SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ÚVOD DO POČÍTAČOVEJ BEZPEČNOSTI
ZADANIE Č.2 - DOCHÁDZKOVÝ SYSTÉM

Adrián Somor, Samuel Gibala, Matúš Kornhauser, Ľuboš Likó, Peter Barnas

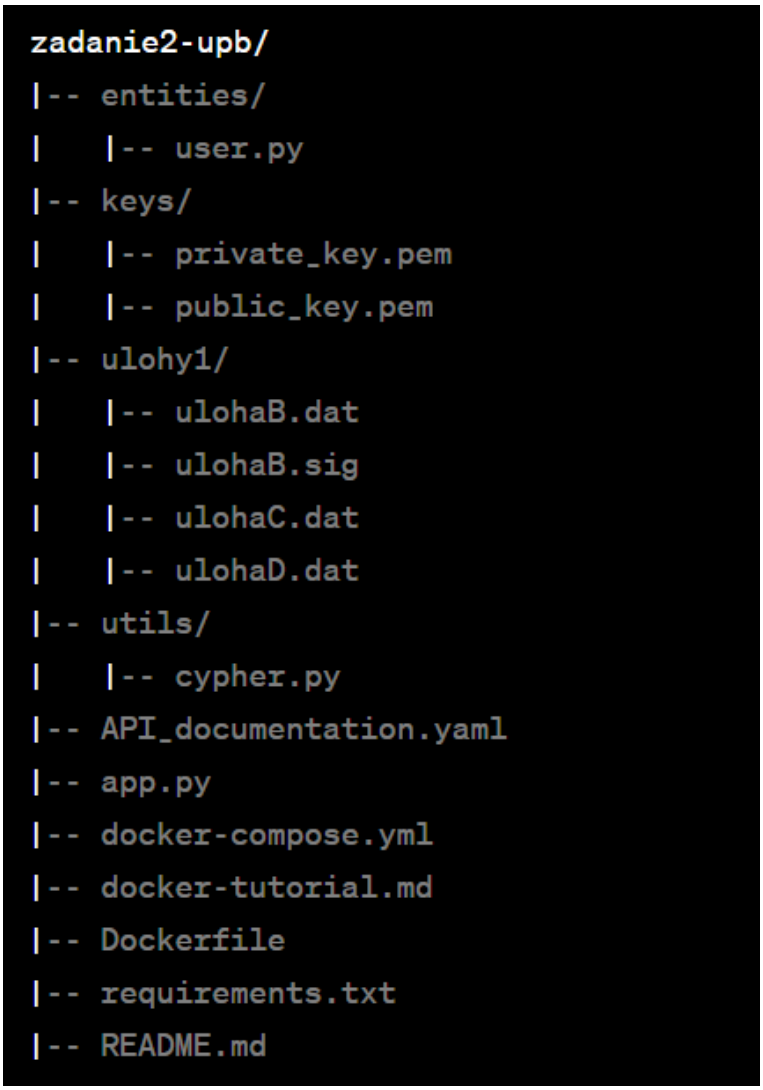
1 Použité technológie

- jazyk: Python (Flask 3.0.0)
- databáza: PostrgreSQL
- kryptografická knižnica: cryptography 41.0.4

Aplikácia vykonáva niekoľko kryptografických operácií a obsahuje funkcie na vytváranie a manipuláciu so zašifrovanými súbormi. Tieto funkcie využívajúce najmä funkcionalitu knižnice *cryptofraphy* sú obsiahnuté v triede *CryptographyUtils* v súbore *utils/crypto.py*. Tieto funkcie zabezpečujú potrebné nástroje pre šifrovanie, dešifrovanie, podpísanie a verifikáciu potrebné pre vypracovanie daného zadania.

Súbory a ich štruktúra sú zobrazené na obr. 1. Súbory a priečinky:

- entities - entity použité v databáze
- keys - privátny a verejný kľúč v pem formáte
- utils - nástroje použité v projekte - momentálne obsahuje súbor s triedou na kryptografiu
- ulohy1 - obsahuje súbory zobrazujúce jednotlivé výsledky zadaných úloh v zadaní 2, úlohy sú pomenované uloha{číslo úlohy}, typ .dat označuje dáta a .sig podpísaný súbor
- API_documentation.yaml - OpenAPI 3 špecifikácia API
- docker-compose.yml - súbor definujúci špecifikáciu a štruktúru docker kontajnera
- docker-tutorial.md - inštrukcie na vytvorenie docker kontajnera
- Dockerfile - inštalačné a štrukturálne docker špecifikácie
- requirements.txt - knižnice potrebné pre fungovanie projektu, ktoré sa automaticky inštalujú do docker kontajnera
- app.py - hlavný súbor aplikácie, kde sú všetky inicializácie, endpointy a vypracovanie požadovaných úloh



