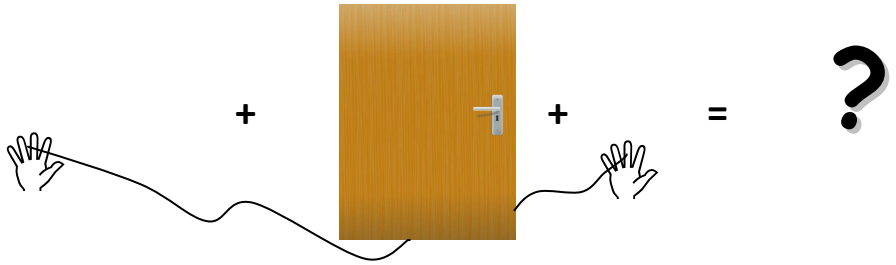
Kommunikation ohne Worte - Ein Kommunikationsprotokoll vereinbaren

Aufgabe 1

Überlegt Euch ein Verfahren um mit ein oder zwei unter einer Tür verlaufenden Schnüre ohne weitere Hilfsmittel und ohne zu sprechen ein Wort zu kommunizieren!



Ihr könnt dazu eine oder zwei Schnüre verwenden. Als Idee könnt Ihr einen der von eurem Lehrer bereit gestellten Übertragungscodes nutzen oder Euch einen eigenen Code ausdenken. Ein Schüler auf der einen Seite der Tür wird nachher ein beliebiges Wort genannt bekommen, dass er dann den Mitschülern auf der anderen Seite der Tür übermittelt.

Notiert auf einem extra Blatt **alle Vereinbarungen**, die Ihr mit Euren Partnern treffen müsst, damit die Kommunikation funktioniert!

Aufgabe 2

Sucht Euch jeweils zu viert eine Tür. Legt zur Datenübertragung eine (oder zwei) Schnüre unter der Tür hindurch. Sendet zwei Mal ein kurzes Wort von einer Seite auf die andere und eine Antwort wieder zurück. Verbessert (falls nötig) eure Vereinbarungen zur Kommunikation!

Aufgabe 3

Einige Gruppen werden nun ihre Verfahren vorstellen. Um die verschiedenen Verfahren vergleichen zu können, beobachtet die Kommunikation der Gruppen und macht Euch Notizen zu folgenden Fragen:

Gruppe	Wie schnell ist die Kommunikation?	Wie häufig können Fehler auftreten?	Wie werden die Buchstaben übermittelt?	Welche weiteren Informationen werden übermittelt?

Bildquelle Tür: Open Clip Art Library, http://www.openclipart.org/detail/20106

Übertragungscode

1) Der Morsecode

A	·−	J	·−−−	S	···	2	··−−−
B	−···	K	−·−	T	−	3	···−−
C	−·−·	L	·−··	U	··−	4	····−
D	−··	M	−−	V	···−	5	·····
E	·	N	−·	W	·−−	6	−····
F	··−·	O	−−−	X	−··−	7	−−···
G	−−·	P	·−−·	Y	−·−−	8	−−−··
H	····	Q	−−·−	Z	−−··	9	−−−−·
I	··	R	·−·	1	·−−−−	0	−−−−−

·	E	··−·	F	−	T	−−	M
··	I	··−−−	2	−·	N	−−·	G
···	S	·−	A	−··	D	−−··	Z
····	H	·−·	R	−···	B	−−···	7
·····	5	·−··	L	−····	6	−−·−	Q
····−	4	·−−	W	−··−	X	−−−	O
···−−	V	·−−·	P	−·−	K	−−−··	8
···−−−	3	·−−−·	J	−·−·	C	−−−−·	9
··−−	U	·−−−−	1	−·−−	Y	−−−−−	0

2) Zuordnung von Buchstaben zu Binärzahlen

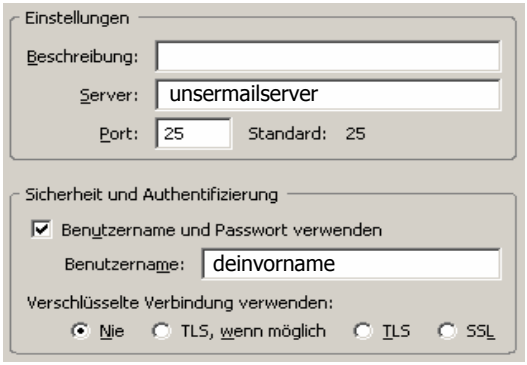
A 1	H 1000	O 1111	V 10110
B 10	I 1001	P 10000	W 10111
C 11	J 1010	Q 10001	X 11000
D 100	K 1011	R 10010	Y 11001
E 101	L 1100	S 10011	Z 11010
F 110	M 1101	T 10100	
G 111	N 1110	U 10101	

3) Zuordnung von Buchstaben zu Zahlenpaaren

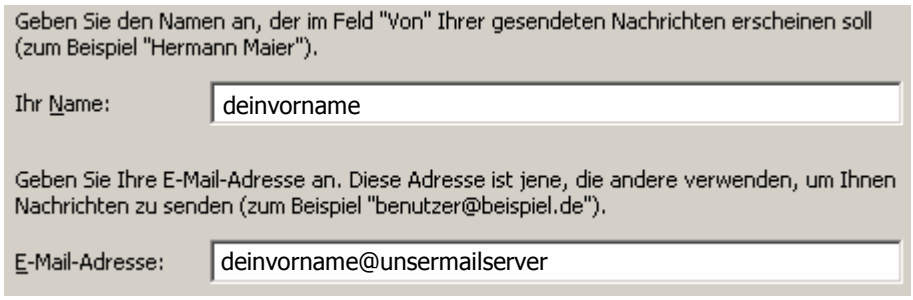
A (1,1)	H (2,3)	O (3,5)	V (5,2)
B (1,2)	I (2,4)	P (4,1)	W (5,3)
C (1,3)	J (2,5)	Q (4,2)	X (5,4)
D (1,4)	K (3,1)	R (4,3)	Y (5,5)
E (1,5)	L (3,2)	S (4,4)	Z (6,1)
F (2,1)	M (3,3)	T (4,5)	
G (2,2)	N (3,4)	U (5,1)	

Anleitung: E-Mail-Postfach in *Thunderbird* einrichten in 12 Schritten

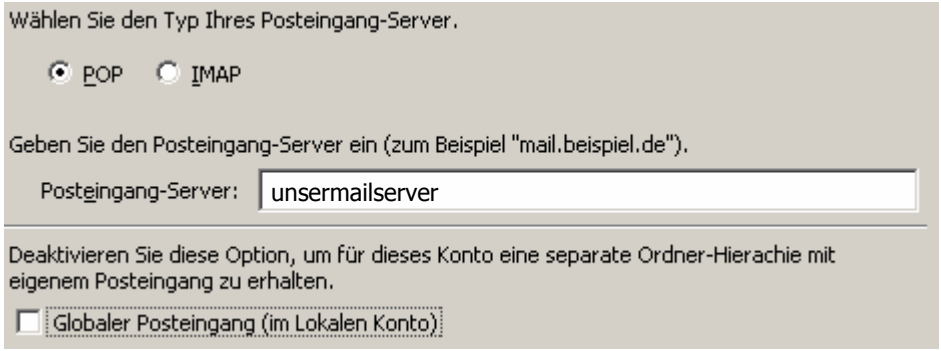
1. Starte das Programm *Thunderbird*!
2. Wähle im Menu „Extras > Konten“!
3. Wähle nun in der linken Spalte unten „Postausgangs-Server (SMTP)“ und dann rechts den Knopf „Hinzufügen...“ aus!
4. Gib im Feld „Server“ **unsermailserver** und im Feld „Benutzername“ **deinen Vornamen** ein **und** bestätige mit dem „OK“-Knopf (siehe Abb. rechts)!



