

Střední průmyslová škola Třebíč

Maturitní práce

Hashtesting

Profilová část maturitní zkoušky

Studijní obor: Informační technologie

Třída: ITA4

Školní rok: 2024/2025 Kamil Franek

Zadání práce

ABSTRAKT

Maturitní práce na téma hashování, jeho využití a typy hashovacích algoritmů. Zabývá se problematikou spojenou s hashováním a vysvětlení použití hashování v IT. Tento dokument popisuje použitou technologii, praktiky a vytváření samotného programu a všeho okolo. Výsledný program disponuje základními i rozšířenými funkcemi práce s hashem a soubory pro zvýšení efektivity práce. Dále disponuje okénkem pro informace, vysvětlení rozdílů mezi hashovacími algoritmy, rozdíl mezi použitím a nepoužitím soli a dále za účelem zvýšení chápání daného téma pro uživatele.

KLÍČOVÁ SLOVA

Maturitní práce, Password JailBreak, Výuka, Hash

ABSTRACT

\*abstrakt anglicky\*

KEYWORDS

Graduation thesis, Password JailBreak, Teaching, Hash

PODĚKOVÁNÍ

Děkuji Ing. Ladislavu Havlátu a oponentu Ing. Drahomíru Škárkovi za cenné připomínky a rady, které mi poskytli při vypracování maturitní práce.

V Třebíči dne 25. října 2024 podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval/a samostatně a uvedl/a v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil/a.

V Třebíči dne 25. října 2024

podpis autora

Obsah

[Úvod 6](#_Toc147495618)

[1 Struktura maturitní práce 7](#_Toc147495619)

[1.1 Titulní list 7](#_Toc147495620)

[1.2 Zadání maturitní práce 7](#_Toc147495621)

[1.3 Abstrakt 7](#_Toc147495622)

[1.4 Klíčová slova 7](#_Toc147495623)

[1.5 Poděkování a prohlášení 8](#_Toc147495624)

[1.5.1 Poděkování 8](#_Toc147495625)

[1.5.1 Prohlášení 8](#_Toc147495626)

[2 Textová část 9](#_Toc147495627)

[2.1 Styly 9](#_Toc147495628)

[2.1.1 Řádkování 9](#_Toc147495629)

[2.1.2 Zvýrazňování textu 9](#_Toc147495630)

[2.1.3 Členění textu 9](#_Toc147495631)

[2.1.4 Číslování stran 9](#_Toc147495632)

[2.2 Vytvoření obsahu 10](#_Toc147495633)

[2.3 Psaní úvodu 10](#_Toc147495634)

[2.4 Struktura odstavců 11](#_Toc147495635)

[2.5 Obrázky, tabulky a rovnice 11](#_Toc147495636)

[2.6 Řazení a struktura kapitol 13](#_Toc147495637)

[2.7 Závěr 13](#_Toc147495638)

[2.8 Seznam použitých zdrojů 13](#_Toc147495639)

[2.9 Seznam použitých symbolů a zkratek 13](#_Toc147495640)

[2.10 Seznamy použitých obrázků a tabulek 14](#_Toc147495641)

[2.11 Seznam příloh 14](#_Toc147495642)

[Závěr 15](#_Toc147495643)

[Seznam použitých zdrojů 16](#_Toc147495644)

[Seznam použitých symbolů a zkratek 17](#_Toc147495645)

[Seznam obrázků 18](#_Toc147495646)

[Seznam tabulek 19](#_Toc147495647)

[Seznam příloh 20](#_Toc147495648)

Úvod

Cílem této práce a programu je zjednodušení práce s hashema, ukázku rozdílů mezi hashovacími algoritmy, používání soli a pepře a podrobné vysvětlení, postupné hashování pomocí mezikroků, silné a slabé stránky hashů a kde a proč se používají. Dále program obsahuje test prolomení hesla (jailbreak) pomocí hrubého útoku. V dokumentu jsou popsány použité programy a technologie, jak jsou použité a proč jsou použité. V praktické části je popsána celá cesta dělání programu, hlavní problémy, trable a vysvětlení fungování celého programu s ukázkami samotného kódu, testování a různé obrázky z pracovního postupu. V závěru jsou popsány moje pocity z práce na projektu a spokojenost s finální verzí programu.

