k verzi 1.2.

Dokumentace SmichovPass

2023

Vypracoval: Kryštof Bruthans, 1.C

Obsah

Požadavky

Pro správnou funkčnost programu jsou vyžadovány následující specifikace:

OS: Windows 10/11

Vstupní zařízení: myš, klávesnice

Výstupní zařízení: monitor

Úložiště: 21 kB volného místa

Ostatní požadavky na hardware jsou shodné s minimálními požadavky na spuštění operačního systému Windows.

Pozn.: Toto jsou minimální prověřené požadavky, tudíž je možné spustit program např. i na starších verzích Windows, avšak není zaručené, že funkce programu budou pracovat správně.

Uživatelská dokumentace

Děkujeme za stažení a používání programu SmichovPass!

Vytváření hesel

1. Vyplňte pole „Název stránky“ – to slouží pro Vaší orientaci, abyste věděli, které přihlašovací údaje náleží jaké aplikaci/službě.
2. Vyplňte pole „Uživatelské pole“ – to slouží k identifikaci vašeho přihlašovacího jména k aplikaci/službě.
3. Vyplňte pole „Heslo“ – to je samotné heslo, které se společně s předchozími daty uloží do zabezpečeného, zašifrovaného formátu.
4. Klikněte na tlačítko „Přidat heslo“.

POZOR

Ani jedno z polí NESMÍ obsahovat čárku (,) ani středník (;). Program nebude jinak fungovat správně. Toto platí **i pro upravování** hesel.

Úprava hesel

Pro úpravu jakéhokoli hesla v datapoli stačí na požadovaný prvek kliknout dvakrát. To vám umožní upravovat jeho hodnotu.

POZOR

Ani jedno z polí NESMÍ obsahovat čárku (,) ani středník (;). Program nebude jinak fungovat správně. Toto platí **i pro vytváření** hesel.

Mazání hesel

1. Zvolte řádek (záznam), který chcete smazat.
2. Stiskněte tlačítko „Odebrat zvolené heslo“.

Kopírovat heslo

Pro komfort uživatelů je zavedeno i kopírování hesel jedním kliknutím, aby při uložení složitého hesla bylo jeho použití co nejjednodušší a nejpohodlnější.

1. Zvolte řádek s heslem, které chcete zkopírovat.
2. Klikněte na tlačítko „Kopírovat zvolené heslo do schránky“.

Ukládání hesel do souboru

1. ZVOLTE SI KLÍČ – tento krok je velmi důležitý, protože při nezadání klíče **není možné** soubor znovu otevřít. Forma klíče je pouze na Vás (jakékoli znaky; písmena, číslice, čárky, tečky…), avšak **nedoporučuje se do hesla dávat mezery**. Tento klíč bude sloužit jako vaše ultimátní „heslo“ ke **všem heslům v souboru.**
2. Klikněte na „Uložit soubor“.
3. Ujistěte se o správnosti zvoleného klíče a na upozornění odpovězte „ano“.
4. Zvolte si umístění souboru.

Načítání hesel ze souboru

1. ZADEJTE KLÍČ – ten jste si zvolili při vytváření souboru s hesly, před rozšifrováním se ujistěte o jeho správnosti.
2. Stikněte tlačítko „Načíst soubor“.
3. Ujistěte se o správnosti klíče a odpovězte na upozornění „ano“.
4. Zvolte soubor s hesly, který chcete načíst.

Architektura a design programu

Základní informace

Každý prvek programu je prvkem ze sady rozhraní Windows Forms. Užívané fonty v programu jsou dva: Bahnschrift a MS Sans Serif.

Každý prvek, který provádí funkci jinou než estetickou (tlačítka, textová pole, datapole) mají svůj kód definovaný v *App.Designer.cs*. Na metodu provedenou po kliknutí či jakékoli jiné interakci s ovládacím prvkem je odkázáno v designeru.

Tyto metody lze nalézt v *App.cs*. Každá taková metoda vždy nevrací hodnotu (její návratový typ – void).

Ukládání záznamů

Každý záznam o stránce, jménu a heslu je uložený v objektu *record* (z angl. záznam). Ten obsahuje veřejné proměnné *Stranka, Jmeno, Heslo* typu string (řetězec). Tyto záznamy jsou uložené v seznamu *database* typu *List<record>*, ty jsou poté v metode *LoadDataTable* převedeny na formát *DataTable*, který je nahrán do objektu *PasswordViewer* typu *DataGridView*.