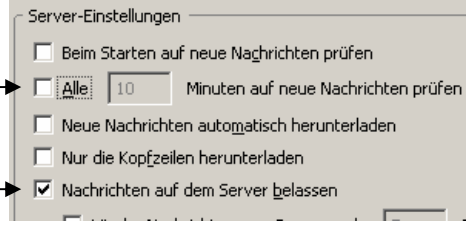
5. Markiere den erstellten Postausgangsserver und mache ihn durch einen Klick auf den Knopf „Standard setzen“ zum Standard-Postausgangsserver!
6. Wähle nun unter dem linksseitigen Menü den Knopf „Konto hinzufügen“ und die Option „E-Mail-Konto“!
7. Gib erneut deinen Vornamen und deine E-Mail-Adresse (**deinvorname@unsermailserver**) ein und bestätige mit „Weiter“:



8. - Wähle nun als Posteingangs-Server „POP“ aus,
   - gib als Posteingangs-Server **unsermailserver** ein,
   - entferne den Haken vor der Option „Globaler Posteingang“
   - und bestätige mit „Weiter“:

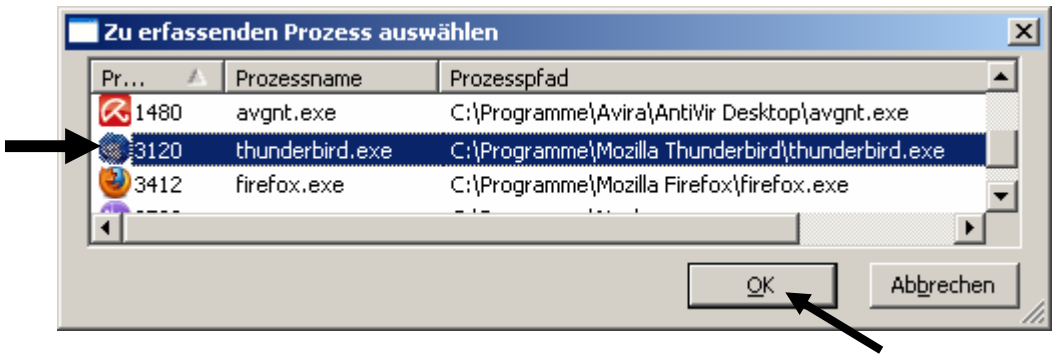


9. Gib deinen Benutzernamen beim Posteingangs-Server, also **deinen Vornamen**, ein!
10. Gib eine Konten-Bezeichnung ein und bestätige mit „Weiter“!
11. Schließe den Vorgang mit „Fertig stellen“ ab!
12. Wähle nun im linksseitigen Menü unter dem neu erstellten E-Mail-Konto den Unterpunkt „Server-Einstellungen“ aus!
    Damit keine wichtigen E-Mails verloren gehen solltest Du die Option „Nachrichten auf dem Server belassen“ aktivieren, die Option „Alle ___… Minuten auf neue Nachrichten prüfen“ sollte besser deaktiviert werden (siehe Abbildung rechts).

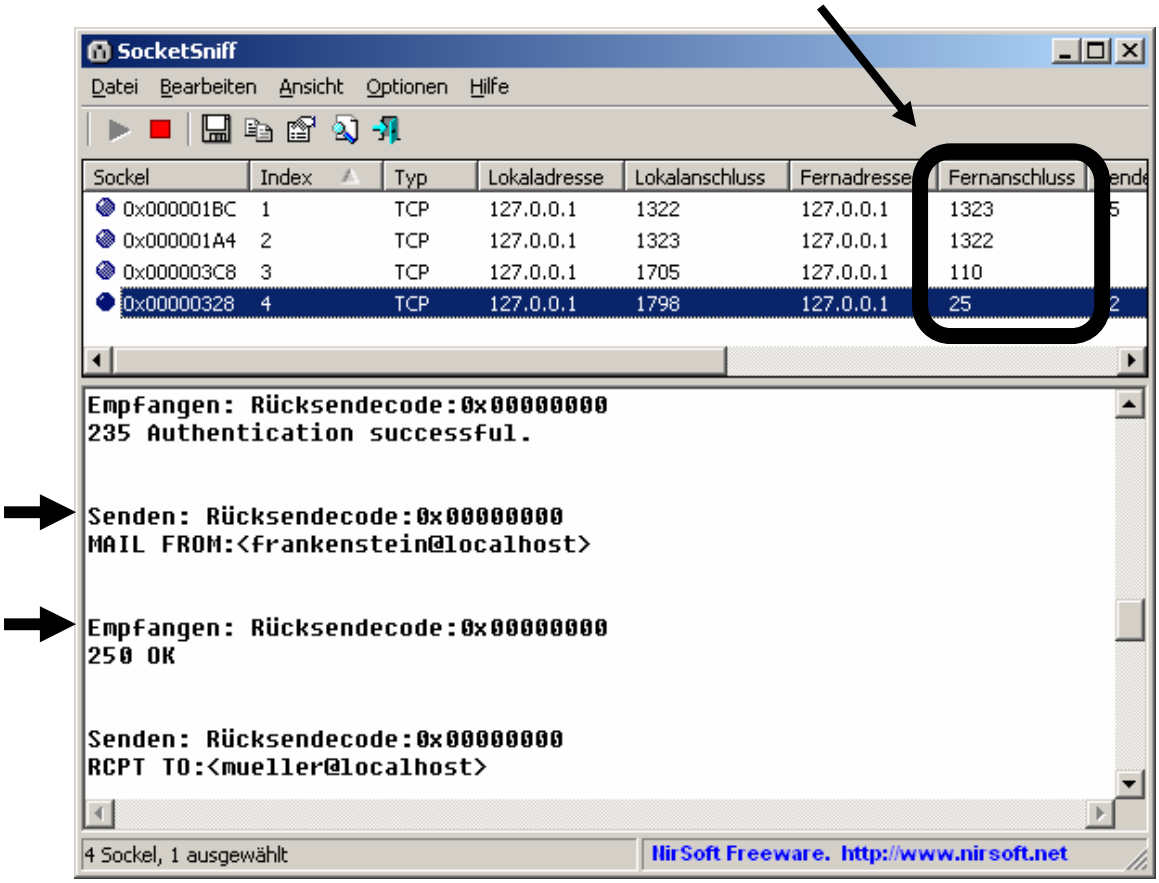


Anleitung: Netzwerk-Kommunikation mit *Socket Sniff* analysieren

1. Starte **zuerst** dein **E-Mail-Programm** (z.B. *Thunderbird*)!
2. Starte **dann** das Programm ***Socket Sniff***!
3. Beim Starten des Programms öffnet sich ein Fenster mit dem Titel „Zu erfassenden Prozess auswählen“. Wähle hier das gestartete E-Mail-Programm aus und bestätige mit „OK“:



4. Interessant ist nun vor allem die Kommunikation mit dem **Fernanschluss 25** zum Versenden von E-Mails und dem **Fernanschluss 110** zum Empfangen von E-Mails (Ein Fernanschluss wird im Englischen als **port** bezeichnet).
   Dabei sind vom Server empfangene Mitteilungen mit „Empfangen: “ markiert, von deinem Computer gesendete Mitteilungen dagegen mit „Senden: “.



