CRACKING OF THE SUBSTITUTION CIPHERS IN CLASSICAL CRYPTOGRAPHY

Martin Kulich
Bachelor Degree Programme (3), FIT BUT
E-mail: xkulic01@stud.fit.vutbr.cz

Supervised by: Jaroslav Stružka
E-mail: istruzka@fit.vutbr.cz

Abstract: This project provides a tool for automatic monoalphabetic substitution ciphers cracking using most common words dictionary and language characteristics. It is focused on specifics of czech language, especially declension and its influence on dictionary size. Language characteristics are used for reducing dictionary size and ability to crack ciphers where words are not separated.

Keywords: ciphers, cracking, dictionary search

1 ÚVOD

Monoalfabetická substituční šifra je poměrně jednoduchou šifrou, kdy dochází k náhradě znaku za znak v poměru 1:1. Šifra je určena k ručnímu šifrování a dešifrování a dnes se již v praxi nepoužívá, neboť byla, jako všechny šifry klasické kryptografie, nahrazena šiframi blokovými.

Převod znaku otevřeného textu na znak šifrového textu můžeme vyjádřit jako injektivní zobrazení $\pi : \Sigma_P \rightarrow \Sigma_C$, kde Σ_P je abeceda otevřeného textu a Σ_C je abeceda šifrového textu. Při luštění se pokoušíme nalézt inverzní zobrazení $\pi^{-1} : \Sigma_C \rightarrow \Sigma_P$. [5] Toto zobrazení můžeme nazývat *klíčem*.

Ruční luštění využívá především porovnávání charakteristik jazyka a šifrového textu, zejména pak frekvenční analýzy jednotlivých znaků, bigramů a trigramů. Následně se luštitel snaží odhadnout častá slova, z nich odvodit další znaky a tak stále dokola, dokud se mu nepodaří šifru vyluštit.

Při tvorbě algoritmů pro luštění monoalfabetické substituční šifry bylo vyzkoušeno několik přístupů. Ať už se jednalo o algoritmus, pracující pouze na základě charakteristik jazyka a textu [4], nebo algoritmus pracující výhradně se slovníkovým útokem [2], anebo o použití genetických algoritmů [3]. Všechny však mají své nedostatky (časová náročnost výpočtu, omezení na šifrový text apod.).

Cílem práce je prozkoumat možnost spojení slovníkového útoku s optimalizací pomocí charakteristik jazyka. Druhým cílem je zjistit optimální velikost slovníku pro český jazyk; předchozí pokusy pracovaly všechny s angličtinou, která používá ve všech pádech stejný tvar slova.

2 METODA PROHLEDÁVÁNÍ

Problém hledání klíče lze vyjádřit jako problém prohledávání stavového prostoru s omezujícími podmínkami (*constraint satisfaction*). Tento přístup již využil M. Lucks [2] při slovníkovém útoku. Lucksův algoritmus však umí pracovat pouze s šifrovým textem děleným na slova, a proto není v praxi použitelný. Šifrové zprávy byly obvykle posílány telegrafním spojením a jako takové neobsahovaly mezery.

Algoritmus však dosahuje dobrých výsledků, především co se týká přesnosti a rychlosti luštění. Využívá slepé prohledávání do hloubky (*DFS – Depth First Search*), kdy je každý uzel rozgenerován na několik variant slova s omezujícími podmínkami:

- známá délka slova,
- již známé části klíče (např. víme, že znak `Q` šifrového textu odpovídá znaku `T` otevřeného textu),
- opakované znaky ve slově (např. druhý a pátý znak slova jsou stejné).

Budeme-li nyní uvažovat šifrový text, který není dělený na jednotlivá slova, např.:

```
BLRYNWSVHDPGSRBNOYFPXPLP,
```

byla by výše popsaná metoda výpočetně velmi náročná. Zatímco v případě, kdy známe délku slova, můžeme odložit vyhodnocení šifrového slova, které by mělo velký počet možných řešení, v případě neděleného textu musíme uvažovat všechny možné délky slova a nelze si vybírat, odkud začít: lze jen od počátku, anebo od konce šifrového textu. Proto musí přijít ke slovu různé heuristiky a optimalizace, kterým se budu blíže věnovat v části 4.

3 VYHLEDÁVÁNÍ VE SLOVNÍKU

Abychom mohli vyhledávat podle výše uvedených omezení, je třeba slovník upravit pro co nejrychlejší vyhledávání. Slovník bude implementován jako relační databáze s možností indexování, což umožní potřebné rychlé vyhledávání.

Jako zdroj pro slovník jsou využita data z mnoha internetových stránek; záznam obsahuje vždy slovo (přesněji tvar slova) a počet využití.

