Programátorská příručka

Prolomení polyalfabetické šifry

V Brně 29. května Vojtěch Matulík

Skript napsaný v jazyce Python s názvem *projekt.py*, který lze nalézt v příloze, slouží k zašifrování anglicky psaného textu pomocí polyalfabetické šifry (konkrétně vigenerovy šifry) a dále i k odšifrování anglicky psaného textu zašifrovaného vigenerovou šifrou bez znalosti klíče.

Ke spuštění skriptu je potřeba nainstalovat si všechny knihovny, které skript používá. Skript byl naprogramován za použití interpreteru verze 3.11.9 64-bit (Microsoft Store). Je potřeba mít všechny přílohy .zip souboru (soubor, který byl vložen do odevzdávárny) ve stejné složce po rozbalení .zip souboru a spuštění skriptu.

Nyní k popisu jednotlivých funkcí v kódu. Funkce vigenere\_cipher slouží k šifrování nebo dešifrování textu pomocí Vigenèrovy šifry. Má tři parametry: text, což je vstupní text, který chceme šifrovat nebo dešifrovat, key, což je klíč používaný při šifrování nebo dešifrování, a mode, který určuje, zda se má text šifrovat (hodnota 'encrypt') nebo dešifrovat (hodnota 'decrypt'). Klíč je výchozí prázdný řetězec, což znamená, že musí být vždy specifikován, jinak nebude šifrování nebo dešifrování možné. Nejprve funkce inicializuje prázdný seznam final\_text, který bude obsahovat výsledný text po šifrování nebo dešifrování. text\_chars je seznam jednotlivých znaků vstupního textu a non\_alpha\_count je čítač pro sledování nealfabetických znaků ve vstupním textu. Funkce pak prochází každý znak ve vstupním textu pomocí smyčky for. Pro každý znak zkontroluje, zda je písmenem pomocí metody isalpha. Pokud je znak písmenem, získá odpovídající znak z klíče pomocí výrazu key[(i - non\_alpha\_count) % len(key)]. Tento výraz zajistí, že se klíč opakuje v případě, že je kratší než vstupní text, a zároveň se posouvá správně i v případě, že ve vstupním textu jsou nealfabetické znaky. Pokud je mode nastaven na 'encrypt', funkce provede šifrování tím, že sečte numerické hodnoty aktuálního znaku a znaku z klíče, odečte dvojnásobek hodnoty 'A' a výsledek vezme modulo 26, aby se zajistilo, že výsledná hodnota zůstane v rozsahu 0–25. Pokud je mode nastaven na 'decrypt', funkce provede dešifrování tím, že od numerické hodnoty aktuálního znaku odečte numerickou hodnotu znaku z klíče, přičte 26 a výsledek vezme modulo 26. Výsledná hodnota se poté převede zpět na znak přidáním hodnoty 'A' a tento znak se přidá do seznamu final\_text. Pokud aktuální znak není písmenem, přidá se do seznamu final\_text beze změny a zvýší se hodnota non\_alpha\_count. Nakonec funkce spojí všechny znaky v seznamu final\_text do jednoho řetězce a tento řetězec vrátí jako výsledek.

