění šifrovaného textu a klíče. [5] Příkladem je Vigenèrova šifra, která byla po 300 let považována za neprolomitelnou.[6]

2.1.3 Polygrafická substituce

Pakliže jsou znaky otevřeného textu rozděleny do skupin, a každá skupina je následně substituována jedním, nebo více znaky, jedná se o polygrafickou substituci. Příkladem je šifra Playfair, která substituuje dvojice znaků. [7]

2.2 Transpoziční šifry

Pro zašifrování dat mění transpoziční šifry pozici znaků otevřeného textu dle příslušného algoritmu a klíče. Jak v šifrovaném, tak v otevřeném textu tak budou stejné znaky. Transpozice je často využívána společně se substitucí pro ztížení kryptoanalýzy, oproti jednoduché transpozici či substituci jsou tyto šifry mnohem bezpečnější.

2.3 Blokové šifry

Blokové šifry rozdělují otevřený text na bloky dat o stejném počtu bitů, následně jsou data zašifrována postupně po jednotlivých blocích. Velikost bloků dat neovlivňuje bezpečnost zašifrovaných dat, často je však velikost bloků 64 bitů. Taková velikost je méně náchylnější k prolomení slovníkovým útokem a je jednodušší pro implementaci jakožto násobek 8. pro moderní procesory. [8] V případě že data nelze rozdělit na bloky o ekvivalentní délce, jsou data rozdělena bez rozdílu a poslední blok je doplněn opakujícími se bity po dosažení požadované velikosti bloku.

2.4 Proudové šifry

Šifry tohoto typu šifrují otevřený text po jednotlivých bitech nebo bajtech. Proudové šifry jsou oproti blokovým rychlejší a vhodné nejen pro šifrování zpráv, ale i pro šifrování telekomunikačních přenosů, zejména pak, dochází-li ke ztrátám přenosu.[9]

2.5 Symetrické šifry

Symetrické šifry využívají stejný klíč pro zašifrování a dešifrování. Při komunikaci tak obě strany potřebují stejný klíč, ten nelze sdílet v nezašifrované komunikaci, neboť při odhalení klíče jedné strany je komunikace pro případného útočníka dešifrovatelná. Symetrické šifry jsou však méně náročné na výpočetní operace a poskytují tak rychlejší šifrování a dešifrování. [9] Moderní šifrovací algoritmy častěji užívají nesymetrického šifrování – rozdílného klíče pro šifrování a dešifrování. Pro útočníka je tak získání jednoho klíče méně výhodné, oproti získání klíče symetrické šifry. Všechny šifry implementované v této bakalářské práci se řadí mezi šifry symetrické.[10]

2.6 Asymetrické šifry

Asymetrické šifrování odstraňuje problém odhalení jednoho sdíleného klíče. Asymetrické šifrování užívá dvou rozdílných klíčů, jeden pro zašifrování zprávy (veřejný klíč), druhý pro dešifrování zprávy (soukromý klíč). Veřejný klíč není třeba tajit, protože jej útočník nemůže využít k dešifrování zprávy, a lze ho tak přeposílat i pomocí nezabezpečené komunikace. Ze znalosti soukromého klíče lze odvodit veřejný klíč, ovšem ne naopak. Pro komunikaci dvou stran má obvykle každá z nich vlastní soukromý i veřejný klíč, s protistranou vždy sdílí jen klíč veřejný, což zajistí zabezpečenou komunikaci. Nevýhodou asymetrického šifrování je vyšší výpočetní náročnost celého procesu, příkladem asymetrické šifry je například moderní šifra RSA. [9][10]

3 VYBRANÉ ŠIFRY KLASICKÉ KRYPTOGRAFIE

V této kapitole budou popsány vybrané šifrovací algoritmy, jejich principy a příklady. Jednotlivé algoritmy jsem vybíral pro co nejlepší zastoupení substitučních a transpozičních šifer.

3.1 Substituční šifry

3.1.1 Afinní šifra

Tato šifra se řadí mezi jednoduché substituční šifry. Afinní šifra umožňuje více možností transformace než Caesarova šifra a je tak složitější k prolomení jednoduchou kryptoanalýzou. Ačkoliv počet transformací zajistí určitou odolnost vůči útokům hrubou silou, je náchylná k útokům frekvenční analýzou. Místo jednoduchého posuvu využívá modulární aritmetiku. [11]

Každému znaku v abecedě je přiřazeno číslo, pro zašifrování každého znaku je pak využito transformace:

$$E(x) = (ax + b) * mod(m)$$

Kde:

- *E(x)* je zašifrovaný znak s indexem *x*
- *a* je první část klíče
- *x* je index znaku který zašifrujeme
- *b* je druhá část klíče
- *m* je délka abecedy

První část klíče *a* může být pouze prvočíslo s délkou abecedy *m*. Za předpokladu, že užíváme abecedu anglických znaků s délkou 26 znaků, můžeme za *a* zvolit kterékoliv z čísel: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 a 25.

Pomocí tohoto algoritmu zašifrujeme zprávu *AFFINE* při volbě parametrů *a=5, b=9*.

$$A \Rightarrow 0 \Rightarrow 5*0 + 9 \bmod (26) = 9 \Rightarrow J$$

$$F \Rightarrow 5 \Rightarrow 5*5 + 9 \bmod (26) = 8 \Rightarrow I$$

$$F \Rightarrow 5 \Rightarrow 5*5 + 9 \bmod (26) = 8 \Rightarrow I$$

$$I \Rightarrow 8 \Rightarrow 5*8 + 9 \bmod (26) = 23 \Rightarrow X$$

$$N \Rightarrow 13 \Rightarrow 5*13 + 9 \bmod (26) = 22 \Rightarrow W$$

$$E \Rightarrow 4 \Rightarrow 5*4 + 9 \bmod (26) = 3 \Rightarrow D$$

Tabulka 1.: Nahoře abeceda otevřeného textu, dole abeceda zašifrovaná klíčem s parametry a=5, b=9.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	O	T	Y	D	I	N	S	X	C	H	M	R	W	B	G	L	Q	V	A	F	K	P	U	Z	E

Pro dešifrování je využito následující transformace:

$$D(x) = a^{-1}(x - b) mod(m)$$

Kde:

- *D(x)* je dešifrovaný znak s indexem *x*
- a^{-1} je multiplikativní inverze *a* [12]
- *x* je index znaku který zašifrujeme
- *b* je druhá část klíče
- *m* je délka abecedy

[13] [14]

3.1.2 Playfair

Šifra Playfair je polyalfabetická substituční polní šifra známá zejména pro svou jednoduchost a prvenství v šifrování zpráv pomocí bigramů – rozdělení otevřeného textu na dvojici znaků, jejichž substituce probíhá na stejném principu pro celou dvojici. Díky své rychlosti a jednoduchosti byla užívána za první i druhé světové války. K zašifrování i dešifrování není třeba žádná výpočetní technika, znalost modulární aritmetiky nebo složitých algoritmů, je však relativně snadno prolomitelná, zejména pokud je známa část klíče nebo otevřeného textu. [15]

Pro šifrování je třeba sestavit tabulku o velikosti 5x5 pro 25 znaků, aby mohla být abeceda vepsána do tabulky je třeba jeden znak odstranit, zpravidla je odstraněno písmeno *j*, v případě, že se *j* vyskytuje v otevřeném textu je nahrazeno písmenem *i*. Do tabulky je následně zapsáno do řádků klíčové slovo, v němž se nesmí opakovat jednotlivé znaky. Po doplnění klíčového slova se doplní zbytek abecedy v libovolném pořadí, kromě znaků obsažených v klíčovém slově, to zajistí, že každý znak je v tabulce pouze jednou.

Otevřený text je rozdělen do dvojic znaků – bigramů. V případě lichého počtu znaků v otevřeném textu je doplněn o další znak, zpravidla *z*. Každý bigram musí tvořit rozdílné znaky, tvoří-li bigram dva stejné znaky, je mezi ně vložen jiný znak např. *z*. [7]

Např. otevřený text *HELLOTHERE* bude převeden na následující bigramy:

HE LZ LO TH ER EZ. Pro zašifrování použijeme klíč *CIPHER*.

Tabulka 2.: Sestavená tabulka pro šifrování bigramů.

C	I	P	H	E
R	A	B	D	F
G	K	L	M	N
O	Q	S	T	U
V	W	X	Y	Z

Substituci v playfairově šifře lze provádět třemi způsoby v závislosti na pozici znaků bigramu v tabulce.

1. Pokud jsou znaky bigramu na stejném řádku
   Každý znak bigramu se substituován znakem napravo od něj. V případě že je šifrován poslední znak řádku, je substituován prvním znakem řádku.
   První bigram je tedy šifrován: *H→E*, *E→C*.

Tabulka 3.: Šifrování prvního bigramu dle řádkové substituce.

C	I	P	H	E
R	A	B	D	F
G	K	L	M	N
O	Q	S	T	U
V	W	X	Y	Z

2. Pokud jsou znaky bigramu ve stejném sloupci

   Každý znak bigramu je substituován znakem na pozici pod ním. V případě že je šifrován poslední znak sloupce, je substituován prvním znakem sloupce.

   Čtvrtý bigram je tedy šifrován: *T*→*Y*, *H*→*D*.

Tabulka 4.: Šifrování čtvrtého bigramu dle sloupcové substituce.

C	I	P	H	E
R	A	B	D	F
G	K	L	M	N
O	Q	S	T	U
V	W	X	Y	Z

3. Pokud ani jedna z předchozích možností neplatí

   Nesdílí-li znaky řádek ani sloupec, je vytvořen pomyslný obdélník nebo čtverec, kdy znaky bigramu tvoří úhlopříčně rohy tohoto útvaru, znaky jsou substituovány znaky ve zbývajících rozích čtverce/obdélníku.

   Druhý bigram je tedy šifrován: L→*N*, *Z*→*X*.

Tabulka 5.: Šifrování druhého bigramu dle úhlopříčné substituce.

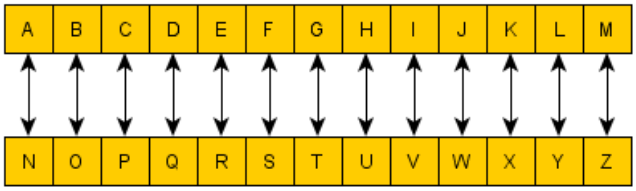
C	I	P	H	E
R	A	B	D	F
G	K	L	M	N
O	Q	S	T	U
V	W	X	Y	Z

Pro dešifrování je použit stejný postup v opačném směru. Příjemce musí znát klíčové slovo pro doplnění do tabulky, popř. pořadí znaků abecedy – pokud byla pozměněna. Šifrovaný text je opět rozdělen na bigramy, tentokrát již bez potřeby doplňování znaků, a rozšifrován dle výše uvedených možností substituce. Substituce se nyní provádí v opačném směru tedy

při řádkové substituci dochází k posuvu vlevo, při sloupcové substituci k posunu vzhůru a při úhlopříčné substituci k záměně úhlopříček.

3.1.3 ROT13

Řazena mezi monoalfabetické substituční šifry, ROT13 je často přirovnávána k Caesarově šifře. Zatímco Caesarova šifra provádí substituci s rotací o zvolené délce v abecedě, ROT13 provádí rotaci o délce 13 znaků. Díky jednoduchému algoritmu ROT13 nepotřebuje klíč a je jednoduchá na pochopení i implementaci, je však velmi snadno prolomitelná a není vhodná k šifrování důležitých zpráv. [16]



Obrázek 2.: Substituce v ROT13[17]

Pro zašifrování otevřeného textu: *HELLOWORLD* lze použít Obrázek č. 2. pro získání šifrovaného textu: *URYYBJBEYQ*.