Obr. 1: Štruktúra

2 Vypracovanie úloh

2.1 Úlohy 1-3

Ako už bolo spomenuté, použitý bol Python framework Flask v spojení s databázou PostgreSQL. Štruktúra je zadefinovaná v *docker-compose.yml*. V *Dockerfile* je zadefinovaný názov adresára v pracovnom adresári backend časti kontajnera spolu so súborom, z ktorého sa majú inštalovať potrebné knižnice. V našom prípade sa jedná o *requirements.txt*.

2.2 Úloha 4

Pri jednotlivých častiach úlohy 4 a používajú symetrické a asymetrické kľúče. Významy týchto pojmov nie je potrebné vysvetľovať, dôležité je ale poznamenať, ako ich v našej implementácii vytvárame a spravujeme. Symetrický kľúč je generovaný funkciou *generate_key()* ako náhodný 256 bitový kľúč a následne je uložený do premennej *symmetric_key* triedy *CryptographyUtils*, kde sa uchováva vždy posledný generovaný kľúč.

Asymetrické kľúče (súkromný a verejný) sú generované metódou *generate_asymmetric_keys()*, ktorá po vytvorení uloží kľúče do priečinka *keys* v .pem formáte. Ak by však kľúče už boli vytvorené, program ich pri spustení metódou *load_keys(public_key_filename, private_key_filename)* načíta do premenných *private_key* a *public_key* spomínanej triedy, aby s nimi bolo možné následne pracovať

2.2.1 Extrakcia dát z databázy

Na extrakciu dát z databázy bola použitá knižnica psycopg. Psycopg je najpopulárnejší adaptér databázy PostgreSQL pre programovací jazyk Python. Jeho hlavnými vlastnosťami sú kompletná implementácia špecifikácie Python DB API 2.0 a bezpečnosť vlákien (niekoľko vlákien môže zdieľať rovnaké pripojenie). Bol navrhnutý pre silne viacvláknové aplikácie, ktoré vytvárajú a rušia veľa kurzorov a vykonávajú veľký počet súbežných INSERT alebo UPDATE operácií. Psycopg 2 je väčšinou implementovaný v jazyku C ako wrapper libpq, čo vedie k jeho efektívnosti a bezpečnosti. Obsahuje kurzory na strane klienta a servera, asynchrónnu komunikáciu a iné. [1]

2.2.2 Šifrovanie symetrickým kľúčom

Aplikácia používa na šifrovanie údajov symetrický kľúč, ktorý sa generuje funkciou *generate_key()* zo spomínanej triedy *CryptographyUtils*. Metóda *encrypt_message(message, output_filename)* zašifruje správu podľa aktuálne používaného kľúča. Správa s otvoreným

textom je vyplnená tak, aby zodpovedala veľkosti bloku (zvyčajne 16 bajtov). Správa sa zašifruje pomocou algoritmu AES v režime Cipher Block Chaining (CBC) pomocou poskytnutého symetrického kľúča a náhodne generovaného inicializačného vektora. Výsledný zašifrovaný súbor následne uloží na miesto definované premennou *output_filename*.

2.2.3 Šifrovanie symetrického kľúča

Na zašifrovanie symetrického kľúča verejným kľúčom príjemcu je použitá metóda *encrypt_symmetric_key(public_key_file, output_filename)*. Verejný kľúč príjemcu sa načíta zo súboru. Symetrický kľúč sa zašifruje pomocou verejného kľúča príjemcu, pričom sa použije optimálne asymetrické šifrovanie (OAEP) s hašovacím algoritmom SHA-256.