# Teorie hashovaní

Hashování je matematický algoritmus pro převod dat do předem určitého dlouhého výstupu podle algoritmu. Hashe se používání k uschování důležitých informací (hesla), která nejsou potřeba šifrovat (vrátit data zpátky), dělání kontroly a integrity dat (checksum), vytváření a ověřování elektronického podpisu, hledání škodlivého malwaru antivirovým programem, k hledání úseků DNA sekvencí atd.  
Hashe mají několik výtečných vlastností: vstupní data můžou být jakkoliv dlouhé, minimální změna v datech znamená velký rozdíl ve výstupech, nedá se získat z výstupních dat vstupní data (maximálně pomocí brute force útoku) a s větší délkou se exponenciálně zmenšuje šance na stejnost výstupních hodnot při jiném vstupu.

„Formálně jde o funkci *h*, která převádí vstupní posloupnost [bitů](https://cs.wikipedia.org/wiki/Bit) (či [bytů](https://cs.wikipedia.org/wiki/Bajt)) na posloupnost pevné délky *n* bitů. Z definice plyne existence [*kolizí*](https://cs.wikipedia.org/wiki/Kolize_(informatika)), to znamená dvojic vstupních dat (*x*,*y*), *x* ≠ *y*, takových, že *h*(*x*) = *h*(*y*), tj. dvojice různých vstupních dat může mít stejný otisk. Kolize jsou nežádoucí, ale v principu se jim nelze vyhnout, protože počet možných různých vstupních zpráv je větší než počet možných různých otisků. Vhodnou volbou funkce lze snížit pravděpodobnost, že nastane kolize pro podobná data. “ – [Wikipedie](https://cs.wikipedia.org/wiki/Ha%C5%A1ovac%C3%AD_funkce)

## Používané Hashe

### MD5

### SHA1

### SHA256

### SHA512

### RIPEMD160

### CRC32

CRC32 je hashovací algorithmus určený k dělání checksum souborů a dat. Skoro vůbec se nepoužívá pro ukládání hesel. Chtěl bych tento algoritmus použít, protože je používán v několika velice důležitých protokolech: „ISO 3309 (HDLC), ANSI X3.66 (ADCCP), FIPS PUB 71, FED-STD-1003, ITU-T V.42, ISO/IEC/IEEE 802-3 (Ethernet), SATA, MPEG-2, PKZIP, Gzip, Bzip2, POSIX cksum, PNG, ZMODEM atd.“ – [Wikipedia](https://en.wikipedia.org/wiki/Cyclic_redundancy_check)

# Použité programy a technologie

Každý správný projekt potřebuje použití několika programů či stránek pro zlepšení práce na projektu. Každý použitý program či stránka jsou popsané k čemu slouží, teorii, popřípadě vysvětlení jak fungují a proč jsem je použil

## C#

## Unit-testy

Unit-testy jsou věc, kterou jsem si sám přidal do projektu z jednoho hlavního důvodu. Kontrola celého programu jestli funguje a je mnohem jednodušší dělat debugging programu. Neměl jsem v plánu dělat „Test Driven Developement“ a taky jsem podle toho nefungoval. Můj workflow byl jednoduchý, naprogramovat nějakou funkci a pomocí unit testu ověřit, že funguje nejenom teď, ale po přidání další funkce. Mám zkušenost, že jsem něco naprogramoval a po měsíci to zas nefungovalo.

Díky unit testům, které jsem si na začátku udělal, jsem zjistil, že mi kvůli jedné malé chybce ve switchi nefungoval MD5 algorithmus.

