

Elektronická podpora výuky předmětu Kryptologie - šifry v prostředí webMathematica

Electronic Support of the Course Cryptology - Cipher in
Environment webMathematica

Ondřej Dlesk

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej DLESK**

Osobní číslo: **A09099**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **prezenční**

Téma práce: **Elektronická podpora výuky předmětu Kryptologie - šifry v prostředí WebMathematica**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Vytvořte webovou prezentaci předmětu Kryptologie.
3. Vytvořte ukázky šifer v prostředí WebMathematica.
4. Věnujte pozornost zabezpečení aplikace.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. SINGH, Simon. *Kniha kódů a šifer*. Argo, 2003. ISBN: 80-7203-499-5.
2. JANEČEK, J. *Odhalená tajemství šifrovacích klíčů minulosti*. Naše Vojsko, 1994.
3. VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Albatros, 2006. ISBN 80-00-01888-8
4. HANŽL, T. *Šifry a hry s nimi*. Portál, 2007. ISBN 978-80-7367-196-9.
5. KATZ, Jonathan. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall, 1 edition. 2007. 552 s. ISBN 978-1584885511.
6. MURPHY, Sean. *Kryptografie - Průvodce pro každého*. Dokořán, 2006. 157 s. ISBN 80-7363-074-5.
7. HOSTE, Jim. *Mathematica DeMYSTiFied*. McGraw-Hill Professional, 2008. 408 s. ISBN 978-0071591447.

Vedoucí bakalářské práce:

Ing. Michal Pluháček

Ústav matematiky

Datum zadání bakalářské práce:

24. února 2013

Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ka ředitel ústavu

ABSTRAKT

Práce se zabývá vybranými klasickými metodami kryptologie, mezi které patří šifrování a dešifrování především substitučními a transpozičními metodami. Důležitou součástí práce jsou také webové stránky, ve kterých jsou implementovány metody vybraných šifer. Zmíněné šifry byly nejdříve naprogramovány v programu Mathematica a poté implementovány do prostředí webMathematica. Stránky slouží jako elektronická podpora výuky kryptologie, aby si každý student mohl vyzkoušet danou šifru v internetovém prohlížeči a zkontrolovat správnost jeho šifer. Za předpokladu základní znalosti XHTML jazyka a programování v prostředí Mathematica by čtenář po přečtení této práce měl znát základy teorie uvedených šifer a základní práci v prostředí webMathematica.

Výsledky této práce se budou využívat jako studijní materiály při výuce na Univerzitě Tomáše Bati na Fakultě Aplikované Informatiky.

Klíčová slova: Mathematica, webMathematica, kryptologie, šifrování, dešifrování, XHTML

ABSTRACT

This thesis is addressing the classic methods of cryptology, which contain cyphering and deciphering and especially the substitutional and transpositional methods. Another important part of the thesis is a website, which implements several methods of the selected cyphers. These cyphers were first programmed in the Mathematica software and later implemented into the webMathematica environment. The website serves as an electronic support for study of cryptology, so that every student may experience the given cypher inside an internet browser and check the correctness of his or her cyphers. Assuming that the student has basic knowledge of the XHTML language and the Mathematica programming environment, he or she should know the basics of the cyphers mentioned above and the basics of the webMathematica environment, after having read this thesis.

Keywords: Mathematica, webMathematica, cryptology ,cyphering, decyphering, XHTML

Mé poděkování patří zejména, vedoucímu bakalářské práce, kterým je Ing. Michal Pluháček, za rady, připomínky a návrhy týkající se jak teoretické, tak praktické části bakalářské práce.

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 a jeden výtisk bude uložen u vedoucího práce;
- 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 s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- 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

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 SLOVNÍK POJMŮ	11
2 KRYPTOLOGIE	13
2.1 KRYPTOGRAFIE	14
2.2 KRYPTOANALÝZA	15
2.3 STEGANOGRAFIE	15
3 KRYPTOLOGICKÉ METODY	16
3.1 TRANSPOZICE	16
3.1.1 Transpoziční šifra v jedné tabulce se dvěma klíči	16
3.1.1.1 Šifrování.....	16
3.1.1.2 Dešifrování.....	17
3.1.1.3 Příklad.....	17
3.1.2 Transpoziční šifra ve dvou tabulkách se dvěma klíči	19
3.1.2.1 Šifrování.....	19
3.1.2.2 Dešifrování.....	19
3.1.2.3 Příklad.....	20
3.2 SUBSTITUCE	22
3.2.1 Afinní šifra	22
3.2.1.1 Šifrování.....	22
3.2.1.2 Dešifrování.....	22
3.2.1.3 Příklad.....	23
3.2.2 Caesarova šifra	23
3.2.2.1 Šifrování.....	23
3.2.2.2 Dešifrování.....	24
3.2.2.3 Příklad.....	24
3.2.3 Monoalfabetická substituční šifra	24
3.2.3.1 Šifrování.....	25
3.2.3.2 Dešifrování.....	25
3.2.3.3 Příklad.....	25
3.2.4 PlayFairova šifra	26
3.2.4.1 Šifrování.....	26
3.2.4.2 Dešifrování.....	27
3.2.4.3 Příklad.....	27
3.2.5 Vernamova šifra – analogová.....	28
3.2.5.1 Šifrování.....	28
3.2.5.2 Dešifrování.....	28
3.2.5.3 Příklad.....	28
3.2.6 Vernamova šifra – binární.....	29
3.2.6.1 Šifrování.....	29
3.2.6.2 Dešifrování.....	29
3.2.6.3 Příklad.....	30
3.2.7 Vigenérova šifra	30
3.2.7.1 Šifrování.....	31
3.2.7.2 Dešifrování.....	31
3.2.7.3 Příklad.....	32

3.3	ASYMETRICKÉ ŠIFROVÁNÍ	32
3.3.1	RSA šifra	33
3.3.1.1	Šifrování.....	34
3.3.1.2	Dešifrování.....	35
3.3.1.3	Příklad.....	35
4	ÚVOD DO PROGRAMU WEBMATHEMATICA	36
II	PRAKTICKÁ ČÁST	37
5	WEBMATHEMATICA	38
5.1	PROGRAMOVÁNÍ STRÁNEK	38
5.1.1	Základní kostra dokumentu	39
5.1.2	Vkládání funkčních bloků kódu z Mathematicy	39
5.1.3	Vstupní data	40
5.1.4	Zobrazení výsledku	41
5.1.5	Bezpečnost	41
6	WEBOVÉ STRÁNKY	42
6.1	ROZVRŽENÍ STRÁNEK	42
6.2	OVLÁDÁNÍ STRÁNEK	43
6.3	PŘÍKLAD ZDROJOVÉHO KÓDU STRÁNKY	44
6.3.1	Popis zdrojového kódu.....	46
ZÁVĚR	51	
ZÁVĚR V ANGLIČTINĚ.....	52	
SEZNAM POUŽITÉ LITERATURY.....	53	
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	55	
SEZNAM OBRÁZKŮ	56	
SEZNAM PŘÍLOH.....	58	

ÚVOD

Potřebu skrývat nebo utajovat určité informace lidstvo pociťuje snad od samého počátku svého vzniku. Utajované informace jsou a byly důležité v různých odvětvích lidského úsilí, od milostné komunikace až po válečné konflikty. Jedna z prvních zmínek o šifrování neboli kryptologii a utajení zprávy je v Kamasútře, která je datována do 4. století, její autor však čerpal z pramenů až o 800 let starších.

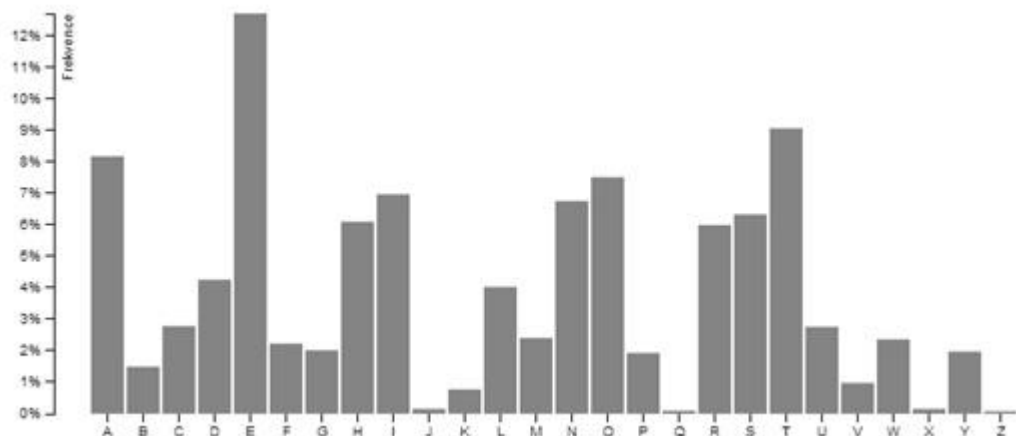
Kryptologie je rozsáhlý vědní obor, do kterého spadá kryptografie, kryptoanalýza a steganografie. Kryptologie se může dále dělit na klasickou a moderní kryptologii. Klasická kryptologie trvala přibližně do 1. poloviny 20. století a vyznačovala se tím, že k šifrování zprávy bylo potřeba jen tužky, papíru případně další jednoduché pomůcky. Moderní kryptologie začala vznikat během 1. poloviny 20. století, kdy se začaly vyvíjet šifrovací stroje umožňující složitější, především matematické šifrovací postupy. V dnešní moderní kryptologii se nepoužívají zádě speciální šifrovací stroje nýbrž klasické počítače. Tato práce se zabývá především klasickou kryptologií. [1, 10]

Na Fakultě Aplikované Informatiky Univerzity Tomáše Bati ve Zlíně se problematice klasické kryptologie věnuje předmět pod názvem Kryptologie, ve kterém se studenti učí principy vybraných šifer, které dále programují v programu Mathematica. Nástavbou tohoto programu je webMathematica, což je nástroj umožňující přenos projektů z prostředí Mathematica do webového prohlížeče.

I. TEORETICKÁ ČÁST

1 SLOVNÍK POJMŮ

- *ASCII*: American Standard Code for Information Interchange - standard pro převod znaků do číselné podoby.
- *Asymetrické šifry*: pro šifrování se používá veřejný klíč a pro dešifrování klíč soukromý.
- *Blokové šifry*: šifrování se provádí po blocích bitů.
- *Dešifrování*: převod šifrovaného textu zpět na text otevřený.
- *Dešifrovaný text*: převedený text z textu šifrovaného, také otevřený text.
- *Frekvenční analýza*: každé písmeno je v každém jazyce zastoupeno s jinou četností. Pokud víme, v jakém jazyce je zpráva napsána, můžeme u některých šifrovacích postupů, v závislosti na četnosti znaků, určit, který znak zastupuje který. Tato četnost se zobrazuje pomocí histogramu.
- *Histogram*: zobrazení četnosti znaků textu do grafu. Na ose „x“ jsou znaky abecedy a na ose „y“ procentuální zastoupení oněch znaků.