ROT13 lze stejně jako Caesarovu šifru prolomit pomocí frekvenční analýzy, jediným rozdílem je že Caesarova šifra má celkem 25 možných posunutí kdežto ROT13 pouze jedno. V případě, že útočník ví, že se jedná o ROT13 či Caesarovu šifru, je možné použít útok hrubou silou – vyzkoušení všech možných posunů. Jeden způsob, jak zvýšit bezpečnost ROT13 šifry by bylo přidat do abecedy náhodné, avšak unikátní znaky.

3.1.4 Vigenèrova šifra

Známá jako polyalfabetická substituční šifra, Vigenèrova šifra byla neprolomena přes 300 let, jedná se o kombinaci Caesarových šifer, kdy celý otevřený text byl posunut o určitou vzdálenost v abecedě, Vigenèrova šifra posouvá každý znak v otevřeném textu odlišným posunem – tedy každý znak je šifrován dle jinak uspořádané abecedy. [18]

KEYWORD LETTERS \ PLAINTEXT LETTERS	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Obrázek 3.: Vigenèrův čtverec pro šifrování [18]

Pro každý znak abecedy je určená jiná substituční abeceda, šifrování probíhá po jednotlivých znacích, zvolený sloupec odpovídá právě šifrovanému znaku otevřeného textu a řádek odpovídá znaku klíčového slova na stejném indexu. Klíčové slovo musí mít stejnou délku jako otevřený text, z praktických důvodu je však volen klíč kratší, který je opakován, aby dosáhl požadované délky. Např. pro otevřený text: *HELLOWORLD* a zvolený klíč *KEY* je třeba klíč upravit na: *KEYKEYKEYK*.

Tabulka 6.: Vigenèrův čtverec pro šifrování HELLOWORLD s klíčem KEY

-	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Na obrázku č. 3. je patrné, že opakující se znak je možno zašifrovat dvěma způsoby v závislosti na klíči tedy: *I→M* při klíči *E* a *I→W* při klíči *O*. Stejně tak mohou být náhodou dva odlišné znaky být zašifrovány na stejný znak.

V tabulce č. 6. je znázorněno šifrování otevřeného textu: *HELLOWORLD* s klíčem *KEY*, zašifrovaný text bude: *RIJVSUYVJN*.

Při užití více abeced pro jednu zprávu je tedy frekvenční analýza nepoužitelná, útok hrubou silou je možný, avšak velmi nepraktický a náročný, počet možných klíčů je m^n kde m je délka otevřeného textu a n počet unikátních znaků klíče [19]. Pro prolomení šifrované zprávy:

CVHSXAKJESIRKDCHMWBVDAM o délce 23. znaků a délce klíče 12. znaků je počet možných klíčů téměř 22 biliard.

Pro prolomení šifry však lze využít Kasiského analýzu kterou se budeme zabývat později. Frekvenční analýza jako taková si s šifrou neporadí, nicméně v kombinaci s Kasiského analýzou není obtížné šifru prolomit. [19]

3.2 Transpoziční šifry

3.2.1 ADFGVX

Šifra ADFGVX je variantou původní polní šifry ADFGX používané německou armádou během první světové války. Jedná se o transpoziční šifru, která kombinuje Polybiův čtverec [20] a sloupcovou transpozici.

Šifra je založena na sestavení tabulky o 6. sloupcích a 6. řádcích, kdy jednotlivým sloupcům a řádkům jsou přiřazena písmena *A, D, F, G, V, X*. Vzniklá tabulka velikosti 6x6 pojme 36 znaků, tedy 26 písmen abecedy a číslice 0 až 9, původní varianta ADFGX utvořila tabulku o pouze 25 znacích, nebylo s ní tedy možno šifrovat číslice, což vzhledem k jejímu využití na bojišti byla kritická vlastnost. Znaky abecedy jsou do tabulky doplňovány v náhodném pořadí, samotná abeceda je tak dalším klíčem. Při použití ADFGX je třeba zkrátit abecedu o jeden znak kvůli limitu 25 znaků, nejčastěji se sjednotí znaky *i* a *j* čímž je splněna maximální povolená délka při zachování akceptovatelné čitelnosti šifrované a následně dešifrované zprávy. [21]

Pro zašifrování zprávy *ATTACKATONCE* zvolíme náhodně generovanou abecedu o délce 36 znaků a klíč *CHARGE*.

Tabulka 7.: Náhodná abeceda pro získání druhé tabulky ADFGVX.

	A	D	F	G	V	X
A	8	K	D	6	U	B
D	9	F	2	Q	4	R
F	J	Z	H	O	P	7
G	5	N	1	E	X	W
V	0	Y	C	L	G	A
X	T	I	V	3	M	S

V závislosti, ve kterém řádku a sloupci se každý znak otevřeného textu nachází, je zvolena dvojice znaků symbolizující příslušný řádek a sloupec.

Tabulka 8.: Přiřazená písmena řádků a sloupců pro každý znak otevřeného textu.

A	T	T	A	C	K	A	T	O	N	C	E
VX	XA	XA	VX	VF	AD	VX	XA	FG	GD	VF	GG

Dále je vytvořena nová tabulka z písmen reprezentujících otevřený text, společně se zvoleným klíčem vepsaným nad tuto tabulku.

Tabulka 9.: Nová tabulka s klíčem k zašifrování.

C	H	A	R	G	E
V	X	X	A	X	A
V	X	V	F	A	D
V	X	X	A	F	G
G	D	V	F	G	G

Sloupce tabulky jsou dále seřazeny dle abecedního pořadí klíčového slova.

Tabulka 10.: Transponovaná tabulka dle abecedního pořadí znaků klíče.

A	C	E	G	H	R
X	V	A	X	X	A
V	V	D	A	X	F
X	V	G	F	X	A
V	G	G	G	D	F

Zbývá již pouze odečíst zašifrovaný text po jednotlivých sloupcích, zašifrovaná zpráva je tedy: *XVXVVV VGADGG XAFGXX XDAFAF*

Pro dešifrování je třeba vyplnit tabulku zašifrovaným textem a se znalostí klíče přetransformovat sloupce ke zpětnému získání tabulky č. 9. Čtením jednotlivých řádků a následným rozdělením na dvojice poté za znalosti uspořádání jednotlivých znaků v tabulce č. 8. můžeme zpětně dešifrovat jednotlivé znaky. [22]

Slabinou šifry ADFGVX je snadná identifikace šifry díky opakování těchto znaků a rozčlenění šifrovaného textu na dvojice kdy každá dvojice představuje jeden znak je snadné určit přibližnou délku otevřeného textu.

3.2.2 Railfence

Řadí se mezi jednoduché transpoziční šifry, je také známa pod názvem „Zigzag“ díky principu šifrování. Jednoduché transpoziční šifry jako Railfence nebo Route šifra fungují na stejném principu transpozice dle určitého tvaru. [23]

Pro zašifrování je otevřený text zapsán do tabulky, kdy každý znak je posunut o pozici napravo o řádek níže. Počet řádků je určen klíčem. Po dosažení nejspodnějšího řádku tabulky se směr otočí a vyplňování znaků pokračuje do prvního řádku po vyčerpání otevřeného textu.

Tabulka 11.: Šifrovací tabulky Railfence šifry

A				C				O			
	T		A		K		T		N		E
		T				A				C	

Na příkladě v tabulce č. 11. je šifrovací tabulka pro otevřený text *ATTACKATONCE* s klíčem 3. Šifrovaný text lze získat čtením z šifrovací tabulky po řádcích: *ACOTAKTNETAC.*

Pro dešifrování je třeba vytvořit tabulku o počtu řádků odpovídající hodnotě klíče a počtu sloupců odpovídající délce šifrovaného textu. Do prvního řádku je vepsán první znak zašifrovaného textu, dále jsou do tabulky doplňovány libovolné znaky, dokud algoritmus znovu nedojde k prvnímu řádku, do něj je dosazen další znak ze zašifrovaného textu. [24]

Tabulka 12.: Doplnění prvního řádku pro dešifrování Railfence

A				C				O			
	*		*		*		*		*		*
		*				*				*	

Horním řádkem se nyní stává řádek č. 2. a celý postup je opakován, dokud není celá tabulka vyplněna. Pro dešifrování lze zprávu přečíst v původním „Zigzag“ tvaru.

Railfence je velmi neefektivní zejména pro krátké zprávy a lze ji snadno dešifrovat i bez využití počítače. Zlepšit bezpečnost lze např. přidáním mezer do abecedy, mezery tak budou vepisovány do tabulky stejně jako ostatní znaky. Tato úprava mírně pozmění tvar šifrovaného textu a ztíží identifikaci šifry.

Tabulka 13.: Šifrovací tabulka Railfence s mezerami započítanými v abecedě

A				C				T				C	
	T		A		K		A		-		N		E
		T				-				O			

Výsledný šifrovaný text se tak pozmění na: *ACTCTAKA NET O.*

3.2.3 Route

Podobně jako Railfence, šifra typu Route je jednoduchá transpoziční šifra. Šifrovat lze dle jakéhokoliv dohodnutého tvaru či postupu v tabulce, jejíž velikost je stanovena klíčem.

Nejčastěji voleným postupem je sloupcová transpozice. Klíč může určovat počet řádků nebo počet sloupců tabulky, otevřený text je pak zapsán do tabulky, v případě že délka otevřeného textu je menší než velikost tabulky, text je doplněn o znak *X*. Šifrovaný text je čten po sloupcích. Např. zašifrování textu: *HELLOWORLD* s klíčem: 3.[25]

Tabulka 14.: Šifrovací tabulka Route šifry při sloupcové transpozici.

H	E	L
L	O	W
O	R	L
D	X	X

Zašifrovaný text lze snadno odečíst: *HLODEORXLWLX.*

Pro dešifrování stačí znát velikost šifrovací tabulky a doplnění šifrovaného textu po sloupcích.

Tato varianta je však snadno dešifrovatelná bez jakýchkoliv nástrojů nebo výpočetní techniky. Lze zvolit o něco komplikovanější postup šifrováním ve tvaru spirály.

Zašifrování otevřeného textu: *HELLOWORLD* s klíčem o velikosti 4. metodou spirály.

Klíč nyní udává počet řádků, záleží však pouze na domluvě stran, který rozměr tabulky bude klíč udávat. Pro ztížení je otevřený text zapisován od pravého spodního rohu tabulky směrem vzhůru a od posledního sloupce tabulky k prvnímu. Rozměry tabulky a otevřeného textu se liší, tabulka je znova doplněna o znaky *X*.

Tabulka 15.: Šifrovací tabulka Route šifry při transpozici obrácené spirály.

X	R	L
X	O	L
D	W	E
L	O	H

Šifrovaný text je však čten od prvního řádku prvního sloupce ve směru hodinových ručiček. Šifrovaný text tedy bude: *XRLLEHOLDXOW*.

Pro dešifrování opět stačí znát velikost tabulky a doplnění znaků šifrovaného textu do tvaru spirály.

Šifrovaný text je obtížnější k rozluštění za použití tužky a papíru, zranitelnost šifry vůči vyhledávaní anagramů počítačem však zůstává. Zlepšit bezpečnost lze kombinací Route šifry s jinou substituční šifrou, obdobně jako tomu je u šifry ADFGVX kombinující substituci i transpozici. [26]

3.2.4 Scytale

Scytale nebo také *skutálē* je překlad slova cylindr ze starověké řečtiny, právě v této době byla šifra vyvinuta a využívána pro komunikaci v armádě. Řadí se mezi transpoziční šifry. Pro šifrování byl používán právě cylindr o určitém počtu stran a průměru, na cylindr byla namotána látka, na kterou byla v řádku zapsána zpráva. Po odmotání látky byla zpráva na ní nečitelná a dešifrovatelná pro druhou stranu za použití cylindru o stejných rozměrech. [27]

Pro moderní implementaci je místo cylindru využíván klíč určující počet znaků ve sloupci, lze vytvořit tabulku do níž bude zpráva vepsána. Pro zašifrování zprávy: *SENDHELP* lze sestrojit šifrovací tabulku:

Tabulka 16.: Šifrovací tabulka Scytale pro klíč 3.