Das *Post Office Protocol (POP3)* zum Empfangen von E-Mails

1. Konfiguriere dein Postfach in *Thunderbird* gemäß Anleitung, falls nicht bereits geschehen!
2. Starte die Erfassung des Netzwerkverkehrs von *Thunderbird* mit dem Netzwerkanalysewerkzeug *Socket Sniff* (siehe separate Anleitung zu *Socket Sniff*)!
3. Bitte jemanden, dir eine E-Mail zu senden und empfange diese E-Mail!
   Verwende dann das in *Socket Sniff* erstellte Protokoll des Netzwerkverkehrs, um die Kommunikation zwischen Server und Client zu rekonstruieren:
   Bringen Sie auf beiliegendem Interaktionsdiagramm die unten stehenden Nachrichten in die korrekte Reihenfolge und ordnen Sie den Schritten sinnvolle Bezeichnungen zu (ausschneiden und aufkleben oder in Felder eintragen und hier ausstreichen)!
   Hinweis:
   Angaben in spitzen Klammern (z.B. <benutzername> oder <Absender-Adresse>) werden jeweils durch gültige Einträge ersetzt (z.B. mschneider oder martin_schneider@web.de).
4. Vergleiche mit einem Experten für SMTP die beiden Protokolle!
   Notiert dabei Gemeinsamkeiten der beiden Protokolle, und leitet daraus typische Eigenschaften von Protokollen ableiten ab!

Passwort überprüfen

E-Mail herunterladen

Anzahl gespeicherter E-Mails ermitteln

Verbindung beenden

Benutzer anmelden

Index der letzten E-Mail ermitteln

```
LIST
```

```
STAT
```

```
+OK mailbox locked
<Anzahl gespeicherter
Nachrichten>
```

```
PASS <Passwort>
```

```
QUIT
```

```
+OK <Informationen
über den Mailserver>
```

```
RETR <Index der zu
empfangenden E-Mail>
```

```
+OK mailbox has
<Anzahl> messages
<Liste Index Größe>
```

```
+OK bye
```

```
+OK <Index und Größe
der letzten E-Mail>
```

```
+OK  <Aufforderung
Passwort zu senden>
```

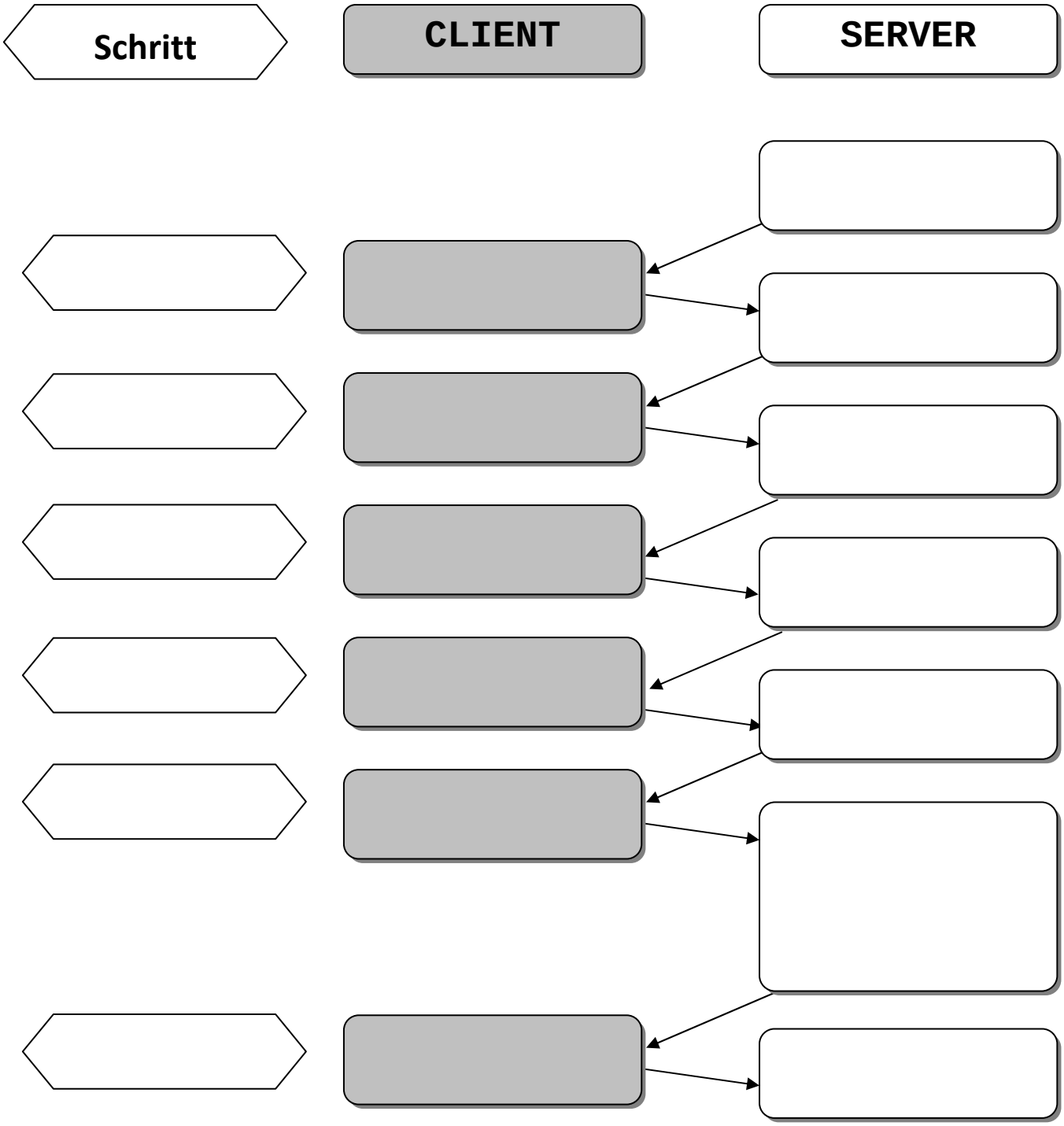
```
USER <Benutzername>
```

```
+OK message follows
Date: <Datum und Uhrzeit>
From: <Absender-Adresse>
To: <Empfänger-Adresse>
Subject: <Betreffzeile>

<Text der Nachricht
.
```

Das *Post Office Protocol (POP3)* zum Empfangen von E-Mails

- Interaktionsdiagramm -



Das *Simple Mail Transfer Protocol (SMTP)* zum Versenden von E-Mails

1. Konfiguriere dein Postfach in *Thunderbird* gemäß Anleitung, falls nicht bereits geschehen!
2. Starte die Erfassung des Netzwerkverkehrs von *Thunderbird* mit dem Netzwerkanalysewerkzeug *Socket Sniff* (siehe separate Anleitung zu *Socket Sniff*)!
3. Sende jemandem eine E-Mail zu! Verwende dann das in *Socket Sniff* erstellte Protokoll des Netzwerkverkehrs, um die Kommunikation zwischen Server und Client zu rekonstruieren:
   Bringe auf dem Interaktionsdiagramm die unten stehenden Nachrichten in die korrekte Reihenfolge und ordne den Schritten sinnvolle Bezeichnungen zu (ausschneiden und aufkleben oder in Felder eintragen und hier ausstreichen)!
   Hinweise:
   - Angaben in spitzen Klammern (z.B. <benutzername> oder <Absender-Adresse>) werden jeweils durch gültige Einträge ersetzt (z.B. mschneider oder martin_schneider@web.de).
   - Mit ***)** markierte Nachrichten sind in Base64 codiert und können z.B. auf http://decodebase64.com/ in eine von Menschen lesbare Darstellung umgewandelt werden.
4. Vergleiche mit einem Experten für POP3 die beiden Protokolle!
   Notiert dabei Gemeinsamkeiten der beiden Protokolle, und leitet daraus typische Eigenschaften von Protokollen ableiten ab!