2.2.4 Dešifrovanie zašifrovaného súboru

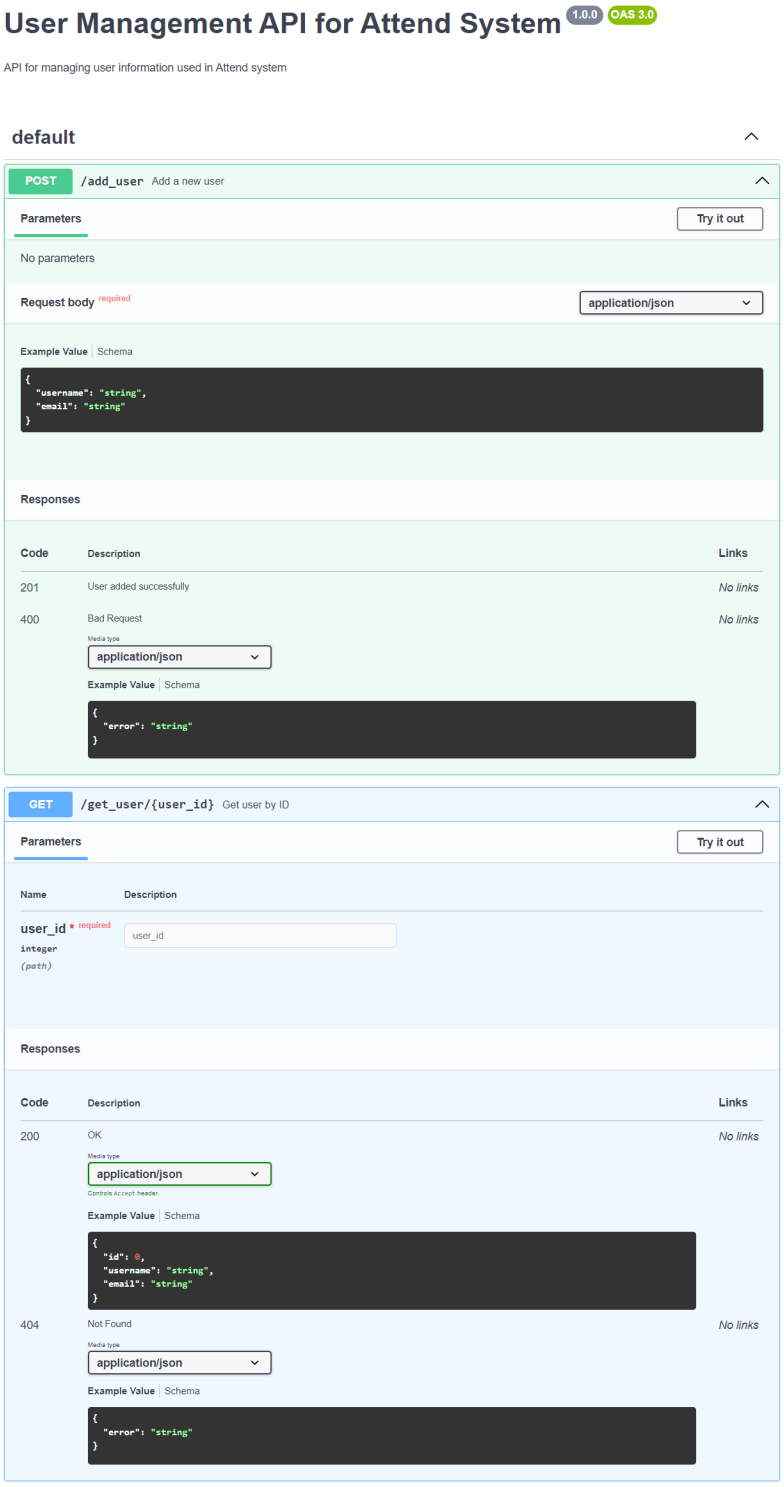
Na dešifrovanie zašifrovaného súboru pomocou symetrického kľúča je použitá metóda *decrypt_file(input_filepath, output_filepath, key)*. Metóda načíta vstupný súbor, ktorý obsahuje inicializačný vektor a zašifrované údaje. Inicializuje proces dešifrovania AES so symetrickým kľúčom a inicializačným vektorom. Šifrovaný text sa dešifruje, čím sa získa otvorený text.