S	E	N
D	H	E
L	P	X

Při přečtení výsledných řádků získáme zašifrovanou zprávu: *SDLEHPNEX*. Kde *X* je opět doplňkem pro dorovnání délky otevřeného textu do velikosti tabulky.

Pro dešifrování stačí znát velikost tabulky danou klíčem a doplnit do ní zašifrovanou zprávu po sloupcích.

Pro útok na šifru Scytale lze použít útoku hrubou silou, velikost klíče je omezena délkou zprávy, která zůstává po zašifrování stejná.

4 MOŽNOSTI KRYPTOANALÝZY

Úkolem kryptoanalýzy je analýza šifrovaných textů, algoritmů používaných k jejich zašifrování a hledání slabin v šifrovacích algoritmech pro prolomení šifrovaných dat. Důležitou částí je množství útočníkovi dostupných informací. [28]

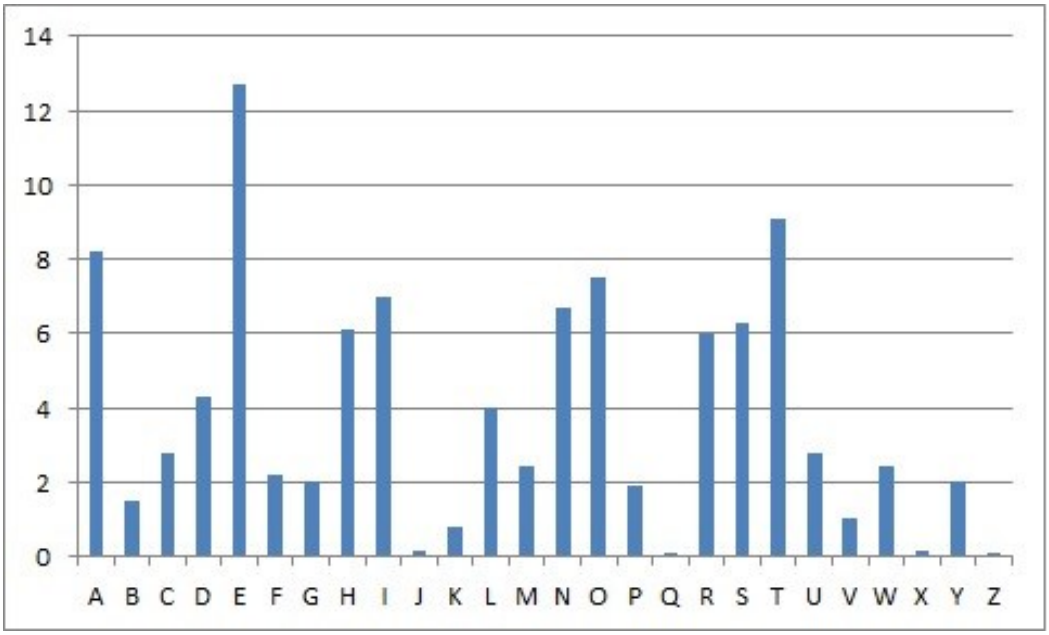
- Ciphertext-only
  Útočník zná pouze zašifrované texty. Bez možnosti ovlivnit vstup či výstup je tato možnost nejobtížnější k dešifrování.
- Known-plaintext
  Útočník zná jak zašifrované texty, tak otevřené texty, jediná neznámá je tak šifrovací klíč, popř. použitý šifrovací algoritmus.
- Chosen-plaintext
  Útočník má možnost zvolit otevřený text, který bude zašifrován a také znám útočníkovi.
- Adaptive chosen-plaintext
  Útočník má možnost volby otevřeného textu na základě předchozích otevřených textů a z nich zašifrovaných textů.
- Chosen-ciphertext
  Útočník má k dispozici dešifrované texty zvolených šifrovaných textů.
- Related-key
  Útočník má možnost získat zašifrované texty, které byly zašifrovány rozdílnými klíči, které však útočník nezná.

Existuje mnoho kryptoanalytických metod využívajících výše uvedené, tato práce je zaměřena na klasické šifry, pro které jsem zvolil metody popsané v této kapitole.

4.1 Frekvenční analýza

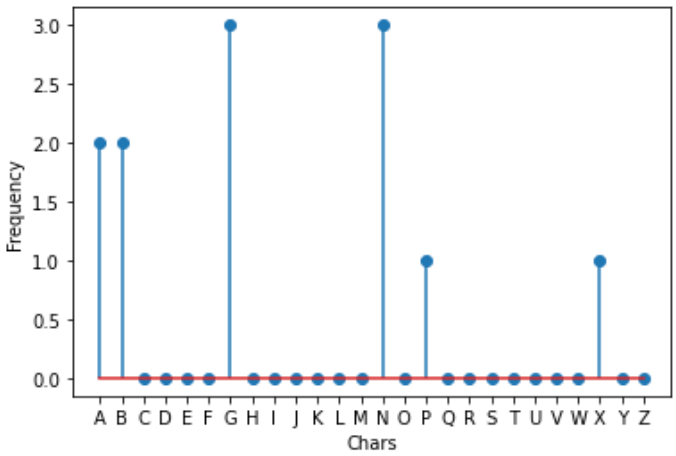
Frekvenční analýza zjišťuje četnost výskytu jednotlivých znaků, či skupin znaků v textu. Každý jazyk má své základní rozložení znaků odlišné, avšak s dostatečně dlouhým textem nezáleží na kontextu daného textu, frekvence znaků bude odpovídat „normálnímu rozložení“. Např. v anglickém jazyce mají nejvyšší frekvenci výskytu písmena *E, T, A*, *O*. Pro češtinu jsou to písmena *O, E, N, A*. Pro kryptoanalytické účely tato vlastnost znamená, že stačí znát jazyk ve kterém byl otevřený text napsán k úspěšnému rozšifrování i bez znalosti klíče. Frekvenční analýza je tak využívána k zjištění četnosti znaků v zašifrovaném textu,

kdy je tato četnost porovnána se standardním rozložením znaků v daném jazyce a je možno určit o kolik znaků je abeceda posunuta, nebo jaké znaky byly substituovány. Tohle však platí pouze monoalfabetické substituční šifry. Jak již víme, substituční šifry nahrazují znaky v otevřeném textu jinými znaky téže abecedy, tedy v případě frekvenční analýza takového textu musí být četnost znaků „posunuta“ na jiný znak. V případě polyalfabetické substituce tento postup neplatí, zde může být např. znak *E* nahrazen v jedné instanci znakem *I* a jindy znakem *Q*, frekvenční analýza tak neukáže posun ve výskytu znaků a není možno je porovnat oproti standardnímu výskytu. [29]



Obrázek 4.: Standardní rozložení znaků pro anglický text [29]

Pro frekvenční analýzu je nutno použít text o co největší možné délce. Pro kratší texty je frekvenční analýza velmi nepřesná a neúplná.



Obrázek 5.: Rozložení znaků pro text ATTACKATNOON zašifrováno ROT13

Frekvenční analýza není omezena pouze na jednotlivé znaky, lze analyzovat i tzv. N-gramy, tedy *n*-tice znaků, nejčastěji bigramy či trigramy.

TH : 2.71	EN : 1.13	NG : 0.89
HE : 2.33	AT : 1.12	AL : 0.88
IN : 2.03	ED : 1.08	IT : 0.88
ER : 1.78	ND : 1.07	AS : 0.87
AN : 1.61	TO : 1.07	IS : 0.86
RE : 1.41	OR : 1.06	HA : 0.83
ES : 1.32	EA : 1.00	ET : 0.76
ON : 1.32	TI : 0.99	SE : 0.73
ST : 1.25	AR : 0.98	OU : 0.72
NT : 1.17	TE : 0.98	OF : 0.71

Obrázek 6.: Nejčastěji vyskytující se bigramy pro anglický text s frekvencemi výskytu [30]

THE : 1.81	ERE : 0.31	HES : 0.24
AND : 0.73	TIO : 0.31	VER : 0.24
ING : 0.72	TER : 0.30	HIS : 0.24
ENT : 0.42	EST : 0.28	OFT : 0.22
ION : 0.42	ERS : 0.28	ITH : 0.21
HER : 0.36	ATI : 0.26	FTH : 0.21
FOR : 0.34	HAT : 0.26	STH : 0.21
THA : 0.33	ATE : 0.25	OTH : 0.21
NTH : 0.33	ALL : 0.25	RES : 0.21
INT : 0.32	ETH : 0.24	ONT : 0.20

Obrázek 7.: Nejčastěji vyskytující se trigramy pro anglický text s frekvencemi výskytu [30]

Analýzu bigramů lze do jisté míry použít například pro analýzu šifry typu Playfair či ADFGVX, zde jde však frekvenční analýza pouze součástí možného řešení vzhledem k dalším vlastnostem těchto šifer. V závislosti na výsledku frekvenční analýzy lze určit, jestli se jedná o substituční šifru nebo transpoziční. Pouhá transpozice zachová počet jednotlivých znaků v textu, pouze ve změněném pořadí, v takovém případě musí frekvenční analýza šifrovaného textu odpovídat frekvenční analýze otevřeného textu. [31]

4.2 Kasiského metoda

Kasiského metoda je nástrojem užívaným k prolomení polyalfabetických substitučních šifer, konkrétně k určení délky klíčového slova použitého k zašifrování. Princip spočívá v hledání opakujících se *n*-gramů v zašifrovaném textu, kdy je změřena vzdálenost stejných *n*-gramů,

která napoví délku šifrovacího klíče. Opakování *n*-gramů napovídá opakujícímu se šifrovacímu klíči, jako tomu je ve Vigenèrově šifře, která je Kasiského metodou snadno rozluštitelná. Např. vzdálenost dvou trigramů zašifrovaného textu je 12, klíč by se tedy měl opakovat v rozmezí maximálně 12 znaků kdy pomocí faktorizace lze určit možné délky tedy: *1, 2, 3, 4, 6, 12*. Tuto informaci lze zpřesnit nalezením více *n*-gramů. Podobně jako u frekvenční analýzy, delší zašifrovaný text poskytne více dat k analýze a zpřesnění výsledků. Při faktorizaci vzdáleností mezi *n*-gramy je obvykle mnoho výsledků, je proto vhodné považovat faktor s nejvyšším počtem opakování pro všechny *n*-gramy jako správnou délku. Po nalezení délky, nebo alespoň přibližné délky klíče lze v prolomení šifry pokračovat dalšími způsoby, jako útokem hrubou silou nebo slovníkovým útokem. [32]

4.3 Index koincidence

Index koincidence vyjadřuje pravděpodobnost shody dvou znaků při porovnávání dvou textů znak po znaku. Hodnota je určena pravděpodobností výskytu znaku v abecedě prvního textu a pravděpodobností, že znak bude porovnán se znakem z druhého textu, který má taktéž vlastní pravděpodobnost výskytu ve své abecedě. Každý jazyk má tak vlastní index koincidence vzhledem k odlišnému počtu, a frekvenci výskytu znaků v jeho abecedě. Vzorec pro výpočet je pak:

$$IC = \frac{\sum_{i=1}^{c} n_i(n_i - 1)}{N(N-1)/c}$$

- IC – index koincidence
- n_i – četnost výskytů znaku v textu *i*
- N – počet znaků v textu
- c – počet znaků v abecedě [33]

Při výběru náhodného znaku z anglické abecedy obsahující 26 znaků, a poté dalšímu výběru náhodného znaku z téže abecedy je šance na výběr stejného znaku 1/26, tedy 0.0385. V případě, že však vybereme dva znaky náhodně z anglicky psaného textu se šance zvýší na 0.0667, pro anglický jazyk je tak index koincidence $IC = \frac{0.0667}{0.0385} = \mathbf{1.73.}$

Pro náhodně psaný text se index koincidence pohybuje mezi hodnotou 1.5 a 2.0. Pakliže by byla frekvence výskytu znaků určitého jazyka stejná pro všechny znaky, blížil by se tento index hodnotě 1.0.[34]

Pro kryptoanalytické účely je index koincidence vhodný např. pro určení typu šifry. Při zašifrování pomocí transpozice nebo monoalfabetické substituce bude index koincidence pro zašifrovaný text vyšší, než pro náhodný text. Při nižším indexu koincidence se předpokládá šifrování polyalfabetickou šifrou, protože jeden znak může být zašifrován více způsoby, což zvýší pravděpodobnost výskytu dalších znaků a sníží pravděpodobnost volby stejného znaku pro index koincidence.

