Úvod do počítačovej bezpečnosti

Zadanie 2 - Implementácia aplikácie na šifrovanie súborov

Cieľom zadanie bolo vytvoriť aplikáciu, ktorá umožní používateľovi šifrovať a dešifrovať súbory symetrickou šifrou s náhodne generovaným kľúčom.

A. Špecifikácia

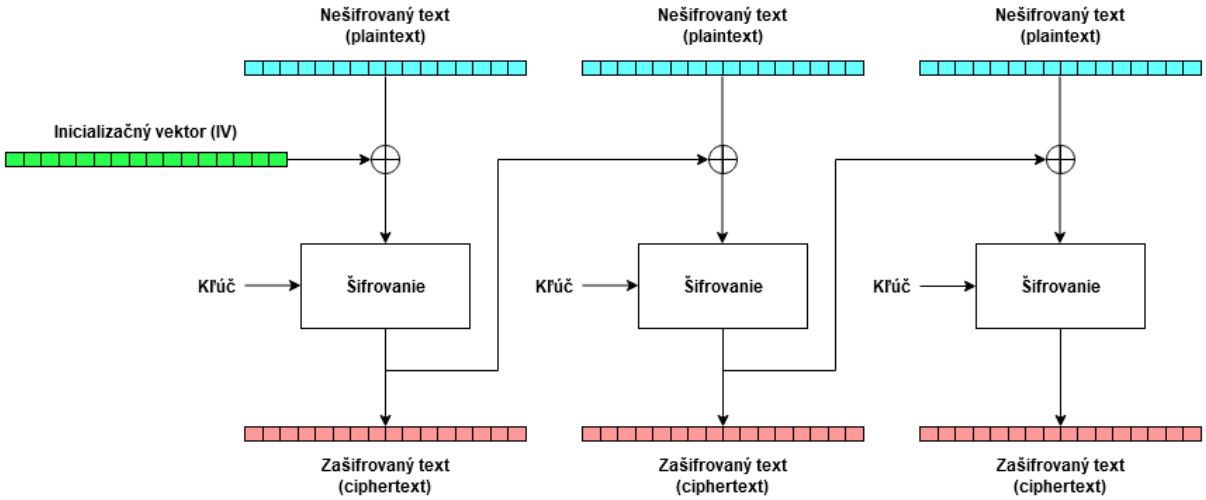
- Programovací jazyk: Java
- Knižnice: `java.security` (security framework)
  `javax.crypto` (kryptografické operácie)
- Zdroje: https://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html