Pozn. toto je z důvodu mnohých problémů při přímém nahrávání database do zobrazovače datapole DataGridView, který lépe pracuje se zdrojem typu DataTable.

Před/po kryptografii je potřeba převést všechny záznamy do jednoho dlouhého řetězce a naopak. To je prováděno přímo v metodách *LoadButton_Click* a *CreateButton_Click*. Každá proměnná záznamu je od sebe oddělena čárkou, záznamy jsou oddělené středníkem.

Generovaný řetězec je dále v dokumentu označován jako „surová data záznamů“.

Kryptografie

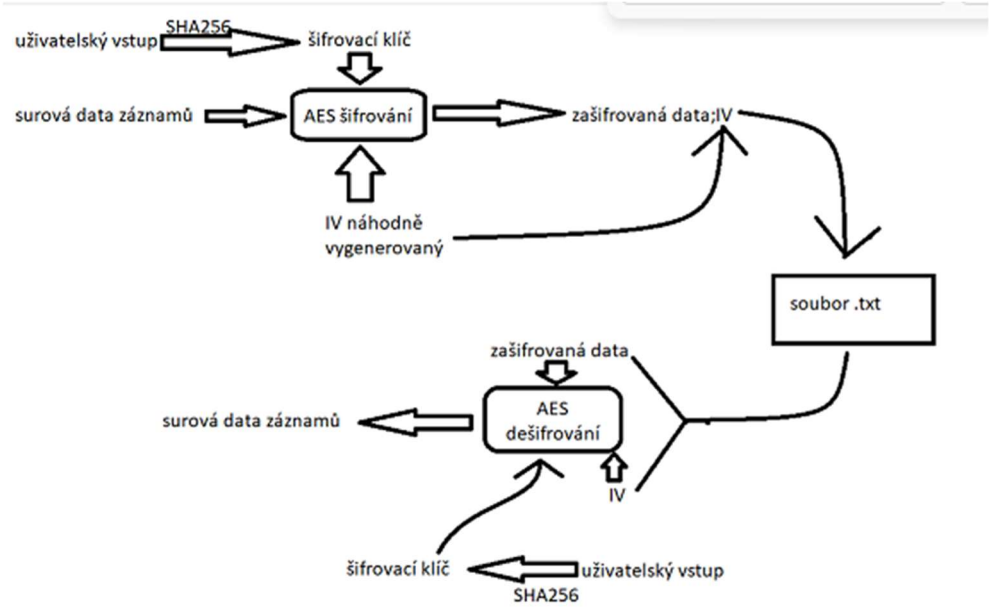
SmichovPass používá pro šifrování standard AES (Advanced Encryption Standard). Ještě před samotným je klíč zvolený uživatelem na hash pomocí algoritmu SHA256. Tento hash je používán jako opravdový klíč používaný při samotném šifrování (tento krok zajiš´tuje vyšší bezpečnost šifrovaného obsahu).

Před šifrováním je ještě vygenerován inicializační vektor (IV), který slouží jako bezpečnostní prvek (v praxi zajišťuje, že dvě data zašifrovaná stejným klíčem nejsou stejná kvůli náhodně vygenerovanému IV; tento IV je potřeba k následnému dešifrování, takže je obsažen v souboru .txt společně se zašifrovanými daty).

Poté je pomocí knihovny *System.Security.Cryptography* použit AES pro zašifrování surových dat záznamů. Výstupní zašifrovaná data jsou jeden řetězec.

V konečném kroku je k řetězci se zašifrovanými daty přidán středník a za ním IV použitý k zašifrování těchto dat. Tento výsledný řetězec je zapsán jako text do souboru .txt.

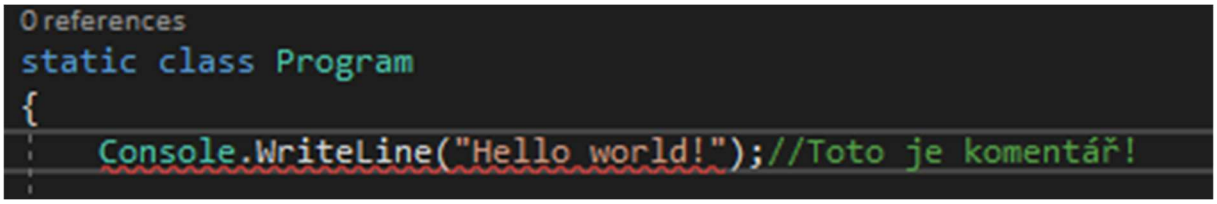
Dešifrování funguje obdobným způsobem, nejdříve je zahashován pomocí SHA256 klíč uživatele, poté od sebe oddělen IV a zašifrovaná data a pomocí algoritmu v knihovně *System.Security.Cryptography* jsou data dešifrována na surová data, která jsou nahrána do *database* a metoda *LoadDataTable* data načte do datapole.