Pro rozluštění substituční šifry pomocí indexu koincidence je třeba odhadnout délku klíče. Za předpokladu že délka klíče je *n*, šifrovaný text je rozdělen do „stringů“ tak aby platilo:

$$S_i = C_i C_{i+n} C_{i+2n} \ldots.$$

- S_i – jednotlivé „stringy“ pro analýzu
- C_i – znak na indexu $i+n$ zašifrovaného textu
- n – odhadovaná délka klíče [34]

Za předpokladu, že je odhadnutá délka klíče správná, bude index koincidence každého z utvořených „stringů“ okolo 0.0667 za předpokladu, že zašifrovaný text byl psán v angličtině. Pakliže index koincidence dosahuje podobných hodnot pro více délek klíče, nejvyšší hodnota je považována za správnou délku. [35] [36]

4.4 Útok hrubou silou

Tato metoda kryptoanalýzy je jednou z nejznámějších, ale také nejprimitivnějších metod k prolamování šifer. Jedná se o hádání a zkoušení všech možných kombinací klíče. Funguje však pouze za předpokladu, že šifra má malé množství možných klíčů, které je možné vyzkoušet v přijatelném čase. Čas potřebný k prolomení klíče je úměrný 2^k, kdy *k* je délka klíče v bitech, je však také závislý na specifikacích počítače, resp. Počtu možností, které je počítač schopen vyhodnotit za jednotku času. Ačkoliv si i běžné stolní počítače hravě poradí s jednoduchými substitučními šiframi, některé šifry jsou vůči tomuto útoku stále relativně odolné, a je třeba využít některé z výše uvedených kryptoanalytických metod. [37] [38]

4.5 Slovníkový útok

Zvláštní variantou útoku hrubou silou je slovníkový útok. Princip náhodného hádání zůstává stejný, avšak místo náhodného znaku je za šifrovací klíč dosazen výraz ze „slovníku“ – seznamu často užívaných slov či frází pro určitý jazyk.

II. PRAKTICKÁ ČÁST

5 NÁSTROJE

Následující nástroje byly zvoleny pro svou jednoduchost, rychlost, a možnosti snadného rozšíření aplikace do budoucna.

5.1 Python

Python je moderní, vysokoúrovňový programovací jazyk známý pro svou jednoduchou syntaxi, širokou škálu využití, a skvělou komptabilitou s mnoha frameworky. Pro práci byla využita verze 3.10. Jazyk byl zvolen pro svou flexibilitu, ale také mnoho knihoven ulehčujících práci s daty.

5.2 Volba vývojového prostředí

Anaconda je distribuce Pythonu pro vědecké účely obsahující mnoho užitečných modulů. Pro práci byla vybrána pro jednoduchý proces instalace dodatečných modulů ulehčujících implementaci. Jako vývojové prostředí byl zvolen Spyder verze 5.1.5 díky své všestrannosti a dobré optimalizaci práce s velkým objemem dat. Pro vývoj snadného a responzivního GUI byl využit framework Qt.

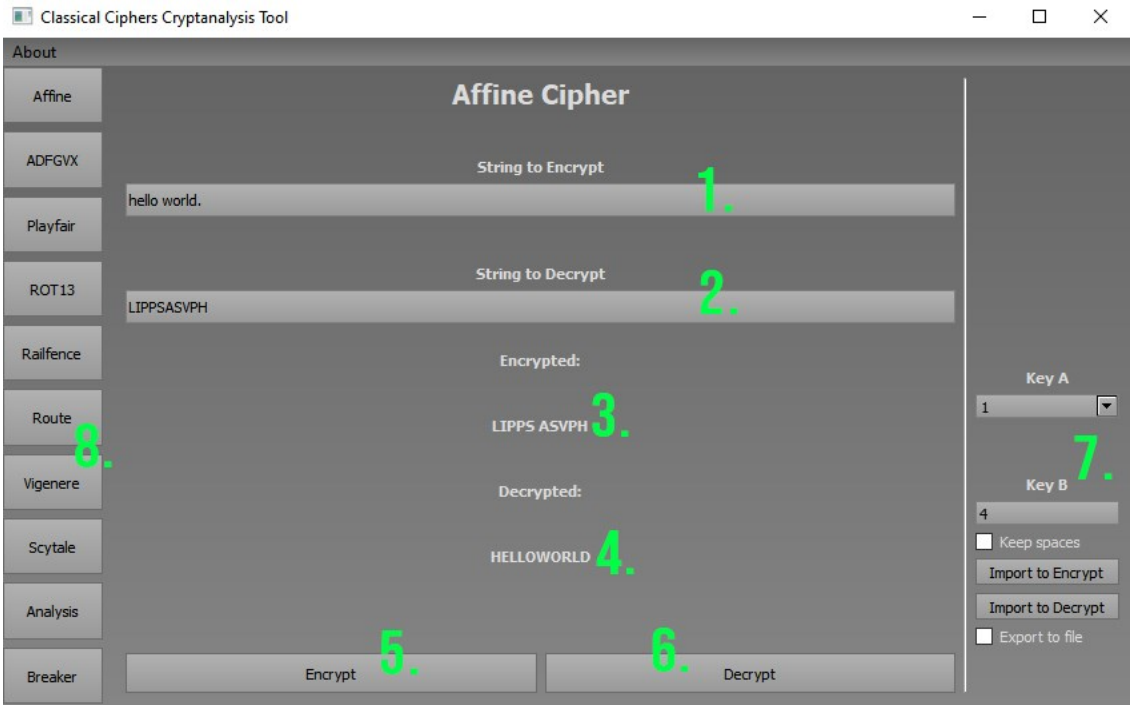
6 ŠIFROVACÍ APLIKACE

Výsledná aplikace slouží k šifrování a dešifrování pomocí algoritmů popsaných v teoretické části práce, analýze k určení typu užité šifry pouze ze znalosti zašifrovaného textu a následnému prolamování.

6.1 Funkcionality programu

6.1.1 Šifrování

Po spuštění aplikace je třeba zvolit požadovanou akci. Každá šifra je implementována ve vlastní záložce kde uživatel může provádět šifrování i dešifrování.



Obrázek 8.: Šifrovací aplikace – záložka pro šifrování Afinní šifrou s popisem prvků

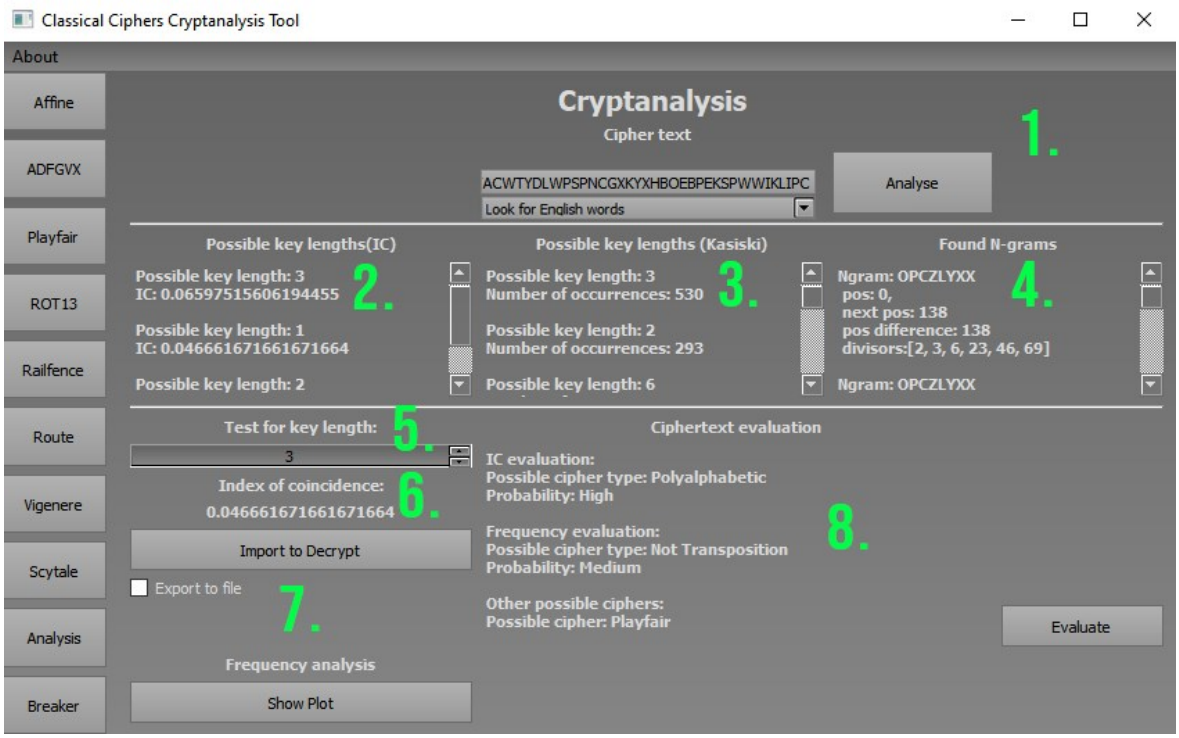
1. Zadávací pole pro otevřený text k zašifrování
2. Zadávací pole pro zašifrovaný text k dešifrování
3. Zašifrovaný text
4. Dešifrovaný text
5. Tlačítko ke spuštění šifrování textu zadaného do pole 1.
6. Tlačítko ke spuštění dešifrování textu zadaného do pole 2.
7. Oblast obsahující zadávání šifrovacího klíče a volbu nahrání šifry ze souboru
8. Tlačítka pro volbu činnosti (záložky) programu

Aplikace umí upravit text zadaný do pole k šifrování či dešifrování, zbavuje se tak všech nealfanumerických znaků, tato úprava probíhá po stisknutí tlačítek *Encrypt* či *Decrypt* pro příslušná pole. V případě, že uživatel stiskne tlačítko *Import to Encrypt* nebo *Import to Decrypt*, je otevřeno okno umožňující načíst soubor ve formátu *.txt*, obsahující text k zašifrování nebo dešifrování. Dále je text automaticky převeden do formátu *uppercase*. Takto upravený text je připraven k zašifrování/dešifrování a program pokračuje ke kontrole správnosti zadání šifrovacího klíče.