Funkce find\_key\_length slouží k nalezení délky klíče použitého při šifrování textu pomocí Vigenèrovy šifry. Tento proces je založen na analýze opakujících se slov v textu a jejich vzdálenostech, což umožňuje odhadnout pravděpodobnou délku klíče. Funkce se skládá z několika kroků, které jsou popsány níže. Nejprve se text rozdělí na jednotlivá slova pomocí metody split(), což vytvoří seznam slov. Následně se pomocí defaultdict z knihovny collections vytváří slovník word\_positions, kde se sledují pozice každého slova v textu. Pro každé slovo v textu se jeho pozice (index) přidá do seznamu v odpovídajícím záznamu ve slovníku. Poté se filtrují pouze ta slova, která se opakují, tedy mají délku seznamu pozic větší než 1. Dále se provede další filtr, aby se zohlednila pouze slova s délkou alespoň 3 znaky, což pomáhá odstranit běžná krátká slova, jako je například "the" v angličtině. Následuje odstranění všech nealfabetických znaků z textu pomocí regulárního výrazu re.sub(r'[^a-zA-Z]', '', text), aby se usnadnila další analýza. Pro každé opakující se slovo se pak pomocí regulárního výrazu hledají všechny jeho výskyty v textu a jejich začátky se zaznamenávají do slovníku starts. Poté se najde slovo s největším počtem výskytů. Pro toto slovo se vypočítají vzdálenosti mezi jednotlivými výskyty tím, že se od sebe odečítají pozice začátků výskytů. Tyto vzdálenosti se uloží do seznamu final\_lengths. Seznam final\_lengths se následně rozdělí na trojice a pro každou trojici se vypočítá největší společný dělitel (NSD) pomocí funkce math.gcd. Výsledky těchto výpočtů se uloží do seznamu gcd\_results. Konečně se pomocí Counter z knihovny collections zjistí nejčastější hodnota v seznamu gcd\_results, což je pravděpodobná délka klíče. Pokud je tato hodnota 1, vezme se druhá nejčastější hodnota, protože délka klíče 1 je nevhodná. Funkce vrátí nejčastější (nebo druhou nejčastější, pokud je první hodnota 1) hodnotu ze seznamu gcd\_results, což je odhadovaná délka klíče.

Funkce frequency\_analysis provádí frekvenční analýzu textu, což je metoda používaná k odhalení délky klíče a samotného klíče při dešifrování Vigenèrovy šifry. Tento proces zahrnuje několik kroků, které jsou popsány níže. Nejprve funkce odstraní všechny nealfabetické znaky z textu pomocí regulárního výrazu re.sub(r'[^a-zA-Z]', '', text). Tím se zajistí, že zůstanou pouze písmena, což je nezbytné pro správnou frekvenční analýzu. Poté inicializuje seznam blocks, který obsahuje gcd prázdných řetězců, kde gcd je délka klíče nalezená předchozí funkcí. Tento seznam bude sloužit k rozdělení textu na bloky podle délky klíče. Následně funkce distribuuje jednotlivé znaky textu do příslušných bloků. To se provádí pomocí smyčky, která pro každý znak vypočítá index bloku jako zbytek po dělení indexu znaku v textu délkou klíče (i % gcd). Každý znak je pak přidán do odpovídajícího bloku. Pro každý blok pak funkce provede frekvenční analýzu, aby našla nejčastěji se vyskytující znak v každém bloku. Toho dosáhne tak, že pro každý znak v bloku počítá jeho výskyt a ukládá ho do slovníku char\_occurrences. Následně najde znak s nejvyšším počtem výskytů pomocí max(char\_occurrences, key=char\_occurrences.get) a přidá ho do seznamu most\_common\_chars. Každý znak v tomto seznamu představuje nejčastěji se vyskytující znak v odpovídajícím bloku. Abychom získali pravděpodobný klíč, upraví funkce tyto znaky tak, že od jejich numerické hodnoty odečte 4 (což odpovídá tomu, že 'E' je v angličtině nejčastější písmeno). Numerická hodnota znaku je vypočítána jako ord(char) - ord('A'). Poté se provede operace modulo 26, aby se zajistilo správné zabalení kolem abecedy. Upravené hodnoty jsou pak převedeny zpět na znaky a přidány do seznamu adjusted\_values. Nakonec funkce spojí upravené znaky do jednoho řetězce a vrátí tento řetězec jako výsledek. Tento řetězec je pravděpodobný klíč použitý při šifrování textu.

Funkce get\_text\_from\_file umožňuje bezpečné a správné načtení textu z jakéhokoli souboru tím, že automaticky detekuje kódování souboru a přizpůsobí se mu. To je zvláště užitečné při práci s textovými soubory různého původu, kde kódování může být neznámé nebo různé.

Nakonec následuje logika vkládání vstupu do terminálu, vypisování do terminálu a vypisování do .txt souborů.