Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Dipl.-Ing. Stephan Günther, M.Sc.
Nadine Herold, M.Sc.
Dipl.-Inf. Stephan Posselt

# Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme
## Übungsblatt 11 (1. Juli – 5. Juli 2013)

**Hinweis:** Die mit * gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

## Aufgabe 1 Asymmetrische Kryptographie: El Gamal

In der Vorlesung haben wir DH76 als Beispiel für ein symmetrische Schlüsselaustauschverfahren kennengelernt. Mit Hilfe von DH76 konnten zwei Kommunikationspartner Alice und Bob ein gemeinsames Geheimnis errechnen, welches wiederum als Schlüssel für ein symmetrische Verschlüsselungsverfahren wie RC4 verwendet werden kann.
In dieser Aufgabe wollen wir ein asymmetrisches Verfahren untersuchen, bei dem Alice und Bob jeweils einen privaten und einen öffentlichen Schlüssel besitzen. Sei $a$ der öffentliche und $u$ der private Schlüssel von Alice. Will Bob eine Nachricht an Alice senden, so nutzt er $a$ zur Verschlüsselung seiner Nachricht. Zur Entschlüsselung wird der private Schlüssel $u$ benötigt, den nur Alice kennt. Im Detail funktioniert dies wie folgt:

1. Alice wählt eine große Primzahl $p$ und eine dazu passende primitive Kongruenzwurzel $g$.
2. Alice wählt ihren privaten Schlüssel $u \in \{0, 1, \ldots, p-1\}$. Dieser bleibt geheim.
3. Alice berechnet $a = g^u \bmod p$.
4. Alice veröffentlicht das Tripel $(a, p, g)$ als ihren öffentlichen Schlüssel.

Bob verfährt analog. Im Anschluss möchte Bob eine Nachricht $m$ an Alice schicken. Diese Nachricht sei eine Dezimalzahl $m \in \{0, 1, \ldots, p-1\}$. Um diese zu verschlüsseln, verfährt Bob wie folgt:

1. Bob besorgt sich den öffentlichen Schlüssel von Alice.
2. Bob wählt gleichverteilt ein $v \in \{0, 1, \ldots, p-1\}$ und berechnet $b = g^v \bmod p$ und $k = a^v \bmod p$. Bei $k$ handelt es sich um einen sog. transienten Schlüssel, welcher zur Verschlüsselung der Nachricht dient.
3. Bob berechnet den Ciphertext $c = km \bmod p$.
4. Bob sendet das Tupel $(b, c)$ an Alice.

Zur Entschlüsselung geht Alice wie folgt vor:

1. Alice berechnet $k^{-1} = b^{-u} \bmod p$.
2. Anschließend kann Alice $m = k^{-1}c \bmod p$ bestimmen.

a)* Weswegen kann Eve aus $(b, c)$ nicht ohne weiteres auf $m$ schließen?

$b$ an sich hat erst einmal nichts mit der Nachricht $m$ zu tun. Aus $c$ alleine kann Eve keine Rückschlüsse auf $m$ ziehen, da sie $k$ nicht kennt und es (abhängig von der Größe der Zahlen) sehr viele unterschiedliche $k$ gibt, so dass $c = km \bmod p$ gilt.

b)* Zeigen Sie, dass $m = k^{-1}c \bmod p$ gilt.
**Hinweis:** $(x \bmod p)(y \bmod p) \bmod p = xy \bmod p$

$$\begin{aligned} k^{-1}c \bmod p &= (b^{-u} \bmod p)(km \bmod p) \bmod p \\ &= (g^{-uv} \bmod p)(a^v \bmod p)(m \bmod p) \bmod p \\ &= (g^{-uv} \bmod p)(g^{uv} \bmod p)(m \bmod p) \bmod p \\ &= g^{-uv} g^{uv} m \bmod p \\ &= m \bmod p \\ &= m \quad (\text{da } m \in \{0, 1, \ldots, p-1\}) \end{aligned}$$

Bob möchte nun die binäre Nachricht $m = \texttt{10101111}$ an Alice schicken. Der öffentliche Schlüssel von Alice sei $(a, p, g) = (9, 17, 12)$.

c)* Zerlegen Sie die Nachricht $m$ geeignet in zwei Teilnachrichten $m_1$,$m_2$ und berechnen Sie die zugehörigen Ciphertexte $c_1$,$c_2$.