Zadávání šifrovacího klíče je vždy prováděno na pravé části okna za svislou dělící čárou, tato oblast také obsahuje dodatečná nastavení pro každou šifru na základě jejich potřeb jako např. zadávání vlastní šifrovací matice pro šifru ADFGVX či možnost zachovat mezery v otevřeném textu i po zašifrování pro všechny šifry. Pro klíč Afinní šifry konkrétně platí omezení zmiňovaná v teoretické části, každé pole v této části tak akceptuje pouze číselnou hodnotu jako klíč. Tohle omezení je pozměněno pro každou šifru na základě omezení klíče, např. Vigenèrova šifra akceptuje pouze písmena abecedy A-Z. Kontrola správnosti klíče opět probíhá po stisknutí tlačítek *Encrypt* či *Decrypt*. V případě, že klíč není správného formátu či délky, aplikace zobrazí uživateli chybové okno indikující špatně zadaný klíč. Při zobrazení chybové hlášky aplikace vloží „základní“ klíč správného formátu do zadávacího pole ve správném formátu, pro Afinní šifru se například jedná o číslici 5. Poté je možné opět kliknout na tlačítko *Encrypt* či *Decrypt* pro příslušnou akci a pokračovat. V případě úspěšného šifrování je zašifrovaný text vložen po odrážku *Encrypted:* a zároveň do pole č. 2. pro dešifrování. Tato vlastnost umožňuje lepší čitelnost a zároveň možnost výsledný text kopírovat pro další využití uživatelem. Pokud byla zaškrtnuta volba *Export to file*, bude po šifrování nebo dešifrování vytvořen textový soubor obsahující výsledný text.

6.1.2 Analýza

Analýza zašifrovaného textu byla navržena pro přímé rozpoznání, nebo alespoň přiblížení typu šifry užité k zašifrování. Pro analýzu byl využit index koincidence, jak k zjištění délky klíče, tak k určení typu šifry a Kasiského metoda pro určení délky klíče a nalezení opakujících se N-gramů a jejich vyhodnocení. Je také možné zobrazit frekvenční analýzu šifrovaného textu, v této části aplikace však nelze manipulovat nebo upravovat rozložení znaků k možnému prolomení šifry.



Obrázek 9.: Šifrovací aplikace – záložka pro analýzu šifrovaného textu a popis prvků

1. Pole pro zadávání zašifrovaného textu k analýze, výběru jazyka a spouštěcí tlačítko
2. Okno pro výsledek analýzy – délka klíče indexem koincidence
3. Okno pro výsledek analýzy – délka klíče Kasiského metodou
4. Okno pro výsledek analýzy – nalezené N-gramy v zašifrovaném textu
5. Pole pro volbu maximální délky klíče pro analýzu indexem koincidence
6. Index koincidence pro zašifrovaný text
7. Tlačítka pro zobrazení grafu frekvenční analýzy a volbu nahrání šifry ze souboru
8. Okno pro celkové vyhodnocení analýzy

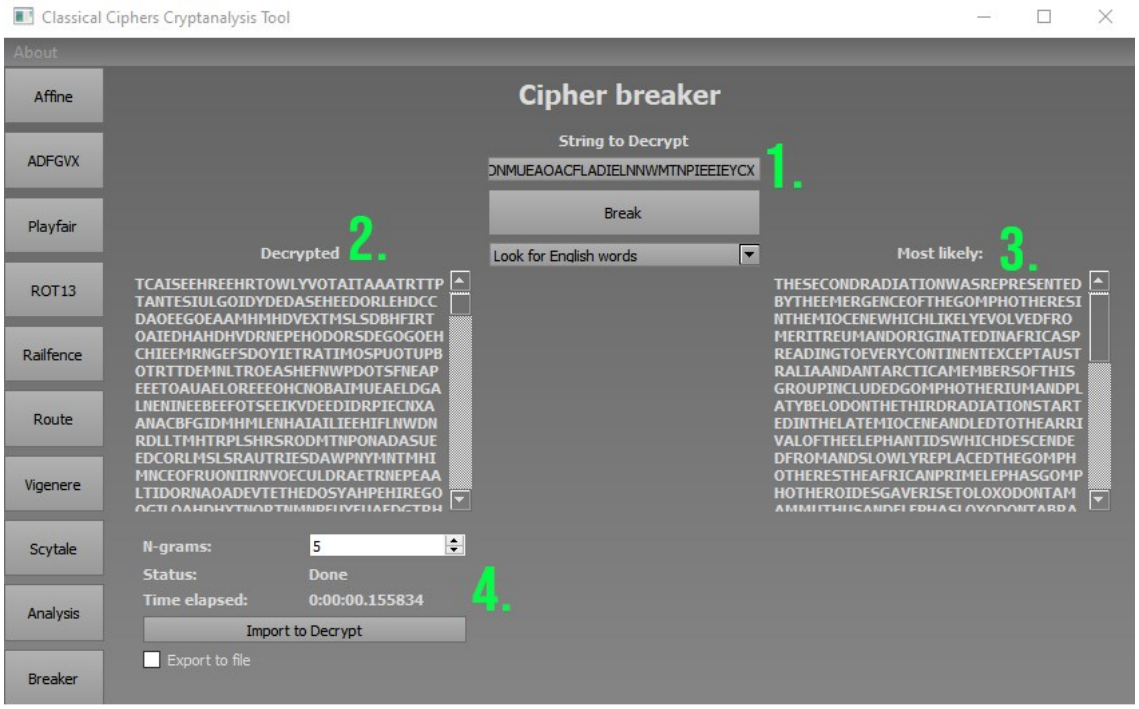
Podobně jako je tomu u šifrovacích záložek, i zde je vstupní text v poli 1. upraven na uppercase. Pro přesnou analýzu je však vhodné vkládat zašifrovaný text ze kterého byly odstraněny mezery a číslice. Po stisku tlačítka *Analyse* se spustí analýza a výsledky jsou vypsány do oken 2, 3, 4 a 6. Pravděpodobná délka klíče dle indexu koincidence v okně č. 2. je seřazena dle nejvyššího indexu koincidence pro danou délku klíče. Limit délky klíče, pro který má aplikace hledat a vyhodnocovat index koincidence je zadáván polem č. 5. úprava tohoto pole aktualizuje obsah pole č. 2. v reálném čase a není tak třeba znovu spouštět analýzu. Okno č. 3. slouží jako ověření okna č. 2. Na obrázku č. 9. je ukázka analýzy textu o délce 1437 znaků zašifrovaného Vigenèrovou šifrou klíčem dlouhým 4 znaky. Z příkladu lze

pozorovat, proč je vhodné délku klíče určovat více metodami, neboť určení délky klíče na základě nejvyššího výskytu N-gramů o určité délce nemusí být vždy zcela přesné. Okno č. 4. zobrazuje nalezené N-gramy, jejich vzdálenost a pozici v zašifrovaném textu k ověření výsledků Kasiského metody při určení délky klíče. Po stisknutí tlačítka *Evaluate* je zobrazeno konečné vyhodnocení analýzy v okně č. 8. Výpis obsahuje určení typu šifrovacího algoritmu dle indexu koincidence a Kasiského metody, společně s pravděpodobností správnosti tohoto odhadu. Pravděpodobnost odhadu bude dále popsána v popisu analytického algoritmu. V závislosti na vlastnostech šifrovaného textu je také navržen konkrétní algoritmus, který mohl být použit k zašifrování textu, není zde uváděna míra pravděpodobnosti vzhledem k volatilitě přesnosti tohoto odhadu.

Analýza je prováděna pro zašifrovaný text, jehož otevřený text byl psán jak v češtině, tak v angličtině, při použití zašifrovaného textu jiného jazyka se bude lišit index koincidence což zapříčiní změnu ve vyhodnocení délky klíče a určení šifrovacího algoritmu.

6.1.3 Prolamování

V záložce *Breaker* je nejdříve nutno zvolit typ šifry, který chceme dešifrovat. Po výběru šifry jsou elementy záložky zobrazeny v závislosti na tom, které z nich zvolená šifra využívá.



Obrázek 10.: Šifrovací aplikace – záložka pro prolamování šifrovaného textu a popis prvků

1. Pole pro zadávání zašifrovaného textu k analýze, volba jazyka a spouštěcí tlačítko
2. Okno pro zobrazení dešifrovaného textu s klíčem, popř. metodou šifrování
3. Okno zobrazující pravděpodobně nejsprávnější výsledky dešifrování s délkou klíče
4. Okno k zadávání počtu N-gramů pro vyhledávání v okně č. 3. a čas trvání prolamování a tlačítko pro volbu nahrávání šifer ze souboru

Výše uvedený příklad je ukázkou prolomení Route šifry. Do okna č. 2. jsou zapsány veškeré výsledky prolamování této šifry, společně s typem transpozice a délkou klíče. Okno č. 3. zobrazuje nejpravděpodobněji správný výsledek šifrování. Výsledné dešifrované texty z okna č.2. jsou porovnány se slovníkem anglických, nebo českých slov o délce *n,* v závislosti na výběru slovníku před tlačítkem *Break*, Délka *n* je určena v okně č. 4., a v případě shody je pak tento výsledek zapsán do okna č. 3. jako pravděpodobně správný výsledek. Pro nalezení nejsprávnějšího řešení je vhodné volit co nejvyšší *n*. Při nastavení nízkého *n* existuje dobrá šance, že i v nesprávně dešifrovaném textu bude nalezena shoda se slovníkem, tedy že malé množství po sobě jdoucích znaků bude stejné pro oba texty.

6.2 Struktura programu

Zdrojové kódy aplikace jsou rozděleny pro každou funkcionalitu aplikace zvlášť. Každý šifrovací algoritmus je vložen do vlastního souboru jako kompletní řešení včetně šifrování i dešifrování. Každý zdrojový kód šifry importuje *usefull_functions.py* - soubor obsahující funkce pro úpravu vstupního textu, které jsou společné pro více šifer, ale také vlastní GUI. Tyto soubory jsou volány ze souboru *main.py* do nějž jsou importovány všechny podsoubory mezi nimiž je přepínáno dle potřeby. Níže ukázka kódu v souboru *main.py* pro inicializaci tlačítka pro volbu Afinní šifry a přechod na danou záložku.

```
from src.affine import Affine

class MainWindow(QMainWindow):

    def __init__(self):
        super().__init__()
        uic.loadUi('bpdraftgui.ui', self)
        self.stackedWidget.setCurrentWidget(self.centralwidget)
        self.affineBtn.clicked.connect(self.gotoAffine)

    def gotoAffine(self):
        self.affine=Affine()
        self.stackedWidget.addWidget(self.affine)
        self.stackedWidget.setCurrentWidget(self.affine)
```

Soubor *main.py* tak potřebuje v jednu chvíli otevírat pouze GUI s tlačítky pro volbu šifrování implementované pomocí třídy *QtStackedWidget,* na kterou je „promítáno“ GUI importované přímo šifrovacím algoritmem, analytickým nástrojem či lamačem šifer.

6.3 Algoritmy pro analýzu

Analýza využívá indexu koincidence, frekvenční analýzy a Kasiského metody – algoritmů popsaných v teoretické části. Je však nutno vyhodnotit data získaná z těchto algoritmů. Pro posouzení typu šifry na základě indexu koincidence je porovnána jeho hodnota s hraničními hodnotami. [35]. Při hodnotě blízké 0.070 je šifrovaný text pravděpodobně zašifrován transpoziční nebo monoalfabetickou substituční šifrou. Pro nízký index koincidence platí, že se jedná o polyalfabetickou šifru, stačí tak porovnat hodnoty indexu koincidence zašifrovaného textu s hraničními hodnotami a získáme poměrně přesný odhad o typu šifry včetně přesnosti.