Prüfe Empfängeradresse

Begrüßung

E-Mail versenden

Benutzer anmelden

Verbindung beenden

Prüfe Absenderadresse

Passwort überprüfen

```
From: <Absender-Adresse>
To: <Empfänger-Adresse>
Subject: <Betreff-Zeile>

<Text der Nachricht>
.
```

DATA

EHLO <Name des Client-Rechners>

250 <Bestätigung>

221 <Verabschiedung>

RCPT TO: <Empfängeradresse>

250 <Bestätigung>

220 <Informationen über den Mailserver>

<Passwort> *)

QUIT

334 Password: *)

250 <Bestätigung>

<Benutzername> *)

AUTH LOGIN

334 Username: *)

250 <Bestätigung>

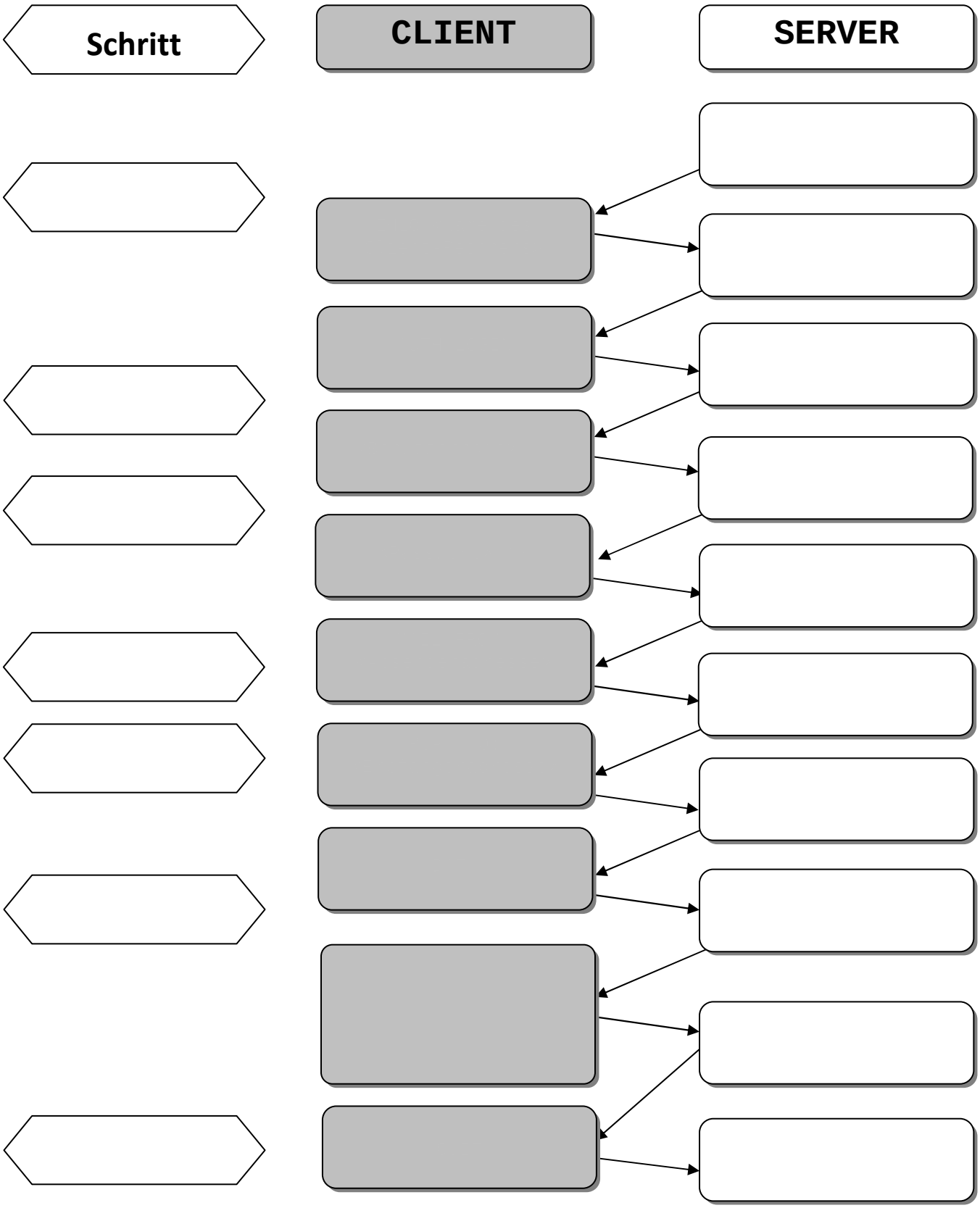
354 <Aufforderung E-Mail zu senden> <Endzeichen>

