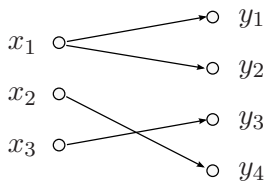
1. Übungsblatt
Kryptographie und Mediensicherheit (SoSe 2016)

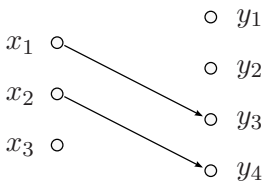
Bauhaus-Universität Weimar, Professur für Mediensicherheit
Prof. Dr. Stefan Lucks, Jakob Wenzel
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching`
Abgabe: Bis zum 25.04.2016, 13:30 Uhr zu Beginn der Übung oder per E-Mail an `jakob.wenzel(at)uni-weimar.de`.
Lösungen sind bevorzugt in LaTeX zu verfassen. Ein Template finden Sie auf der Übungsseite der Veranstaltung.

Aufgabe 1 – Wiederholung Funktion und Permutation (6 Punkte)
Wiederholen Sie selbständig die Begriffe Funktion, Injektion, Surjektion, Bijektion, Permutation und Transposition. Im Folgenden seien $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ und $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\}$ zwei Mengen und $f : \mathcal{X} \to \mathcal{Y}$ eine Funktion. Beantworten Sie die folgenden Fragen jeweils immer mit einer *kurzen* Begründung.

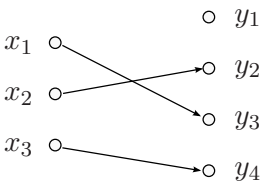
a) Welche der folgenden Diagramme repräsentieren eine Funktion?



(a1)

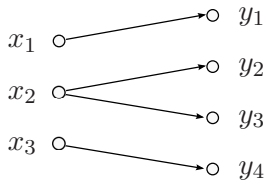


(a2)

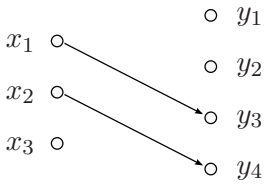


(a3)

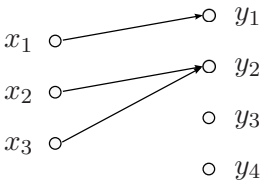
b) Welche der folgenden Diagramme repräsentieren eine injektive Funktion?



(b1)

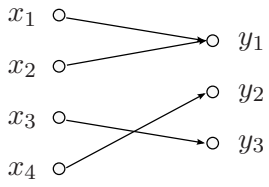


(b2)

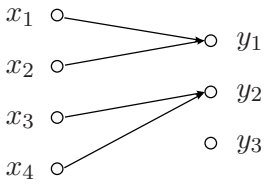


(b3)

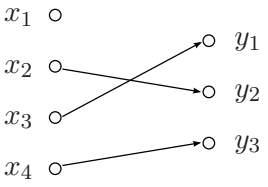
c) Welche der folgenden Diagramme repräsentieren eine surjektive Funktion?



(c1)



(c2)



(c3)

d) Kann f eine Bijektion sein, wenn $n < m$? Kann f eine Bijektion sein wenn $n > m$?

e) Kann f eine Permutation sein, wenn $n < m$? Kann f eine Permutation sein, wenn $n > m$?

f) Erhält eine Permutation stets das Hamming-Gewicht der Eingaben? Wie verhält es sich für eine Transposition?

Aufgabe 2 – Komplexitätsrechnung (3 Punkte)
Angenommen Alice verfügt über ein Programm, dass auf ihrem Rechner innerhalb von 12 Minuten 2^{32} Schleifendurchläufe schafft.

a) Rechnen Sie hoch, wie viele Durchläufe das Programm innerhalb von 1 Stunde, 1 Tag, 1 Woche, 1 Monat und 1 Jahr schafft.

b) Unter einem "Brute-Force"-Angriff auf ein Verschlüsselungsverfahren versteht man das vollständige Ausprobieren aller möglichen Schlüssel. Angenommen pro Schleifendurchlauf kann 1 Schlüssel getestet werden. Welche maximale Schlüssellänge (in Bits) kann Alice mit dem Programm – auf ihrem Rechner – innerhalb von 1 Stunde, 1 Tag, 1 Woche, 1 Monat bzw. 1 Jahr durch Brute-Force-Angriffe knacken?

c) Wie lange bräuchte Alice, unter der Annahme aus Teil b), um eine Chiffre mit einer Schlüssellänge von 40, 64 bzw. 128 Bits zu knacken.