Pro frekvenční analýzu je přidáno omezení minimální délky znaků pro analýzu, protože je využíváno četnosti *n*-gramů k identifikaci, jestli se jedná o transpoziční šifru. Při nízkém počtu znaků by tak byl odhad typu šifry velmi nepřesný. Níže ukázka kódu pro určení typu šifry dle frekvenční analýzy.

```
for i in range(13):
     freqalphabetletters.append(cipherCounter[i][0])
if self.languagePick.currentText()=='Look for English words':
         default=["E","T","A","O","I","N","S","H","R","D","L","C","U"]
if self.languagePick.currentText()=='Look for Czech words':
         default=["O","E","N","A","T","V","S","I","L","K","R","P","M"]
points=0
conText.append("\nFrequency evaluation:\n")

for i in range(len(default)):
     if freqalphabetletters[i] in default:
         points+=1
if points>=10:
    conText.append("Possible cipher type: Transposition \n")
    conText.append("Probability: Very high \n")
if points>=7:
    conText.append("Possible cipher type: Transposition \n")
    conText.append("Probability: High \n")
if points<=6:
    conText.append("Possible cipher type: Not Transposition \n")
    conText.append("Probability: Medium \n")
if points<=3:
    conText.append("Possible cipher type: Not Transposition \n")
    conText.append("Probability: Very high \n")
else:
    conText.append("Ciphertext not long enough \n")
```

Bylo vybráno 13 nejčetnějších znaků v anglické a české abecedě dle obrázku č. 4., poté je v zašifrovaném textu zjištěn výskyt těchto znaků. Výskyt každého znaku je pak ohodnocen jedním bodem, které jsou poté vyhodnoceny na stupnici určující, zdali se jedná o transpoziční šifru. Transpoziční šifra zachová v zašifrovaném textu stejné znaky v jiném pořadí, při výskytu většiny nejužívanějších znaků v češtině či angličtině, tento typ šifry je tak snadno identifikovatelný.

Identifikace konkrétních šifrovacích algoritmů není příliš přesná, pro identifikaci Playfair či Hillovy šifry je pouze zjištění dělitelnosti dvěmi a třemi. Šifra ADFGVX je často identifikovatelná na první pohled – šifrovaný text obsahuje pouze tyto znaky. Ve chvíli, kdy je nalezen jiný znak, nemůže se jednat o ADFGVX.

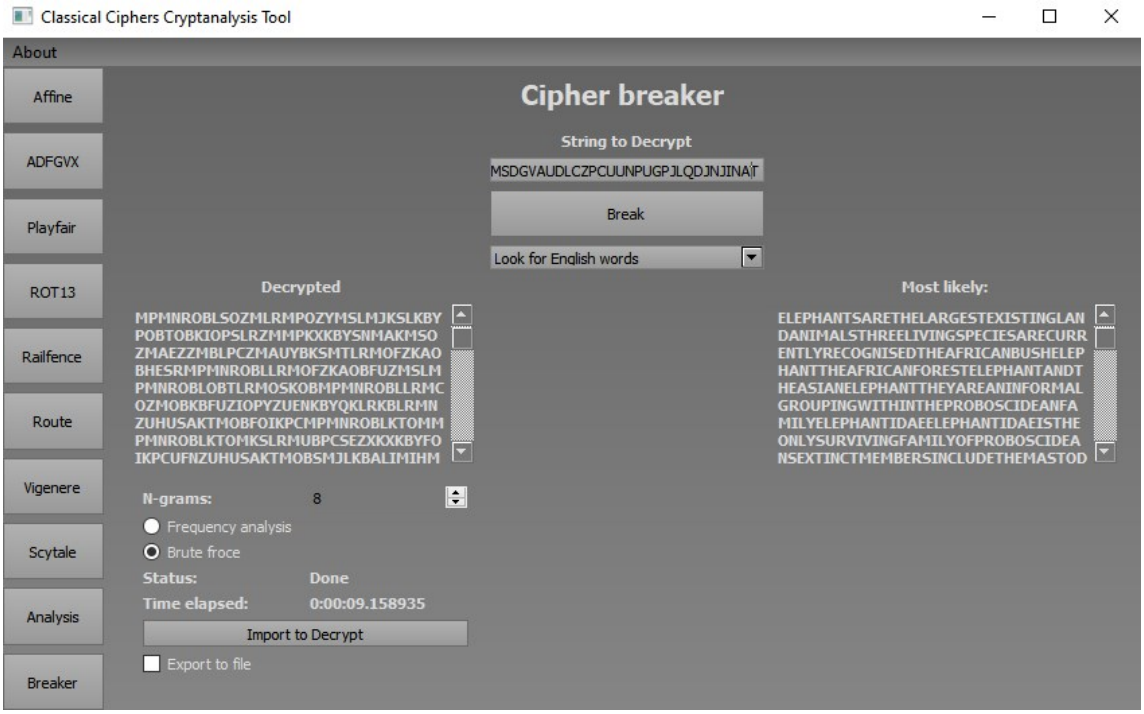
6.4 Algoritmy pro prolamování

K prolamování šifer jsem v této bakalářské práci přistupoval s předpoklady:

- Není znám klíč, otevřený text nebo jeho část
- Je znám šifrovací algoritmus, který zašifroval zprávu a jazyk nezašifrované zprávy

6.4.1 Útok hrubou silou

Šifry s malým počtem možných a jednoduchých klíčů jsou velmi náchylně k útoku hrubou silou, např. pro Afinní šifru stačí vyzkoušet všechny možnosti klíče, kterých je pro abecedu A-Z díky omezením klíče pouze 311 možných. Zašifrovaný text o libovolné délce je i touto metodou prolomen v řádech vteřin. Na obrázku níže příklad prolomení šifrovaného textu o délce 2080 znaků.



Obrázek 11.: Prolomení Afinní šifry hrubou silou

Prolamování šifry ROT bylo implementováno pro všechny hodnoty posunu v anglické abecedě, je tedy provedeno všech 26 posunů, opět v řádech několika vteřin jsou zobrazeny všechny možnosti.

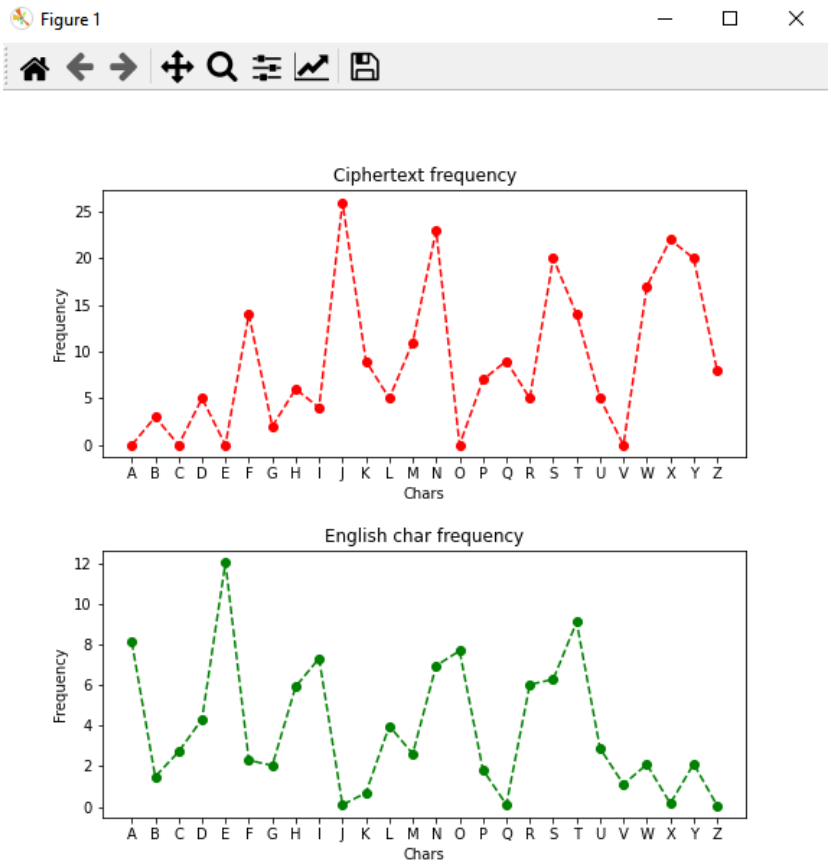
Počet možných klíčů šifry Railfence je omezen počtem šifrovaných znaků, protože klíč udává počet řádků, do kterých je možné text vepsat, délka výpočtu sice stoupá s délkou zašifrované zprávy, avšak pro většinu zpráv je rychlost prolamování stále přijatelná.

Pro prolomení transpoziční šifry Route je třeba postup opakovat pro každou přijatelnou velikost matice – velikost klíče, do nichž je šifrovaný text zapsán, a přečten pomocí dešifrovacího algoritmu pro každý tvar šifrování, v tomto případě se jedná o sloupcovou transpozici a o transpozici ve spirále. Podobně jako u šifry Railfence, pro většinu zpráv je prolamování otázkou vteřin.

Poslední šifrou využívající této metody je šifra Scytale. Narozdíl od Afinní šifry, Scytale má pouze jednu část klíče udávající počet sloupců a maximální počet klíčů je tedy limitován počtem znaků abecedy. Prolamování šifrovaného textu o 3700 znacích tak netrvá ani půl vteřiny.

6.4.2 Útok frekvenční analýzou

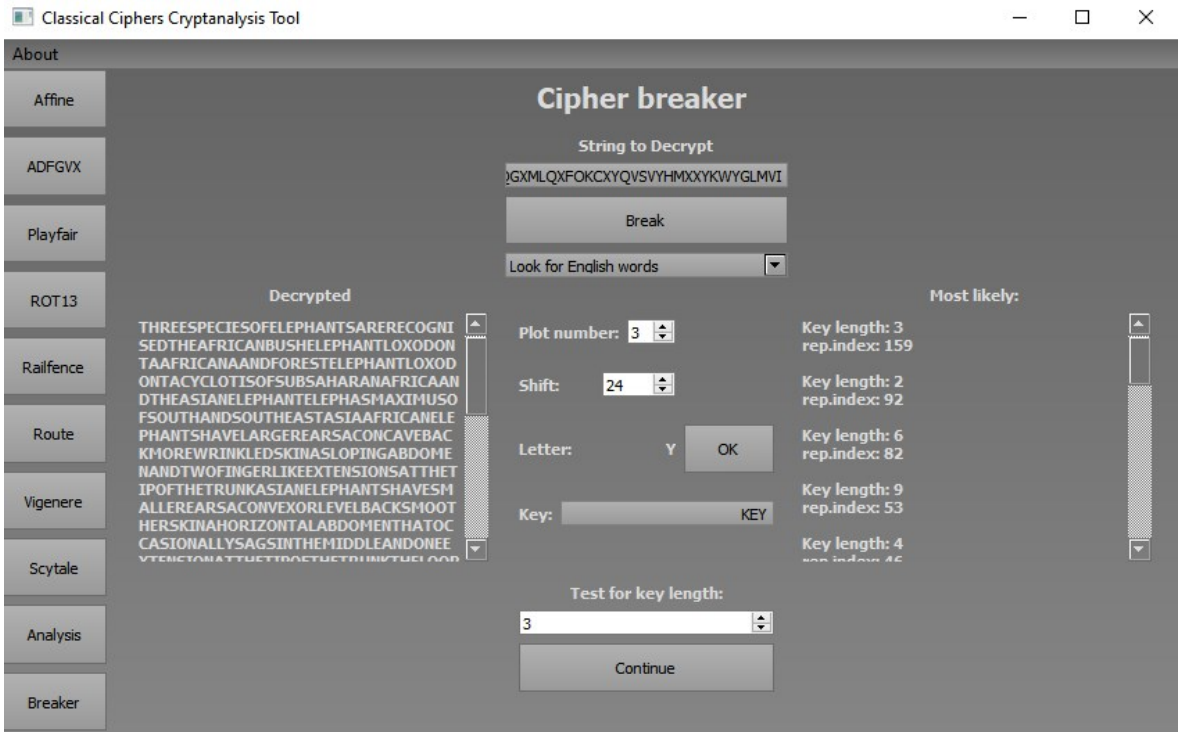
O něco zajímavější je útok pomocí frekvenční analýzy, tato metoda je implementována pro Afinní šifru, může však být nápomocná i pro řešení Vigenérovy šifry při využití Kasiského metody. Nevýhodou tohoto útoku je potřeba uživatele pro rozhodnutí posunů znaků/čísel klíčového slova, tato potřeba je však v aplikaci značně zjednodušena vykreslením grafu srovnávajícím četnost znaků. Jak již bylo uvedeno v teoretické části, pro efektivní prolomení je třeba šifrovaný text o co největší délce.