Es sei $m_1 = \texttt{1010}_{(2)} = 10$, $m_2 = \texttt{1111}_{(2)} = 15$. Wir wählen außerdem $v_1 = 7$ und $v_2 = 15$.

$$\begin{aligned} b_1 &= g^{v_1} \bmod p = 12^7 \bmod 17 = 7 \\ k_1 &= a^{v_1} \bmod p = 9^7 \bmod 17 = 2 \\ c_1 &= k_1 m_1 \bmod p = 2 \cdot 10 \bmod 17 = 3 \end{aligned}$$

$$\begin{aligned} b_2 &= 10 \\ k_2 &= 2 \\ c_2 &= 13 \end{aligned}$$

d) Geben Sie die beiden Nachrichten an, die Bob an Alice überträgt

Bob muss zusammen mit $c_1$ und $c_2$ auch jeweils $b_1$ und $b_2$ übertragen: $(7, 3)$,$(10, 13)$.

e) Zeigen Sie, dass Alice $c_1$,$c_2$ mit ihrem privaten Schlüssel $u = 10$ wieder entschlüsseln kann.
**Hinweis:** $k^{-1} \bmod p = k^{p-2} \bmod p$ wenn $p$ prim ist (Euler).

$$\begin{aligned} k_1^{-1} &= b_1^{-u} \bmod p \\ &= b_1^{-1} b^{-u+1} \bmod p \\ &= b_1^{p-2} b^{-u+1} \bmod p \\ &= b_1^{p-u-1} \bmod p \\ &= b_1^{17-10-1} \bmod p \\ &= b_1^6 \bmod p = 9 \\ m_1 &= k_1^{-1} c_1 \bmod p = 9 \cdot 3 \bmod 17 = 10 \end{aligned}$$

$m_2$ analog.

f)* Wie bewerten Sie El Gamal hinsichtlich der Authentizität der Kommunikationspartner?

Das hier eingeführte Verfahren garantiert keine Authentizität, da es beispielsweise anfällig gegenüber Man-in-the-Middle Attacken ist. Allerdings gibt es eine Erweiterung (Signaturverfahren), welche Authentizität gewährleistet (mehr dazu in der Vorlesung Netzsicherheit).

g)* Da die Verschlüsselung mittels El Gamal im Vergleich zu symmetrischen Verfahren rechenaufwendig ist, wäre eine Kombination beider Verfahren wünschenswert. Wie könnte solch ein hybrides Verfahren funktionieren?

El Gamal (oder ein anderes asymmetrisches Verfahren) wird häufig nur zum verschlüsselten Austausch eines gemeinsamen Geheimnisses verwendet, welches dann wiederum als Schlüssel für ein symmetrisches Verfahren genutzt wird.

## Aufgabe 2 Kompression: Huffman-Kodierung

Gegeben sei das Alphabet $\mathcal{A} = \{a, b, c, d\}$ und die Nachricht

$$m = \texttt{aabccdacababbbbcbddbbbaababdbcbabdbcadba} \in \mathcal{A}^{40}.$$

a)* Bestimmen Sie die Auftrittswahrscheinlichkeiten $p_{i\in\mathcal{A}}$ der einzelnen Zeichen in $m$.

Aus den Zeichenhäufigkeiten ergibt sich:

$$p_a = \frac{11}{40},\ p_b = \frac{17}{40},\ p_c = \frac{6}{40},\ p_d = \frac{6}{40}$$

b) Bestimmen Sie die den Informationsgehalt $I(p_{i\in\mathcal{A}})$ der einzelnen Zeichen.

Für den Informationsgehalt erhalten wir:

$$\begin{aligned} I(p_a) &= -\log_2(p_a) \approx 1.86\,\text{bit} \\ I(p_b) &= -\log_2(p_b) \approx 1.23\,\text{bit} \\ I(p_c) &= -\log_2(p_c) \approx 2.74\,\text{bit} \\ I(p_d) &= -\log_2(p_d) \approx 2.74\,\text{bit} \end{aligned}$$

c) Die Nachricht $m$ stamme aus einer Nachrichtenquelle $X$. Bestimmen Sie auf Basis der bisherigen Ergebnisse die Quellenentropie $H(X)$.

