

WPP	Praktikum IT-Sicherheit	Hübner / Behnke
WS 2015	Aufgabe 2 – Symmetrische Verschlüsselungstechniken	Seite 1 von 2

1. Pseudozufallszahlengenerierung

Implementieren Sie in JAVA einen Pseudozufallszahlengenerator nach der Linearen Kongruenzmethode!

Stellen Sie eine Klasse **LCG** mit einer Methode **nextValue()** zur Verfügung. Diese Methode soll nach der linearen Kongruenzmethode einen „Zufallswert“ liefern. Der Startwert des Pseudozufallszahlengenerators (X_0) soll dem Konstruktor der **LCG** – Klasse als Parameter übergeben werden.

- Verwenden Sie eine Parameterkombination für a , b und N aus der Datei „LinearerKongruenzgenerator-Infos.pdf“!
- Achten Sie grundsätzlich auf die Verwendung eines geeigneten Datentyps (z.B. „long“), um mögliche Überläufe zu vermeiden!

2. Stromchiffre

a. Schreiben Sie unter Verwendung der **LCG** – Klasse aus Teil 1 ein JAVA-Programm **HC1** („HAW-Chiffre 1“), welches als Eingabeparameter von der Standardeingabe einen numerischen Schlüssel (Startwert) sowie den Pfad für eine zu verschlüsselnde / entschlüsselnde Datei erhält. Ihr Programm soll jedes Byte der Datei mit einem – ausgehend vom übergebenen Startwert – „zufällig“ erzeugten Schlüsselbyte mittels XOR verknüpfen und das Ergebnis in eine neue Chiffredatei ausgeben.

b. Testen Sie Ihre Stromchiffre **HC1**, indem Sie eine Klartextdatei verschlüsseln und die erzeugte Chiffredatei anschließend durch einen nochmaligen Aufruf von **HC1** wieder entschlüsseln. Verifizieren Sie (z.B. mittels „diff“), dass beide Dateien identische Inhalte besitzen.

- Arbeiten Sie mit *Input/Outputstreams* und vermeiden Sie die Verwendung von „*Buffered Reader*“ oder „*Buffered Writer*“ – Klassen!

3. TripleDES als Blockchiffre

a. Implementieren Sie in JAVA ein Programm **TripleDES** mit der Aufgabe, alle Bytes einer beliebigen Datei mit dem TripleDES-Verfahren zu verschlüsseln oder zu entschlüsseln. Ihr Programm soll mit drei verschiedenen DES-Schlüsseln arbeiten und in der Reihenfolge EDE (encrypt-decrypt-encrypt) das DES-Verfahren dreimal hintereinander durchlaufen. Implementieren Sie als Blockchiffre-Betriebsart¹ den CFB-Modus.

¹ Die Betriebsart bezieht sich hier auf das gesamte TripleDES-Verfahren, nicht auf die einzelnen DES-Aufrufe!

WPP	Praktikum IT-Sicherheit	Hübner / Behnke
WS 2015	Aufgabe 2 – Symmetrische Verschlüsselungstechniken	Seite 2 von 2

Kommandozeilen-Parameter:

1. Dateiname einer zu verschlüsselnden/entschlüsselnden Datei
2. Dateiname einer Schlüssel-Datei mit folgendem Inhalt:
Byte 1-24: 24 Schlüsselbytes (3 DES-Schlüssel mit je 8 Byte, wobei von jedem Byte jeweils 7 Bit verwendet werden)
Byte 25-32: 8 Bytes für den Initialisierungsvektor zum Betrieb im CFB - Modus
3. Dateiname der Ausgabedatei
4. Status-String zur Angabe der gewünschten Operation:
encrypt – Verschlüsselung der Datei
decrypt – Entschlüsselung der Datei

b. Testen Sie Ihre Implementierung, indem Sie die Datei **3DESTest.enc** entschlüsseln (Tipp: Ergebnis ist eine PDF-Datei). Schlüssel und Initialisierungsvektor finden Sie in der Datei **3DESTest.key**.

➤ Benutzen Sie als DES-Implementierung die mitgelieferte Datei *DES.java* !

➤ Beispielcode zum blockweisen Kopieren einer Datei:

```

FileInputStream in;
FileOutputStream out;
byte[] buffer = new byte[1024];
int len;
while ((len = in.read(buffer)) > 0) {
    out.write(buffer, 0, len);
}

```

Viel Spaß!