Obrázek 12.: Grafy k prolomení Afinní šifry frekvenční analýzou

Z porovnání grafů je zřejmé, že šifrovaná abeceda bylo posunuta o 5 znaků, tedy při zpětném posunutí bude graf zašifrovaného textu zhruba odpovídat grafu standartního rozložení. Zpětné posunutí je prováděno zadáváním hodnoty požadovaného posunutí přímo v GUI, tímto je zároveň překreslen graf k zobrazení aktuálního posunutí a zároveň je text dešifrován pro dané posunutí, pro kratší texty je tak možnost alespoň odhadnout klíč i z neúplného grafu. Graf četnosti je na obrázku č. 12. sestaven pro šifrovaný text o délce 235 znaků, delší text by tak identifikaci hodnoty posunu ještě zjednodušil.

Pro Vigenèrovu šifru je mimo frekvenční analýzu využita také Kasiského metoda pro zjištění délky klíče. Na základě jejího vyhodnocení je třeba vybrat délku klíče s co nejvyšším indexem opakování, neboť na jeho délce závisí počet bloků, pro které se bude frekvenční analýza provádět.



Obrázek 13.: Prolamování Vigenèrovy šifry

Dále je postupováno podobně jako v případě Afinní šifry, při posunu je určen odpovídající znak abecedy jakožto znak klíče a tlačítkem *OK* je předán k dešifrování.

6.4.3 Slovníkový útok

Tato metoda je méně flexibilní oproti útoku hrubou silou. Zatímco útok hrubou silou lze lehce modifikovat pro několik typů šifer, slovníkový útok je možné využít pouze tehdy, pokud jsme se jistí že prolamované klíčové slovo je slovem v daném jazyce. Modifikovanou verzi tohoto útoku jsem využil pro prolomení Playfair šifry. K prolomení byly využity slovníky [36] obsahující 1000 nejpoužívanějších českých a anglických slov, pomocí každého z nich byl vytvořen klíč obsahující klíčové slovo na začátku, doplněný o ostatní znaky abecedy neobsažené v klíčovém slově. Implementovaná metoda bude pokračovat až do vyčerpání všech slov ve slovníku, což v případě delšího slovníku může být časově velmi náročné, zároveň se ale zvýší šance na objevení správného klíče. Pro nalezení správného výsledku je

opět využita funkce hledající N-gramy v dešifrovaném textu která vypíše nejpravděpodobnější řešení.

Níže uvedený kód je ukázka funkce pro tvorbu klíče pro slovníkový útok na šifru Playfair.

```
if self.languagePick.currentText()=='Look for English words':
       keyword=open("src/eng1000words.txt","r").read().splitlines()
if self.languagePick.currentText()=='Look for Czech words':
       keyword=open("src/czech1000words.txt","r").read().splitlines()
alphabet = list('ABCDEFGHIKLMNOPQRSTUVWXYZ')
global playfairRes
global playfairKey
playfairRes=[]
playfairKey=[]
res=[]
for k in range(len(keyword)):
    self.status.setText("Working..")
    keyphrase=[]
    for i in range(len(keyword[k])):
        if keyword[k][i]=='j' or keyword[k][i]=='J':
             keyword[k]=keyword[k].replace('j','I')
        if keyword[k][i].upper() not in keyphrase:
              keyphrase.append(keyword[k][i].upper())
     for i in range(len(alphabet)):
         if alphabet[i] not in keyphrase:
              keyphrase.append(alphabet[i])
     key=''.join(keyphrase)
     dec=Playfair(key=key).decipher(ciph)
     self.status.setText("Done")
     playfairRes.append(dec)
     playfairKey.append('Key: '+key)
     res.append(dec)
     res.append('Key: '+key)
self.plainTextLine.setText('\n\n'.join(res))
```

7 VÝSLEDKY ANALÝZY

Poslední kapitola se zabývá efektivitou jednotlivých kryptoanalytických algoritmů pro analýzu i prolamování pro zašifrované texty o různých délkách.

Různé texty o délkách 12, 122, 321, 553, 1070 a 2080 znaků byly zašifrovány všemi implementovanými algoritmy, poté byly zašifrované texty analyzovány pro ověření, zda záložka *Analysis* zvládne správně určit typ šifry. Při prolamování šifer byly měřeny a zaznamenány jejich časy, v případě, že šifra může být prolomena více způsoby, byly implementovány časy všech metod. V případě prolomení frekvenční analýzou není čas uváděn, protože rychlost prolamováni závisí na rychlosti hledání posunů uživatelem. Pro šifrované texty, které nebyly prolomeny, je místo času vepsána značka "*x*". Šifra je považována za identifikovanou za předpokladu, že si výsledky indexu koincidence a frekvenční analýzy neodporují, např. Afinní šifra o délce 2080 znaků je identifikována indexem koincidence jako monoalfabetická nebo transpoziční a frekvenční analýzou jako transpoziční, přestože se jedná o substituční šifru, tento výsledek může nastat, pokud jsou frekvenční analýzy otevřeného a šifrovaného textu podobné, přestože nejsou stejné. Pro Afinní šifru o délce 12 znaků by však frekvenční analýza byla nepřesná, a lze tedy vycházet pouze z indexu koincidence, který značí, že se jedná o monoalfabetickou nebo transpoziční šifru, v tomto případě lze řešení považovat za úspěšné, protože alespoň jeden ukazatel naznačuje typ šifry.

Tabulka 17.: Výsledky kryptoanalýzy Afinní šifry

-	-	Prolomení	Čas prolamování [s]
Délka textu	Identifikace	frekvenční analýza	Bruteforce
12	ANO	ANO	1,68
122	ANO	ANO	2,35
321	ANO	ANO	2,73
553	ANO	ANO	3,22
1070	ANO	ANO	4,82
2080	NE	ANO	9,15

Tabulka 18.: Výsledky kryptoanalýzy šifry ROT13

-	-	Čas prolamování [s]
Délka textu	Identifikace	Bruteforce
12	ANO	0,001
122	ANO	0,002
321	ANO	0,006
553	ANO	0,01
1070	ANO	0,017
2080	ANO	0,035

Tabulka 19.: Výsledky kryptoanalýzy šifry Route

-	-	Čas prolamování [s]
Délka textu	Identifikace	Bruteforce
12	ANO	0,02
122	NE	0,12
321	ANO	0,06
553	ANO	0,13
1070	ANO	0,18
2080	ANO	0,18

Tabulka 20.: Výsledky kryptoanalýzy Vigenèrovy šifry

-	-	Úspěšnost
Délka textu	Identifikace	Frekvenční analýza
12	NE	ANO
122	ANO	ANO
321	ANO	ANO
553	NE	ANO
1070	NE	ANO
2080	ANO	ANO

Tabulka 21.: Výsledky kryptoanalýzy šifry Scytale

-	-	Čas prolamování [s]
Délka textu	Identifikace	Bruteforce
12	ANO	0,001
122	NE	0,005
321	ANO	0,007
553	ANO	0,009
1070	ANO	0,017
2080	ANO	0,017

Tabulka 22.: Výsledky kryptoanalýzy pro šifru Railfence

-	-	Čas prolamování [s]
Délka textu	Identifikace	Bruteforce
12	ANO	0,01
122	ANO	2,19
321	ANO	52,53
553	ANO	237,98
1070	ANO	1709,11
2080	ANO	12702,81

Tabulka 23.: Výsledky kryptoanalýzy pro šifru Playfair

-	-	Čas prolamování [s]
Délka textu	Identifikace	Slovníkový útok
12	NE	0,02
122	ANO	0,11
321	ANO	0,28
553	ANO	0,46
1070	ANO	0,91
2080	NE	1,81

ZÁVĚR

Cílem bakalářské práce bylo nastudovat, vybrat a implementovat metody pro kryptoanalýzu klasických šifer do desktopové aplikace.

V první části jsem teoreticky popsal vybrané šifrovací algoritmy, postupy pro šifrování, dešifrování a tvorbě šifrovacích klíčů. Pro každou šifru jsem uvedl krátkou ukázku šifrovacího a dešifrovacího procesu na modelovém příkladě. V této části jsem se také zmínil o výhodách a nevýhodách jednotlivých šifer a jejich slabinách. Teoreticky jsem popsal vybrané metody kryptoanalýzy pro klasické šifry, principy fungování těchto algoritmů a jejich využití pro kryptoanalýzu. Teoretická část této bakalářské práce slouží k přiblížení problematiky klasické kryptografie a kryptoanalýzy.

Druhá část se zabývá uplatněním znalostí a principů z teoretické části k implementaci vybraných algoritmů pro naprogramování aplikace schopné šifrovat či dešifrovat text, analyzovat a odhadnout typ užité šifry ze znalosti zašifrovaného textu. Popsal jsem zde strukturu programu a GUI, ukázky a podrobné vysvětlení jednotlivých útoků a jejich implementace do aplikace. Dále jsem popsal algoritmy užité pro analýzu a identifikaci jednotlivých šifer. Pro tuto část práce jsem zaznamenal velkou fluktuaci úspěšnosti identifikace šifrovacích algoritmů. Z konečného shrnutí výsledků kryptoanalýzy plyne, že nejméně se dařilo identifikovat Vigenèrovu šifru s úspěšností pouze 50 %.

Prolamování šifer na základě využitého šifrovacího algoritmu se ukázalo jako nejobtížnější část této práce. Pro prolamování jsem využíval co nejvíce metod, většina zvolených šifer však byla prolomena i nejjednoduššími metodami v řádech vteřin, výjimku tvořila šifra Railfence. Při konečném testování trvalo prolomení šifrovaného textu o délce 2080 znaků přes 3 hodiny. Z vybraných šifer se nepodařilo prolomit pouze jedinou, a to ADFGVX. Proces prolomení ADFGVX pro mě byl velmi obtížně automatizovatelný, a ani po implementaci několika různých algoritmů k prolomení se mi šifru nepodařilo prolomit.

Vytvořená aplikace shrnuje šifrovací principy a slabiny vybraných šifer a implementuje metody klasické kryptoanalýzy pro prolamování a analýzu těchto šifer. Jejím přínosem je možnost seznámení se základy klasické kryptoanalýzy a čtenářovo prohloubení znalostí kryptologie.

SEZNAM POUŽITÉ LITERATURY

[1] ŠENKEŘÍK, Roman. Přednášky z předmětu Kryptologie. 2019-2021

[2] Základní pojmy v kryptologii. www.wikisoftia.cz [online]. 2013 [cit. 2022-05-09]. Dostupné z: https://wikisofia.cz/wiki/Z%C3%A1kladn%C3%AD_rozd%C4%9Blen%C3%AD_kryptologie

[3] SINGH, Simon. Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii. Praha: Dokořán, 2003, 382 s. Aliter. ISBN 8072034995.

[4] RODRIGUEZ-CLARK, Daniel. Monoalphabetic Substitution Ciphers. Https://crypto.interactive-maths.com/ [online]. https://www.interactive-maths.com/ [cit. 2022-05-09]. Dostupné z: https://crypto.interactive-maths.com/monoalphabetic-substitution-ciphers.html

[5] Polyalphabetic Substitution Ciphers. Crypto-it [online]. 2020 [cit. 2022-05-09]. Dostupné z: http://www.crypto-it.net/eng/simple/polyalphabetic-substitution-ciphers.html

[6] Blaise de Vigenere. www.geocaching.com [online]. [cit. 2022-05-09]. Dostupné z: https://www.geocaching.com/geocache/GC8590W_blaise-de-vigenere?guid=6af0699b-7a0d-45a5-87f7-6a7f0e693d97%C2%A8

[7] Playfair Cipher with Examples. Geeksforgeeks [online]. 2021, 24.12.2021 [cit. 2022-05-09]. Dostupné z: https://www.geeksforgeeks.org/playfair-cipher-with-examples/

[8] Block Cipher. https://www.tutorialspoint.com/ [online]. [cit. 2022-05-09]. Dostupné z: https://www.tutorialspoint.com/cryptography/block_cipher.htm

[9] JAŠEK, Roman. Informační a datová bezpečnost. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta managementu a ekonomiky, 2006, 140 s. ISBN 8073184567.