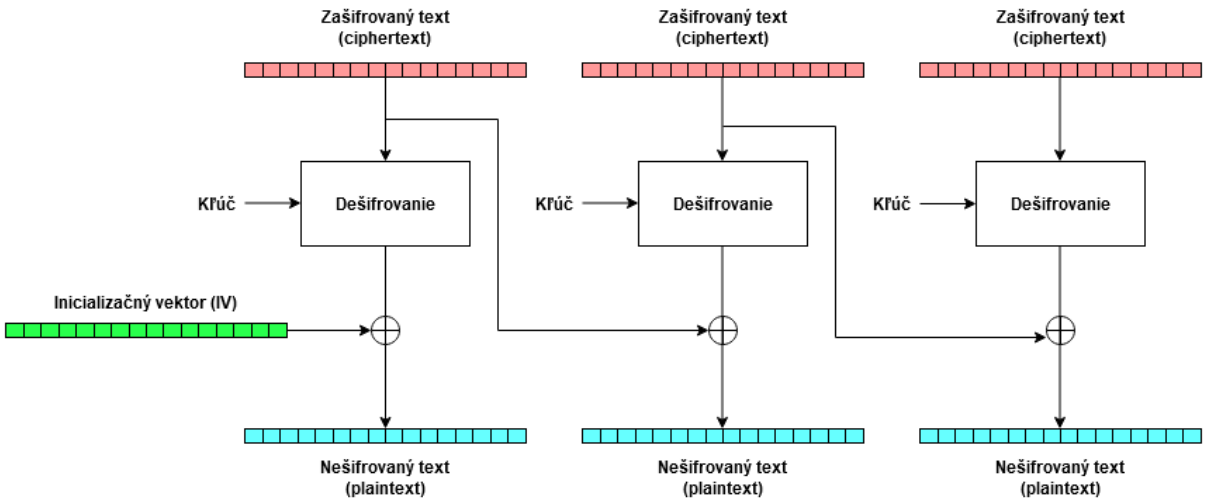
B. Použitá šifra

Aplikácia využíva na šifrovanie šifrovací algoritmus AES v CBC móde s kľúčom veľkosti 256 bitov. Advanced Encryption Standard (AES) je 128-bitová bloková šifra (každý blok pozostáva zo 16 bajtov) podporujúca kľúče s 128, 192 a 256 bitmi. Je bezpečnejšia ako predchádzajúci štandard DES (Data Encryption Standard), ktorý používa kľúče s dĺžkou len 64 bitov (z toho 8 je kontrolných a 56 efektívnych) alebo 3DES (Triple-DES) upravená verzia DES šifry používajúca šifrovací algoritmu DES trikrát na každý dátový blok (veľkosti kľúčov: 168, 112 or 56 bitov). Veľkosti kľúčov v prípade DES algoritmu a v niektorý prípadoch aj pri 3DES algoritme sa už nepovažujú za primerané vzhľadom na moderné kryptoanalytické techniky a výkon počítačov.

Symetrické šifry používajú na šifrovanie a dešifrovanie rovnaký kľúč, ktorý musí poznať odosielateľ aj príjemca.

Bloková šifra AES v móde CBC zámena, že pred zašifrovaním sa odpovedajúce bloky otvoreného textu XORujú s predchádzajúcim blokom zašifrovaného textu. To znamená, že jednotlivé bloky sú na sebe závislé. Aby sme dešifrovali konkrétny blok, musíme dešifrovať aj všetky predošlé. Vzhľadom na to, že prvý blok nemá žiadneho predchodcu nie je k dispozícii blok, s ktorým by ho bolo možné XORovať. Z tohto dôvodu sa jeden blok vygeneruje náhodne a pridá sa k zašifrovaným dátam ako nultý blok, konkrétne nazývaný inicializačný vektor (IV). Tento vektor sa používa na zašifrovanie prvého bloku a má veľkosť pravé jedného bloku AES šifry (16 bajtov resp. 128 bitov).





C. Rýchlosť šifrovania

Rýchlosť šifrovania bola testovaná na súbore s veľkosťou 1GB. Aplikácia dokázala súbor zašifrovať z 5. pokusov priemerne za 15,4s.

1. meranie	2. meranie	3. meranie	4. meranie	5. meranie	Priemer
14s	15s	16s	14s	18s	15,4s

D. Spustenie aplikácie

Aby bolo možné aplikáciu spustiť je potrebné mať na zariadení nainštalovanú JAVU. Aplikácia nemá vlastné grafické používateľské rozhranie (GUI), preto je pre jej spustenie potrebné otvoriť konzolu, prejsť do adresára kde sa nachádza zadanie2.jar súbor a príkazom **`java -jar zadanie2.jar`** (Windows, Mac OS, Linux) aplikáciu spustiť.

E. Používateľská príručka

1. Návod na použitie aplikácie

```
D:\Zadanie2>java -jar zadanie2.jar
Aplikácia bola úspešne spustená...

Zadajte názov resp. absolútnu cestu k súboru (Nešifrovaný súbor):
```

Po úspešnom spustení aplikácie je užívateľ vyzvaný zadať názov súboru, ktorý sa má zašifrovať. V prípade, že sa súbor nachádza v rovnakom adresári stačí zadať názov inak je potrebné uviesť absolútnu cestu k súboru aby ho aplikácia vedela nájsť.

```
Zadajte názov resp. absolútnu cestu k súboru (Nešifrovaný súbor):
Súbor sa nepodarilo nájsť, skúste zadať názov súboru ešte raz:
```

V prípade, že súbor nebude možné nájsť, aplikácia Vám umožní názov zadať opätovne.

```
Zadajte názov resp. absolútnu cestu k súboru (Nešifrovaný súbor): test.txt

Prebieha šifrovanie...
Šifrovanie bolo dokončené (7 ms)...
Šifrovaný súbor 'D:\Zadanie2\test.encrypted' bol vytvorený.
Kľúč 'D:\Zadanie2\test.key' bol vytvorený.

Pre dešifrovanie stlačte ENTER...
```

Ak aplikácia súbor nájde a bude ho vedieť otvoriť automaticky začne so šifrovaním. Po skončení šifrovania budú vytvorené dva súbory: **<názov>.encrypted**, ktorý reprezentuje šifrovaný súbor a **<názov>.key** kde je uložený šifrovací kľúč. Pre pokračovanie na proces dešifrovanie je potrebné stlačiť ENTER.