Aufgabe 3 – Bedingte Wahrscheinlichkeiten (4 Punkte)
Wiederholen Sie selbständig Zufallsvariablen, bedingte Wahrscheinlichkeiten und Bayes Theorem. Sie haben zwei Würfel vor sich, einen fairen und einen, bei dem ein Gewicht dafür sorgt, dass er immer mit jeweils gleicher Wahrscheinlichkeit entweder auf der 1 oder der 6 landet. Beide sind äußerlich nicht voneinander zu unterscheiden. Angenommen, Sie nehmen einen der Würfel und würfeln eine 6.

a) Wie hoch ist die Wahrscheinlichkeit, dass Sie den manipulierten Würfel gewählt haben?

b) Sie würfeln noch einmal mit dem selben Würfel. Wie hoch ist die Wahrscheinlichkeit beim nächsten Wurf auch eine 6 zu würfeln?

Aufgabe 4 – Two-Time Pad (4 Punkte)
Angenommen, die One-Time-Pad-Verschlüsselung der Nachricht "Sende 100 Euro an Bob" sei `2b139fdd1af18fc5e6d82ea722b396b6244b5f5213`. Die Klartextbuchstaben sind 8-Bit-ASCII-kodiert und der Chiffretext in Hex. Geben Sie den Chiffretext (unter dem gleichen geheimen Schlüssel) zur Nachricht "Sende 500 Euro an Eve" an und begründen Sie Ihre Lösung.

Aufgabe 5 – Substitutionschiffre (4 Punkte)
Der folgende Chiffretext wurde mit einer Substitutionschiffre verschlüsselt. Klartext- und Chiffretextmenge bestehen jeweils aus den 26 Kleinbuchstaben `a-z`:

```
nkccd bolox xkmrg krbro sdexn obuox xdxsc qoryo bdjew cmryo xcdox nocco xnobw
oxcmr pkors qscdg oxxke mrnob cdyvj kepns ococc dbolo xwosc dswwe xnono btoxs
qoxsc dnsok wgoxs qcdox fyxcy vmrow cdbol oxobp eovvd csxn
```

Sie können annehmen, dass der dazugehörende Klartext deutschsprachig ist. Die folgende Tabelle zeigt Ihnen die Häufigkeitsverteilung in deutschsprachigen Texten [1]:

Buchstabe	E	N	I	S	R	A	T	D	H	U
Rel. Häufigkeit	17,40	9,78	7,55	7,27	7,00	6,51	6,15	5,08	4,76	4,35
Buchstabe	L	C	G	M	O	B	W	F	K	Z
Rel. Häufigkeit	3,44	3,06	3,01	2,53	2,51	1,89	1,89	1,66	1,21	1,13
Buchstabe	P	V	J	Y	X	Q				
Rel. Häufigkeit	0,79	0,67	0,27	0,04	0,03	0,02				

a) Ermitteln Sie – z.B. mit einem kleinen Python-Programm oder durch Zählen – die Häufigkeitsverteilung der einzelnen Zeichen im Kryptogramm.

b) Entschlüsseln Sie den Chiffretext. *Hinweis: Die Häufigkeitsverteilung der Buchstaben im Chiffretext weicht etwas von der obigen Verteilung ab. Zwei aufeinanderfolgende Fünferblöcke lauten im Klartext "erkenntnis".*

Aufgabe 6 – Caesar-Chiffre – Programmieraufgabe (4 Punkte)
Schreiben Sie ein Programm in Python, welches die Caesar-Chiffre implementiert. Ihr Programm soll dabei folgende Eingaben als Kommandozeilenparameter entgegennehmen:

- Eine Datei die den Klartext enthält und
- einen Schlüssel K.

