Grundlagen der Kryptographie

Praktikum 2

Prof. Dr. Christoph Saatjohann

28.10.2025 - 18.11.2025

1 One-Time-Pad mit einem LCG

Gegeben sei ein One-Time-Pad (OTP) von dem Sie wissen dass dieser nicht mit einem echten eimaligen Zufallsstring als Schlüssel, sondern mit folgendem Linearen Kongruenzgenerator (LCG) genutzt wird:
$s_{i+1} = A * s_i + B \bmod 2^8$

Mittels dieser Chiffre wurde eine geheime Nachricht verschlüsselt, die Sie gerne entschlüsseln würden. Sie wissen das eine Datei mit folgendem 3 Header-Bytes (p_1 = erstes Plaintextbyte) verschlüsselt wurde:
$p_1 = 0x4C$
$p_2 = 0x43$
$p_3 = 0x47$

Die abgefangene verschlüsselte Nachricht lautet (hexadezimal):

```
05a5206fb5d13c944dea1a7acdb482139122b76a3077f1c8ba7cd674e9257c12c8b6
af9586e1dbbda13fb18328d42f2537519282859586a6449a7590dd5d1e326dc349fd
```

Hinweise:

- Die Datei ist ASCII-basiert
- Der Modulus des LCGs ist bewusst auf 2^8 gesetzt worden. Sie können Zeichen für Zeichen bzw. Byte für Byte, berechnen.
- Sie können Python, C, etc. oder auch online Tools nutzen, z.B.:

  Cyberchef: `https://gchq.github.io/CyberChef`

  WolframAlpha: (`https://www.wolframalpha.com/`

  Insbesondere für die Berechnung der modularen Inverse bieten sich solche Tools an: WolframAlpha Widget: `https://www.wolframalpha.com/widgets/view.jsp?id=a9d64f006accc458a887ceb71eca63c6`

planetcalc: `https://planetcalc.com/3311/`

2 ChaCha20 - Quarter Round (QR)

2.1 Berechnung per Hand

Berechnen Sie die Ausgaben der ChaCha20-QuarterRound (QR) mit folgenden Eingangs-Werten:
$a = 0x00000001$ $b = 0x00000000$ $c = 0x00000000$ $d = 0x00000000$

Führen Sie die Berechnung nachvollziehbar per Hand durch.

2.2 Programmierung der QR

Implementieren Sie die QR von ChaCha20 in C. Nutzen Sie, wo es Ihnen sinnvoll erscheint, einen 32 Bit Datentyp, bspw. `uint32` (aus stdint.h).

Überprüfen Sie Ihre Berechnungen aus der ersten Aufgabe.