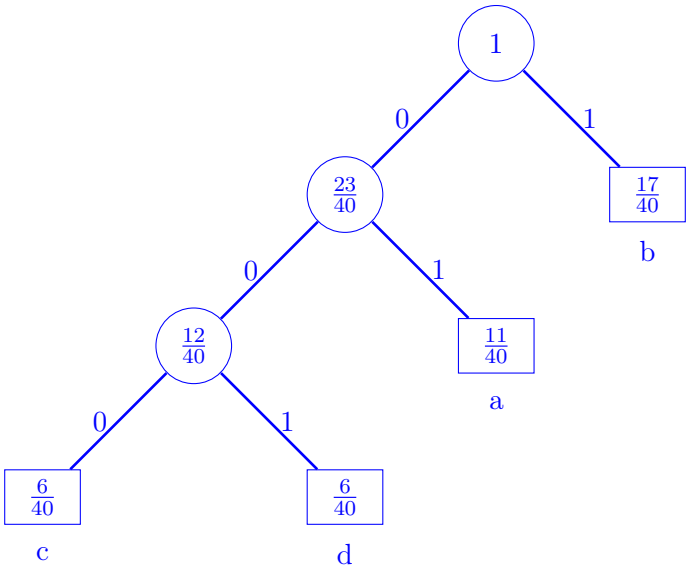
Die Quellenentropie ist nichts weiter als die mit den Auftrittswahrscheinlichkeiten gewichtete Summe des Informationsgehalts der Einzelzeichen:

$$H(X) = \sum_{i\in\mathcal{A}} p_i I(p_i) \approx 1.86\,\text{bit}$$

Dies bedeutet, dass sich die Zeichen der Quelle $X$ mit durchschnittlich 1.86 bit pro Zeichen kodieren lassen.

d) Bestimmen Sie nun einen binären Huffman-Code $C$ für diese Nachrichtenquelle.

Siehe Vorlesungsfolien. Beginnend bei den beiden Zeichen mit der geringsten Auftrittswahrscehinlichkeit wird ein Baum beginnend bei den Blättern (den Zeichen) konstruiert. Dabei werden in jedem Schritt stets die beiden Knoten bzw. Blätter zusammengefasst, so dass die Summe deren Auftrittswahrscheinlichkeiten über alle Knoten bzw. Blätter minimal ist:




Die Kanten werden mit 0 bzw. 1 beschriftet. Der Code lässt sich nun einfach ablesen, indem man von der Wurzel ausgehend die Kantenbeschriftungen abliest: $C = \{a \mapsto \texttt{01}, b \mapsto \texttt{1}, c \mapsto \texttt{000}, d \mapsto \texttt{001}\}$

Zeichen mit hoher Auftrittswahrscheinlichkeiten erhalten kurze Codewörter. Außerdem lässt sich leicht überprüfen, dass $C$ präfixfrei ist: Kein Codewort ist ein Präfix eines anderen Codeworts. Dies erleichtert die Dekodierung.

e) Bestimmen Sie die durchschnittliche Codewortlänge von $C$.

Die durchschnittliche Codewortlänge ergibt sich aus der mit den Auftrittswahrscheinlichkeiten gewichteten Summe der Codewortlängen. Sei $l(c)$ die Länge eines Codeworts in $C$ und $c(i)$ die Funktion, welche ein Zeichen $i \in \mathcal{A}$ auf ein Codewort aus $C$ abbildet. Dann erhalten wir:

$$\bar{l}_C = \sum_{i \in \mathcal{A}} p_i \cdot l(c(i)) \approx 1.88$$

f) Vergleichen Sie die durchschnittliche Codewortlänge von $C$ mit der Codewortlänge eines uniformen[1] Binärcodes.

Der kürzeste uniforme Code hat eine durchschnittle Codewortlänge von $\bar{l}_U = 2$. Die Ersparnis beträgt also etwa 6 %.

[1] Ein Code heißt *uniform*, wenn alle Codewörter dieselbe Länge aufweisen.