Afinní šifra

Afinní šifra je substituční šifra, která vyměňuje jednotlivá písmena za jiná. Jedno písmeno se vždy transformuje na jiné (například A bude pokaždé K). Kvůli této vlastnosti bu zachována frekvence písmen před a po transformaci (jenom budou posunuty vrcholy). Důsledkem je, že je tato šifra náchylná na útok pomocí frekvenční analýzy.

Šifrování

Šifrování vstupního řetězce probíhá dle uvedeného vzorce. Parametry “a” a “b” jsou zadány.

$E(x) = (a * x + b) \% 26$
Kde:
“a” je multiplikativní parametr (část klíče) “a”
“b” je aditivní parametr (část klíče) “b”
“x” je index písmena v abecedě (A = 0, B = 1, …)
E(x) je zašifrovaný znak na indexu “x”

Dešifrování

Dešifrování vstupního řetězce probíhá dle uvedeného vzorce. Parametry “a” a “b” jsou zadány.

$D(x) = a^{-1}(x – b) \% 26$
Kde:
“a^{-1}” je multiplikativní inverze parametru (část klíče) “a”
“b” je aditivní parametr (část klíče) “b”
“x” je index písmena v abecedě (A = 0, B = 1, …)
D(x) je dešifrovaný znak na indexu “x”

Crack

Prolomení šifry bez znalosti klíče (parametry “a” a “b” nejsou zadány) je založeno na frekvenční analýze. Frekvenční analýza je založena na četnosti výskytu jednotlivých písmen v přirozeném jazyce (pro každý jazyk je jiná). Pokud víme, v jakém jazyce je zašifrovaný text, můžeme pomocí frekvenční analýzy provést odhad širovacích parametrů. Princip spočívá v tom, že písmena s vysokým výskytem v zašifrovaném textu budou pravděpodobně odpovídat nějakému znaku s vysokým výskytem (v našem případě) v češtině.

Implementace

V projektu jsou implementovány dva způsoby útoku pomocí frekvenční analýzy. Oba vyžadují znalost frekvenčního histogramu češtiny (hodnoty převzaty z https://nlp.fi.muni.cz/cs/FrekvenceSlovLemmat).

První přístup si určí dvě nejčastější písmena v češtině (jedná se o písmena “e” a ”a”) a první dvě nejčastější písmena v zašifrovaném textu. (V mém případě bylo nutné prohodit první písmeno za druhé). Nyní můžeme pomocí indexů těchto písmen sestavit soustavu rovnic

```
c1 == p1 * a + b % 26;
c2 == p2 * a + b % 26;
```

Kde:
“c1, c2” jsou indexy zašifrovaných písmen
“p1, p2” jsou indexy odpovídajících písmen v otevřeném textu

Z této soustavy můžeme spočítat parametry “a” a “b”. Pokud budeme mít štěstí, budou to opravdu parametry, pomocí kterých byl text zašifrován a můžeme ho pomocí nich i rozšifrovat. Výhoda toho přístupu je rychlost, ale nevýhoda je horší robustnost, protože zvolená písmena si nemusí odpovídat. (například v mém případě byla prohozena frekvence prvního a druhého nejčastějšího písmene). Tato metoda je implementována ve funkci *crack().*

Druhý přístup kombinuje frekvenční analýzu a brute force. Zkusíme zašifrovaný vstupní text rozšifrovat pomocí všech kombinací parametrů “a” a ”b” (je jich 12 * 26 = 312) a pro každý takový “rozšifrovaný” text provedeme frekvenční analýzu (uděláme histogram s relativními četnostmi). Tento histogram porovnáme s histogramem češtiny a spočítame, jak moc se liší. K tomu používam součet absolutních odchylek.

$$\text{Loss} = \sum_{i}^{26} \text{abs}(\text{histogram\_textu}[i] - \text{czech\_frequency}[i])$$

Průběžně si ukládám parametry “a” a “b” nejlepšího “rozšifrovaného” textu. Pro projití všech možností budou pravděpodobně parametry “a” a “b”, které vytvořily “rozšifrovaný” text nejpodobnější češtině správnými (de)šifrovacími parametry. Výhodou tohoto přístupu je vyšší robustnost (není třeba žádný odhad písmen), ale nevýhoda je vyšší časová složitost, protože vpodstatě děláme brute force útok. Tato metoda je implementována ve funkci *crack_BF().*