

**Správce hesel s dvoufaktorovou autentizací**

**Autor:** Michal Žernovič (230923), David Zeman(231304), Oliver Bielik (231229), Vojtěch Vaculík (230338)

**Obor:** BPC-IBE  
**Predmet:** BPC-AKR

Obsah

[Úvod 3](#_Toc85991563)

[Ciele projektu 4](#_Toc85991564)

[Teoretická časť 5](#_Toc85991565)

[Dvojfázové overenie 5](#_Toc85991566)

[Hash 5](#_Toc85991567)

[AES 5](#_Toc85991568)

[Stav riešenia 7](#_Toc85991569)

[Autori 8](#_Toc85991570)

[Záver 9](#_Toc85991571)

[Prílohy 10](#_Toc85991572)

# Úvod

V dobe kedy je potrebný účet pre každú službu ktorú používame sa viac a viac rozširujú tzv. Password Managers. Ide o programy, v ktorých si môžeme uložiť heslá a dostať sa k nim pomocou jedného „master“ hesla. Vďaka takýmto programom môžeme zredukovať rôzne risky – môžeme používať komplexné heslá a nemusíme si ich pamätať, stačí nám master heslo. Tieto programy ponúkajú aj možnosť vygenerovať dlhé a bezpečné heslá a automaticky ich uložiť, takže užívateľ nemá dôvod používať to isté heslo na viac službách zo strachu, že by viac hesiel zabudol.

Prirodzene takýto software musí byť aj dostatočne chránený rôznymi kryptografickými funkciami a algoritmami. V tomto projekte vytvoríme jednoduchú aplikáciu na ukladanie hesiel s patričnými bezpečnostnými opatreniami.

# Ciele projektu

Cílem našeho projektu je dosažení plně funkční aplikace, napsané v programovacím jazyce Python. Aplikace bude sloužit k úschově hesel a bude využívat dvoufaktorovou autentizaci pro přístup k uloženým heslům. Jedním z ideálních typů dvoufaktorové autorizace je využití aplikace od společnosti Google, tedy jejich autentizátor, který po přidání aplikace do telefonu nebo počítače vygeneruje náhodný kód, kterým se provede autorizace. Dále, všechny uschované soubory s hesly budou zabezpečeny dostatečně silným šifrováním. Uživatel bude mít na výběr ze tří různých šifrovacích algoritmů, které budou použity pro šifrování jím požadovaných hesel. Všechny změny provedené v databázi s hesly budou logovány (bude uchováván záznam) a v určitých časových intervalech bude prováděna integrita zašifrovaných souborů za pomocí hashovacích funkcí. V případě efektivního vypracování této aplikace také plánujeme implementovat GUI, které ulehčí práci.

Shrnuli, tedy všechny tyto body, tak naším hlavním cílem je dosažení aplikace, která efektivně a bezpečně uchovává všechna hesla, která se dostanou do databáze tohoto programu. Přístup do této aplikace bude chráněn dvoufaktorovou autentizací, kde si uživatel bude moci vybrat z několika možných variant a použití hashovací funkce SHA-256 pro kontrolu integrity a narušení uložených souborů.

# Teoretická časť

## Dvojfázové overenie

Dvojfázové overenie je v dnešnej dobe veľmi rozšírený spôsob overenia, či je daný užívateľ oprávnený prihlásiť sa do služby. Typicky to býva SMS kód, alebo e-mail ktorý príde vlastníkovi účtu na iné zariadenie ako to z ktorého sa prihlasuje. Dvojfázovosť tohto overenia rozumieme v tom, že sú jednotlivé kroky od seba nezávislé.