Technická dokumentace

Technická dokumentace se nachází přímo v kódu v podobě komentářů. Komentáře jsou vždy na jedné lince a začínají znaky „//“.

Příklad:



Ovládací prvky programu a jejich metody

Tlačítko *LoadButton* po stisknutí zjistí jakékoli úpravy uživatelem do *dataTable* a zobrazí okno pro potvrzení klíče, poté inicializuje okno pro výběr souboru, ze kterého má číst. Po vybrání provede dešifrování pomocí AES a klíče zadaného uživatelem (*více v kapitole **Kryptografie***), následně převede formát surových dat do záznamů typu *record* a ty nahraje do *database*.

Tlačitíko *CreateButton* po stisknutí zjistí jakékoli úpravy uživatelem do *dataTable* a zobrazí okno pro potvrzení klíče, poté inicializuje okno pro určení, kam má výsledný soubor zapsat. Nejdříve převede všechny záznamy *database* do surových dat, která zašifruje pomocí AES a klíče zadaného uživatelem (*více v kapitole **Kryptografie***). Výsledný řetězec zapíše do zvoleného souboru.

Tlačítko *AddRecord* po stisnutí přečte ze *StrankaBox, JmenoBox* a *HesloBox* text ve formě řetězců, incializuje nový záznam, který zaplní těmito řetězci a ty přidá do *database* a zavolá metodu *LoadDataTable*.

Tlačítko *CopyButton* po stisknutí zkopíruje heslo zvoleného záznamu v datapoli do schránky.

Tlačítko *RemoveRecord* po stisknutí smaže zvolený záznam v datapoli z *database* a zavolá metodu *LoadDataTable*.

Metoda *LoadDataTable* inicializuje dataTable nahraje každý záznam z *database* do něj a následně aktualizuje zobrazovač datapole(*PasswordViewer, typ DataGridView*).

Objekt *PasswordViewer* je typu *DataGridView* a zobrazuje data z *dataTable* uživateli.

Budoucí funkce, rozvoj

Export samostatných záznamů

Export samotných záznamů a poté jejich šifrování by mohlo být užitečné z důvodu:

1. Dvojité zálohování jen určitého hesla/seznamu hesel.
2. Sdílení hesel – stačí jen vědět společné heslo a poté si jakýmkoli prostřednictvím bezpečně přeposílat soubory s hesly (sociální sítě, e-mail).
3. Aby se dosáhlo synchronizace při sdílení těchto hesel, bylo by potřeba vytvořit unikátní hash a datum a ty připojit ke každému záznamu. Hash pro identifikaci hesla a datum pro udržování nejaktuálnější formy přihlašovacích údajů.

Propojení s cloudem

Propojení s cloudem přímo v aplikaci by znamenalo zálohování hesel přímo v reálném čase. V aplikaci si zvolíte službu (Google disk, OneDrive), místo uložení a je hotovo. Poté pomocí desktopové, mobilní nebo webové aplikaci stačí zadat přihlašovací údaje (jméno, klíč a popř. IP adresu vlastního úložiště) a synchronizace proběhne automaticky.

Pozn. Toto je už nyní možné při uložení souboru s hesly přímo na OneDrive nebo nahrání souboru na Google disk, ale propojením je myšlena kompletní integrace SmichovPass do těchto služeb a jejich automatická domluva.

Využívání více algoritmů pro šifrování

Přidání možnosti do aplikace zvolit si i jiné šifrovací algoritmy dle potřeby (např. Triple DES, RSA, Blowfish; v budoucnosti kvantových počítačů poté BB84 nebo E91). To znamená, že pro určitá hesla stačí velmi rychlý standard „Blowfish“, zatímco pro hesla citlivější by se využil standard „AES“ a pro velmi efektivní sdílení souborů asymetrický standard „RSA“.

Zavedení vícefaktorového ověření

Zatím program využívá jen jeden faktor – „něco, co víte“ (klíč). Další dva faktory – „něco, co jsme“ (biometrika) a „něco, co máme“ (např. flash disk s autentizačním tokenem) mohou být implementovány do programu pro další zvýšení bezpečnosti a prevenci nežádaného přístupu.