Obr. 1: Histogram textu [8]

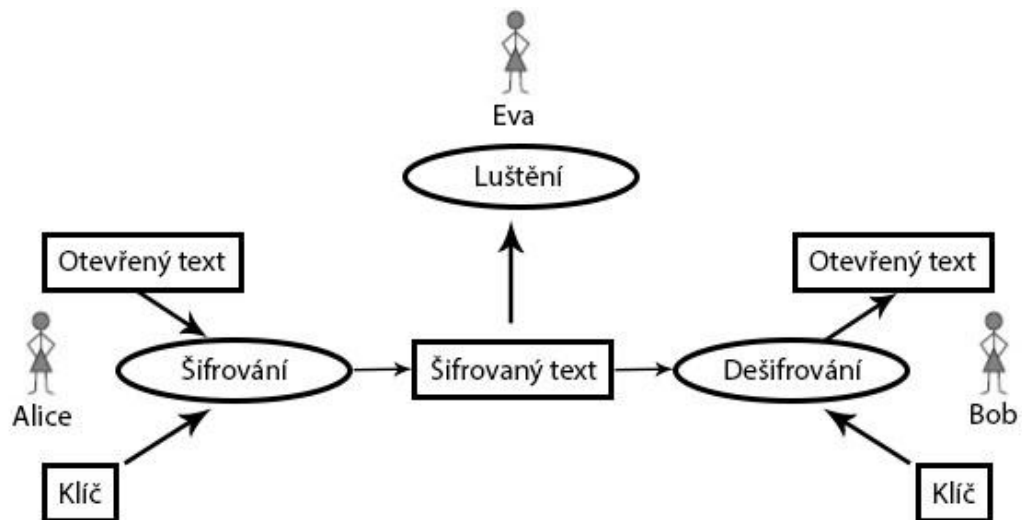
- *Homofonní substituce*: jeden znak může být šifrován různými znaky.
- *Klíč*: element, který změní obecný algoritmus šifrování v konkrétní postup. Entita určující flexibilitu šifrování.

- *Kryptoanalýza*: věda o tom, jak bez znalosti klíče odvodit otevřený text z textu šifrovaného.
- *Kryptografie*: věda o šifrování zpráv a zatajení jejich smyslu.
- *Kryptologie*: věda o utajení zprávy ve všech formách zahrnující kryptografii a kryptoanalýzu.
- *Monoalfabetická substituce*: šifrování pracuje jen s jedinou šifrovou abecedou.
- *Otevřený text*: původní zpráva, zpráva před zašifrováním.
- *Polyalfabetická substituce*: šifrování pracuje s několika abecedami.
- *Proudové šifry*: šifrování se provádí po jednotlivých bitech.
- *Soukromý klíč*: klíč používaný příjemcem k dešifrování zprávy u asymetrického šifrování.
- *Steganografie*: nauka o skrývání existence zprávy, na rozdíl od kryptologie, což je nauka o skrývání obsahu zprávy.
- *Substituce*: nahrazení znaku jiným znakem nebo skupinou znaků.
- *Symetrické šifry*: šifrování a dešifrování používá jeden a ten samý klíč.
- *Šifra*: jakýkoliv systém ukrývající smysl zprávy tak, že původní písmeno, podle určitých pravidel, nahradí jiným písmenem. Systém by měl obsahovat určitou flexibilitu (klíč).
- *Šifrová abeceda*: přeskupení originální abecedy, které určuje, jak je každé písmeno šifrováno. Může být sestavena ze znaků, čísel, dokonce i obrázků.
- *Šifrování*: postup, který převádí otevřený text na text šifrovaný.
- *Šifrovaný text*: otevřený text, který projde šifrovacím procesem. Na první pohled nesmyslný.
- *Transpozice*: záměna pořadí znaků v otevřeném textu.
- *Útok hrubou silou*: útočník zkouší všechny možné varianty klíče.
- *Veřejný klíč*: klíč používaný veřejností k zašifrování zprávy pro příjemce, který ji svým soukromým klíčem dešifruje. [1]

2 KRYPTOLOGIE

Kryptologie je věda zabývající se utajením zpráv, ve formách zahrnujících kryptografii, kryptoanalýzu a steganografii. Kryptologie se po staletí vyvíjela od nejjednodušších klasických šifer až do dnešní moderní kryptologie, což jsou složité matematické postupy jak šifrování textu tak i jeho luštění.

Níže uvedený obrázek nám znázorňuje základní šifrovanou komunikaci, kde Alice šifrovaně komunikuje s Bobem. Alice nejdříve otevřený text zašifruje pomocí předem domluveného klíče, tuto zprávu pošle Bobovi, který ji pomocí klíče zase dešifruje. Může také nastat situace, kdy šifrovaný text zachytí narušitel Eva, která nemá klíč. Pokud tedy chce Eva znát otevřený text, musí šifrovaný text pomocí kryptoanalytických metod rozluštit.



Obr. 2: Šifrovaná komunikace

Zde je souhrnný přehled šifer a jejich rozdělení podle principů šifrovacích metod. Významy jednotlivých pojmů jsou vysvětleny ve výše uvedeném slovníku pojmů.

- Klasické šifry
 - Transpoziční šifrování
 - Substituční šifrování
 - Monoalfabetická substituce
 - Homofonní substituce
 - Polyalfabetická substituce
- Mechanické šifry
- Moderní šifry
 - Symetrické šifrování
 - Blokované šifrování
 - Proudové šifrování
 - Asymetrické šifrování

[6]

2.1 Kryptografie

Samotné slovo kryptografie je sloučenina dvou řeckých slov kryptós – skrytý a gráphein – psát.

Kryptografie se často používá jako alternativa ke kryptologii. Jedná se o vědu, zabývající se utajením obsahu zpráv, převodem do podoby, která je bez dodatečné informace nečitelná. Tato dodatečná informace je známá jako klíč. O tento převod se stará určitá šifrovací metoda, kdy se jeden znak nahradí jiným znakem či skupinou znaků, tato zpráva se pak nazývá šifrovaný text. Zpětným převodem šifrovaného textu na text otevřený se zabývá metoda zvaná dešifrování.

Dalo by se říct, že za vývoj kryptografie vděčíme kryptoanalýze. Jakmile byla vyvinuta nová šifra, nedalo to spát lidské zvědavosti a kryptoanalytici se snažili tuto šifru prolomit, jakmile ji prolomili, šifra se stala bezcennou a bylo zapotřebí vyvinout jinou, dokonalejší šifru. [1]

2.2 Kryptoanalýza

Kryptoanalýza je postup, kdy se kryptoanalytici snaží ze šifrovaného textu zjistit text otevřený. Zkoumají buďto šifrovací metodu a snaží se nalézt její slabinu, pomocí které by prolomili šifrovaný text. Další možností je zkoumání samotného šifrovaného textu, kdy se na základě zkoumání snaží převést šifrovaný text na otevřený text. Dalo by se říct, že kryptoanalýza také zkoumá odolnost kryptografických metod.

Kryptoanalytické metody:

- Frekvenční analýza (účinná na substitučních šifrách)
- Útok hrubou silou
- Přeskupování řádků a sloupců v tabulce (transpoziční šifry)
- Slovníková metoda hledání klíče
- „Náhodné“ přeskupování písmen nebo zkoušení klíče

2.3 Steganografie

Slovo steganografie taktéž pochází z řeckých slov, kterými jsou steganós – schovaný a gráphein – psát. Steganografie se zabývá utajováním komunikace a samotných zpráv. Hlavní rozdíl mezi steganografií a kryptografií je, že kryptografie se snaží ukrýt obsah zprávy, kdežto steganografie se snaží ukrýt existenci zprávy.

Způsobů ukrývání textu je několik. Patří sem neviditelné inkousty, které jdou vidět například pod určitým spektrem světla nebo po zahřátí, voskové tabulky, kdy byl text ukryt pod vrstvu vosku, ukrytí textu do zvukové stopy nebo obrázku a mikrotečky. [10]

Nevýhodou steganografie je, že pokud narušitel komunikace odhalí skrytý text, je pro něj okamžitě čitelný. Proto se steganografie často kombinuje s kryptografií. Otevřený text se nejdříve zašifruje a poté se ukryje.

3 KRYPTOLOGICKÉ METODY

Jak již bylo řečeno, kryptologické metody se dělí do několika skupin, podle principu jednotlivých metod. V této práci se zabývám především transpozičními a substitučními metodami.

3.1 Transpozice

Transpoziční šifra neboli transpozice je jedna ze základních kryptografických operací. Jedná se o symetrickou šifru (pro šifrování i dešifrování se používá tentýž klíč). Principem transpoziční šifry je záměna pořadí neboli zamíchání jednotlivých znaků textu dle určitého pravidla. Výhodou je jednoduchost postupu šifrování a dešifrování. Nevýhodou je poměrně snadná analýza šifrovaného textu, snadné odhalení jazyka otevřeného textu pomocí frekvenční analýzy.

Transpoziční šifru lze prolomit například pomocí metody s využitím vycpávky. Vycpávka (padding) = „X“ (doplnění „X“ na konec textu aby se zarovnal na plný obdélník) Pokud víme, že se jedná o transpoziční šifru, je nejefektivnější tento typ útoku. Ze šifrovaného textu můžeme vyčíst počet pozic mezi paddingy. Pokud má šifrovaný text např. 35 znaků a vzdálenost paddingů je 7, můžeme říct, že původní obdélník měl 5 sloupců a 7 řádků. [8]

3.1.1 Transpoziční šifra v jedné tabulce se dvěma klíči

Transpoziční šifra je jednou ze základních kryptografických operací, která mění pořadí znaků v otevřeném textu. Otevřený text nejdříve zapíšeme do tabulky, ve které setřídíme řádky a sloupce podle klíčů.

3.1.1.1 Šifrování

Jak již bylo řečeno, otevřený text se po řádcích zapíše do tabulky, která má tolik sloupců, kolik má první klíč znaků. Pokud není tabulka zcela vyplněna, doplníme ji znaky „X“ nebo jinými méně používanými znaky. Druhý klíč musíme zvolit tak, aby měl tolik znaků, kolik má tato tabulka řádků. Dále postupujeme tak, že abecedně seřadíme sloupce dle prvního klíče a řádky dle klíče druhého. Pokud se v klíči vyskytují stejné znaky, rozhoduje pořadí těchto znaků v klíči. Šifrovaný text potom čteme po sloupcích.

3.1.1.2 Dešifrování

Dešifrování probíhá podobným způsobem, jen s tím rozdílem, že tabulka má tolik sloupců, kolik má druhý klíč znaků. Tedy, zapíšeme šifrovaný text do tabulky dle kritéria v předchozí větě, dále abecedně seřadíme sloupce dle druhého klíče a řádky dle klíče druhého. Dešifrovaný text potom čteme po sloupcích.