3.1 TEXT DĚLENÝ NA SLOVA

Při vyhledávání v textu děleném na slova lze cele využít Lucksova algoritmu s přidaným počátečním odhadem nejčastěji se vyskytujících znaků, příp. bigramů. To zajistí počáteční zmenšení stavového prostoru a z toho plynoucí rychlejší prohledávání.

3.2 NEDĚLENÝ TEXT

U textu neděleného na slova není možné použít délku slova jako omezující podmínku pro vyhledávání v databázi. Dokonce nelze ani optimalizovat, které slovo vyhodnocovat jako první. Vyhodnocování proto musí začít buď od začátku, anebo od konce šifrového textu, příp. z obou stran. Z podstaty problému je délka slova l další proměnnou, která je rozgenerovávána. Jistě by nebylo rozumné iniciovat délku slova na hodnotu $l = 1$ a iterovat ji. U mnoha slov by bylo vyzkoušeno mnoho krátkých variant slov, které by ke kýženému výsledku nevedly.

4 OPTIMALIZACE A HEURISTIKY

Při zmenšování velikosti stavového prostoru jsou využity charakteristiky jazyka, používané především při ručním luštění. V tabulce 1 můžeme vidět četnosti jednotlivých písmen v českém jazyce. Tabulka je založena na [1], písmeno `ch` je však rozloženo na dva znaky a četnosti jsou přepočítány.

Dále je potřeba optimalizovat odhad délky slova l. Výhodnější než prostá inkrementace od 1, je iniciovat délku l na nějakou střední hodnotu. Tu lze zjistit jako vážený průměr délky všech slov ve slovníku. Vzhledem k tomu, že délka slova i počet užití jsou u každého záznamu slovníku známy, je určení střední délky poměrně jednoduchou záležitostí.

znak	výskyt	znak	výskyt	znak	výskyt
o	8,201 %	u	3,100 %	ý	0,932 %
e	7,753 %	í	3,072 %	š	0,809 %
a	6,632 %	c	2,582 %	ů	0,564 %
n	6,610 %	h	2,280 %	f	0,390 %
t	5,499 %	á	2,108 %	g	0,339 %
s	4,574 %	z	2,102 %	ú	0,143 %
i	4,526 %	j	1,963 %	x	0,092 %
v	4,335 %	y	1,735 %	ň	0,073 %
l	4,056 %	b	1,649 %	w	0,071 %
r	3,938 %	ě	1,476 %	ť	0,038 %
k	3,715 %	ř	1,175 %	ó	0,032 %
d	3,577 %	é	1,166 %	ď	0,019 %
p	3,420 %	ž	1,012 %	q	0,006 %
m	3,230 %	č	1,007 %		

Tabulka 1: Tabulka výskytu znaků v českém jazyce řazená podle výskytu

5 ZÁVĚR

Práce navrhuje možné automatické luštění monoalfabetických substitučních šifer. Předmětem zkoumání bude úspěšnost luštění, potřebná velikost slovníku a počáteční hodnoty výše zmíněných heuristik. V současné době je návrh implementován. Hledání optimalizací bude jistě náročnou částí testování, věřím však, že přinese zlepšení oproti neoptimalizovanému řešení, příp. oproti řešení uveřejněném v [2].

REFERENCE

[1] HANČAR, P.: Frekvence písmen, bigramů, trigramů, délka slov. [online], Naposledy editováno 23. 3. 2008.
URL <http://nlp.fi.muni.cz/cs/Frekvence_pismen_bigramu_trigramu_delka_slov>

[2] LUCKS, M.: A constraint satisfaction algorithm for the automated decryption of simple substitution ciphers. In *Proceedings on Advances in cryptology*, CRYPTO '88, New York, NY, USA: Springer-Verlag New York, Inc., 1990, ISBN 0-387-97196-3, s. 132–144.
URL <http://dl.acm.org/citation.cfm?id=88314.88360>

[3] ORANCHAK, D.: Evolutionary algorithm for decryption of monoalphabetic homophonic substitution ciphers encoded as constraint satisfaction problems. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, New York, NY, USA: ACM, 2008, ISBN 978-1-60558-130-9, s. 1717–1718.
URL <http://doi.acm.org/10.1145/1389095.1389425>

[4] PELEG, S.; ROSENFELD, A.: Breaking substitution ciphers using a relaxation algorithm. *Commun. ACM*, ročník 22, November 1979: s. 598–605, ISSN 0001-0782.
URL <http://doi.acm.org/10.1145/359168.359174>

[5] STINSON, D.: *Cryptography: Theory and Practice*. CRC/C&H, druhé vydání, 2002, ISBN 1584882069.