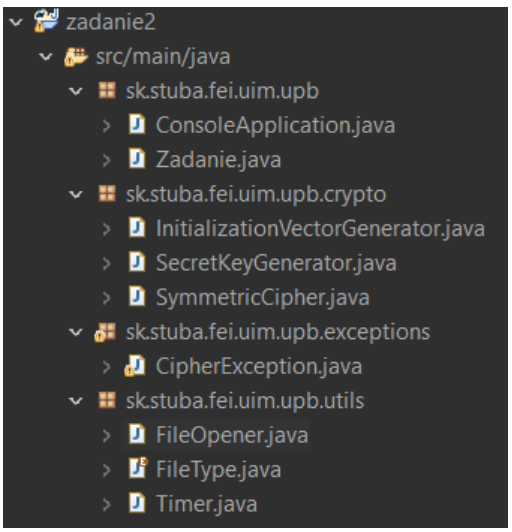
```
Pre dešifrovanie stlačte ENTER...

Zadajte názov resp. absolútnu cestu k súboru (Šifrovaný súbor): test.encrypted
Zadajte názov resp. absolútnu cestu k súboru (Kľúč): test.key

Prebieha dešifrovanie...
Dešifrovanie bolo dokončené (170100 ns)...
Dešifrovaný súbor 'D:\Zadanie2\test.decrypted' bol vytvorený.
```

Pre dešifrovanie je potrebné zadať názov respektíve cestu k zašifrovanému súboru a k súboru kde je uložený šifrovací kľúč (**je nutné aby mal zašifrovaný súbor príponu .encrypted a súbor s kľúčom príponu .key** inak aplikácia bude od Vás súbory vyžadovať opätovne). Po skončení dešifrovania bude vytvorený súbor **<názov>.decrypted**. Tento súbor by mal byť zhodný so vstupným súborom, ktorý sme šifrovali. Pre skontrolovanie zhody stačí len manuálne prepísať príponu súboru na takú akú mal pôvodný súbor a prezrieť obsah.

2. Návod na importovanie projektu do IDE

Aplikácia bola vyvíjaná v Eclipse IDE pričom bola vytvorená ako Gradle projekt, pre prípad, že by ste chceli otvárať kód aplikácie v IDE odporúčam aplikáciu importovať ako „existing gradle project“ aby náhodou nevznikali nejaké komplikácie. V prípade, že by ste si chceli nanovo vygenerovať .jar stačí použiť gradle task „`jar`“, ktorý celý kód skompiluje vybuilduje a zabalí do `zadanie2.jar`, takto vygenerovaný súbor je potom možné nájsť v `zadanie2\build\libs` adresári a spustiť ho.

3. Štruktúra projektu



4. Stručný popis hlavných tried:

Trieda	Funkcionalita
ConsoleApplication	Hlavná trieda aplikácie
FileOpener	Vyhľadáva a otvára súbory
SymmetricCipher	Šifrovanie a dešifrovanie súborov
SecretKeyGenerator	Generátor náhodného šifrovacieho kľúča
InitializationVectorGenerator	Generátor inicializačného vektora
Timer	Časovač šifrovania/dešifrovanie

Šifra (SymmetricCipher)

- Na šifrovanie bola použitá trieda `javax.crypto.Cipher`, ktorá poskytuje funkcionalitu na šifrovanie a dešifrovanie konkrétnych kryptografických šifier
- Šifrovací algoritmus: AES (Advanced Encryption Standard)
- Mód šifrovacieho algoritmus: CBC (Cipher Block Chaining)
- Výplň šifrovacieho algoritmu: PKCS5Padding

Kľuč (SecretKeyGenerator)

- Na generovanie šifrovacieho kľúča bola využitá trieda `javax.crypto.KeyGenerator`, ktorá poskytuje funkcionalitu na generovanie tajných symetrických kľúčov.
- Na to aby bol kľuč silný, bezpečný a naozaj náhodný bola využitá trieda `java.security.SecureRandom`, ktorá poskytuje kryptograficky silný generátor náhodných čísel.
- Algoritmus generovaného kľúča je AES s veľkosťou 256 bitov

Inicializačný vektor (InitializationVectorGenerator)

- Na generovanie inicializačného vektora bola využitá trieda `javax.crypto.spec.IvParameterSpec`, ktorá poskytuje funkcionalitu na špecifikovanie inicializačných vektorov.
- Vzhľadom na predvolenú veľkosť bloku 128 bitov pre AES algoritmus potrebujeme 16 bajtový inicializačný vektor

F. Bezpečnostné odporúčania

- Nezverejňovať kľúč tretím stranám a neoprávneným osobám
- Nezanechávať kľúč na rôznych úložiskách (HDD, USB,...) bez zabezpečenia
- Pri šifrovaní je doležíte klásť dôraz na bezpečnostné odporúčania