3.1.1.3 Příklad

Šifrování

Otevřený text: TRANSPOZICNISIFRA

Klíč 1: CAJK

Klíč 2: CAJE

	C	A	J	I	K
C	T	R	A	N	S
A	P	O	Z	I	C
J	N	I	S	I	F
E	R	A	X	X	X

Obr. 3: Zápis do tabulky

	A	C	I	J	K
C	R	T	N	A	S
A	O	P	I	Z	C
J	I	N	I	S	F
E	A	R	X	X	X

Obr. 4: Šifrování podle prvního klíče

	A	C	I	J	K
A	O	P	I	Z	C
C	R	T	N	A	S
E	A	R	X	X	X
J	I	N	I	S	F

Obr. 5: Šifrování podle druhého klíče

Šifrovaný text: ORAIP TRNIN XIZAX SCSXF

Dešifrování

Šifrovaný text: ORAIP TRNIN XIZAX SCSXF

Klíč 1: CAJIK

Klíč 2: CAJE

	C	A	J	E
C	O	R	A	I
A	P	T	R	N
J	I	N	X	I
I	Z	A	X	S
K	C	S	X	F

Obr. 6: Zápis do tabulky

	A	C	E	J
C	R	O	I	A
A	T	P	N	R
J	N	I	I	X
I	A	Z	S	X
K	S	C	F	X

Obr. 7: Dešifrování podle druhého klíče

	A	C	E	J
A	T	P	N	R
C	R	O	I	A
I	A	Z	S	X
J	N	I	I	X
K	S	C	F	X

Obr. 8: Dešifrování podle prvního klíče

Dešifrovaný text: TRANSPOZICNISIFRAXXX

3.1.2 Transpoziční šifra ve dvou tabulkách se dvěma klíči

Tato transpoziční šifra je velice podobná předchozí jednoduché transpoziční šifře jen s rozdílem, že využívá dvou tabulek, ve kterých se v každé šifruje jiným klíčem. Nejdříve se otevřený text zapíše do první tabulky, zašifruje se pomocí prvního klíče, mezitext se zapíše do druhé tabulky a znovu se zašifruje dle druhého klíče.

3.1.2.1 Šifrování

Otevřený text se zapíše do první tabulky, která má tolik sloupců, kolik má první klíč znaků. Pokud není tabulka zcela zaplněna, doplní se znaky „X“ nebo jinými méně používanými znaky. Sloupce této tabulky se abecedně seřadí dle prvního klíče. Mezitext čteme po sloupcích, který dále zapíšeme po řádcích do druhé tabulky, která má tolik sloupců, kolik má druhý klíč znaků. Pozor, tuto tabulku již nesmíme doplňovat! Délka druhého klíče musí být beze zbytku dělitelná s délkou mezitextu. Sloupce této tabulky se abecedně seřadí dle druhého klíče. Šifrovaný text čteme po sloupcích z druhé šifrované tabulky.

3.1.2.2 Dešifrování

Šifrovaný text zapíšeme do první tabulky, která má tolik sloupců, kolik má druhý klíč znaků. Sloupce této tabulky abecedně seřadíme dle prvního klíče. Mezitext čteme po sloupcích a zapíšeme jej po řádcích do druhé tabulky, která má tolik sloupců, kolik má první klíč znaků. Sloupce této tabulky abecedně seřadíme dle prvního klíče. Dešifrovaný text čteme po řádcích z druhé dešifrované tabulky.

3.1.2.3 Příklad

Šifrování

Otevřený text: TRANSPOZICNISIFRA

Klíč 1: CAJIK

Klíč 2: CAJE

C	A	J	I	K
T	R	A	N	S
P	O	Z	I	C
N	I	S	I	F
R	A	X	X	X

Obr. 9: Zápis do tabulky

A	C	I	J	K
R	T	N	A	S
O	P	I	Z	C
I	N	I	S	F
A	R	X	X	X

Obr. 10: Šifrování podle prvního klíče

C	A	J	E
R	O	I	A
T	P	N	R
N	I	I	X
A	Z	S	X
S	C	F	X

Obr. 11: $(1. \text{šifrování})^{-1}$

A	C	E	J
O	R	A	I
P	T	R	N
I	N	X	I
Z	A	X	S
C	S	X	F

Obr. 12: Šifrování podle druhého klíče

Šifrovaný text: OPIZC RTNAS ARXXX INISF

Dešifrování

Šifrovaný text: OPIZC RTNAS ARXXX INISF

Klíč 1: CAJIK

Klíč 2: CAJE

C	A	J	E
O	R	A	I
P	T	R	N
I	N	X	I
Z	A	X	S
C	S	X	F

Obr. 13: Zápis do tabulky

A	C	E	J
R	O	I	A
T	P	N	R
N	I	I	X
A	Z	S	X
S	C	F	X

Obr. 14: Dešifrování podle druhého klíče

C	A	J	I	K
R	T	N	A	S
O	P	I	Z	C
I	N	I	S	F
A	R	X	X	X

Obr. 15: (1. dešifrování)⁻¹

A	C	I	J	K
T	R	A	N	S
P	O	Z	I	C
N	I	S	I	F
R	A	X	X	X

Obr. 16: Dešifrování podle prvního klíče

Dešifrovaný text: TRANSPOZICNISIFRAXXX

3.2 Substitute

Substituční šifra neboli substitute je další ze základních kryptografických operací. Jedná se o symetrickou šifru (pro šifrování i dešifrování se používá tentýž klíč). Principem substituční šifry je nahrazení znaků jedné abecedy znaky abecedy druhé. V závislosti na typu šifry to mohou být znaky, skupiny znaků nebo další zástupné symboly.

Substituční šifru lze prolomit pomocí frekvenční analýzy. V každém jazyce se každé písmeno využívá s jinou četností. V českém jazyce se nejvíce využívá „O“ (8%) v jazyce anglickém má nejčastější zastoupení písmeno „E“ (12%). [1, 8]

3.2.1 Afinní šifra

Afinní šifra nebo také lineární posun nebo také lineární transformace. Tato šifra, do určité míry, eliminuje hlavní nevýhodu Caesarovy šifry, kterou je málo možností transformace daného písmene a tím je velmi jednoduchá kryptoanalýza.

3.2.1.1 Šifrování

Šifrování se provádí pomocí jednoduché rovnice

$$C_i = a \cdot T_i + b \cdot \text{mod}(m) \quad (1)$$

C_i ... i-té písmeno šifrovaného textu

T_i ... i-té písmeno otevřeného textu

a ... parametr a

b ... parametr b

m ... modulo (obvykle 26) [8]

3.2.1.2 Dešifrování

Dešifrování se provádí pomocí podobné rovnice

$$T_i = a^{-1} \cdot (C_i - b) \cdot \text{mod}(m) \quad (2)$$

a^{-1} ... multiplikativní inverze a v Z_m [8]

3.2.1.3 Příklad

Šifrování

Otevřený text: AFINNISIFRA

Parametr a: 3

Parametr b: 5

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Šifrová abeceda: FILORUXADGJMPSVYBEHKNQWZC

Například: A = 0 -> 3 * 0 + 5 = 5 => F

Šifrovaný text: FUDSS DHDUE F

Dešifrování

Šifrovaný text: FUDSS DHDUE F

Parametr a: 3

Parametr b: 5

Šifrová abeceda: FILORUXADGJMPSVYBEHKNQWZC

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Dešifrovaný text: AFINNISIFRA

3.2.2 Caesarova šifra

Caesarova šifra je jednou ze základních substitučních šifer. Princip šifry spočívá v posunu každého znaku otevřeného textu o určitý počet znaků v abecedě. Původní Caesarova šifra měla konstantní posun roven 3, dále se tato šifra používala s libovolným posunutím v abecedě (1-25). [1]

3.2.2.1 Šifrování

Šifrování se provádí dle následujícího vzorce:

$$C_i = (T_i + k) \cdot \text{mod}(m) \quad (3)$$

C_i ... i-té písmeno šifrovaného textu

T_i ... i-té písmeno otevřeného textu

k ... posun

m ... délka abecedy

3.2.2.2 Dešifrování

Princip dešifrování je přesně opačný, písmena se v abecedě posouvají opačným směrem než při šifrování.

$$T_i = (C_i - k) .mod(m) [8] \quad (4)$$

3.2.2.3 Příklad

Šifrování

Otevřený text: CAESAROVASIFRA

Klíč: 3

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Šifrová abeceda: DEFGHIJKLMNOPQRSTUVWXYZABC

Šifrovaný text: FDHVD URYDV LIUD

Dešifrování

Šifrovaný text: FDHVD URYDV LIUD

Klíč: 3

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Šifrová abeceda: DEFGHIJKLMNOPQRSTUVWXYZABC

Dešifrovaný text: CAESAROVASIFRA

3.2.3 Monoalfabetická substituční šifra

Monoalfabetická šifra je nejjednodušší typ substituční šifry, kdy se každé písmeno otevřeného textu nahradí písmenem šifrované abecedy. Šifrovaná abeceda může být posloupnost, písmen, znaků dokonce i obrázků.

Před samotným šifrováním se musíme nejdříve vytvořit takzvanou tabulku substitucí. Tuto tabulku můžeme vytvořit libovolným způsobem, avšak musíme dodržet, že zobrazení musí být bijektivní. Každé písmeno otevřeného textu musí mít jedinečný obraz v šifrovaném textu a naopak. [11]

3.2.3.1 Šifrování

Nejprve si musíme upravit otevřený text, tak aby vyhovoval našemu šifrování. V našem případě umí šifrovací funkce šifrovat jen znaky anglické abecedy, tedy znaky bez diakritiky a interpunkce.

V dalším kroku si vytvoříme tabulku substitucí. Nejdříve si zvolíme klíč, ten zapíšeme do tabulky jako první a dále pokračujeme znaky abecedy ovšem bez znaků, které jsou již obsazeny v klíči.

Nakonec šifrujeme otevřený text znak po znaku. Přečteme znak, nalezneme jej v tabulce substitucí a nahradíme odpovídajícím znakem. [11]

3.2.3.2 Dešifrování

Dešifrování probíhá totožně jako šifrování. Nejdříve si podle klíče vytvoříme tabulku substitucí a dále pokračujeme znak po znaku šifrovaného textu a nahrazujeme znaky pomocí substituční tabulky.

3.2.3.3 Příklad

Šifrování

Otevřený text: MONOALFABETICKSUBSTITUTE

Klíč: KLIC

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Šifrová abeceda: KLICABDEFGHJMNOPQRSTUVWXYZ

Šifrovaný text: MONOK JBKLA TFIHS ULSTF TUIA

Dešifrování

Šifrovaný text: MONOK JBKLA TFIHK SULST FTUIA

Klíč: KLIC

Šifrová abeceda: KLICABDEFGHJMNOPQRSTUVWXYZ

Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Dešifrovaný text: MONOALFABETICKASUBSTITUTE

3.2.4 PlayFairova šifra

Playfairova šifra je polyalfabetická substituce, která není pojmenovaná podle jejího autora (Charles Wheatston), ale podle jejího největšího propagátora (Lyon Playfair). Tato šifra je odolná vůči frekvenční analýze. [6]

3.2.4.1 Šifrování

Úprava otevřeného textu a klíče

- Otevřený text a klíč zbavíme diakritiky a interpunkce, v anglické verzi, pokud *OT* obsahuje písmeno „J“, nahradíme jej písmenem „I“. V české verzi se nahrazuje písmeno „W“ písmenem „V“.
- Takto upravený otevřený text rozdělíme do párů (bigramů).
- Jestliže se v bigramu vyskytnou stejná písmena, musí být rozdělena písmenem „X“ nebo „Z“.
- Pokud není poslední bigram kompletní, doplní se na konec taktéž písmeno „X“ nebo „Z“.

