– Praktikumsaufgabe 2 –

Thema: *Knacken historischer Chiffren, triviale XOR-Verschlüsselung*

Zielstellung: Ciphertext-only-Angriff auf eine monoalphabetische Chiffre, Nachvollzug des Kasiski-Tests, Festigung der Programmierung mittels C, Nutzung eines Werkzeugs zum Ver- und Entschlüsseln (CrypTool), Gewinnung elementarer Erkenntnisse zur XOR-Verschlüsselung

1. Entschlüsseln Sie *geheimmono.txt* per Hand, über das folgendes bekannt ist:

   - Es handelt sich um eine monoalphabetische Chiffre.
   - Der Klartext wurde in Deutsch abgefasst.

   Hinweise:

   - Sie benötigen ein kleines Programm, das Ihnen die Häufigkeit der einzelnen Buchstaben im Chiffrat ausgibt. Dies leistet *countchars.c*.
   - Die Tabelle der Buchstabenhäufigkeiten finden Sie im 2. Teil der Vorlesungsunterlagen auf Seite 7.

   Quelle: Albrecht Beutelspacher: *Kryptologie*. Vieweg+Teubner, 2009. Aufgabe 23, S.22

2. Bestimmen Sie die Schlüssellänge des Vigenère-verschlüsselten Chiffrates *vigenere.txt* mittels Kasiski-Test. Entwickeln Sie für die Bestimmung der Abstände gleicher Morpheme im Chiffrat ein kleines C-Programm.

   Hinweis: Um eine Zeichenkette in einer anderen zu suchen, gibt es die Funktion `strstr()`.

3. Machen Sie sich mit dem Werkzeug CrypTool vertraut; rufen Sie es folgendermaßen auf: `~robge/is-tools/cryptool` und lösen Sie danach die folgenden Aufgaben:

   a) Überprüfen Sie die Lösung aus Aufgabe 1.

   b) Entschlüsseln Sie das Chiffrat aus Aufgabe 2.

   c)* Versuchen Sie, ohne zusätzliche Informationen das Chiffrat in *unbekannt.txt* zu entschlüsseln[1].

4. Die Firma *Sooper Dooper Encryption, Inc.* bietet für 499,95$ pro Installation eine Software namens *SuperCrypt* an, die Dateien sehr effizient ver- und entschlüsseln soll. Ihr Chef arbeitet nach dem Motto „Viel hilft viel.“ und möchte gern zwanzig SuperCrypt-

[1] Es handelt sich um das Rätsel eines so genannten Mystery-Geocaches der höchsten Schwierigkeitsstufe.

Lizenzen erwerben. Um sich abzusichern, bittet er Sie, *SuperCrypt* zu testen.

a) Laden Sie *supercrypt* herunter[2], setzen Sie dessen Ausführungsrechte und überzeugen Sie sich, dass es Dateien verschlüsseln kann.

b) Eine Known-Plaintext-Attacke auf XOR-Verschlüsselung ist besonders einfach, denn es gilt:

$$\begin{aligned} C &= P \oplus K && \mid \oplus P \\ C \oplus P &= P \oplus K \oplus P && \mid P \oplus P \text{ ist stets wahr} \\ C \oplus P &= K \end{aligned}$$

Wenn man also einen Plaintext P mit einem zugehörigen Chiffrat C (als Schlüssel) verschlüsselt, erhält man offenbar den Schlüssel, der zur Erzeugung von C verwendet wurde als Ergebnis zurück.
Nutzen Sie die simple Implementierung *xor.c*, um diesen Angriff auf *SuperCrypt* auszuführen. Wie lautet der (fest in *SuperCrypt* eingebaute) Schlüssel?

5.* Das Brechen einer XOR-Verschlüsselung mittels *Ciphertext-Only-Attack* ist ebenfalls recht einfach, wenn man sich vor Augen hält, dass es sich dabei um eine polyalphabetische Chiffrierung analog zur Vigenere-Chiffre handelt.

Im ersten Schritt bestimmt man wiederum die Länge des Schlüssels: Zunächst bestimmt man den Friedman-Index für das Chiffrat und eine um n Stellen verschobene Version des Chiffrats. Für $n = 1$ bedeutet das also, dass das Zeichen an Position 0 mit dem Zeichen an Position 1 verglichen wird, danach das Zeichen an Position 1 mit dem Zeichen an Position 2 usw. Für $n = 2$ wird das Zeichen an Position 0 mit dem Zeichen an Position 2 verglichen, dann das Zeichen an Position 1 mit dem an Position 3 usw. Die Anzahl Paare mit gleichen Zeichen wird für jedes n bestimmt. Für ein bestimmtes n sowie dessen Vielfache ist die Anzahl signifikant erhöht; dieser Wert n ist die gesuchte Schlüssellänge.

Implementieren Sie ein C-Programm zur Bestimmung der Schlüssellänge XOR-verschlüsselter Texte und bestimmen Sie die Länge für das Chiffrat *xorchiffrat.txt*

6. Entschlüsseln Sie per Hand die mittels PlayFair (Kennwort: „Halbzeit“) verschlüsselte Phrase „GIFCRZSPYKDPEIZI“.

[2] Sie laden aber bitte niemals ausführbare Dateien aus nicht vertrauenswürdigen Quellen herunter und führen diese aus.