

Pflichtenheft zu Caesar Verschlüsselung

Rosario-Francesco Polito
Tobias Dolfen

4. Februar 2019

Inhaltsverzeichnis

1	Zielbestimmung	3
1.1	Musskriterien	3
1.2	Kannkriterien	3
1.3	Abgrenzungskriterien	3
2	Einsatz	4
2.1	Anwendungsbereiche	4
2.2	Zielgruppen	4
2.3	Betriebsbedingungen	4
3	Umgebung	5
3.1	Software	5
3.2	Hardware	5
4	Funktionalität	6
5	Benutzungsoberfläche	7

1 Zielbestimmung

1.1 Musskriterien

Das Programm muss in C# geschrieben sein. Die Hauptfunktionen des Programms bestehen daraus, einen eingetippten Text via Ceasar-Verschlüsselung mit dem vom Benutzer angegebenen Schlüssel zu verschlüsseln oder zu entschlüsseln. Die Letzte Funktion besteht darin einen verschlüsselten Text ohne Angabe eines Schlüssels zu entschlüsseln.

1.2 Kannkriterien

Es kann die Funktion eingebaut werden den zu verschlüsselnden Text aus einer Textdatei (.txt) einzubinden um auf einfache Art und Weise mehrere Zeilen zu ver- bzw. entschlüsseln.

1.3 Abgrenzungskriterien

Es wird bewusst darauf verzichtet eine grafische Nutzeroberfläche zu implementieren.

2 Einsatz

2.1 Anwendungsbereiche

Die Ceasar-Verschlüsselung wird genutzt um einen Text unleserlich zu machen und somit Fremden den Informationsdiebstahl zu erschweren.

2.2 Zielgruppen

Die Zielgruppe besteht aus jeder Person, welche eine Nachricht verschicken will, ohne dass unbefugte Personen die Möglichkeit haben die übermittelten Informationen mitzulesen.

2.3 Betriebsbedingungen

Betriebsbedingungen sind ein C# fähiger Computer und ein Text, welcher verschlüsselt oder entschlüsselt werden soll. Dieser Text muss - sofern man die „Textdatei verschlüsseln“ oder die „Textdatei entschlüsseln“ Funktionen nutzen möchte als (digitale) Textdatei im .txt Format vorliegen.

3 Umgebung

3.1 Software

Die Softwarevoraussetzung besteht aus dem .Net Framework, welche gebraucht wird um C#-Programme ausführen zu können.

Zum entwickeln wird wahlweise JetBrains Rider¹ oder VisualStudio ² / VisualStudio Code³ benutzt. Zur Versionskontrolle wird Git genutzt. Die Daten werden dabei in einem Repository auf den Servern von GitHub(<https://github.com>) gespeichert.

3.2 Hardware

Um .Net Programme ausführen zu können, wird ein Monitor mit einer Auflösung von mindestens 800x600 Pixeln, 1GB RAM und 1.5GB Festplattenspeicher benötigt.

¹<https://www.jetbrains.com/rider/>

²<https://visualstudio.microsoft.com/de/>

³<https://code.visualstudio.com>

4 Funktionalität

Encrypt-Funktion: Die Encrypt-Funktion fragt den Benutzer nach einem (klar) Text/Satz, welcher von ihm in die Konsole eingetippt wird. Anschließend fragt das Programm nach einem Schlüssel, welcher ebenfalls in die Konsole eingetippt wird. Das Programm gibt anschließend den Satz/Text auf der Konsole aus.

Decrypt-Funktion: Die Decrypt-Funktion fragt den Benutzer nach einem (verschlüsselten) Text/Satz, welcher von ihm in die Konsole eingetippt wird. Anschließend fragt das Programm nach einem Schlüssel, welcher ebenfalls in die Konsole eingetippt wird. Das Programm gibt anschließend den (klar) Text/Satz auf der Konsole aus.

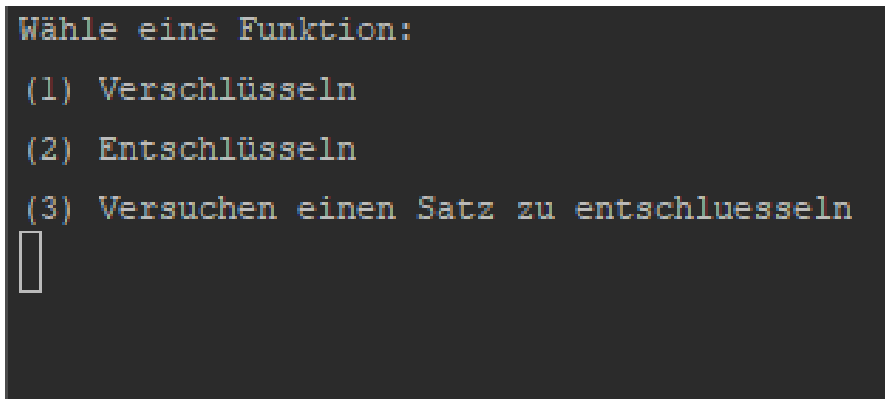
Bruteforce-Funktion: Die Bruteforce-Funktion fragt den Benutzer nach einem (verschlüsselten) Text/Satz, welcher von ihm in die Konsole eingetippt wird. Anschließend gibt das Programm den (vermeintlich entschlüsselten) Satz genau so oft aus, wie Schlüssel vorhanden sind. Anders ist nicht zu gewährleisten, dass mindestens eine der angezeigten Möglichkeiten die Richtige ist.

Textdatei-Verschlüsseln-Funktion: Die Textdatei-Verschlüsseln-Funktion fragt den Benutzer nach einer Textdatei, welche entschlüsselt werden soll. Der Benutzer gibt den Pfad zur Textdatei per Konsoleneingabe an. Anschließend wird nach einem Schlüssel gefragt. Das Programm öffnet jetzt die Textdatei, liest Zeichen für Zeichen ein und verschiebt diese um den angegebenen Wert Schlüssel des Schlüssels nach rechts, setzt die Zeichen zusammen und speichert den Text zeilenweise ab.

Textdatei-Entschlüsseln-Funktion: Die Textdatei-Entschlüsseln-Funktion fragt den Benutzer nach einer Textdatei, welche entschlüsselt werden soll. Der Benutzer gibt den Pfad zur Textdatei per Konsoleneingabe an. Anschließend wird nach einem Schlüssel gefragt. Das Programm öffnet jetzt die Textdatei, liest Zeichen für Zeichen ein und verschiebt diese um den angegebenen Schlüssel nach links, setzt die Zeichen zusammen und speichert den Text zeilenweise ab.

5 Benutzungsoberfläche

Die Benutzeroberfläche besteht aus der Konsole, welche mit Hilfe eines Dialogfensters den Benutzer durch das Programm leitet.

A screenshot of a terminal window with a dark background. The text is displayed in a monospaced font. The prompt 'Wähle eine Funktion:' is followed by three numbered options: '(1) Verschlüsseln', '(2) Entschlüsseln', and '(3) Versuchen einen Satz zu entschlüsseln'. A white cursor is positioned at the start of a new line below the third option.

```
Wähle eine Funktion:  
(1) Verschlüsseln  
(2) Entschlüsseln  
(3) Versuchen einen Satz zu entschlüsseln  
█
```

Abbildung 5.1: Benutzeroberfläche