Playfairův čtverec

Nejprve si zvolíme klíč, ten by měl mít minimálně 5 znaků, přičemž se žádný znak nesmí opakovat. Klíč zapíšeme do prvního řádku čtvercové matice, která má rozměry 5 x 5. Dále pokračujeme v psaní abecedy, ve které vynecháme již použitá písmena v klíči.

Samotné šifrování

Šifrování je založeno na skutečnosti, že písmena každého bigramu se mohou vyskytovat ve třech pozicích vůči sobě v Playfarově čtverci a to ve stejném řádku, sloupci nebo v jiném řádku a sloupci. Šifrování pak probíhá dle následujících pravidel:

- Stejný řádek: písmeno se nahradí písmenem vpravo od něj, pokud je písmeno na konci řádku, nahradí se písmenem prvním ve stejném řádku.
- Stejný sloupec: písmeno se nahradí písmenem pod ním, pokud je písmeno poslední ve sloupci, nahradí se písmenem prvním ve stejném sloupci.

- Jiný řádek a sloupec: písmeno se nahradí písmenem ležícím v průsečíku řádku daného písmena a sloupce druhého písmena. [6]

3.2.4.2 Dešifrování

Dešifrování probíhá podobně jako šifrování. Nejdříve šifrovaný text rozdělíme do bigramů, poté vytvoříme Playfairův čtverec a pokračujeme podle stejných pravidel jako při šifrování.

3.2.4.3 Příklad

Šifrování

Otevřený text: PLAYFAIROVASIFRA

Klíč: KLIC

Rozdělení po dvojicích: PL AY FA IR OV AS IF RA

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

Obr. 17: Playfairův čtverec

Šifrovaný text: MACZG CLSHY IUCEU L

Dešifrování

Šifrovaný text: MACZG CLSHY IUCEU L

Klíč: KLIC

Rozdělení po dvojicích: MA CZ GC LS HY IU CE UL

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

Obr. 18: Playfairův čtverec

Dešifrovaný text: PLAYFAIROVASIFRA

3.2.5 Vernamova šifra – analogová

Jedná se o symetrickou proudovou šifru, jejíž princip spočívá v posunu písmen otevřeného textu o náhodný počet pozic v abecedě. Z důvodu náhodného posunu se jedná prakticky o nerozluštitelnou šifru. Počet pozic, o které se písmeno posune v abecedě, je dán klíčem. Klíč proto musí být stejně dlouhý jako OT. Klíč může být v číselné podobě nebo také ve znakové podobě.

3.2.5.1 Šifrování

Máme zadán otevřený text, který chceme zašifrovat. K tomuto textu si musíme zvolit stejně dlouhý klíč. Pokud se jedná o číselný klíč, každá *i*-tá cifra nám udává posun *i*-tého znaku otevřeného textu v abecedě. Pokud máme klíč ve znakové podobě, musíme si nejdříve každý znak převést na číselnou podobu (např. ASCII). Jak již bylo řečeno, šifrování probíhá tak, že *i*-tý znak otevřeného textu posuneme o *i*-té číslo klíče v abecedě. [9]

3.2.5.2 Dešifrování

Dešifrování provádíme přesně opačně jako šifrování. Šifrování posouvá znaky otevřeného textu směrem doprava, při dešifrování posouváme znaky šifrovaného textu dle klíče směrem doleva.

3.2.5.3 Příklad

Šifrování

Otevřený text: VERNAMOVASIFRA

Klíč: VERNAMOVASIFRA

OT	V	E	R	N	A	M	O	V	A	S	I	F	R	A
OT	21	4	17	13	0	12	14	21	0	18	8	5	17	0
Klíč	V	E	R	N	A	M	O	V	A	S	I	F	R	A
Klíč	21	4	17	13	0	12	14	21	0	18	8	5	17	0
ST	Q	I	I	A	A	Y	C	Q	A	K	Q	K	I	A

Obr. 19: Vernamova analogová šifra- šifrování

Šifrovaný text: QIIAA YCQAK QKIA

Dešifrování

Šifrovaný text: QIIAA YCQAK QKIA

Klíč: VERNAMOVASIFRA

ST	Q	I	I	A	A	Y	C	Q	A	K	Q	K	I	A
ST	16	8	8	0	0	24	2	16	0	10	16	10	8	0
Klíč	V	E	R	N	A	M	O	V	A	S	I	F	R	A
Klíč	21	4	17	13	0	12	14	21	0	18	8	5	17	0
OT	V	E	R	N	A	M	O	V	A	S	I	F	R	A

Obr. 20: Vernamova analogová šifra- dešifrování

Dešifrovaný text: VERNAMOVASIFRA

3.2.6 Vernamova šifra – binární

Jedná se o symetrickou proudovou šifru, jejíž princip spočívá v binární operaci XOR nad otevřeným textem a klíčem v binární podobě. Aby šifra byla bezpečná, musí klíč splňovat určité požadavky (viz níže). Pokud je klíč správně použit je tato šifra nerozluštitelná.

Klíč

- Klíč musí být stejně dlouhý jako otevřený text v binární podobě.
- Klíč musí být dokonale náhodná posloupnost bitů.
- Klíč může být použit jen jednou. [8]

3.2.6.1 Šifrování

Jak již bylo řečeno, princip šifrování spočívá v binární operaci XOR, kdy tuto operaci použijeme nad i -tým bitem otevřeného textu a i -tým bitem klíče. [8]

3.2.6.2 Dešifrování

Dešifrování se provádí stejným způsobem jako šifrování. Nad bity šifrovaného textu a klíče použijeme XOR, který nám vlastně říká, které bity zprávy se mají změnit a které zachovat.

3.2.6.3 Příklad

Šifrování

Otevřený text: Pes

Klíč: 0111001010011100011100101001110001110 01010011100

OT	OT v bin	Klíč	ST
P	0000000001010000	0111001010011100	0111001011001100
e	0000000001100101	0111001010011100	0111001011111001
s	0000000001110011	0111001010011100	0111001011101111

Obr. 21: Vernamova binární šifra- šifrování

Šifrovaný text: 011100101100110001110010111100101110 01011101111

Dešifrování

Šifrovaný text: 011100101100110001110010111100101110 01011101111

Klíč: 0111001010011100011100101001110001110 01010011100

ST	Klíč	OT v bin	OT
0111001011001100	0111001010011100	0000000001010000	P
0111001011111001	0111001010011100	0000000001100101	e
0111001011101111	0111001010011100	0000000001110011	s

Obr. 22: Vernamova binární šifra- dešifrování

Dešifrovaný text: Pes

3.2.7 Vigenérova šifra

Vigenérova šifra je symetrická polyalfabetická substituční šifra, která využívá až 26 šifrových abeced, přičemž klíč rozhoduje, která abeceda šifry bude použita při zašifrování písmene otevřeného textu. Šifrovaný text vzniká modulárním přičtením klíče k otevřenému textu nebo pomocí takzvaného Vigenérova čtverce, ve kterém je po řádcích zapsáno všech 26 abeced. Velkou výhodou této šifry je odolnost vůči frekvenční analýze. [1]

Vigenérův čtverec

Jak již bylo řečeno, Vigenérův čtverec obsahuje 26 šifrových abeced, každá abeceda je zapsána na samostatném řádku. V prvním řádku a prvním sloupci je

zapsána originální abeceda, v dalších řádcích je abeceda posunuta o jeden znak oproti řádku předchozímu.

3.2.7.1 Šifrování

Pokud je klíč kratší než otevřený text, klíč opakujeme tak dlouho, dokud se jejich délky nevyrovnají.

Modulární přičítání

Šifrování je vyjádřeno vzorcem:

$$C_i = T_i + K_i \cdot \text{mod}(m) \quad (5)$$

C_i ... i-té písmeno šifrovaného textu

T_i ... i-té písmeno otevřeného textu

K_i ... i-té písmeno klíče

m ... délka abecedy

Pomocí Vigenérova čtverce

Metoda pomocí Vigenérova čtverce svojí logikou odpovídá výše uvedenému vzorci. Šifrování probíhá tak, že nejdříve nalezneme znak otevřeného textu v prvním řádku Vigenérova čtverce, dále nalezneme odpovídající písmeno klíče v prvním sloupci, písmeno šifrovaného textu je potom průsečíkem písmene OT a klíče. [8]

3.2.7.2 Dešifrování

Modulární odčítání

Dešifrování probíhá pomocí podobného vzorce jako šifrování, jediný rozdíl je, že nepřičítáme, ale odečítáme.

$$T_i = C_i - K_i \cdot \text{mod}(m) \quad (6)$$

Pomocí Vigenérova čtverce

V řádku odpovídajícímu příslušnému písmenu klíče nalezneme dané písmeno šifrovaného textu, otevřený text pak odpovídá označení sloupce.

3.2.7.3 Příklad

Šifrování

Otevřený text: VIGENEROVASIFRA

Klíč: KLIC

	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

Obr. 23: Vigenérův čtverec [8]

Šifrovaný text: FTOGX PZQFL AKPCI

Dešifrování

Šifrovaný text: FTOGX PZQFL AKPCI

Klíč: KLIC

Dešifrovaný text: VIGENEROVASIFRA

3.3 Asymetrické šifrování

Asymetrické šifry na rozdíl od symetrických šifer používají k zašifrování zprávy jeden klíč a dešifrování klíč druhý. Pro šifrování a dešifrování je tedy třeba znát dvojici klíčů, které se nazývají veřejný a soukromý klíč.

V asymetrickém šifrování mohou nastat tyto dva příklady:

1. Šifrování je provedeno soukromým klíčem a dešifrování veřejným klíčem (digitální podpis). Tímto způsobem však nelze zajistit bezpečnost obsahu zprávy, protože veřejný klíč, jak už vyplývá z názvu, je veřejně dostupný.
2. Šifrování je provedeno veřejným klíčem a dešifrování soukromým klíčem. Tímto způsobem zajišťujeme bezpečnost zprávy, protože soukromý klíč, kterým lze zprávu dešifrovat má pouze příjemce. [7]

3.3.1 RSA šifra

RAS šifra je pojmenována podle jejich autorů **R**ivest, **S**hamir a **A**dleman. Metoda RSA šifry je založena na principu asymetrické kryptografie, která generuje dvojici klíčů. Každý subjekt má svůj soukromý klíč a jemu odpovídající veřejný klíč. Jeden klíč slouží k šifrování a druhý k dešifrování.

Soukromý klíč patří jen jeho vlastníkovvi. Tímto klíčem se provádí dešifrování přijaté zprávy příjemcem.

Veřejné klíče jsou zveřejněné. Teoreticky je možné z veřejného klíče vypočítat soukromý. Dosud je to však výpočetně „neproveditelné“. Tímto klíčem se provádí šifrování.