2.2.5 Podpisovanie a overovanie

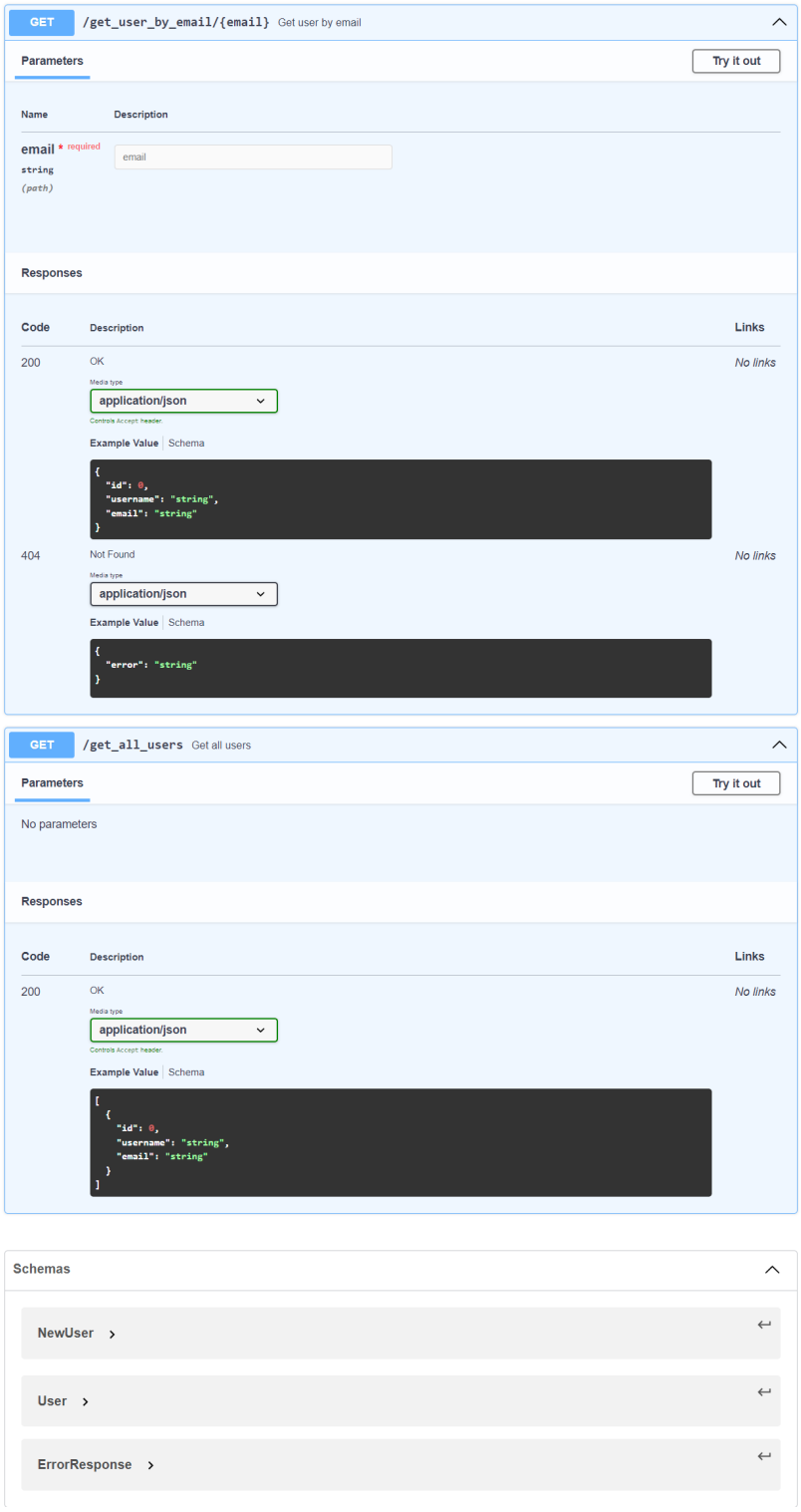
Aplikácia obsahuje aj funkcie na podpisovanie a overovanie súborov. Metóda *sign_with_private_key(message)* sa používa na podpísanie správy súkromným kľúčom. Podpis sa zapíše do súboru s príponou .sig. Metóda *verify_file(filename)* slúži na overenie podpisu súboru pomocou verejného kľúča. Tieto kryptografické operácie sa vykonávajú na zabezpečenie a overenie údajov a štruktúra zašifrovaných súborov zabezpečuje, že údaje možno bezpečne dešifrovať a overiť s istotou ich integrity a správnosti.

3 Endpointy

Nakoľko zadanie neobsahuje časť, kde by bolo explicitne potrebné použitie nejakých endpointov na komunikáciu, môžme túto časť považovať za bezpredmetnú. Z edukačného a hľadiska sme však vytvorili aspoň pár endpointov súvisiacich s dátami v nami použitej databázovej tabuľke. Nakoľko sa jedná o tabuľku kvázi používateľov, endpointy slúžia na pridanie používateľa, získanie používateľa podľa ID, e-mailu a získanie zoznamu všetkých používateľov. Dokumentácia k API vo forme openAPI 3.0 je uvedená v súbore *API_documentation.yaml* a rovnako ju uvádzame aj v krajšej a prehľadnejšej grafickej forme na obr. 2 a 3.



Obr. 2: API dokumentácia 1



Obr. 3: API dokumentácia 2

Zoznam použitej literatúry

1. PSYCOPG2 DEVELOPERS. *psycopg2: PostgreSQL adapter for Python* [online]. 2023. [cit. 2023-10-22]. Dostupné z : `https://pypi.org/project/psycopg2/`.