TECHNISCHE UNIVERSITÄT
CHEMNITZ

Professur
Betriebssysteme

Algorithmen und Programmierung

Komplexe Caesar-Verschlüsselung (Trainingsaufgabe)

Die Caesar-Chiffre gilt als eines der einfachsten aber auch unsichersten Verschlüsselungsverfahren.[1] Zur Verschlüsselung werden Geheimtextbuchstaben aus den Klartextbuchstaben durch eine zyklische Verschiebung im Alphabet generiert.

Um diese Chiffre etwas komplexer zu gestalten, können aus dem erhaltenen Schlüssel k mod 26 auch weitere Schlüssel abgeleitet werden, um die Verschlüsselungsfunktion komplexer gestalten zu können. Ein Beispiel dafür ist die folgende Überführung:

$$\begin{aligned} k_1 &= 3k + 1 \mod 26 \\ k_2 &= k \mod 26 \\ k_3 &= 2k - 5 \mod 26 \end{aligned}$$

Mittels dieser zusätzlichen Schlüssel ist es nun möglich, einzelne Zeichen mehrfach im Geheimtext abzubilden:

$$\begin{aligned} S_k(M) &= S_{k,1}(M) \circ S_{k,2}(M) \circ S_{k,3}(M) \\ \forall i : S_{k,i} &= M + k_i \mod 26 \end{aligned}$$

Wobei S hier für den jeweiligen Geheimtextbuchstaben, M der Klartextbuchstabe und $\circ$ der Konkatenationsoperator ist. So wird aus der Zeichenkette `AB C` der Geheimtext `HCZIDA JEB`.

Aufgabe:

Schreiben Sie eine Quelltextdatei `complexcesar.c`, welche die Funktion
`char* complexCesar(int key, char* input);` implementieren soll. Diese Funktion nimmt den Schlüssel sowie einen C-String mit Zeichen zwischen A und Z als Parameter an und soll diesen mittels der hier definierten Caesar-Chiffre mit Schlüssel $k =$ `key` verschlüsseln.

Achten Sie darauf, dass die Datei zur Abgabe eine `main`-Funktion beinhalten soll. Dazu soll der erste Programmparameter für `key` verwendet werden, der Zweite soll der zu verschlüsselnte Text sein. Die Ausgabe soll lediglich der Geheimtext sein.

[1] `https://de.wikipedia.org/wiki/Caesar-Verschlüsselung`

Hinweise zur Aufgabenstellung

Für die Lösung dieser Aufgabe benötigen Sie folgende Grundkenntnisse:

- Kontrollfluss (`if`)
- Umgang mit Datentyp `char` in C
- Umgang mit C-Strings
- Dynamische Speicherverwaltung (`malloc`)
- Schleifen in C
- Benutzung von `gcc`

Hinweise zur Abgabe

- Erstellen Sie eine ZIP- bzw. TGZ-Archivdatei, welche die geforderten Dateien enthält.
- Fügen Sie dem Archiv keine weiteren Dateien oder Ordner hinzu.
- Reichen Sie Ihre Lösung unter `https://osg.informatik.tu-chemnitz.de/submit` ein.
- Bis zum Abgabeende (Deadline), sofern gegeben, können beliebig neue Lösungen eingereicht werden, die die jeweils älteren Versionen ersetzen.
- Ihr Programm muss auf der Testmaschine übersetzbar sein. Deren Details sind auf dem OpenSubmit-Dashboard verfügbar.
- Ihre Lösung wird automatisch validiert. Sie werden über den Abschluss der Validierung per eMail informiert.