## Aufgaben

**2.1.** Die Stromchiffre aus Definition 2.1 kann einfach verallgemeinert werden, so dass sie auf beliebigen Alphabeten operiert und nicht auf das binäre beschränkt ist. Für Handchiffren, ðChiffren für die manuelle Verschlüsselung ohne Computer, ist es beispielsweise vorteilhaft, wenn die Stromchiffre direkt Buchstaben verschlüsseln kann.

1. Beschreiben Sie die Ver- und Entschlüsselungsfunktion für eine Stromchiffre, die auf den Buchstaben (A, B,..., Z) operiert, wobei diese durch die Zahlen $(0,1,\ldots,25)$ dargestellt werden. Wie sieht der Schlüsselstrom aus?
2. Entschlüsseln Sie den folgenden Text:
   `bsaspp kkuosr`,
   der mit dem folgenden Schlüsselstrom chiffriert wurde:
   `rsidpy dkawoa`
3. Wie wurde der junge Mann umgebracht?

**2.2.** Gegeben sei ein Einmalschlüssel eines One-Time-Pads mit 1 GByte, der auf einer CD-ROM gespeichert ist. Diskutieren Sie die folgenden praktischen Aspekte für die Nutzung des OTPs: Lebensdauer des Schlüssels, Speicherung des Schlüssels während und nach der Lebensdauer, Schlüsselerzeugung, Verteilung des Schlüssels etc.

**2.3.** Wir nehmen ein OTP an, bei dem der Schlüssel nur 128 Bit beträgt. Für größere Klartexte wird der Schlüssel periodisch wiederverwendet. Wie kann diese Verschlüsselung gebrochen werden?

**2.4.** Auf den ersten Blick scheint es möglich, das One-Time-Pad durch vollständige Schlüsselsuche zu brechen, was einen Widerspruch zu der informationstheoretischen Sicherheit des OTPs ist. Gegeben sei eine kurze Nachricht bestehend aus 5 ASCII-Zeichen, d. h. 40 Bit. Dieser Klartext wurde mit 40 Bit eines OTPs verschlüsselt. Beschreiben Sie genau, warum eine vollständige Schlüsselsuche nicht zum Erfolg führt, obwohl genug Rechenleistung für die Suche zur Verfügung steht.

*Bemerkung:* Das Paradox muss aufgelöst werden, d. h. Anworten à la "Das OTP ist beweisbar sicher, daher funktioniert die vollständige Schlüsselsuche nicht." reichen nicht.

**2.5.** Gegeben sei ein LFSR mit den Koeffizienten $(p_2 = 1, p_1 = 0, p_0 = 1)$.

1. Welche Folge wird mit den Startwerten $(s_2 = 1, s_1 = 0, s_0 = 0)$ erzeugt?
2. Welche Folge wird mit den Startwerten $(s_2 = 0, s_1 = 1, s_0 = 1)$ erzeugt?
3. Wie hängen die beiden Folgen zusammen?

**2.6.** Gegeben sei eine Stromchiffre mit kurzer Periode, von der wir wissen, dass sie im Bereich von 150–200 Bit liegt. Wir wissen *nichts* über die Interna der Chiffre, d. h. wir können auch nicht annehmen, dass es sich um ein einfaches LFSR handelt. Bei dem Klartext, der verschlüsselt wird, handelt es sich um ASCII-Zeichen in deutscher Sprache.

Beschreiben Sie genau, wie die Chiffre gebrochen werden kann. Spezifizieren Sie, wie viel Klartext und Geheimtext Oskar kennen muss, und wie er das gesamte Chiffrat entschlüsseln kann.

**2.7.** Gegeben sei ein LFSR vom Grad 8 mit dem Rückkopplungspolynom aus Tabelle 2.4. Berechnen Sie die ersten zwei Bytes, die von dem Schieberegister generiert werden, wenn der Startvektor, d. h. der Anfangszustand der Flipflops, den Hexadezimalwert FF hat.

