Afinitní šifra (Affine cipher)

V tomto projektu se seznámíte s jednoduchou substituční šifrou, konkrétně s Afinitní šifrou, která se používá k šifrování textových zpráv. Její základní princip spočívá v lineární transformaci jednotlivých písmen v textu. Váš program bude umět šifrovat, dešifrovat a prolamovat tuto šifru. Afinitní šifra se skládá ze dvou klíčů "a" a "b". Tyto klíče ovlivňují, jaké písmeno bude z původního textu zakódováno na konkrétní písmeno v zašifrovaném textu.

Šifrování:
Celá rovnice pro zašifrování jednoho písmena je následující:
E(x) = (a * x + b) % 26
kde:
E(x) - je zašifrovaný znak s indexem x
a - je první část klíče
b - je druhá část klíče
x - je index znaku, který zašifrujeme
% - operace modulo

Klíč "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.

Příklad použití Afinitní šifry s koeficienty a = 3 a b = 7: Předpokládejme, že máme zprávu, kterou chceme zašifrovat: "TOTO JE TAJNA ZPRAVA". Každé písmeno převedeme na číslo odpovídající jeho pozici v abecedě (A=0, B=1, C=2, atd.). Následně každé písmeno vynásobíme jeho číselnou hodnotu koeficientem "a" a přičteme konstantu "b". Použitím našich koeficientů a=3 a b=7 dostáváme:
T -> (3*19+7) mod 26 = 12 -> M
O -> (3*14+7) mod 26 = 23 -> X
T -> (3*19+7) mod 26 = 12 -> M
…..
V -> (3*21+7) mod 26 = 18 -> S
A -> (3*0+7) mod 26 = 7 -> H

Zašifrovaný text vypadá takto: MXMX IT MHIUH EAGHSH

Dešifrování:
$D(x) = a^{-1}(x - b) \% 26$
kde:
D(x) - je dešifrovaný znak s indexem x
a^{-1} - je multiplikativní inverze a
x - je index znaku který zašifrujeme
b - je druhá část klíče
Zašifrovaný text: MXMX IT MHIUH EAGHSH
Proces dešifrování:
M -> 9*(12-7) % 26 = 19 -> T
X -> 9*(23-7) % 26 = 14 -> O
M -> 9*(12-7) % 26 = 19 -> T
…. Atd.
Výsledkem je: TOTO JE TAJNA ZPRAVA

Dešifrování bez znalosti klíčů:
Frekvenční analýza je jedna z nejúčinnějších metod na prolomení Afinitní šifry. Princip této metody spočívá v analýze četnosti výskytu písmen v zašifrovaném textu, což umožní odhadnout pravděpodobnost toho, která písmena se v textu nejčastěji vyskytují. Při použití Afinitní šifry jsou písmena v otevřeném textu nahrazována jinými písmeny podle matematického vzorce, který používá dva klíče "a" a "b". Četnost výskytu písmen v zašifrovaném textu zůstává nezměněna, a proto je vhodné použít frekvenční analýzu na její prolomení.

Prvním krokem při použití frekvenční analýzy na Afinitní šifru je určení četnosti jednotlivých písmen v zašifrovaném textu. Tato analýza by měla ukázat, která písmena se nejčastěji vyskytují v textu. V anglickém jazyce jsou to obvykle písmena "e", "t", "a", "o" a "i" (v českém jazyce budou jiná).
Dalším krokem je porovnání četnosti výskytu jednotlivých písmen v zašifrovaném textu s četností výskytu písmen v českém jazyce. Pokud se některé písmeno v zašifrovaném textu vyskytuje podstatně častěji než v českém jazyce, může to být indikací toho, že toto písmeno odpovídá jednomu z nejčastějších písmen v otevřeném textu.

Dalším krokem je použití znalosti o četnosti výskytu písmen k odhadnutí hodnoty klíčů "a" a "b". Pokud se podaří odhadnout několik písmen v zašifrovaném textu, lze na základě těchto znalostí získat přibližné hodnoty klíčů "a" a "b".
TIP: Klíč “a” může mít pouze několik konkrétních hodnot.

Požadavky

Program implementujte v jazyce C/C++, pro implementaci nesmíte využít již existujících řešení v knihovnách. Je zakázáno využit GitHub Copilot, Chat GPT a jiné AI technologie. Pro prolomení Afinitní šifry můžete využit jakoukoliv knihovnu, mimo knihovny s úplnou implementací této problematiky. V dokumentaci popište implementaci Afinitní šifry jen velmi stručně, zaměřte se na její prolomení. Popište, jak váš program provádí frekvenční analýzu při prolamování šifry. Program musí být rychlý, příliš dlouhý výpočet může vést k bodové srážce.