MAIL FROM: <Absenderadresse>

235 <Bestätigung der Anmeldung>

Das *Simple Mail Transfer Protocol (SMTP)* zum Versenden von E-Mails

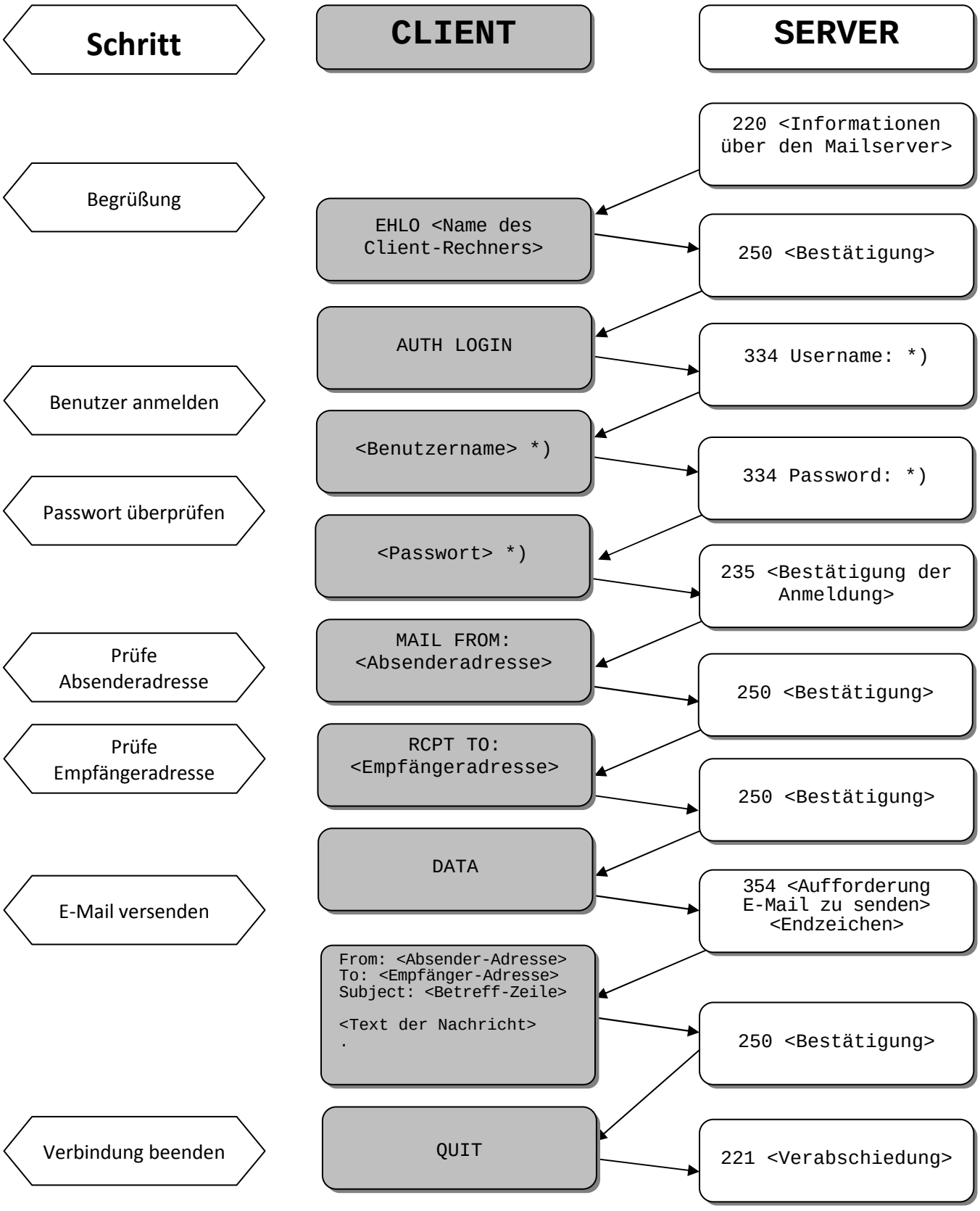
- Interaktionsdiagramm -



*) Diese Nachrichten sind in *Base64* codiert und können z.B. auf http://decodebase64.com/ wieder in eine von Menschen lesbare Darstellung umgewandelt werden.

Das *Simple Mail Transfer Protocol (SMTP)* zum Versenden von E-Mails

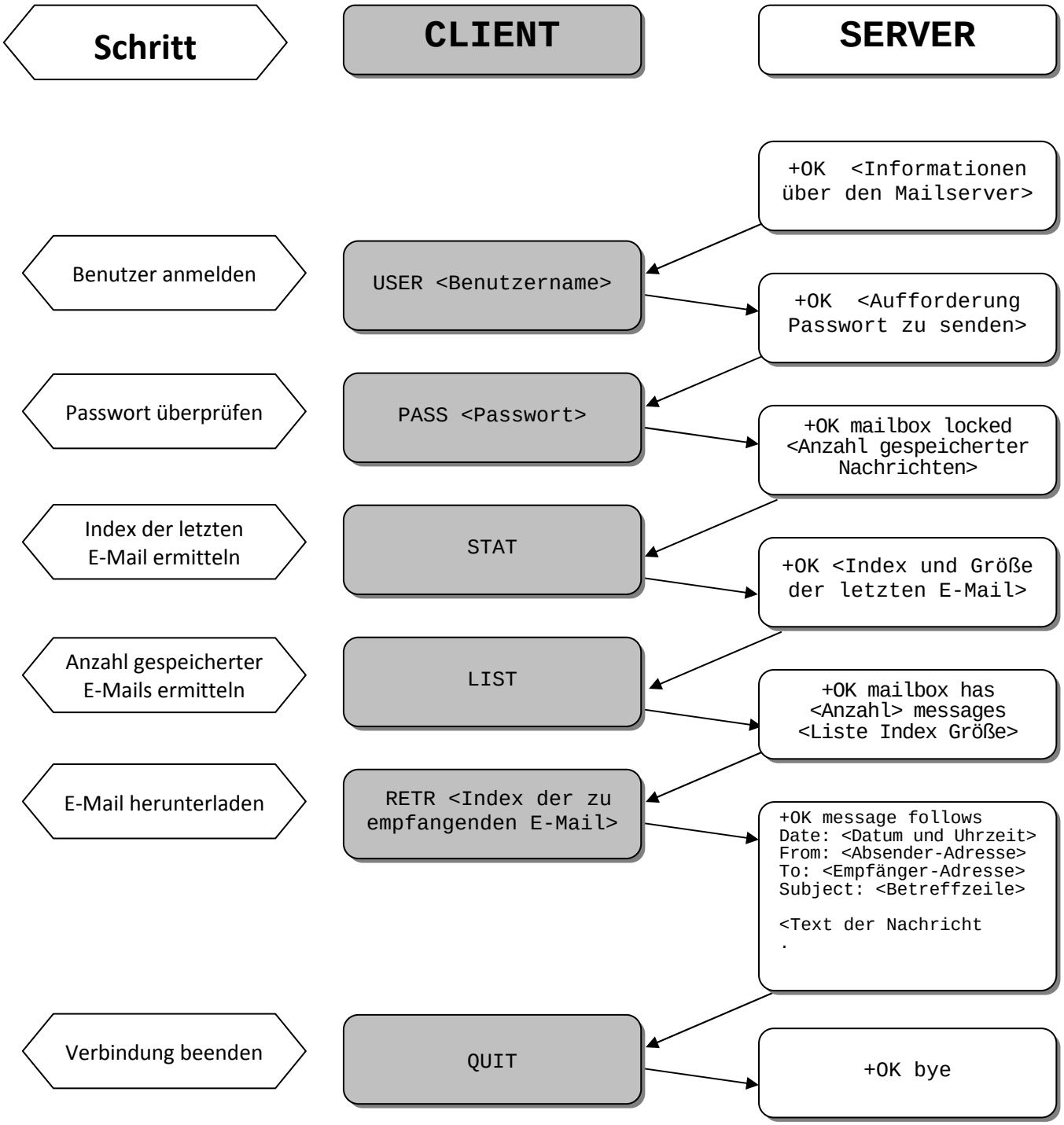
- Musterlösung -



*) Diese Nachrichten sind in *Base64* codiert und können z.B. auf http://decodebase64.com/ wieder in eine von Menschen lesbare Darstellung umgewandelt werden.