Národný úrad štandardov a technológií (NIST) v smernici v roku 2016 zavrhoval SMS kódy ako validný spôsob overenia, avšak rok na to boli vo finalizovanej smernici považované za bezpečné. [1]

## Hash

Jeden z krokov pri ukladaní hesla v kryptografii je použitie hashovacej funkcie. Je to veľmi rýchla operácia, ktorá vytvorí z pôvodného vstupu výstup fixnej dĺžky. Hashovacia funkcia je užitočná najmä pre rýchlejšie vyhľadávanie určitých dát a najmä pri ukladaní hesiel výrazne zbezpečňuje celý proces. Hashovacia funkcia musí spĺňať určité vlastnosti:

* Odolnosť voči získaniu pôvodnej správy
* Odolnosť voči získaniu inej predlohy
* Oolnosť voči kolízii, v zmysle hashovania myslené ako dva odlišné vstupy ktoré vyprodukujú rovnaký výstup. [2]

V histórii sa používali rôzne hashovacie funkcie, napríklad LM Hash používaný v operačnom systéme Microsoft Windows - ide o hash ktorý je už dávno prelomený. Jeden z dôvodov prelomenia bolo obmedzenie hesla na iba 14 znakov a každé heslo nad 7 znakov bolo rozdelené do dvoch blokov, čo výrazne znižovalo zložitosť prelomenia.

V dnešnej dobe je jedna z najpoužívanejších hashovacích funkcia SHA-256 (Secure Hashing Alogirthm), patriaca do rodiny SHA-2. Ide o rodinu hashovacích funkcí, ktorá je zatiaľ považavaná za bezpečnú. Je nástupcom prelomenej SHA-1.

## AES

AES je v súčasnosti jeden z najrozšírenejších šifrových algoritmov. Ide o symetrickú blokovú šifru tj. Pracuje s blokmi rovnakej dĺžky o veľkosti 128bit. Na rozdiel od algoritmu DES používa AES substitučne permutačnú sieť (SP sieť). Kľúče sú väčšinou dĺžky 128, 192 a 256bit.

Proces šifrovania pri AES sa skladá sa z 10 kôl, ktoré obsahujú rôzne matematické operácie. Na začiatku prevedieme operáciu XOR s časťou nášho kľúča a plaintextu. Poradie operácií kôl je nasledovné:

1. Substitute Bytes – Nahradzovanie bajtov pomocou tabuľky nazývanej S-BOX
2. Shift Rows – Posúvanie bajtov v riadkoch
3. Mix Columns – Bajty po stĺpcoch sú matematickou funkciou pozmenené
4. Add Round Key – Pridanie ďalšej časti expandovaného kľúča

Ak máme 128bitový kľúč, prebehne 10 kôl. V desiatom kole je vynechaný proces MixColumns.

# Stav riešenia

# Autori

Na riešení problematiky sa podieľa skupina 4 ľudí, Michal, Oliver, Vojtěch a David. Michal mal za úlohu vypracovať teoretickú časť dokumentu, ktorá je zameraná na úvod k problematike a jej teoretickú časť, ktorú stručne zhrnul na stranách vyššie. Vojtěch mal za úlohu popísať ciele, ktoré chceme dosiahnuť pri riešení problematiky. David v sekcii stav riešenia popisuje aktuálny stav riešenie, ktorý sa bude počas nasledujúcich týždňov meniť, nakoľko sa projekt bude stále posúvať dopredu. Oliver v časti autori opisuje, ako sa každý člen skupiny podielal na tvorbe projektu „Password managera“, či už priamo pri riešení problematiky v jazyku Python alebo pri písaní štúdie. Riešenie problematiky v jazyku Python sme rozdelili nasledvne. Michal má za úlohu navrhnúť GUI a implementovať funkcie, Vojtěch a David spoločne riešia problematiku zabezpečania pomocou HASH funkcie a AES. Oiver ma za úlohu vyriešiť dvojfaktorové overenie a ukladanie údajov do SQL tabuliek.

# Záver

# Prílohy

Zdroje

[1] NIST Softens Guidance on SMS Authentication [online] 2021 [cit. 23-10-2021] https://www.onespan.com/blog/nist-softens-guidance-sms-authentication

[2] Cryptographic hash function [online] 2021 [cit. 23-10-2021]

<https://en.wikipedia.org/wiki/Cryptographic_hash_function>

[3] Advanced Encryption Standard [online] 2021 [cit. 24-10-2021]

https://www.tutorialspoint.com/cryptography/advanced\_encryption\_standard.htm