Formát vstupních a výstupních textů
Vstupní text pro šifrování se znalostí klíče se bude skládat pouze ze znaků velké/malé abecedy a znaku mezery, který ve výstupní textu zachováte. Vstupní text pro šifrování bude v českém jazyce bez diakritiky a interpunkce (Jinak řečeno pouze [a-zA-Z]+ a znak mezery) nebo anglickém v případě jejich mix. Pro šifrování/dešifrování se znalostí klíče je jazyk nepodstatný, podstatné jsou pouze znaky!
Příkladem vstupních textů:
- Naprogramuj affine cipher
- Vysledny kod odevdej do Moodle vcetne dokumentace v souboru pdf

Výstupní text pro šifrování
Výstupní text bude zašifrován pomocí Afinitní šifry a zadaným klíčem.
Příklady zašifrovaných textů:
Naprogramuj affine cipher -> REBLWILEMAX EDDSRY OSBNYL (klíč a=5, b=4)
Vysledny kod odevdej do Moodle vcetne dokumentace v souboru pdf ->
VEMRWTXE OAT ATWVTWL TA UAATRW VQWPXW TAOSUWXPKQW V MASNAJS DTZ (klíč a=3, b=10)

Výstupní text pro dešifrování:
Výstupní text po dešifrování bude napsán pomocí české velké abecedy bez interpunkce a diakritiky.
Přiklad dešifrování:
REBLWILEMAX EDDSRY OSBNYL -> NAPROGRAMUJ AFFINE CIPHER (klíč a=5, b=4)

Vstupní a výstupní text pro dešifrování bez znalosti klíče:
Vstupní soubor bude obsahovat jeden řádek se zašifrovaným textem pomocí Afinitní šifry a klíče. Vaším úkolem bude tuto zprávu dešifrovat pomocí **frekvenční analýzy** a získat klíč. Hodnotit se bude jak **dešifrovaný text,** tak **získané klíče**. Zašifrovaný text bude vždy v **českém jazyce** bez diakritiky a interpunkce (takže zase jen [a-zA-Z]+ a znak mezery).
Výstupní text:
Výstupní text následně uložíte do souboru, který dostanete pomocí parametrů. Výstupní text bude ve stejném formátu jako při znalosti klíče tedy pouze velká písmena a mezery. S tím že na standartní výstup vypíšete odhadované klíče, viz příklady spuštění.

Příklady spuštění:
Program půjde spustit celkem ve třech provedení
1, šifrování
2, dešifrování
3, dešifrování bez znalosti klíče
Program se bude spouštět s několika parametry:
-e – parametr označující šifrování
-d – parametr označující dešifrování
-c – parametr označující dešifrování bez znalosti klíče
-a x – parametr označující klíč “a” (x označuje hodnotu například 3)
-b y – parametr označující klíč “b” (x označuje hodnotu například 7)
-f <název souboru> - parametr označující cestu k souboru.
-o <název souboru> - parametr označující výstupní soubor s otevřeným textem.
Šifrování (-e)
vstup:
./kry -e -a 3 -b 7 “TOTO JE TAJNA ZPRAVA”
výstup:
MXMX IT MHIUH EAGHSH

Dešifrování (-d)
vstup:
./kry -d -a 3 -b 7 “MXMX IT MHIUH EAGHSH”
výstup:
TOTO JE TAJNA ZPRAVA

Dešifrování bez znalosti hesla:
vstup:
./kry -c -f “soubor.txt.enc” -o “soubor.txt”
výstup:
a=3,b=7
Testování
Hodnocení bude následující:
1, Dokumentace - 1 bod
2, Správné šifrování - 1,5 body
3, Správně dešifrování s klíčem - 1,5 body

4, Prolomení šifry bez znalosti klíče pomocí frekvenční analýzy. Zde se bude brát počet správně obnovených slov vůči všech slov v otevřeném textu. - 3 body

Další poznámky:
- Při implementaci můžete využít libovolnou knihovnu (mimo s hotovou implementací dané problematiky) na serveru merlin.fit.vutbr.cz, na kterém se budou vaše programy testovat.
- Při řešení projektu využijte materiály, které máte poskytnuté v rámci přednášek KRY.
- Další zdroje:
  [1] https://math.asu.edu/sites/default/files/affine.pdf
  [2] https://www.dcode.fr/affine-cipher
  [3] https://digilib.k.utb.cz/bitstream/handle/10563/51862/%C5%BEen%C4%8D%C3%A1k_2022_dp.pdf
  [4] https://www.matweb.cz/frekvencni-analyza/
  [5] https://wikisofia.cz/wiki/Frekven%C4%8Dn%C3%AD_anal%C3%BDza

Odevzdání

Do E-learningu (Moodlu) odevzdávejte archiv .zip pojmenovaným vašim osobním číslem například 1233456.zip se zdrojovými kódy vaší aplikace a dokumentací. Aplikace musí obsahovat makefile, který po zavolání příkazu "make" vytvoří spustitelný soubor "kry".

- Projekt odevzdávejte ve formátu ZIP (TAR, 7Zip, RAR a jiné == 0 bodů)
- Využití technologie Chat GPT, Github Copilot a jiné AI technologie == 0 bodů
- Archiv pojmenujte Vašim osobním číslem: 123456.zip
- Archiv v sobě nebude obsahovat žádné složky, složky pojmenované Vašim loginem, src, a podobné == 0 bodů
- Makefile vytvoří program s názvem ./kry.
- Během běhu nevypisujte na stdout zbytečnosti, případné chyby vypisujte na stderr.
- Všechny zadané vstupy při testování budou korektní, není potřeba kontrolovat zadané argumenty nebo formát vstupů.

Konzultace k projektu poskytuje Ing. Daniel Snášel: isnasel@fit.vutbr.cz
Datum odevzdání: 9.4.2023