Das *Post Office Protocol (POP3)* zum Empfangen von E-Mails

- Musterlösung -



Unterhaltung mit einem E-Mail-Server via Telnet

Man kann sich auch ohne E-Mail-Programm direkt mit einem E-Mail-Server „unterhalten“. Dazu ist z.B. das Programm *Telnet* geeignet, das bereits Bestandteil vieler Betriebssysteme ist. Unter Windows kannst Du wie folgt vorgehen:

1. Starte die Eingabeaufforderung! In Windows geht das z.B. mit Start >> Ausführen >> „cmd“ eingeben oder Start >> Programme >> Zubehör >> Eingabeaufforderung
2. Gib den Befehl „telenet“ ein und bestätige mit der Eingabetaste (Return)!
3. Gib den Befehl „open“ gefolgt vom Namen des Servers sowie der Portnummer des E-Mail-Servers (110 für POP3, 25 für SMTP) ein, also z.B. „open pc-r104-13 110“!
4. Nun kannst du wiederholt Befehle des Protokolls an den E-Mail-Server senden und die Antworten lesen. Versuche, gemäß POP3-Protokoll ein E-Mail aus deinem Postfach anzeigen zu lassen! Zum Versenden der E-Mails müssen Username und Passwort nach dem *Base64*-Verfahren kodiert werden, z.B. auf http://decodebase64.com/.
5. Wenn Du genug hast, schließe die Verbindung indem Du „c“ eingibst!
6. Beende das Programm *Telnet* mit der Eingabe „q“!
7. Schließe die Eingabeaufforderung mit der Eingabe „exit“!

Unterhaltung mit einem E-Mail-Server via Telnet

Man kann sich auch ohne E-Mail-Programm direkt mit einem E-Mail-Server „unterhalten“. Dazu ist z.B. das Programm *Telnet* geeignet, das bereits Bestandteil vieler Betriebssysteme ist. Unter Windows kannst Du wie folgt vorgehen:

1. Starte die Eingabeaufforderung! In Windows geht das z.B. mit Start >> Ausführen >> „cmd“ eingeben oder Start >> Programme >> Zubehör >> Eingabeaufforderung
2. Gib den Befehl „telenet“ ein und bestätige mit der Eingabetaste (Return)!
3. Gib den Befehl „open“ gefolgt vom Namen des Servers sowie der Portnummer des E-Mail-Servers (110 für POP3, 25 für SMTP) ein, also z.B. „open pc-r104-13 110“!
4. Nun kannst du wiederholt Befehle des Protokolls an den E-Mail-Server senden und die Antworten lesen. Versuche, gemäß POP3-Protokoll ein E-Mail aus deinem Postfach anzeigen zu lassen! Zum Versenden der E-Mails müssen Username und Passwort nach dem Base64-Verfahren kodiert werden, z.B. auf http://decodebase64.com/.
5. Wenn Du genug hast, schließe die Verbindung indem Du „c“ eingibst!
6. Beende das Programm *Telnet* mit der Eingabe „q“!
7. Schließe die Eingabeaufforderung mit der Eingabe „exit“!

E-Mail (nur?) für Dich – eine Unterrichtsreihe des Projekts *Informatik im Kontext*

Gefahren bei der Kommunikation über das Internet erkennen

In diesem Lernschritt werden die Schülerinnen und Schüler **in einer fiktiven Situation verschiedene reale Gefahren** der Kommunikation im Internet **erleben**. Dazu senden sie sich – wie bereits im Lernschritt zur Rekonstruktion der Protokolle SMTP und POP3 – fiktive Nachrichten über einen auf dem Lehrerrechner (oder einem anderen Rechner mit angeschlossenem Projektor) E-Mail-Server.

Um einen „man-in-the-middle"-Angriff realistisch demonstrieren zu können wird empfohlen, einen Standardrechner als Vermittlungsrechner (Router) umzubauen (siehe separate Anleitung „Einen Windows-Rechner zum Router machen"). Alternativ kann ein ähnlicher Angriff auf dem Lehrerrechner simuliert werden, nur ist die Trennung zwischen den Rollen Kommunikations-infrastruktur und Diensterbringer dann nicht mehr deutlich erkennbar. In jedem Fall reichen die Mittel von *Socket Sniff* nicht aus, um Passwörter, die Schülerinnen und Schüler für die fiktiven E-Mail-Konten nutzern, auszuspähen, da der E-Mail-Server *Hamster* diese ausblendet und eine Aufzeichnung der Kommunikation des Hamsters mit *Socket Sniff* nicht möglich scheint. Es wird daher empfohlen, dass Lehrende auf die Verwendung von *Wireshark* zurückgreifen.

Vorbereitung

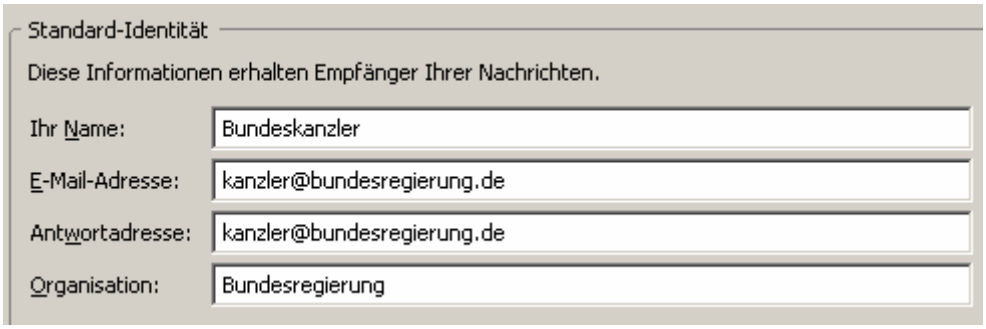
Vor Unterricht sollten Lehrende ggf. den Router vorbereiten (siehe separate Anleitung) und das spätere Versenden einer E-Mail mit gefälschten Absender-Angaben wie folgt vorbereiten:

a. Einen weiteren Schülerrechner starten und sich an dem Rechner anmelden.

b. Einen E-Mail-Client (z. B. *Thunderbird*) starten und für einen gültigen Benutzer des E-Mail-Servers einrichten (siehe Anleitung „E-Mail-Client *Thunderbird* einrichten").

c. Die Einstellungen zur Identität abändern:

- Menu „Extras > Konten" wählen.
- Die Angaben zur Standardidentität ändern, z. B.:



d. Eine E-Mail mit einer vermeintlich vertrauensvollen Information (z. B. Ankündigung einer bevorstehenden Privatisierung eines Staatsunternehmens mit Empfehlung eines Aktienkaufs vor Veröffentlichung des Vorhabens) an den Kurs schreiben und speichern (NOCH NICHT SENDEN – die Postfächer werden ja erst noch erstellt!). Dabei können als Email-Adressen jeweils <vornameDesSchuelers>@<RechnernameDesServers> angenommen werden.

Durchführung

Das folgende **Szenario** lässt sich in ca. 45 Minuten durchspielen. Für eine Sicherung der Sicherheitsanforderungen an Kommunikation Vertraulichkeit, Integrität und Authentizität ist zusätzliche Zeit zu veranschlagen.

1. Benutzer für die Kursteilnehmer neu einrichten, dabei sollten die Schüler ihre Passwörter selbst eingeben (z. B. durch Herumreichen der Tastatur) – die Schülerinnen und Schüler sollten zuvor darauf hingewiesen werden, nicht Passwörter zu wählen, die sie auch für andere Zugänge nutzen. **Warum?** Die Passwörter werden ja später in den POP3-Nachrichten ausgelesen – das wiederum den Schülerinnen und Schüler noch nicht sagen!

Vortäuschen einer falschen Identität auf Client-Seite

2. Die Schülerinnen und Schüler auffordern, die Einstellungen ihrer E-Mail-Clients ggf. anzupassen und sich einige E-Mails zu schicken.

    a. Dabei die unter falscher Identität erstellte E-Mail an den Kurs versenden.

    b. Anschließend mit *Wireshark* auf dem „Router“ das Passwort eines Schülers „abfangen“.

