

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

NKS ZADANIE 1
SEMINÁRNA PRÁCA

Študijný program:	Aplikovaná informatika
Predmet:	I-NKS – Návrh a kryptoanalýza šifier
Prednášajúci:	prof. Ing. Pavol Zajac, PhD.
Cvičiaci:	prof. Ing. Pavol Zajac, PhD.

Bratislava 2023

Bc. Soóky Dávid

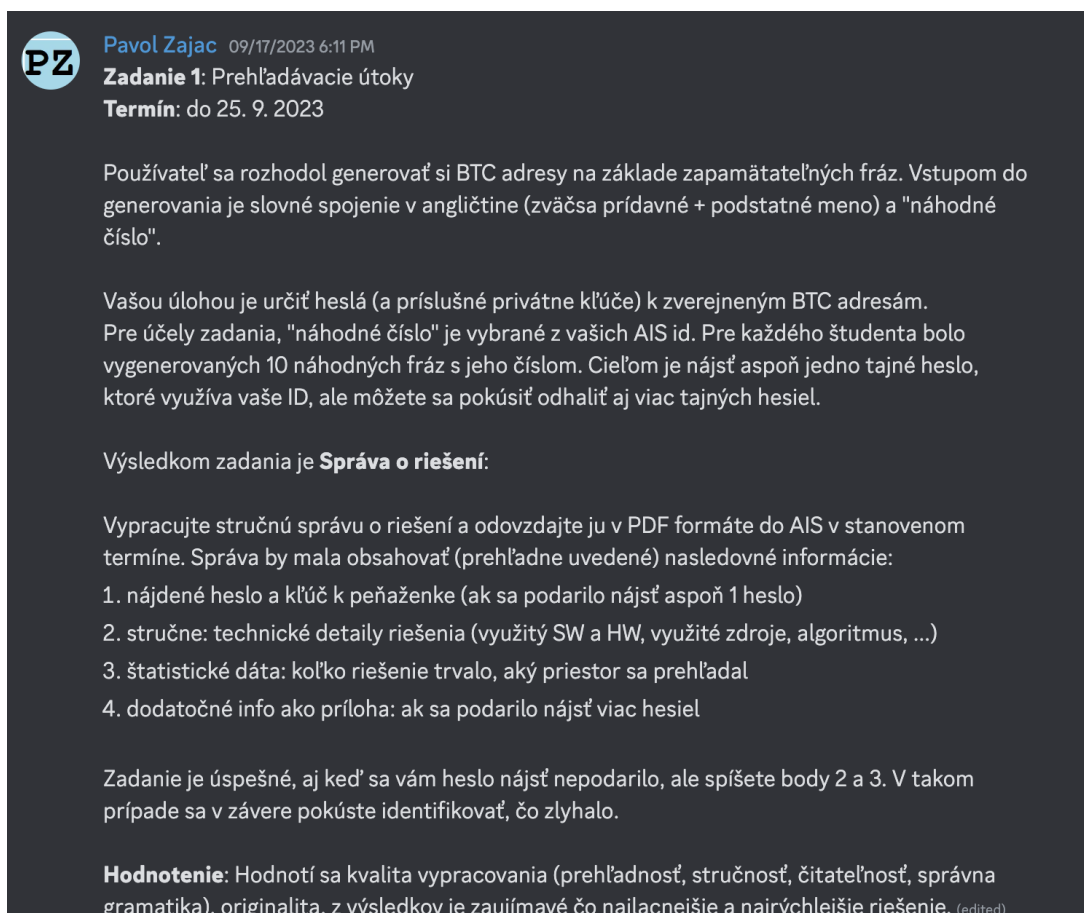
Obsah

1	Zadanie	1
2	Nájdené heslo a kľúč k peňaženke	2
3	Technické detaily riešenia	3
3.1	Hardvér a softvér	3
3.2	Algoritmy a zdroje	3
3.3	Multiprocesing v jazyku Python	4
4	Štatistické dáta	5
4.1	Požadovaný výkon	5
4.2	Chyby, ktoré sa môžu vyskytnúť u ostatných	5

Zoznam obrázkov a tabuliek

Obrázok 1	Zadanie na obrazovke nasnímané z nášho triedného discord kanála.	1
Obrázok 2	Špecifikácie môjho notebooku.	3
Tabuľka 1	štatistická tabuľka použitých algoritmov	5

1 Zadanie



PZ Pavol Zajac 09/17/2023 6:11 PM
Zadanie 1: Prehľadavacie útoky
Termín: do 25. 9. 2023

Používateľ sa rozhodol generovať si BTC adresy na základe zapamätateľných fráz. Vstupom do generovania je slovné spojenie v angličtine (zväčša prídavné + podstatné meno) a "náhodné číslo".

Vašou úlohou je určiť heslá (a príslušné privátne kľúče) k zverejneným BTC adresám. Pre účely zadania, "náhodné číslo" je vybrané z vašich AIS id. Pre každého študenta bolo vygenerovaných 10 náhodných fráz s jeho číslom. Cieľom je nájsť aspoň jedno tajné heslo, ktoré využíva vaše ID, ale môžete sa pokúsiť odhaliť aj viac tajných hesiel.

Výsledkom zadania je **Správa o riešení**:

Vypracujte stručnú správu o riešení a odovzdajte ju v PDF formáte do AIS v stanovenom termíne. Správa by mala obsahovať (prehľadne uvedené) nasledovné informácie:

1. nájdené heslo a kľúč k peňaženke (ak sa podarilo nájsť aspoň 1 heslo)
2. stručne: technické detaily riešenia (využitý SW a HW, využité zdroje, algoritmus, ...)
3. štatistické dáta: koľko riešenie trvalo, aký priestor sa prehľadal
4. dodatočné info ako príloha: ak sa podarilo nájsť viac hesiel

Zadanie je úspešné, aj keď sa vám heslo nájsť nepodarilo, ale spíšete body 2 a 3. V takom prípade sa v závere pokúste identifikovať, čo zlyhalo.