[10] DURČÁK, Pavel. Symetrické a asymetrické šifrování. www.napocitaci.cz [online]. 2018, 18.9.2018 [cit. 2022-05-09]. Dostupné z: https://www.napocitaci.cz/33/symetricke-a-asymetricke-sifrovani-uniqueidgOkE4NvrWuNY54vrLeM677jX7sp3Lu-ZpLp-GVMy1prA/

[11] Afinní šifra. www.algoritmy.net [online]. [cit. 2022-05-09]. Dostupné z: https://www.algoritmy.net/article/49/Afinni-sifra

[12] Modular multiplicative inverse. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-09]. Dostupné z: https://en.wikipedia.org/wiki/Modular_multiplicative_inverse

[13] Implementation of Affine Cipher. Geeksforgeeks [online]. 2021, 04.08.2021 [cit. 2022-05-09]. Dostupné z: https://www.geeksforgeeks.org/implementation-affine-cipher/

[14] ŽERAVÍK, Josef. Klasické kryptografické metody. PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO, 2012. BAKALÁŘSKÁ PRÁCE. UNIVERZITA PALACKÉHO. Vedoucí práce Mgr.Eduard Bartl, Ph.D.

[15] DUMESSA, Nathan. Let's Play Fair: A Deceptively Simple Cipher. Wondersandmarvels [online]. [cit. 2022-05-09]. Dostupné z: https://www.wondersandmarvels.com/2014/11/lets-play-fair.html#:~:text=The%20main%20weakness%20of%20the,very%20short%20amount%20of%20time

[16] ROT13 cipher. Geeksforgeeks [online]. 2022, 27.01.2022 [cit. 2022-05-09]. Dostupné z: https://www.geeksforgeeks.org/rot13-cipher/

[17] ROT13. Algoritmy.net [online]. [cit. 2022-05-11]. Dostupné z: https://www.algoritmy.net/article/41797/ROT13

[18] WEBB, Chris. Vigenère Cipher in Python. Codedrome [online]. 2020, 05.11.2020 [cit. 2022-05-09]. Dostupné z: https://www.codedrome.com/vigenere-cipher-in-python/

[19] Vigenère cipher. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-09]. Dostupné z: https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher#Kasiski_examination

[20] Historie šifer: Polybiův čtverec. Geocaching [online]. 2018, 09/12/2018 [cit. 2022-05-09]. Dostupné z: https://www.geocaching.com/geocache/GC7XN8A_historie-sifer-polybiuv-ctverec

[21] Šifra ADFGVX – ADFGVX cipher. Wikijii [online]. [cit. 2022-05-09]. Dostupné z: https://wikijii.com/wiki/ADFGVX_cipher

[22] ADFGVX cipher. Cryptography.fandom [online]. [cit. 2022-05-09]. Dostupné z: https://cryptography.fandom.com/wiki/ADFGVX_cipher

[23] RODRIGUEZ-CLARK, Daniel. Rail Fence Cipher. Crypto-it [online]. 2020 [cit. 2022-05-09]. Dostupné z: https://crypto.interactive-maths.com/rail-fence-cipher.html

[24] Rail Fence Cipher – Encryption and Decryption. Geeksforgeeks [online]. 2022, 22.03.2022 [cit. 2022-05-09]. Dostupné z: https://www.geeksforgeeks.org/rail-fence-cipher-encryption-decryption/

[25] RODRIGUEZ-CLARK, Daniel. Route Cipher. Crypto-it [online]. 2020 [cit. 2022-05-09]. Dostupné z: https://crypto.interactive-maths.com/rail-fence-cipher.html

[26] Route Cipher: TRANSPOSITION CIPHER. Crypto-it [online]. 2020, 2020.03.09 [cit. 2022-05-09]. Dostupné z: http://www.crypto-it.net/eng/simple/route-cipher.html#:~:text=The%20Route%20Cipher%20is%20a,path%20drawn%20on%20a%20grid

[27] Cryptography/Scytale. Wikibooks [online]. [cit. 2022-05-09]. Dostupné z: https://en.wikibooks.org/wiki/Cryptography/Scytale

[28] Prostý a šifrovaný text. Upadvice [online]. [cit. 2022-05-09]. Dostupné z: https://upadvice.net/cs/prosty-a-sifrovany-text#Co_je_to_kryptoanalyza

[29] RODRIGUEZ-CLARK, Daniel. Frequency Analysis: Breaking the Code: TRANSPOSITION CIPHER. Crypto-it [online]. 2020, 2020.03.09 [cit. 2022-05-09]. Dostupné z: https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html

[30] English Letter Frequencies. Practicalcryptography [online]. [cit. 2022-05-09]. Dostupné z: http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/

[31] BAUMSLAG, Gilbert, Benjamin FINE, Martin KREUZER a Gerhard ROSENBERGER. A Course in Mathematical Cryptography. Berlin/Boston: De Gruyter, 2015, 1 online zdroj. De Gruyter Textbook. ISBN 9783110386165. Dostupné také z: https://proxy.k.utb.cz/login?url=https://www.degruyter.com/openurl?genre=book&isbn=9783110372779

[32] RODRIGUEZ-CLARK, Daniel. Kasiski Analysis: Breaking the Code: TRANSPOSITION CIPHER. Crypto-it [online]. 2020, 2020.03.09 [cit. 2022-05-09]. Dostupné z: https://crypto.interactive-maths.com/kasiski-analysis-breaking-the-code.html

[33] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-05-11]. Dostupné z: https://en.wikipedia.org/wiki/Index_of_coincidence

[34] Keyword Length Estimation with Index of Coincidence. Pages.mtu.edu [online]. [cit. 2022-05-09]. Dostupné z: https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-IOC-Len.html

[35] FOJTOVÁ, Lucie a Karel BURDA, 2010. Softwarová podpora výuky klasické kryptoanalýzy: Software support of education in classical cryptoanalysis. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií.

[36] Index of Coincidence. Dcode.fr [online]. [cit. 2022-05-09]. Dostupné z: https://www.dcode.fr/index-coincidence

[37] Brute-Force Attack. Crypto-it [online]. 2020, 2020.03.09 [cit. 2022-05-09]. Dostupné z: http://www.crypto-it.net/eng/attacks/brute-force.html

[38] KNUDSEN, Lars a Matthew ROBSHAW. The block cipher companion. Berlin: Springer, c2011, xiv, 267 s. Information security and cryptography. Dostupné z: doi:9783642173424

SEZNAM PŘEVZATÉHO KÓDU

Zdrojové soubory: *breakCipher.py* a *analysis.py*

DAVID, Robin. Kasiski-Babbage Cryptanalysis in Python. Robindavid [online]. 2012, 15.06.2012[cit.2022-05-13].Dostupné z: http://www.robindavid.fr/blog/2012/06/15/kasiski-babbage-cryptanalysis-in-python/

1,000 most common US English words. Github [online]. [cit. 2022-05-13]. Dostupné z: https://gist.github.com/deekayen/4148741

Frekvenční seznam (čeština)/ČNK SYN2005/1-1000. Wiktionary [online]. [cit. 2022-05-13].Dostupné:https://cs.wiktionary.org/wiki/P%C5%99%C3%ADloha:Frekven%C4%8Dn%C3%AD_seznam_(%C4%8De%C5%A1tina)/%C4%8CNK_SYN2005/1-1000

Zdrojový soubor: *route.py*

Print matrix in spiral order. Techiedelight [online]. [cit. 2022-05-13]. Dostupné z: https://www.techiedelight.com/print-matrix-spiral-order/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GUI Grafické uživatelské rozhraní

mod modulo

SEZNAM OBRÁZKŮ

Obrázek 1.: Členění kryptografie [1] 11
Obrázek 2.: Substituce v ROT13[17] 18
Obrázek 3.: Vigenèrův čtverec pro šifrování [18] 19
Obrázek 4.: Standardní rozložení znaků pro anglický text [29] 28
Obrázek 5.: Rozložení znaků pro text ATTACKATNOON zašifrováno ROT13 28
Obrázek 6.: Nejčastěji vyskytující se bigramy pro anglický text s frekvencemi výskytu [30] 29
Obrázek 7.: Nejčastěji vyskytující se trigramy pro anglický text s frekvencemi výskytu [30] 29
Obrázek 8.: Šifrovací aplikace – záložka pro šifrování Afinní šifrou s popisem prvků 34
Obrázek 9.: Šifrovací aplikace – záložka pro analýzu šifrovaného textu a popis prvků 36
Obrázek 10.: Šifrovací aplikace – záložka pro prolamování šifrovaného textu a popis prvků 37
Obrázek 11.: Prolomení Afinní šifry hrubou silou 41
Obrázek 12.: Grafy k prolomení Afinní šifry frekvenční analýzou 42
Obrázek 13.: Prolamování Vigenèrovy šifry 43

SEZNAM TABULEK

Tabulka 1.: Nahoře abeceda otevřeného textu, dole abeceda zašifrovaná klíčem s parametry a=5, b=9. 15
Tabulka 2.: Sestavená tabulka pro šifrování bigramů.....16
Tabulka 3.: Šifrování prvního bigramu dle řádkové substituce.....17
Tabulka 4.: Šifrování čtvrtého bigramu dle sloupcové substituce.....17
Tabulka 5.: Šifrování druhého bigramu dle úhlopříčné substituce.....17
Tabulka 6.: Vigenèrův čtverec pro šifrování HELLOWORLD s klíčem KEY.....20
Tabulka 7.: Náhodná abeceda pro získání druhé tabulky ADFGVX. 21
Tabulka 8.: Přiřazená písmena řádků a sloupců pro každý znak otevřeného textu. ...22
Tabulka 9.: Nová tabulka s klíčem k zašifrování.....22
Tabulka 10.: Transponovaná tabulka dle abecedního pořadí znaků klíče. 22
Tabulka 11.: Šifrovací tabulky Railfence šifry.....23
Tabulka 12.: Doplnění prvního řádku pro dešifrování Railfence 23
Tabulka 13.: Šifrovací tabulka Railfence s mezerami započítanými v abecedě.....24
Tabulka 14.: Šifrovací tabulka Route šifry při sloupcové transpozici.....24
Tabulka 15.: Šifrovací tabulka Route šifry při transpozici obrácené spirály.....25
Tabulka 16.: Šifrovací tabulka Scytale pro klíč 3.....25
Tabulka 17.: Výsledky kryptoanalýzy Afinní šifry 45
Tabulka 18.: Výsledky kryptoanalýzy šifry ROT13.....46
Tabulka 19.: Výsledky kryptoanalýzy šifry Route 46
Tabulka 20.: Výsledky kryptoanalýzy Vigenèrovy šifry.....46
Tabulka 21.: Výsledky kryptoanalýzy šifry Scytale.....46
Tabulka 22.: Výsledky kryptoanalýzy pro šifru Railfence 47
Tabulka 23.: Výsledky kryptoanalýzy pro šifru Playfair.....47

SEZNAM PŘÍLOH

Příloha P I: CD

PŘÍLOHA P I: CD

CD obsahuje adresář “prilohy“ obsahující soubor “main.py“, “bpdraftgui.ui“, “ui_mainWindow.py“, “usefull_functions.py“ a podadresáře “src“ a “gui“ obsahující jednotlivé podsoubory programu a GUI. Dalším částí přílohy je elektronická verze této bakalářské práce: *fulltext.pdf*, a seznam otevřených a zašifrovaných textů použitých pro analýzu v souboru *analyza_texty.pdf*.