Bezpečnost šifry spočívá ve velmi složitém rozložení velkého čísla na součin prvočísel. Z čísla $n = p * q$ je v přijatelném čase prakticky nemožné zjistit činitele p a q . Naproti tomu násobení dvou velkých čísel je elementární úloha. [7]

Vytvoření veřejného a soukromého klíče

1. Náhodně vygenerujeme dvě velká prvočísla p , q .
2. Vypočteme číslo n a číslo $\varphi(n)$.
 - n je součin dvou náhodně zvolených prvočísel p a q :

$$n = p \cdot q \quad (7)$$

- $\varphi(n)$ je Eulerova funkce určující počet přirozených čísel nesoudělných s n a menších než n :

$$\varphi(n) = (p - 1) \cdot (q - 1) \quad (8)$$

3. Zvolíme náhodné číslo e , kde $1 < e < \varphi(n)$, tak, že největší společný dělitel $NSD(e, \varphi(n)) = 1$.

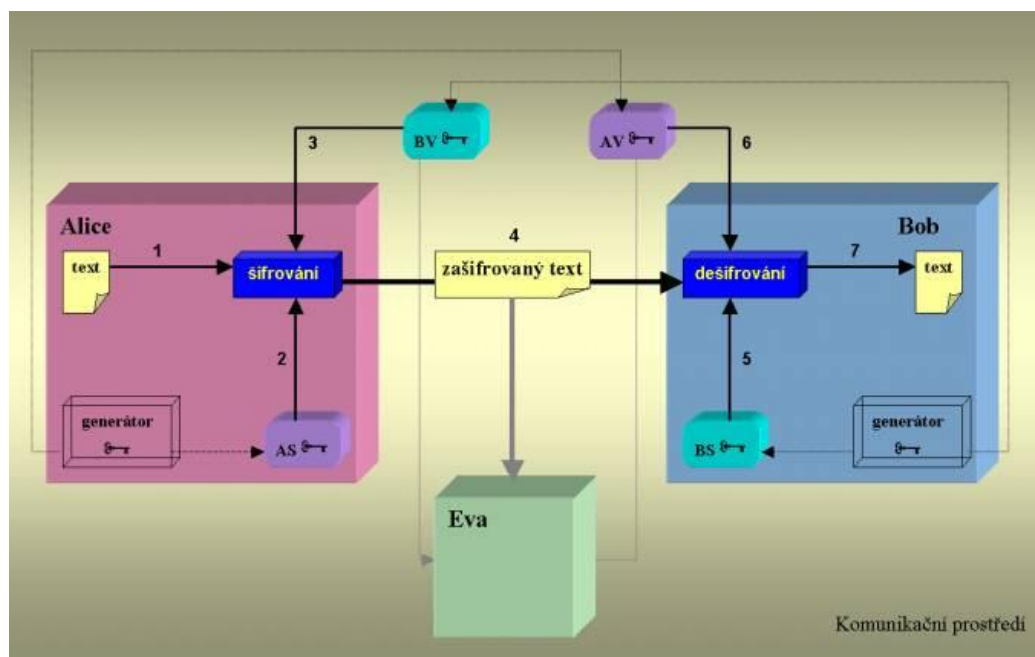
4. Užitím Euklidova algoritmu vypočteme definované číslo d .

- $1 < d < \phi(n)$
- $e \cdot d = \text{mod}(\phi(n))$, nebo také $d = e^{-1} \text{ mod}((p - 1) \cdot (q - 1))$

(9,10)

Soukromý klíč: (n, d)

Veřejný klíč: (n, e)



Obr. 24: Schéma RSA šifrování [7]

3.3.1.1 Šifrování

Jestliže chceme zašifrovat určitou zprávu, musíme tuto zprávu nejdříve převést na číselnou podobu (např. ASCII). Tím převede zprávu do číselné formy, kterou dále rozdělíme na bloky stejné délky m_i , ($0 \leq m \leq n-1$). Pokud nejsou všechny bloky stejně dlouhé, doplníme poslední blok nulami. Všichni účastníci komunikace zveřejní své veřejné klíče. Jednotlivé bloky m_i jsou zašifrovány výpočtem $c_i = m_i \cdot e \cdot \text{mod}(n)$. [7]

(11)

3.3.1.2 Dešifrování

Zprávu dešifrujeme pomocí soukromého klíče. Jednotlivé bloky šifrovaného textu jsou dešifrovány výpočtem $m_i = c_i^d \cdot \text{mod}(n)$. [7] (12)

3.3.1.3 Příklad**Šifrování**

Otevřený text: RSA sifra

p: 137

q: 929

n = p · q: 127 273

$\phi(n) = (p - 1) \cdot (q - 1)$: 126 208

e: 107 479

$d = e^{-1} \text{ mod}((p - 1) \cdot (q - 1))$: 108 263

Veřejný klíč: (127 273, 107 479)

Soukromý klíč: (127 273, 108 263)

OT= R S A _ s i f r a

OT= 82 83 65 32 115 105 102 114 97

m=(82836, 53283, 73708, 26500)

Šifrovaný text: (33365, 105257, 102878, 86070)

Dešifrování

Šifrovaný text: (33365, 105257, 102878, 86070)

Veřejný klíč: (127 273, 107 479)

Dešifrovaný text: RSA sifra

4 ÚVOD DO PROGRAMU WEBMATHEMATICA

WebMathematica nám slouží k interpretaci a převodu notebooků napsaných v klasické Mathematice do webového prohlížeče, kde s tímto notebookem může pracovat přes internet i uživatel bez aplikace Mathematica kdekoli na světě.

Programování v prostředí webMathematica je téměř totožné jako programování v klasické Mathematice. Z pohledu programátora aplikací v těchto prostředích je programování téměř totožné. Kód z klasické Mathematicy se ve webMathematice vloží jen do specifických tagů a funguje stejně. WebMathematica má navíc řadu svých vlastních funkcí a tagů. Pokud tedy umíte pracovat v prostředí Mathematica a máte základní znalosti XHTML, nic vám nebrání vyzkoušet i webMathematicu. Základním a hlavním rozdílem mezi webMathematicou a Mathematicou je ten, že webMathematica pracuje ve webovém prohlížeči a uživatel nemusí vůbec umět programovat, nemusí ani vědět, že s webMathematicou pracuje.

II. PRAKTICKÁ ČÁST

5 WEBMATHEMATICA

WebMathematica je nástavba programu Mathematica pro zobrazování a řešení daných problematik v okně internetového prohlížeče. Pro zvládnutí programování v tomto prostředí je třeba alespoň základní znalosti programování v prostředí Mathematica, dále základní znalost XHTML či CSS.

5.1 Programování stránek

Dříve než uvedu postup, jak se stránky pomocí prostředí webMathematica tvoří, chci zdůraznit, že základní kostru dokumentů tvoří XHTML a CSS.

Před samotným programováním stránek, si musíme nejdříve rozmyslet, jak by stránka měla vypadat, kde bude hlavička, menu a samotný obsah, ve kterém se nám bude zobrazovat popis samotných kryptologických metod, zadávání vstupních parametrů a v neposlední řadě zobrazení výsledků.

Jak již bylo řečeno, kostra dokumentů je tvořena pomocí XHTML a CSS, výjimkou je jen menu, které do zobrazených dokumentů vkládám pomocí JavaScriptu. Je jednodušší, mít celé menu zapsané v jednom externím souboru a vkládat jej do každé stránky, než aby bylo na každé stránce vypsáno zvlášť (podotýkám, že na každé stránce je menu totožné). Další výjimkou jsou hodnoty některých vstupních parametrů (data), které jsou taktéž zapsány JavaScriptem a to z toho důvodu, aby hodnoty ve vstupních polích zůstaly stejné, jako jsme zadali i po obnově stránky či zašifrování textu. Předposlední výjimkou je volba vzhledu, která je implementována taktéž pomocí JavaScriptu, kdy se ve zdrojových stránkách mění pouze stylový soubor. Poslední, ale nejdůležitější výjimkou je vkládání samotných funkcí, tedy kódu z Mathematicy do dokumentů. Tento kód je naprosto stejný jen se vloží do tagů specifických právě pro webMathematicu.

5.1.1 Základní kostra dokumentu

Základní kostra dokumentu a označení vkládání kódu webMathematicy jsou znázorněny na níže uvedeném obrázku Obr. 25.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2 <%@ taglib uri="http://www.wolfram.com/msp" prefix="msp" %>
3 <html>
4   <head>
5     <title>Šifrování v prostředí webMathematica</title>
6     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
7   </head>
8   <body>
9     <form action="Caesar-sif.jsp" method="post">
10      <msp:evaluate>
11      </msp:evaluate>
12    </form>
13  </body>
14 </html>
```

Obr. 25: Základní kostra dokumentu

Popis jednotlivých řádků

1. Typ XHTML dokumentu.
2. Standardní jsp hlavička, která nám říká, že používáme kód webMathematicy a tyto prvky mají předponu msp.
3. - 14. Obsah XHTML dokumentu.
4. - 7. Hlavička XHTML dokumentu.
8. - 13. Tělo XHTML dokumentu.
9. - 12. Formulář, který se po potvrzení odesílá na URL stránky uvedené v parametru *action* metodou *post*.
10. - 11. Tagy, do nichž se vkládá kód z Mathematicy.

5.1.2 Vkládání funkčních bloků kódu z Mathematicy

Pokud chceme do stránek pracujících s prostředím webMathematica vložit kód z Mathematicy, můžeme jej vložit dvěma způsoby.

1. Mezi tagy `<msp:evaluate>` a `</msp:evaluate>`.
2. Nebo do jednoduchého „tagu“ ``${msp:evaluate('kód')}`

Druhá možnost umí pracovat jen s jednoduššími bloky kódu.

5.1.3 Vstupní data

Vstupní data nebo parametry, se kterými chceme, aby naše funkce pracovaly, odesíláme pomocí formulářů, ve kterých máme umístěné například prvky `<input>`, `<textarea>` a jiné formulářové vstupy. Důležitým formulářovým prvkem je tlačítko, pomocí kterého vstupy odešleme ke zpracování. Odesílací formulář je znázorněn na Obr. 26.

```
1 ..
2 <form action="Caesar-sif.jsp" method="post">
3   <input type="text" name="key" >
4   <textarea name="OT"></textarea>
5   <input type="submit" value="Šifrovat">
6   <msp:evaluate>
7     tmp1= $$key;
8     tmp2= $$OT;
9   </msp:evaluate>
10 </form>
11 ..
```

Obr. 26: Ukázka vstupu dat

2. - 10. Obsah samotného formuláře.
3. Vstup formuláře. Atribut *type* nám říká, že se jedná o textové políčko, které se na stránkách zobrazí jako volná bílá plocha, do které můžeme psát. Důležitým atributem je *name*, pomocí tohoto atributu pracujeme s daným políčkem a předáváme jeho hodnotu. Input jako textové pole volíme pro kratší texty.
4. Pokud chceme zadávat delší text, zvolíme jako vstup tag `<textarea>`. S tímto tagem pracujeme stejně jako s `inputem`, ale pozor, je to párový tag.
5. Tlačítko, kterým celý formulář odešleme.
6. - 9. Pokud chceme hodnoty jednotlivých políček uložit do proměnných, provádíme to pomocí dvojitéch dolarů `$$` a jména daného políčka.

jméno_proměnné = \$\$jméno_vstupu_z_formuláře

Takto nám jádro Mathematicy přiřadí do proměnné hodnotu z daného políčka formuláře.

5.1.4 Zobrazení výsledku

Jestliže chceme zobrazit výsledek některé operace nebo jen určitou proměnnou, provede se to stejně jako v klasické Mathematice a to tak, že za tuto danou proměnnou nenapišeme středník. Středník nám tedy udává, jestli se daná proměnná má zobrazit nebo zůstat skrytá. Ukázka výstupů dat je na Obr. 27.