Hodnotenie: Hodnotí sa kvalita vypracovania (prehľadnosť, stručnosť, čitateľnosť, správna gramatika), originalita, z výsledkov je zaujímavé čo najlacnejšie a najrýchlejšie riešenie. (edited)

Obr. 1: Zadanie na obrazovke nasnímané z nášho triedného discord kanála.

2 Nájdené heslo a kľúč k peňaženke

Úspešne som našiel 1 heslo a príslušný kľúč.

Moje heslo bolo

poormethod97935

a kľúč bol

6113c35041829ded7efcede2daae884f0ce1f36fc92541bfc646b8ef5f761555.

Použitie týchto vstupov btc adresa vygenerovaná algoritmom je

1H9xtWiJgs7UzKNaUBX8TTaaPeqNWcAEEF

ktorú možno nájsť v riadke 134. v súbore **z23-01-pk.txt**.

3 Technické detaily riešenia

3.1 Hardvér a softvér

Pri tejto úlohe som používal **python** ako programovací jazyk, \LaTeX na písanie požadovanej dokumentácie a svoj Macbook Pro 13 z roku 2019.



Obr. 2: Špecifikácie môjho notebooku.

3.2 Algoritmy a zdroje

Najprv som chcel len hrubou silou dopracovať k riešeniu pomocou najjednoduchšej metódy to znamená pokusov s **aaaaaaaa** a postupne meniť znaky, kým nenájdeme riešenie. Na seminári sme sa dozvedeli, že prídavné mená a podstatné mená sú tvorené najmenej 4 znakmi a najviac 10 znakmi. Takže musíme začať kontrolovať len reťazce s dĺžkou 8 až 20.

Tento algoritmus by zrejme trval najdlhšie nielen preto, že využíva všetky možné kombinácie, ale má aj obrovskú chybu. Touto chybou je, že naše heslo je kombináciou prídavného mena a podstatného mena, a nie len náhodným poradím znakov. Pri použití tejto metódy hrubej sily by boli takmer všetky naše pokusy zbytočné. Ale predtým, ako som to vyskúšal, som si to rýchlo rozmyslel a uvedomil som si, že existuje jednoduchšia metóda. Vieme, že v hesle máme prídavné meno a podstatné meno. Musíme teda nájsť 2 slovníky, ktoré budú jeden pre prídavné mená a druhý pre podstatné mená.

Oba slovníky som rýchlo našiel na githube. Sú k dispozícii na tomto odkaze.

<https://github.com/hugsy/stuff/tree/main/random-word>

Pomocou týchto slovníkov som práve vytvoril jednoduchý program v jazyku Python, ktorý pomocou dvoch **cyklov for** vyskúša každé prídavné meno s každým podstatným menom.

Kód a použité slovníky v jednom priečinku sú k dispozícii aj na mojom githube, ak by niekoho zaujímalo, ako to funguje. Snažil som sa komentovať každú jeho časť, aby sa dala ľahko čítať.

<https://github.com/Jedwarsh/NKS-Project-1>

3.3 Multiprocesing v jazyku Python

Python je vysokoúrovňový programovací jazyk a z toho vyplýva veľká chyba. Spočiatku využíva iba 1 jadro procesora. Ak máme kód, ktorý nie je založený na predchádzajúcich operáciách, môžeme využiť všetky jadrá procesora, aby náš kód bežal rýchlejšie. V mojom prípade pri použití môjho počítača je vďaka tomu môj kód 4-krát rýchlejší.

4 Štatistické dáta

	1 Príd.+Podst. Meno	Podstatné Meno	Všetko
Hrubá sila	?	?	1.65549×10^{22} rokov
Hrubá sila (multi)	?	?	4.02625×10^{19} rokov
Slovník	0,7 sekund	16 minút	400 hodín
Slovník (multi)	0,17 sekund	4 minút	100 hodín

Tabuľka 1: štatistická tabuľka použitých algoritmov

Pri práci na kóde som úplne zabudol na dĺžku hesla. Ak by som to neurobil, mohol som slovníkový prístup trochu urýchliť odstránením prídavných mien a podstatných mien, ktoré buď neobsahujú dostatok znakov, alebo obsahujú príliš veľa znakov. Ale teraz pri pohľade na súbory by to mohlo priniesť len 1 až 5% zvýšenie výkonu. Výpočty pre algoritmus hrubej sily boli zjavne vykonané s ohľadom na dĺžku hesla.

4.1 Požadovaný výkon

Hoci nie som si istý presnou spotrebou energie môjho počítača, odhadom a podľa môjho prieskumu by to malo byť 60 wattov. Len na základe kalkulačiek náklady na prevádzku tohto stroja a pythonovských kódov spotrebujeme 1,44 kWh denne. Ak nás jedna kWh stojí 0,10 €, odhadované sumy sa dajú ľahko vypočítať.

Viacprocesorový slovníkový prístup by nás stál **0,40 €**.

Prístup so slovníkom s jedným spracovaním by nás stál **1,60€**.

Prístup hrubej sily s použitím multiprocessingu by nás stál **2,118 € $\times 10^{21}$** .

Prístup hrubej sily s použitím singleprocessingu by nás stál **8,70 € $\times 10^{22}$** .

4.2 Chyby, ktoré sa môžu vyskytnúť u ostatných

Používanie súboru prídavných mien a podstatných mien má samozrejme aj svoje chyby. Najdôležitejšou z nich je, že náš slovník nemusí obsahovať slová použité v našom hesle. Našťastie to pre mňa nebol problém.