PROJEKTARBEIT «PERFEKTE KRYPTOGRAPHIE»

Programmierung 2

15. MAI 2024

Inhalt

Autoren

Joel Häfeli
WIVZ 1.51
joel.haefeli@students.fhnw.ch

Emil Blank
WIVZ 1.51
emil.blank@students.fhnw.ch

Einleitung

In unserem Projekt befassen wir uns mit der Anwendung von One-Time Pads, einem der sichersten Kryptographieverfahren, dass seine Ursprünge bis in das Jahr 1882 zurückverfolgen kann. Trotz seines Alters bietet das One-Time Pad, wenn es korrekt angewendet wird, auch heute noch eine nicht zu brechende Verschlüsselungssicherheit. Das Ziel dieses Projekts ist die Entwicklung eines einfachen Systems, das dieses historische Verschlüsselungsverfahren nutzt, um Nachrichten sicher zu verschlüsseln und zu entschlüsseln.
Im Rahmen dieses Projekts haben wir ein System implementiert, das eine Datei als One-Time Pad einliest und vom Benutzer eingegebene Textnachrichten entweder verschlüsselt oder entschlüsselt. Dabei dient nicht nur die technische Umsetzung als Kern unseres Projekts, sondern auch die Betrachtung der Effizienz und Sicherheit der kryptographischen Methoden unter den Aspekten der aktuellen technologischen Standards und Bedrohungen.

Anforderungen

ID	Beschreibung
1	Die Anwendung muss eine Benutzeroberfläche bieten, über die Benutzer eine Textnachricht eingeben können.
2	Die Anwendung muss eine Benutzeroberfläche bieten, über die Benutzer diese Textnachricht verschlüsseln oder entschlüsseln können.
3	Die Anwendung muss eine Benutzeroberfläche bieten, über die die verschlüsselte Nachricht angezeigt wird.
4	Die Anwendung muss eine Benutzeroberfläche bieten, über die eine Textdatei als Pad verwendet werden kann.
5	Die Anwendung muss den Verschlüsselungsalgorithmus des One- Time-Pads verwenden.
6	Die Anwendung darf zur Verschlüsselung nur die Zeichen der ASCII-Tabelle verwenden.
7	Die Anwendung muss fähig sein, Leerzeichen bei einer Verschlüsselung zu entfernen.
8	Die Anwendung muss dem Benutzer eine übersichtliche Benutzeroberfläche bieten.

Tabelle 1 Anforderungen der Anwendung

Praktisches Implementationskonzept

Übersicht: Das vorliegende Programm implementiert eine textbasierte Verschlüsselungs- und Entschlüsselungsanwendung unter Verwendung des One-Time-Pad-Verfahrens, eines Kryptographieverfahrens, das für seine Sicherheit bekannt ist, wenn es korrekt verwendet wird. Die Implementierung erfolgt in Java und nutzt die JavaFX-Bibliothek für die Benutzeroberfläche.

Komponentenstruktur: Das Programm besteht aus drei Hauptkomponenten, die dem Model-View-Controller (MVC)-Muster folgen:

(Riesen, 2020)

Model (CipherModel):

Verantwortlichkeit: Verwaltung der Verschlüsselungslogik und der Daten (One-Time-Pad).

Funktionen:

Liest eine Textdatei, die als One-Time-Pad dient ein und konvertiert deren Inhalt in eine Liste von Integer-Werten.

```
//Konstruktor
1 usage  blankito1
public CipherModel(String padFilePath) throws Exception { //Dateipfad als Parameter
```

Abbildung 1 Textdatei als Parameter

Bietet Methoden zur Verschlüsselung und Entschlüsselung von Nachrichten, wobei die aktuelle Position im Pad berücksichtigt wird.

Setzt die aktuelle Position im Pad zurück, um von vorne zu beginnen.

(Riesen, 2020)

View (CipherView):

Verantwortlichkeit: Darstellung der Benutzeroberfläche.

Funktionen:

Stellt Textfelder zur Eingabe und Anzeige von Nachrichten, Buttons für die Verschlüsselungs- und Entschlüsselungsfunktionen sowie einen Button zum Laden des One-Time-Pads bereit.

Zeigt Statusmeldungen an, die über den Erfolg oder Fehler von Operationen informieren.

(Riesen, 2020)

Controller (CipherController):

Verantwortlichkeit: Vermittlung zwischen View und Model.