```
1 ..  
2 <msp:evaluate>  
3   tmp1=5;  
4   tmp2=10  
5   tmp3=tmp1+tmp2  
6 </msp:evaluate>  
7 ..
```

Obr. 27: ukázka výstupu dat

Na 3. řádku se proměnné *tmp1* přiřadí číslo 5, jakmile si tuto stránku otevřeme v internetovém prohlížeči, tato hodnota se nám ale nezobrazí. Zobrazí se nám až hodnota přiřazené proměnné *tmp2* na 4. řádku a hodnota proměnné *tmp3*, které jsme přiřadili součet předchozích dvou.

Další možností je použití příkazu *Print[]*. Do hranatých závorek se запиše název proměnné nebo funkce, jejíž obsah chceme vypsat na obrazovku. Pozor, tato funkce pracuje ve webMathematice trochu jinak než v klasické Mathematice. V Mathematice se *Print[]* vypíše hned, ve webMathematice se obsahy těchto printů ukládají do schránky a čekají na příkaz, který je vypíše všechny naráz.

MSPGetPrintOutput[]

MSPGetPrintOutput poskytuje způsob, jak získat výstup všech tiskových příkazů, které byly zpracovány podle aktuálního jádra. Vrací seznam řetězců, kde každý řetězec obsahuje formátovaný obsah zprávy. [12]

5.1.5 Bezpečnost

Bezpečnost stránek je zajištěna převodem všech vstupních veličin na text ještě před jakýmkoliv výpočtem nebo uplatněním nějaké funkce na těchto veličinách. Funkce pro převod vstupních hodnot na text je *ToString[]*. Tím zajistíme, pokud by byl zadán do příslušné kolonky libovolný kód z Mathematicy, že se neprovede, ale taktéž se převede na text. Pokud chceme pracovat s klíčem jako s číselnou hodnotou, což v mnoha případech chceme, musíme si tento klíč následně převést zpět na číslo.

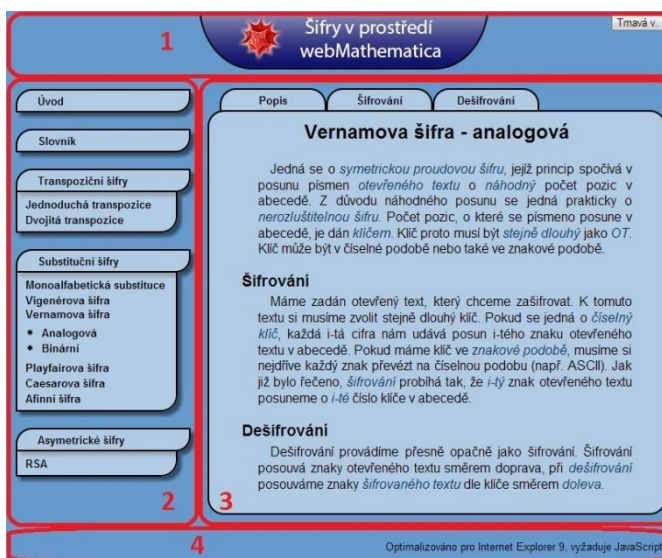
6 WEBOVÉ STRÁNKY

Tyto stránky byly vytvořeny jako praktická část k bakalářské práci, pro studenty, aby si mohli vyzkoušet a porovnat správnosti výsledků jejich naprogramovaných šifer a v neposlední řadě byly stránky vytvořeny jako praktická ukázka programování v prostředí webMathematica. Vytvořil jsem dvě odlišné verze stránek, jednu tmavou a druhou světlou, pro prezentaci či projekci. Tyto dvě verze se liší pouze vzhledem, ale funkčnost či rozvržení zůstávají stejné.

Stránky jsou převážně tvořeny jazykem XHTML a CSS, výjimkou je jen menu, volba vzhledu a pár funkcí zachovávající vstupní hodnoty (JavaScript) a samozřejmě šifrovací a dešifrovací funkce (webMathematica).

6.1 Rozvržení stránek

Stránky jsou pomyslně rozděleny do čtyř oblastí. První oblastí je hlavička nebo také záhlaví (1), kde je napsán název stránek a v pravém horním rohu odkaz na druhou verzi stránek. Druhou oblastí je menu (2) obsahující odkazy na popisy šifrovacích postupů, slovník pojmu a v neposlední řadě odkazy na samotné šifry. V další oblasti (3) si můžeme přečíst ony popisy principů šifer, ale také zde zadáváme vstupní hodnoty šifrovacích funkcí a zobrazují se nám zde také výsledky těchto funkcí. Poslední oblastí je takzvaná patička (4), ve které je jen napsáno, pro jaký prohlížeč jsou tyto stránky napsány a že vyžadují JavaScript. Základní rozvržení stránek je zobrazeno na Obr. 28.



Obr. 28: Rozvržení stránek- 2

6.2 Ovládání stránek

Na níže uvedeném obrázku Obr. 29 jsou zobrazeny základní ovládací prvky webových stránek.

Šifry v prostředí webMathematica 1 Tmavá v.

Úvod 4 Popis Šifrování Dešifrování

Slovník

Transpoziční šifry 2
Jednoduchá transpozice 3
Dvojitá transpozice

Substituční šifry
Monoalfabetická substituce
Vigenérova šifra
Vernamova šifra
• Analogová
• Binární
Playfairova šifra
Caesarova šifra
Afinní šifra

Asymetrické šifry
RSA

Caesarova šifra - šifrování

Otevřený text:
Zadávejte bez diakritiky, diakritické a speciální znaky budou smazány!

5 Caesarova šifra

Klíč (číslo 1-25): 3

Šifrovat

6
Originální abeceda: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Šifrová abeceda: DEFGHIJKLMNOPQRSTUVWXYZABC
Otevřený filtrovaný text: CAESAROVASIFRA
Šifrovaný text: FDHVD URYDV LIUD

Histogram otevřeného textu

Letter	Frequency
A	25
C	7
E	7
F	7
I	7
O	7
R	14
S	14
V	7

Histogram šifrovaného textu

Letter	Frequency
D	25
F	7
H	7
I	7
L	7
R	7
U	14
V	14
Y	7

Optimalizováno pro Internet Explorer 9, vyžaduje JavaScript.

Obr. 29: Rozvržení stránek- 2

Při prvním otevření stránek, se zobrazí úvodní obrazovka, kde se dočtete, o čem tyto stránky jsou plus nějaké základní pojmy. V pravém horním rohu stránek, je umístěno tlačítko (1), které nás přeměruje na druhou verzi stránek. V závislosti na jaké verzi se nyní nacházíme, nás přeměruje buď na světlou verzi, nebo na verzi tmavou. Když se podíváme na menu, zjistíme, že je rozděleno do několika „hlavních menu“ (2) a „submenu“ (3). Toto menu je fixně připevněné, což znamená, že při rolování okna, zůstává na místě. V hlavním menu se můžeme přemístit na již zmíněný úvod, slovník pojmů a na vysvětlení různých principů

šifrování. V submenu se přemístíme na stránky s konkrétními šiframi. Po kliknutí na položku hlavního menu, se načte stránka, kde nalezneme pouze vysvětlení daného pojmu. Po kliknutí na položku submenu se načte stránka s popisem konkrétní šifry. Všimněme si, že nad popisem či metodou šifry se nám zpřístupnily další možnosti výběru (4). V tomto výběru si můžeme vybrat, jestli se chceme přemístit na stránku s popisem, šifrováním či dešifrováním vybrané šifry. Pokud jsme si vybrali záložku šifrování/dešifrování, zobrazí se nám stránka, kde můžeme zadat vstupní hodnoty (5) do dané šifrovací/dešifrovací funkce. Každá šifra má jiné požadavky na vstupní hodnoty, které jsou však na stránkách napsány. Jen bych chtěl upozornit, že otevřený text, šifrovaný text a klíče musíme zadávat bez diakritiky. Samotné funkce si s diakritikou poradí, ale server je nejspíše nastavený tak, že s ní neporadí. Jakmile stiskneme tlačítko šifrovat/dešifrovat, tato funkce se provede a pod čarou nám vypíše výstupní hodnoty (6). Tyto hodnoty jsou otevřený text, šifrovaný text ale také některé mezikroky šifrovacích/dešifrovacích funkcí.

6.3 Příklad zdrojového kódu stránky

Na níže uvedených obrázcích Obr. 30 - 32 je ukázka zdrojového kódu Caesarovy šifry. Tuto šifru jsem zvolil z důvodu, že je nejkratší a k vysvětlení programování ve webMathematice stačí.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <%@ taglib uri="http://www.wolfram.com/msp" prefix="msp" %>
4 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head>
7 <title>Šifrování v prostředí webMathematica</title>
8 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9 <link rel="shortcut icon" href="../../../Resources/Images/favicon.ico" type="image/x-icon"/>
10 <script type="text/javascript" src="../../../Resources/JavaScript/webMathematica.js"></script>
11 <script type="text/javascript" src="js/skin.js"></script>
12 </head>
13 <msp:evaluate>
14     MSPSessionVariable[csklic,Null];
15     MSPSessionVariable[cott,Null];
16     ST=Null;
17     OT1 = Null;
18     key1= Null;
19     If[{$OT!= "" && MSPValueQ[ $$key] ,
20         OT1 = ToString[ $$OT];
21         key1= FromDigits[IntegerDigits[Read[StringToStream[$$key],
22             Number], 10, StringLength[$$key]],10];
23         csklic=key1;
24         cott=OT1;
25     ];
26     If[csklic!=Null && MemberQ[Range[-25, 25], csklic],
27         $$key=ToString[csklic];];
28     If[csklic!=Null csklic== "", $$key="";];
29 </msp:evaluate>