Ein Klartext soll dabei nur aus Buchstaben bestehen. Ihr Programm soll zunächst den Klartext in Kleinbuchstaben umwandeln, bevor dieser verschlüsselt wird. Anschließend soll Ihr Programm den verschlüsselten Klartext (Chiffretext) in 5er-Buchstaben-Gruppen ausgeben (dies macht Sinn, da man so nicht auf die Länge einzelner Wörter schließen kann).

Weitere Hinweise:

- Ihr Programm soll folgende Problematiken sinnvoll behandeln:
  - falsche Anzahl / falscher Typ von Parametern
  - ungültige Zeichen im Klartext sollen ignoriert werden (Sonderzeichen, Zahlen, ...)

[1] Albrecht Beutelspacher, Kryptologie, 7. Aufl., Wiesbaden: Vieweg Verlagsgesellschaft, 2005, ISBN 3-8348-0014-7, Seite 10.

- **Bonus (+1 Punkt):** Entschlüsseln Sie die folgenden beiden Kryptogramme und geben sie sowohl den Klartext als auch den dazugehörigen Schlüssel an.

  1. **Kryptogramm:**

     ```
     gwzhj ktzsi fxjhz wjbfd ytwjz xjfts jynrj ufi
     ```

  2. **Kryptogramm:**

     ```
     hspym cfnpd nsypt pcfdp dozfm wpcze estce ppypy
     ncjae tzyes pntas pcepi etdez elwwj fymcp lvlmw p
     ```

Schicken Sie Ihre Lösung als Anhang einer E-Mail

`caesar_<MatrNr>.py`

an `jakob-wenzel(at)uni-weimar.de` mit dem Betreff `[Krypto SS16] Beleg 1` bis zum 25. April 2016, 13:30 Uhr. Es reicht dabei eine Matrikelnummer aus der Gruppe anzugeben. Die vollständigen Namen und Matrikelnummern sollen als Kommentar im Python-Skript stehen.

Aufgabe 7 – E-Mail-Verschlüsselung (2 Punkte)
E-Mails werden in der Regel im Klartext verschickt. Für die meisten gängigen E-Mail-Programme (Thunderbird/icedove, Outlook, ...) gibt es die Möglichkeit, E-Mails mithilfe von Pretty Good Privacy (PGP) bzw. mithilfe des freien GNU Privacy Guards (GPG) zu verschlüsseln. Zeigen Sie, dass Sie E-Mail-Verschlüsselung beherrschen und schicken Sie eine mit PGP/GPG-verschlüsselte Mail an `jakob.wenzel(at)uni-weimar.de` so dass wir sie auch wieder entschlüsseln können. Der Betreff sollte `[Krypto SS16] Beleg 1` sein. Im Klartext sollten Ihr(e) Name(n), Ihre Matrikelnummer(n) und der folgende Text stehen:

```
Es ist nicht genug zu wissen - man muss auch anwenden.
Es ist nicht genug zu wollen - man muss auch tun.
-- Johann Wolfgang v. Goethe
```

Hinweise:

- Für Thunderbird/icedove finden Sie Hinweise zur Installation und Benutzung des Addons `Enigmail` z.B. unter `http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP` Beachten Sie: `Enigmail` ist nur ein Addon. Sie benötigen noch `GPG`.
- Öffentliche Schlüssel lädt man auf einen vertrauenswürdigen Server hoch (nicht notwendig für diese Aufgabe). Die Standardadressen sind `pool.sks-keyservers.net`, `subkeys.pgp.net`, `sks.mit.edu`, `ldap://certserver.pgp.com`. Auf diesen finden Sie auch die öffentlichen Schlüssel der Professur.