3. Schülerinnen und Schüler im Plenum versammeln und nach erhaltenen E-Mails befragen („Hat alles geklappt?“, „Irgendwelche Probleme oder Unregelmäßigkeiten?“).

    a. Die offensichtlich unautorisierte E-Mail wird zum Anlass genommen, festzustellen: „Hier passieren Dinge, die so nicht passieren sollten!“

    b. Vorschau: „Ich werde jetzt verschiedene Dinge machen.“

    c. Beobachtungsauftrag: „Notiert:
    Welche Gefahren bei der Kommunikation über das Internet lassen sich beobachten?“

4. [Bildschirmprojektion des vorbereiteten Schülerrechners]
    Die Einstellungen zur falschen Identität zeigen (siehe Vorbereitung Punkt 3.c).

    a. „Wie lässt sich feststellen, dass es sich um eine falsche Identität handelt?“
    [unglaubwürdig, unpersönlich: der Verfasser weiß nichts über mich.]
    „Ja, besser wäre, ich wüsste mehr über die Benutzer der Postfächer ...“

Mitlesen vertraulicher Informationen auf dem Kommunikationsweg

    b. Ein weiteres Konto mit den Benutzerdaten des Schülers, dessen Passwort Sie unter Punkt 2.b ermittelt haben, einrichten, E-Mails anzeigen lassen und eine sehr „unhöfliche“ Antwort verfassen.
    “Welche Folgen könnte mein Handeln haben?“
    [Streit, der Empfänger nimmt die Nachricht ernst]
    “Wie könnte ich auf das Passwort gekommen sein?“
    [Verbindung abgehört ...]

    c. [Bildschirmprojektion des „Routers“]
    Das Passwort im Protokoll von *Wireshark* zeigen.

Manipulation auf dem E-Mail-Server

5. [Bildschirmprojektion des „Servers“]

    a. Einen Schüler auffordern, sich mit einem anderen per Email zu verabreden.

    b. Die entstandene .msg-Datei im Ordner Mails\<benutzername> in einem Texteditor öffnen und Zeit und Ort der Verabredung ändern.

    c. Den Empfänger auffordern bitten, sein Postfach auf neue E-Mails zu prüfen.
    “Was machst Du zum verabredeten Termin?“
    [Ich stehe am falschen Ort.]
    “Welche Folgen könnte mein Handeln haben?“
    [Streit, der Empfänger glaubt nicht an eine Manipulation]
    “Wenn sich alle Mitarbeiter des E-Mail-Servers korrekt verhalten – ist dann eine Manipulation ausgeschlossen?“
    [Der Server kann von Fremden gehackt werden.]

Ergebnissicherung

6. Abschließend sollten die konkreten Beobachtungen verallgemeinert werden.

    a. „Was konntet Ihr beobachten?“ Dabei an der Tafele sammeln:

    - **Nachrichten mitlesen**
    - **Nachrichten verändern**
    - **Nachrichten unter falscher Identität verfassen**

    b. „Was sollte man also sicherstellen?“ Dabei an der Tafele ergänzen:

    - **Keiner kann** Nachrichten mitlesen**.**
    - **Keiner kann** Nachrichten verändern**.**
    - **Keiner kann** Nachrichten unter falscher Identität verfassen**.**

    c. „Dafür hat man auch Begriffe gefunden.“ An der Tafele ergänzen:

<u>Anforderungen an eine sichere Kommunikation:</u>

- **Vertraulichkeit:** Keiner kann Nachrichten mitlesen.
- **Integrität:** Keiner kann Nachrichten verändern.
- **Authentizität:** Keiner kann Nachrichten unter falscher Identität verfassen.

Vorschau: „Wir werden zunächst untersuchen, wie sich Vertraulichkeit herstellen lässt! Dieses Problem gibt es nicht erst, seitdem es Computer gibt, die Geheimhaltung von Nachrichten war schon vor tausenden Jahren ein brisantes Thema.“

Anleitung: Einen Windows-Rechner zum Router machen

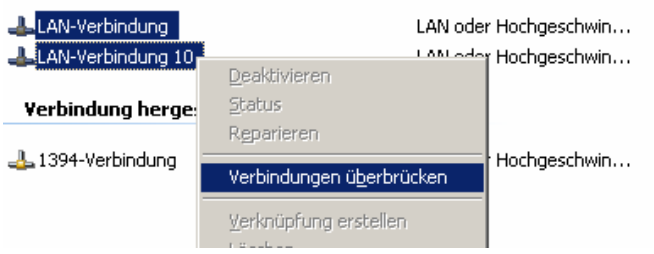
Nachrichten, die zwischen Client und Server verschickt werden, sind auch auf allen Rechnern, die die Nachricht weiterleiten, lesbar und ermöglicht so „man-in-the-middle“-Angriffe. Um dies zu demonstrieren, empfiehlt es sich, einen Standardrechner als Vermittlungsrechner (Router) umzubauen. Dazu werden benötigt:

- eine weitere Netzwerkkarte
- ein Cross-Link-Netzwerkkabel
  (auch als „Kreuzkabel“ oder „cross over“ bekannt, i.d.R. mit dem Symbol X am Stecker markiert)
- optional: ein VGA-Verlängerungskabel um Bildschirme von Schülerrechnern auf dem Projektor zeigen zu können.

Der vorübergehende Umbau des „Routers“ bedarf einer gewissenhaften Vorbereitung, die aufgrund der Verschiedenheiten in Schulen eingesetzter lokaler Netzwerke im Vorfeld der Stunde getestet werden sollte. Bei Erfolg (d.h. wenn man mit den Erfahrungen des Tests „weiß was man tut“) lässt sich die Situation im Zeitraum einer großen Pause aufbauen, der Abbau erfolgt bei selbstheilenden Systemen (z. B. Rembo/MySHN oder Dr. Kaiser-Karten) einfach durch Ausbau der Karte am „Router“, Wieder-Einstecken des ursprünglichen Netzwerkkabels am „Server“ und Neustart der beteiligten Rechner. Für die vorzunehmenden Konfigurationen sind evt. **Administratorrechte** notwendig!

Die folgende Anleitung zur Vorbereitung ist erscheint vielleicht erschreckend umfangreich, soll aber nur jeden Schritt möglichst präzise beschreiben – probieren Sie es aus!

1. Einen Standard-Schülerrechner zum Router umbauen:
   a. Den Rechner starten, am Rechner mit Administratorrechten anmelden und „Systemsteuerung > Netzwerkverbindungen“ öffnen.
   b. Mit rechtem Mausklick auf die Standardnetzwerkverbindung das Kontextmenü der Verbindung aufrufen und dort „Status“ wählen.
   c. Vom Reiter „Netzwerkunterstützung“ IP-Adresse, Subnetzmaske und Standardgateway notieren.
   d. Den Rechner herunterfahren, öffnen und eine zweite Netzwerkkarte einbauen.
   e. Den Rechner starten, ggf. den Versuch über die neue Netzwerkkarte zu booten (BOOTTP ...) mit der Leertaste unterbrechen, am Rechner mit Administratorrechten anmelden und „Systemsteuerung > Netzwerkverbindungen“ öffnen.
   f. Beide Netzwerkverbindungen markieren und über das Kontextmenü die „Verbindungen überbrücken“:



g. Über das Kontextmenü der entstandenen Netzwerkbrücke die „Eigenschaften“ aufrufen, einen Doppelklick auf dem Element „Internetprotokoll (TCP/IP)“ ausführen, die Gruppe „Folgende IP-Adresse verwenden“ aktivieren und dort die zuvor notierten Einstellungen ergänzen.
Warum? Die Netzwerkbrücke erhält neue Netzwerkeinstellungen, die evt. nicht zu den Einstellungen in Ihrem Schulnetzwerk passen.

h. *Wireshark* starten und als Filter „pop“ anwenden (siehe Anleitung zu *Wireshark*).