```

Obr. 30: Příklad zdrojového kódu- Caesarova šifra 1/3

```

30     <msp:evaluate>
31         filtr[OT_] :=
32             Module[{OT1, i},
33                 OT1 = Characters[StringReplace[ToUpperCase[ToString[OT]],
34                     {"Á" -> "A", "Ā" -> "C", "Ď" -> "D", "Ě" -> "E", "Ě" -> "E",
35                     "Ī" -> "I", "Ň" -> "N", "Ó" -> "O", "Ř" -> "R", "Š" -> "S",
36                     "Ť" -> "T", "Ů" -> "U", "Ů" -> "U", "Ÿ" -> "Y", "Ž" -> "Z",
37                     "1" -> "JEDNA", "2" -> "DVA", "3" -> "TRI", "4" -> "ČTYRI",
38                     "5" -> "PĚT", "6" -> "ŠEST", "7" -> "SEDM", "8" -> "OSM",
39                     "9" -> "DEVĚT", "0" -> "NULLA"}]];
40
41                 For[i = 1, i <= Length[OT1], i++,
42                     If[MemberQ[CharacterRange["A", "Z"], OT1[[i]], , OT1[[i]] = ""];
43                 Return[StringJoin[OT1]];
44             ];
45
46         sif[OT_, k_] := Module[{OT1, ST},
47             OT1 = ToCharacterCode[filtr[OT]] - 65;
48             ST = FromCharacterCode[Mod[OT1 + k, 26] + 65];
49             Return[StringJoin[Riffle[Characters[ST], " ", 6]]];
50         ];
51
52         hist[OT_] :=
53             Module[{x, y, z, OT1}, OT1 = StringReplace[filtr[OT], " " -> ""];
54             x = Tally[Sort[Characters[OT1]]];
55             y = x[[All, 2]];
56             z = Round[y/StringLength[OT1]*100];
57             Return[
58                 BarChart[z, ChartLabels -> Placed[x[[All, 1]], Bottom],
59                 LabelStyle -> White, Background -> Black];];
60     </msp:evaluate>
61
62 <body>
63     <div id="main">
64         <div id="header"><button onclick="myFunction()">
65             <script>document.write(sessionStorage.popis);</script></button></div>
66         <div id="menu">
67             <script type="text/javascript" src="js/menu.js"></script>
68         </div>
69         <div id="line_sif">
70             <ul>
71                 <li><div><a href="Caesar-popis.jsp">Popis</a></div></li>
72                 <li><div><a href="Caesar-sif.jsp">Šifrování</a></div></li>
73                 <li><div><a href="Caesar-desif.jsp">Dešifrování</a></div></li>
74             </ul>
75         </div>
76         <div id="content">
77             <h2>Caesarova šifra - šifrování</h2>
78             <div class="code">
79                 <form action="Caesar-sif.jsp" method="post">
80                     <div>
81                         <b>Otevřený text:</b> <br />
82                         Žadávajte bez diakritiky, diakritické a speciální znaky
83                         budou smazány! <br />
84                         ${msp:evaluate('If[cott!=Null, $$OT2=cott;'])}
85                         <textarea id="myTextarea" name="OT" cols="40" rows="3">
86                             <msp:evaluate>
87                                 MSPValue[ $$OT2, "Caesarova sifra"]
88                             </msp:evaluate>
89                         </textarea>
90                     </div>
91                     <div>
92                         <b>Klíč (číslo 1-25): </b>
93                         <input id="myText" type="text" name="key" size="2" value="
94                             <msp:evaluate>
95                                 MSPValue[ $$key, "3"]
96                             </msp:evaluate>"/>
97                     </div>
98                     <div>
99                         <input type="submit" value="Šifrovat" />
100                    </div>
101                </form>
102            </div>
103        </div>

```

Obr. 31: Příklad zdrojového kódu- Caesarova šifra 2/3

```

103     <hr />
104     <div class="code">
105         <msp:evaluate>
106             If[MemberQ[Range[-25, 25], key1] || MemberQ[Range[-25, 25], csklic],
107                 If[csklic===Null, ST=sif[OT1, key1];, ST=sif[cott, csklic];]
108         </msp:evaluate>
109         <c:if test="{m:evaluate('ST!=Null')}">
110             <table style="margin:20px 0 0 0;">
111                 <tr>
112                     <td class="first">Originální abeceda:</td>
113                     <td>{m:evaluate('x=StringJoin[CharacterRange[
114                         "A", "Z"]')]}</td>
115                 </tr>
116                 <tr>
117                     <td>Šifrová abeceda:</td>
118                     <td>{m:evaluate('FromCharCode[Mod[
119                         ToCharCode[x] - 65 +
120                         If[csklic!=Null, csklic, key1], 26] + 65')]}</td>
121                 </tr>
122                 <tr>
123                     <td>Otevřený filtrovaný text:</td>
124                     <td>{m:evaluate('If[StringLength[filtr[OT2]] > 29,
125                         StringJoin[Riffle[Characters[filtr[OT2]], " ", 29]],
126                         filtr[OT2]')]}</td>
127                 </tr>
128                 <tr>
129                     <td>Šifrovaný text:</td>
130                     <td>{m:evaluate('ST')}</td>
131                 </tr>
132             </table>
133             <div class="hist">
134                 <h3>Histogram otevřeného textu</h3>
135                 <msp:evaluate>
136                     MSPShow[hist[OT2]]
137                 </msp:evaluate>
138                 <h3>Histogram šifrovaného textu</h3>
139                 <msp:evaluate>
140                     MSPShow[hist[ST]]
141                 </msp:evaluate>
142             </div>
143         </c:if>
144         <c:if test="{m:evaluate('ST===Null')}">Není zadám <b>OT</b>
145             nebo <b>klíč</b>.</c:if>
146     </div>
147 </div>
148 <div id="footer"></div><div class="ref"> Optimalizováno pro Internet
149     Explorer, vyžaduje JavaScript.</div>
150 </div>
151 </body>
152 </html>

```

Obr. 32: Příklad zdrojového kódu- Caesarova šifra 3/3

6.3.1 Popis zdrojového kódu

Řádky 1-12 přeskočím, byly již vysvětleny v jedné z předchozích kapitol. Zmíním se jen o 11. řádku, který nám na stránku pomocí JavaScriptu vypisuje link s CSS souborem. Tento scriptový soubor bude vysvětlen později.

13. – 29. Blok kódu, ve kterém deklaruji potřebné proměnné a přiřazuji jim dané hodnoty.

14. – 15. Session proměnné, které se ukládají na server. Dále je používám k znovu zobrazení již vypočítaných výsledků při přejetí na jinou stránku a následném návratu na stránku původní.
19. – 28. Podmínky, při kterých se přiřadí vnitřním proměnným hodnoty klíčů, otevřeného a šifrovaného textu. Tyto hodnoty jsou zároveň převedeny na text.
31. – 44. Filtr textu, ve kterém se odstraní nebo nahradí znaky neodpovídající specifikaci dané šifry.
46. – 50. Šifrovací funkce, ve které se nejdříve jednotlivé znaky převedou na číselnou interpretaci, která se dále posune o potřebný počet znaků v abecedě a nakonec se převede zase na znaky. Šifrovaný text je rozdělen po pěticih. Dešifrovací funkce je napsána podobným způsobem.
52. – 59. Funkce, která nám zobrazuje histogram textu, tedy četnost jednotlivých znaků. Tato funkce je implementována pouze u substitučních šifer.
62. – 151. Tělo samotného uživatelem zobrazeného obsahu stránky.
64. – 65. Hlavička, ve které je vykresleno tlačítko pro změnu vzhledu stránky. Kliknutím na toto tlačítko se nám zavolá JavaScriptová funkce, která je spolu s popisem tlačítka zapsána v *skin.js* souboru, na který se odkazujeme na 11. řádku.
67. JavaScriptové volání menu, které je vysvětleno níže.
70. – 74. Další menu, které nám odkazuje na popis, šifrování či dešifrování dané šifry.
79. – 101. Formulář, ve kterém pomocí tagů *input* a *textarea*, zadáváme a odesíláme vstupní hodnoty šifrovací/dešifrovací funkce.
84. Podmínka, která vyhodnocuje, jestli se má do tagu *textarea* vypsát defaultní text nebo text již někdy zadaný v souvislosti se zachováním jednou zadaných hodnot a překliknutí na jinou stránku.

105. – 108. Blok s podmínkou, která vyhodnocuje správný formát klíče, pokud má jeden z klíčů správný formát (aktuálně zadaný klíč nebo klíč v Session), vyhodnotí se další podmínka, která zkoumá, jestli je klíč v Session nebo jestli se má použít aktuálně zadaný. Podle vyhodnocení podmínky se do proměnné *ST* запиše výsledek šifrovací funkce s potřebnými parametry.
109. – 143. Podmínka, která zkoumá, jestli je v proměnné *ST* zapsán výsledek šifrovací funkce či není. Pokud ano, zobrazí se tabulka s výsledky.
111. – 115. Řádek, na který se vygeneruje originální abeceda A až Z.
116. – 121. Řádek, na který se generuje šifrová abeceda v závislosti na klíči.
128. – 131. Řádek se šifrovaným textem.
133. – 142. Zobrazení histogramů otevřeného a šifrovaného textu.
144. – 145. Podmínka pro chybové hlášení, pokud není zadán klíč nebo otevřený text.

Obrázek 33 nám ukazuje zdrojový kód menu.

```

1 document.write("<div class='menu-00_'><a href='index.jsp'>Úvod</a></div>");
2 document.write("<div class='menu-00_'><a href='Slovník.html'>Slovník</a></div>");
3 document.write("<div class='menu-01_'><a href='Trans.html'>Transpoziciční šifry</a></div>");
4 document.write("<div class='menu-02_'>");
5     document.write("<ul>");
6         document.write("<li><a href='JT-popis.jsp'>Jednoduchá transpozice</a></li>");
7         document.write("<li><a href='DT-popis.jsp'>Dvojitá transpozice</a></li>");
8     document.write("</ul>");
9 document.write("</div>");
10 document.write("<div class='menu-03_'></div>");
11 document.write("<div class='menu-01_'><a href='Sub.html'>Substituční šifry</a></div>");
12 document.write("<div class='menu-02_'>");
13     document.write("<ul>");
14         document.write("<li><a href='Mono-popis.jsp'>Monoalfabetická substituce</a></li>");
15         document.write("<li><a href='Viegener-popis.jsp'>Viegenérova šifra</a></li>");
16         document.write("<li><a href='Vernam-analog-popis.jsp'>Vernamova šifra</a>");
17         document.write("<ul style='list-style-type: disc;left:115px;text-align:right;'>");
18             document.write("<li style='width:70px;'><a href='Vernam-analog-popis.jsp'>Analogová</a></li>");
19             document.write("<li style='width:70px;'><a href='Vernam-bin-popis.jsp'>Binární</a></li>");
20         document.write("</ul></li>");
21         document.write("<li><a href='Play-popis.jsp'>Playfairova šifra</a></li>");
22         document.write("<li><a href='Caesar-popis.jsp'>Caesarova šifra</a></li>");
23         document.write("<li><a href='Afinni-popis.jsp'>Afinní šifra</a></li>");
24     document.write("</ul>");
25 document.write("</div>");
26 document.write("<div class='menu-03_'></div>");
27 document.write("<div class='menu-01_'><a href='Asymetr.html'>Asymetrické šifry</a></div>");
28 document.write("<div class='menu-02_'>");
29     document.write("<ul>");
30         document.write("<li><a href='RSA-popis.jsp'>RSA</a></li>");
31     document.write("</ul>");
32 document.write("</div>");
33 document.write("<div class='menu-03_'></div>");
34 document.write("<div class='menu-04_'></div>");

```

Obr. 33: Popis zdrojového kódu- menu.js

Na každém řádku tohoto zdrojového kódu vlastně na stránku JavaScriptově vpisují části textu, tento text jsou HTML tagy, které nám danou část stránky rozdělují do několika divů, ve kterých jsou dále zapsány nečíslované seznamy. Každá položka seznamu je odkaz na další konkrétní stránku. JavaScriptově vypisování menu jsem zvolil z toho důvodu, že je jednodušší mít několik řádků zapsaných v jednom externím souboru a jeden řádek v každé další stránce, než aby na každé stránce byl totožný blok řádků kódu.