Funktionen:

Reagiert auf Benutzerinteraktionen wie das Betätigen von Buttons und ruft entsprechend Methoden im Model auf.

Aktualisiert die View basierend auf den Rückmeldungen und Ergebnissen des Models.

(Riesen, 2020)

Implementierungsdetails:

File Handling: Das One-Time-Pad wird aus einer Textdatei geladen, die über einen Dateiauswahldialog ausgewählt wird. Die Datei wird zeilenweise gelesen und in eine Liste von Integer-Werten umgewandelt, die die Grundlage für die Verschlüsselungsoperationen bilden.

Verschlüsselung und Entschlüsselung

Verschlüsselung: Jedes Zeichen der Eingabenachricht wird in einen Integer-Wert umgewandelt, mit einem Wert aus dem One-Time-Pad addiert (Modulo 95) und wieder in ein Zeichen umgewandelt.

Entschlüsselung: Jedes Zeichen der verschlüsselten Nachricht wird in einen Integer-Wert umgewandelt, vom entsprechenden Wert des One-Time-Pads subtrahiert (Modulo 95) und wieder in ein Zeichen umgewandelt.

Fehlerbehandlung: Die Anwendung überprüft, ob das One-Time-Pad ausgeschöpft ist, und wirft eine Ausnahme, wenn weitere Zeichen ohne Zurücksetzen des Pads verarbeitet werden sollen. Fehlermeldungen werden in der Benutzeroberfläche angezeigt.

```
int messageValue = character - 32; //ASCII Bereich 32-126 -> -32 = 0-94
int padValue = oneTimePad.get(currentPosition++) - 32;
char encryptedChar = (char) (((messageValue + padValue) % 95) + 32); //sicherstellen das Ergebnis im druckbaren Bereich bleibt
encrypted.append(encryptedChar); //verschlüsselter char wird an Strinbuilder angehängt
```

Abbildung 2 Beispiel aus der Methode encrypt

Zusammenarbeit zwischen den Komponenten:

Der Controller initiiert das Laden des Pads und steuert die Verschlüsselungs- und Entschlüsselungsoperationen basierend auf Benutzereingaben.

Die View informiert den Controller über Benutzeraktionen und aktualisiert die Darstellung basierend auf den Ergebnissen und Statusupdates vom Controller.

Das Model führt die logischen Operationen aus und informiert den Controller über Erfolg oder Misserfolg von Operationen.

UML- Diagramm:

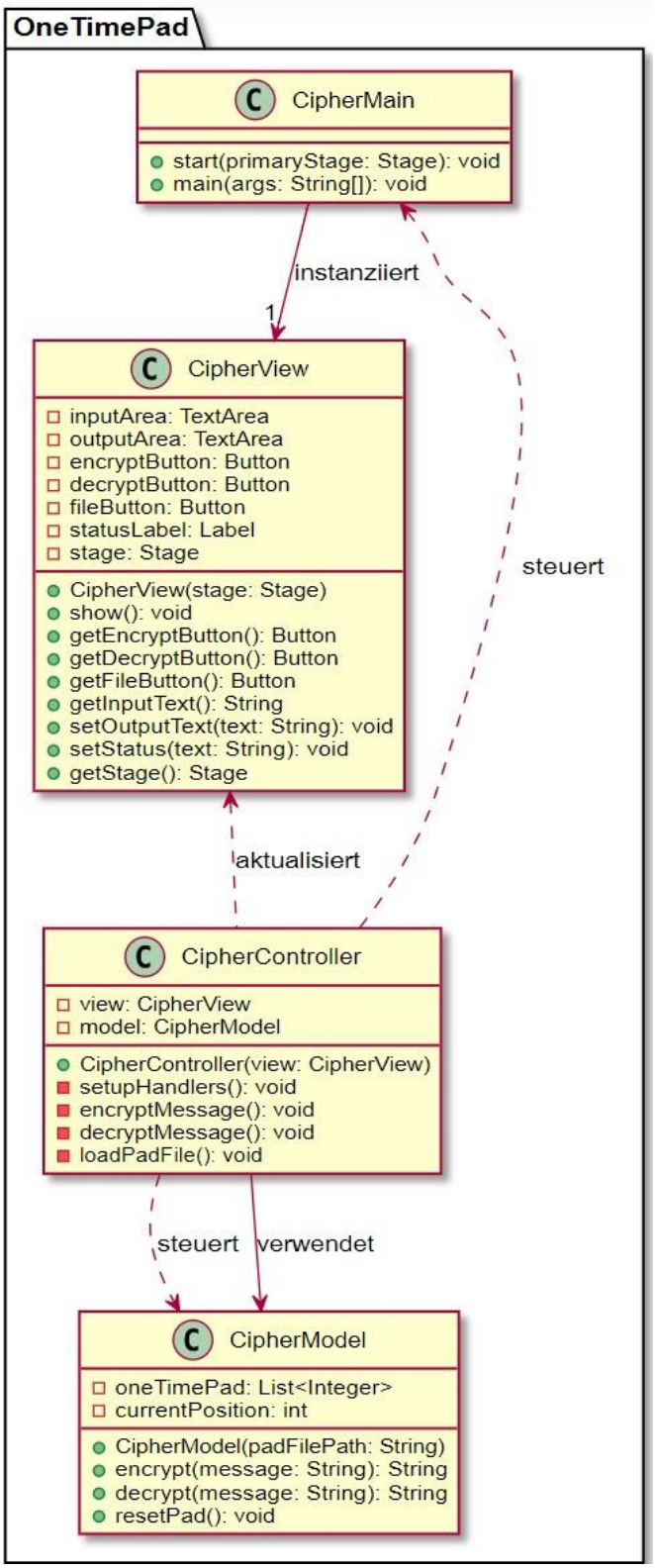

Abbildung 3 UML-Diagramm der Anwendung

Ergebnisse

Die Implementierung des One- Time- Pads in Java unter Verwendung der JavaFX-Bibliothek für die Benutzeroberfläche hat zu einer funktionierenden Anwendung geführt, die in der Lage ist, Nachrichten sicher zu verschlüsseln und zu entschlüsseln. Im folgenden Abschnitt werden die Ergebnisse dieser Implementierung diskutiert.

Funktionalität:

Die Kernfunktionalität dieser Anwendung, also die Verschlüsselung und Entschlüsselung von Nachrichten funktioniert wie erwartet. Die Anwendung verwendet die Methode «encrypt» um damit jedes Zeichen der Nachricht mit einem Zeichen des One- Time- Pads zu kombinieren. Da das Pad zufällig generiert und mindestens so lang wie die Nachricht ist, ist die Sicherheit der Verschlüsselung theoretisch gewährleistet.
Die Methode «decrypt» kehrt dabei die Verschlüsselung um.
Die Anwendung ermöglicht es dem Benutzer, ein neues Pad aus einer Datei zu laden. Dies erhöht die Flexibilität und minimiert die Wiederverwendung von Pads, welche ein Sicherheitsrisiko darstellen würde.

Einschränkungen:

Pad-Länge: Das Pad muss mindestens so lang sein wie die Nachricht. Dies erfordert die Vorbereitung eines ausreichend langen Pads, bevor Nachrichten sicher verschlüsselt werden können.
Benutzerfehler: Benutzer können ein zu kurzes Pad laden, was die Sicherheit der Verschlüsselung gefährden könnte, wenn das Pad für mehrere Nachrichten verwendet wird.

Zusammenfassung

Schlussfolgerungen:

Die Projektarbeit zur Implementierung einer One- Time- Pad Verschlüsselungsanwendung hat uns gezeigt, dass die Prinzipien der Kryptografie in einer praktischen, Benutzerfreundlichen Anwendung umgesetzt werden können. Die Entwicklung und das Testing der Anwendung zeigen, dass mit einem korrekt implementierten One- Time- Pad Verfahren eine sehr hohe Sicherheitsstufe erreicht werden kann, solange das Pad zufällig bleibt und nur einmal verwendet wird.

Weiterführende Untersuchungen:

Automatisierte Pad-Generierung:
Es wäre Interessant, automatisierte Methoden zu untersuchen welche Pads generieren, die noch sicherer und zufälliger sind und auch lange genug, um sie praktisch einsetzbar zu machen.

Multi-Faktor-Verschlüsselung:
Kombinieren des One- Time- Pad Verfahrens mit anderen Verschlüsselungsmethoden, um eine noch höhere Sicherheit zu erreichen.

Quellen und Literatur

Tabellen:

Abbildungen:

Literatur:

Riesen, K. (2020). *Java in 14 Wochen: Ein Lehrbuch für Studierende der Wirtschaftsinformatik*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-30313-6