## Visual Studio 2022

Visual studio a .NET Framework pro Windows jsou hlavní aplikace, které jsem se rozhodnul používat. Hlavní důvod je jednoduchost k přidání dalších knihoven pomocí NuGet (díky „System.IO.Hashing“ mám přístup k hashovacímu algorithmu CRC32) a hlavní hashovací programy (všechny kromě CRC32, které používám v programu) vychází z knihovny „System.Security.Cryptography“, kde mám jednoduchý přístup k metodám.

## Visual Code

## Git/Github/Github Desktop

## Freelo.io

## Word

## Photoshop

## Online hashers

Online hashers jsou funkční algoritmy pro různé hashe, většinou ve formě stránky, které jsem primárně používal k porovnávání vygenerovaných hashů, abych věděl, jestli mi program funguje. Existuje spousty programů a stránek, já jsem většinu času používal <https://www.browserling.com/tools/all-hashes>, protože je tam většina hashovacích algoritmů, které používám v programu (MD5, všechny SHA).

# Praktická část

Pod tuto kapitolu patří všechno moje snažení se o správný chod programu.

## Hasher

Hasher je nejzákladnější kód celého programu. Zahrnuje samotné hashovaní několika způsoby a zjednodušuje tím budoucí programování ve formulářích. Obsahuje generaci všech hashů, náhodnou generaci soli a pepře, a dokonce i správné použití. Všechno je napsané v metodách a samotné hashování je rozděleno do 4 metod s několika přetíženími. Hash() akorát hashuje text pomocí určitého algoritmu. Ten je vybírán pomocí enum v každé z metod. Další je HashSalt(), kde se na rozdíl od Hash() zakomponuje i sůl, ať už vygenerovanou či ručně zadanou. HashPepper() dělá podobnou věc jako HashSalt(), ovšem pepř nevypisuje. Také má vlastnost ručného zadání či náhodné generace pomocí RandomNumberGenerator a StringBuiler. Obojí zakomponované v .NET Frameworku.

## Unit-Testy

## File Manager

## Settings

# Fungování programu

Závěr

Vytvořená šablona maturitních prací obsahuje formální požadavky maturitních prací na SPŠT Třebíč. Jedná se zejména o upravené styly v dokumentu, podrobný popis jednotlivých částí maturitní práce a jejího obsahu, snadno editovatelné záhlaví a zápatí s automatickým číslováním stránek a propojení stylů se seznamy a obsahem.

Seznam použitých zdrojů

1. Citace.com. *Výklad normy ČSN ISO 690*. Online. Brno: Citace.com, 2023. Dostupné z: <https://www.citace.com/Vyklad-CSN-ISO-690-2022.pdf>. [cit. 2023-10-04].
2. Didacticus. *Normostrana: kolik má znaků, jak zjistit jejich počet a další důležité informace.* Online. Praha: Didacticus, c2011-2020. Dostupné z: <https://didacticus.cz/normostrana>. [cit. 2023-09-11].
3. Ústav pro jazyk český AV ČR. *Tečka.* Online. Internetová jazyková příručka. Praha: Ústav pro jazyk český AV ČR, 2008-2023. Dostupné z: <https://prirucka.ujc.cas.cz/>. [cit. 2023-10-04].

Seznam použitých symbolů a zkratek

|  |  |  |
| --- | --- | --- |
| Symbol | Veličina | Jednotka |
| S | Entropie |  |
| Q | Teplo |  |
| T | Termodynamická teplota |  |
| t | Čas |  |
|  |  |  |

Seznam obrázků

[Obr. 2.1 Obsah 10](#_Toc147493921)

[Obr. 2.2 Příklad umístění legendy obrázku 12](#_Toc147493922)

Seznam tabulek

[Tab. 2.1 Legenda k tabulce 12](#_Toc147493615)

Seznam příloh

Prázdná šablona maturitní práce