JavaScriptový soubor na Obr. 34 nám definuje skupinu pravidel, podle kterých se mění po stisknutí tlačítka vzhled stránek.

```
1 if(!sessionStorage.skin)
2     {
3         sessionStorage.skin="dark";
4         sessionStorage.popis="Světlá v.";
5         sessionStorage.link="<link rel='stylesheet' type='text/css' href='dark.css' />";
6     }
7 function myFunction()
8     {
9         if(!sessionStorage.skin)
10        {
11            sessionStorage.skin="dark";
12            sessionStorage.popis="Světlá v.";
13            sessionStorage.link="<link rel='stylesheet' type='text/css' href='dark.css' />";
14        }
15        else
16        {
17            if(sessionStorage.skin=="dark" )
18            {
19                sessionStorage.skin="light";
20                sessionStorage.popis="Tmavá v.";
21                sessionStorage.link="<link rel='stylesheet' type='text/css' href='light.css' />";
22            }
23            else
24            {
25                if (sessionStorage.skin=="light")
26                {
27                    sessionStorage.skin="dark";
28                    sessionStorage.popis="Světlá v.";
29                    sessionStorage.link="<link rel='stylesheet' type='text/css' href='dark.css' />";
30                }
31            }
32        }
33        document.location.reload(true);
34    }
35    document.write(sessionStorage.link);
```

Obr. 34: Popis zdrojového kódu- skin.js

1. – 6. Pokud není nastaven nebo definován *sessionStorage.skin* nastaví se *sessionStorage.skin* na hodnotu *dark*, *sessionStorage.popis* na hodnotu *Světlá v.* a *sessionStorage.link* na hodnotu `<link rel='stylesheet' type='text/css' href='dark.css' />` což je link na CSS soubor pro tmavý vzhled stránky. *SessionStorage.libovolný_název* je proměnná, která si uchovává svoji hodnotu do doby, než se tento session zruší (čili do opuštění stránek/vypnutí prohlížeče).

- 7. – 35. Funkce pro obsluhu události *onclick* na tlačítku, které nám mění vzhled stránek.
- 9. – 14. Stejná podmínka jako na řádcích 1 – 6.
- 15. – 22. Pokud není předchozí podmínka splněna, vyhodnocuje se další podmínka, která testuje, jestli je *sessionStorage.skin* nastaven na hodnotu *dark*. Pokud ano, další session jsou nastaveny na v kódu zapsané hodnoty.
- 23. – 29. Pokud není splněna ani předchozí podmínka, vyhodnocuje se podmínka poslední. Ta dělá to samé jako předchozí podmínka, jen hodnoty session jsou opačné.
- 33. Znovunačtení stránky.
- 34. Zapsání do dokumentu link s CSS souborem.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo vytvoření webové prezentace s důrazem na implementaci prostředí webMathematica, ve kterém jsou naprogramované nástroje, pomocí kterých si studenti mohou vyzkoušet principy vybraných šifer přímo ve webovém prohlížeči a zkontrolovat tak výsledky svých naprogramovaných šifer. Ve webové prezentaci se studenti také mohou seznámit s hlavními principy klasických kryptologických metod a pojmy, které jsou s nimi spojené. Jsou zde zapsány principy substitučních a transpozičních metod, také jsem zmínil jednu metodu z moderní kryptologie, kterou je algoritmus RSA. Všechny popisy a pojmy spojené s danými kryptologickými metodami, které jsou zmíněné ve webové prezentaci, jsou také zapsány v této tištěné podobě. V této formě práce jsou také popsány postupy tvoření stránek a základní pojmy spojené s prostředím webMathematica.

Webová prezentace je optimalizována pro Internet Explorer. Dále je optimalizována z hlediska výkonu, tím, že jsem pro tvoření grafického rozhraní jedné (světlé) verze nepoužil téměř žádný obrázek. Zabezpečení je provedeno znemožněním vykonání jakéhokoliv bloku kódu, který uživatel zadá do vstupních polí.

Vývoj aplikací v prostředí webMathematica je zajímavý už jen z toho důvodu, že není třeba žádného speciálního softwaru, aby si student/uživatel mohl vyzkoušet dané programy. Jediné co uživatel potřebuje je webový prohlížeč a připojení k internetu.

Praktické využití této práce spočívá v tom, že se bude využívat při výuce na Fakultě Aplikované Informatiky Univerzity Tomáše Bati.

ZÁVĚR V ANGLIČTINĚ

The main goal of this thesis was to create an online presentation with emphasis to implementation of the webMathematica environment, which is used for programming of the tools which students use to experience selected cyphers inside a web browser and to be able to verify the correctness of the cyphers they programmed. Using the online presentation, the students are also able to get to know the main principles of the classic cryptological methods and terms connected to them. The main principle of the substitutional and transpositional methods are described there, I also mention a method from modern cryptology, the RSA algorithm. Every description and term related to the given cryptological method, that is mentioned in the online presentation, are also written down in this printed form. There are also procedures of webpage creation and basic terms related to the webMathematica environment listed here.

Website presentation is optimized for Internet Explorer. It is also optimized for performance, because I used almost no images for creating graphical interface in one of the versions (light version). Security is done by disabling execution of any code block, user types in the input fields.

Developing applications in webMatematica environment is interesting especially for the reason, that student/user doesn't need any other special software to test those applications. The only requirement is web browser with internet connection. This work has a practical use as a teaching tool at the Faculty of Applied Informatics, Thomas Bata University

SEZNAM POUŽITÉ LITERATURY

Monografie

- [1] SINGH, Simon. *Knih kódu a šifer*. 1. vyd. Praha: Argo, 2003. ISBN: 80-7203-499-5.
- [2] MURPHY, Sean. *Kryptografie - Průvodce pro každého*. Praha: Dokořán, 2006. 157 s. ISBN 80-7363-074-5.
- [3] VONDRUŠKA, Pavel. *Kryptologie, šifrování a tajná písma*. 1. vyd. Praha: Albatros, 2006. 340 s. ISBN 80-00-01888-8.
- [4] JANEČEK, Jiří: *Odhalená tajemství šifrovacích klíčů minulosti*, Naše vojsko, Praha, 1994
- [5] Tomáš HANŽL, T; PELÁNEK, R; VÝBORNÝ, O. *Šifry a hry s nimi*. 1. vyd. Praha: Portál, 2007. ISBN 978-80-7367-196-9.

Internetové zdroje

- [6] ŠTRÁFELDA, J. *Šifrování a signály*. [online]. 2010 [cit. 2013-4-28]. Dostupný z WWW: <<http://www.shaman.cz/sifrovani/>>.
- [7] FRÁŇA, J; MIKULECKÝ, P; SKALSKÝ, M. *Algoritmus RSA*. [online]. 2003 [cit. 2013-4-28]. Dostupný z WWW: <<http://kryptologie.uhk.cz/6.htm>>.
- [8] MIČKA, P. *Algoritmy.net*. [online]. 2008 [cit. 2013-4-28]. Dostupný z WWW: <<http://www.algoritmy.net/>>.
- [9] ŠIFRY A ŠIFROVÁNÍ. *Vernamova šifra*. [online]. 20013 [cit. 2013-4-28]. Dostupný z WWW: <http://www.tajemstvi-kodu.wz.cz/vernamov_sifra.html>.
- [10] THEEPOCHTIMES. *Tajemství šifer – po stopách kryptografie a steganografie*. [online] 2008 [cit. 2013-4-28]. Dostupný z WWW: <<http://www.velkaepocha.sk/200806125316/Tajemstvi-sifer-po-stopachkryptografie-a-steganografie.html>>.
- [11] COMTEL. *Monoalfabetické substituční šifry*. [online]. 20013 [cit. 2013-4-28]. Dostupný z WWW: <www.comtel.cz/files/download.php?id=4091>.

[12] WOLFRAM. *MSPGetPrintOutput*. [online]. 2013 [cit. 2013-4-28]. Dostupný z WWW:

<<http://reference.wolfram.com/mathematica/webMathematica/ref/MSPGetPrintOutput.html>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

OT Otevřený text.

ST Šifrovaný text.

SEZNAM OBRÁZKŮ

Obr. 1: Histogram textu [8].....	11
Obr. 2: Šifrovaná komunikace	13
Obr. 3: Zápis do tabulky	17
Obr. 4: Šifrování podle prvního klíče	17
Obr. 5: Šifrování podle druhého klíče	17
Obr. 6: Zápis do tabulky	18
Obr. 7: Dešifrování podle druhého klíče	18
Obr. 8: Dešifrování podle prvního klíče	18
Obr. 9: Zápis do tabulky	20
Obr. 10: Šifrování podle prvního klíče	20
Obr. 11: (1. šifrování) ⁻¹	20
Obr. 12: Šifrování podle druhého klíče	20
Obr. 13: Zápis do tabulky	21
Obr. 14: Dešifrování podle druhého klíče	21
Obr. 15: (1. dešifrování) ⁻¹	21
Obr. 16: Dešifrování podle prvního klíče	21
Obr. 17: Playfairův čtverec	27
Obr. 18: Playfairův čtverec	27
Obr. 19: Vernamova analogová šifra- šifrování	28
Obr. 20: Vernamova analogová šifra- dešifrování.....	29
Obr. 21: Vernamova binární šifra- šifrování	30
Obr. 22: Vernamova binární šifra- dešifrování.....	30
Obr. 23: Vigenérův čtverec [8]	32
Obr. 24: Schéma RSA šifrování [7].....	34
Obr. 25: Základní kostra dokumentu	39
Obr. 26: Ukázka vstupu dat	40
Obr. 27: ukázka výstupu dat	41
Obr. 28: Rozvržení stránek- 2.....	42
Obr. 29: Rozvržení stránek- 2.....	43
Obr. 30: Příklad zdrojového kódu- Caesarova šifra 1/3	44
Obr. 31: Příklad zdrojového kódu- Caesarova šifra 2/3	45
Obr. 32: Příklad zdrojového kódu- Caesarova šifra 3/3	46

Obr. 33: Popis zdrojového kódu- menu.js 48
Obr. 34: Popis zdrojového kódu- skin.js..... 49

SEZNAM PŘÍLOH

CD s webovou prezentací a bakalářskou prací

- **Webová prezentace:**

Afinni-desif.jsp

Afinni-popis.jsp

Afinni-sif.jsp

Asymetr.html

BP2.png

BP2_01.png

BP2_02.png

BP2_03.png

BP2_04.png

BP2_05.png

Caesar-desif.jsp

Caesar-popis.jsp

Caesar-sif.jsp

dark.css

desif.png

DT-desif.jsp

DT-popis.jsp

DT-sif.jsp

index.jsp

JT-desif.jsp

JT-popis.jsp

JT-sif.jsp

light.css

menu.js
menu_01.png
menu_02.png
menu_03.png
menu_04.png
Mono-desif.jsp
Mono-popis.jsp
Mono-sif.jsp
Play-desif.jsp
Play-popis.jsp
Play-sif.jsp
popis.png
pozadi.jpg
pozadi2.jpg
RSA-desif.jsp
RSA-popis.jsp
RSA-sif.jsp
sif.png
skin.js
Slovník.html
Sub.html
Thumbs.db
Trans.html
uvod.png
Vernam-analog-desif.jsp
Vernam-analog-popis.jsp

Vernam-analog-sif.jsp

Vernam-bin-desif.jsp

Vernam-bin-popis.jsp

Vernam-bin-sif.jsp

Viegener-desif.jsp

Viegener-popis.jsp

Viegener-sif.jsp

- **Dokumenty**

dlesk_2013_bp.pdf