**2.8.** In dieser Aufgabe betrachten wir LFSRs etwas detaillierter. Es gibt drei Arten von LFSRs:

- LFSRs, die Maximalfolgen erzeugen. Diese LFSR basieren auf *primitiven Polynomen*.
- LFSRs, welche keine Maximalfolgen erzeugen, deren Sequenzlänge aber unabhängig von dem Startwert des Registers ist. Diese LFSRs basieren auf *irreduziblen Polynomen*. Man beachte, dass alle primitiven Polynome auch irreduzibel sind, das Umgekehrte gilt allerdings nicht.
- LFSRs, die keine Maximalfolgen erzeugen und bei denen die Sequenzlänge von dem Startwert des Registers abhängt. Diese Schieberegister basieren auf *reduziblen* Polynomen.

Wir betrachten nur Beispiele für die verschiedenen LFSR-Arten. Bestimmen Sie *alle* Sequenzen, die von den folgenden Schieberegistern erzeugt werden:

1. $x^4 + x + 1$
2. $x^4 + x^2 + 1$
3. $x^4 + x^3 + x^2 + x + 1$

Beachten Sie, dass für jedes Register die Summe der Längen aller Sequenzen $2^m - 1$ ergeben muss. Zeichnen Sie für jedes Polynom das zugehörige LFSR. Welches Polynom ist primitiv, irreduzibel bzw. reduzibel?

**2.9.** Gegeben sei eine Stromchiffre, die aus einem einzelnen LFSR als Schlüsselstromgenerator besteht. Das Schieberegister hat die Länge 256.

1. Wie viele Klartext-Geheimtext-Paare werden für einen Angriff benötigt?
2. Beschreiben Sie im Detail alle Schritte, die für einen Angriff notwendig sind. Zeigen Sie die Gleichung, die der Angreifer am Ende lösen muss.
3. Was ist der Schlüssel bei dieser Chiffre? Warum macht es keinen Sinn, den Startwert des Registers als Schlüssel zu verwenden?

**2.10.** Ziel ist es, eine LFSR-basierte Stromchiffre anzugreifen, bei der der Angreifer einen Teil des Klartexts kennt. Die Klartextbits sind:

```
1001 0010 0110 1101 1001 0010 0110
```

Der Angreifer beobachtet auf dem Kanal das folgende Chiffrat:

```
1011 1100 0011 0001 0010 1011 0001
```

1. Welchen Grad $m$ hat das LFSR?
2. Was ist der Startvektor?
3. Bestimmen Sie die Rückkopplungskoeffizienten.
4. Zeichnen Sie ein Diagramm des LFSR und verifizieren Sie die erzeugte Ausgangssequenz.

**2.11.** In dieser Aufgabe wird ebenfalls eine Stromchiffre angegriffen, die aus einem einzelnen LFSR besteht. Um Buchstaben zu verschlüsseln, wird für die 26 Großbuchstaben und für die Ziffern 0, 1, 2, 3, 4, 5 die folgende Codierung mit 5 Bit verwendet:

$$\begin{aligned} A \leftrightarrow 0 &= 00000_2 \\ &\vdots \\ Z \leftrightarrow 25 &= 11001_2 \\ 0 \leftrightarrow 26 &= 11010_2 \\ &\vdots \\ 5 \leftrightarrow 31 &= 11111_2 \end{aligned}$$

Dem Angreifer sind die folgende Details über das System bekannt:

- Das Schieberegister hat den Grad $m = 6$.
- Jede Nachricht beginnt mit dem Header `WPI`.

Auf dem Kanal wird das folgende Chiffrat mitgeschnitten (das vierte Symbol ist eine Null):

```
j5a0edj2b
```

1. Wie lautet der Startvektor?
2. Was sind die Rückkopplungskoeffizienten des LFSR?
3. Schreiben Sie ein Programm in einer beliebigen Programmiersprache, welches die gesamte Sequenz erzeugt, und bestimmen Sie den Klartext.
4. Wo lebt das Wesen nach `WPI`?
5. Wie würde man den Angriff klassifizieren?

**2.12.** Geben sei die Chiffre Trivium, bei der der Initialisierungsvektor $IV$ und der Schlüssel nur aus Nullen bestehen. Berechnen Sie die ersten 70 Bits $s_1, \ldots, s_{70}$, die während der Initialisierungsphase erzeugt werden. (Man beachte, dass dies nur interne Bits sind, die nicht für die Verschlüsselung von Klartext benutzt werden, da in der Initialisierungsphase 1152 Bits erzeugt werden, bevor der eigentliche Schlüsselstrom ausgegeben wird.)