2. Den „Server“ (Lehrerrechner, auf dem der E-Mail-Server gestartet wird) über den „Router“ ans Netzwerk anschließen:

   a. Das bisherige Netzwerkkabel des „Server“ entfernen und den Rechner durch ein **Cross-Link**-Netzwerkkabel mit der neu eingebauten Netzwerkkarte des „Routers“ verbinden.

   b. Den „Server“ starten und mit Administratorrechten anmelden.

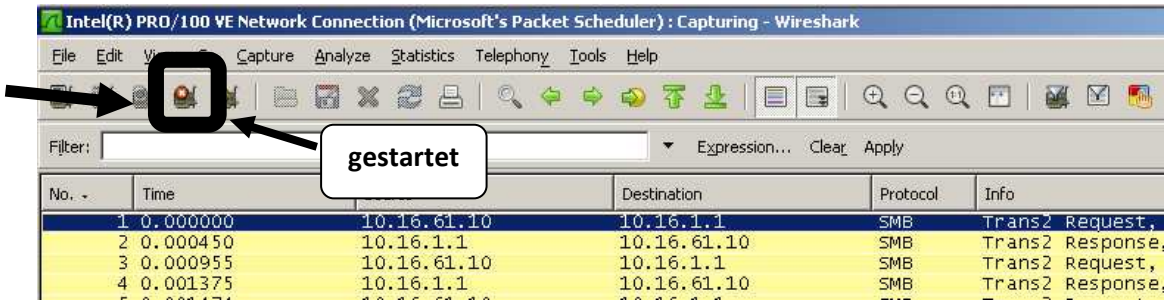
   c. Den E-Mail-Server starten.

Anleitung: Netzwerk-Kommunikation mit *Wireshark* analysieren

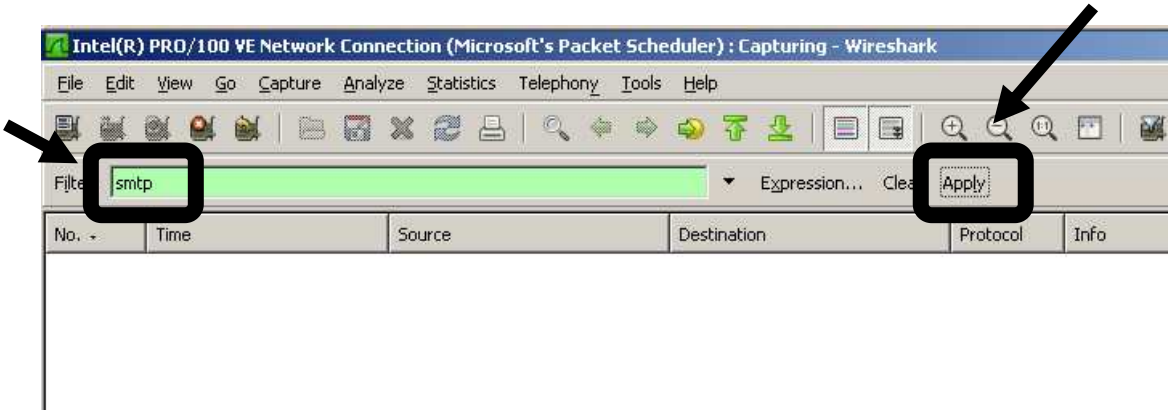
1. Starte *Wireshark*!

2. Klicke die Netzwerkkarte deines Computers (Network **Interface**) an!

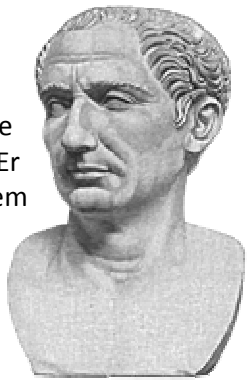
3. Falls die Aufnahme nicht automatisch beginnt, starte die Aufnahme über den Menüpunkt „Capture“ > „Start“ manuell!



4. Gib im Filter direkt unterhalb der Menüleiste **smtp** oder **pop** ein und bestätige mit „Apply“!



Das Geheimnis des römischen Kaisers Caesar



Der römische Feldherr und Kaiser Gaius Julius Caesar (100 – 44 v. Chr.) verschlüsselte angeblich seine geheimen militärischen Botschaften nach folgendem Verfahren: Er verwendete ein Geheimtextalphabet (GTA), welches um drei Stellen gegenüber dem Klartextalphabet (KTA) verschoben war. Jeder Buchstabe des Klartextes wurde durch einen Buchstaben aus dem Geheimtextalphabet ersetzt:

Klartext: E IN STRENG GEHEIMER TEXT

KTA	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
GTA	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Geheimtext: HLQVWUHQJJHKHLPHUWHAW

Vermutlich verwendete Caesar die Verschiebung um genau drei Stellen, weil sein Name mit dem dritten Buchstaben des Alphabetes - dem C - begann. Kaiser Augustus (31 v. Chr. - 14 n. Chr.) soll dagegen die Verschiebung um eine Stelle bevorzugt haben. Zur höheren Sicherheit verwendet man nur Großbuchstaben, keine Umlaute und lässt Leerzeichen, Satzzeichen usw. weg.

Um einen Text zu verschlüsseln, legt man 2 Doppelstreifen übereinander und verschiebt den unteren Streifen entsprechend (siehe Abbildung oben). So lässt sich für jedes Zeichen des Klartexts das entsprechende Zeichen des Geheimtextalphabets auf dem unteren Streifen ablesen.

Zum Entschlüsseln suche den Buchstaben auf dem unteren Streifen und lese den Buchstaben des Klartexts auf dem oberen Streifen ab.

Aufgaben

1. Tausche mit einem Partner kleine Geheimnisse per E-Mail aus. Damit niemand außer euch beiden die Geheimnisse erfährt, solltet ihr die E-Mails nach dem Cäser-Verfahren verschlüsseln. Gehe dabei wie folgt vor:
   a. Vereinbare zu allererst mit deinem Partner, um wie viele Buchstaben ihr die Alphabete verschieben wollt (→ „Abstand zwischen KTA und GTA“)!
   b. Denke dir ein kleines Geheimnis aus und schreibe es in einem kurzen Satz auf!
   c. Verschlüssle den Satz mit Hilfe der Albertischeibe und sende den verschlüsselten Satz deinem Partner in einer E-Mail.
   d. Entschlüssle das Geheimnis deines Partners, das er dir in einer E-Mail geschickt hat, mit Hilfe der Albertischeibe!
2. *Georg der Gangster* hat im Internet folgende vermutlich mit dem Caesar-Verfahren verschlüsselte Nachricht abgefangen: MGLRYXDIGEIWEV
   a. Versuche, den Code auch ohne Kenntnis des Abstands zwischen KTA und GTA zu knacken! *Hinweis:* Probiere doch einfach mal ein paar Abstände mit einem Teil der Nachricht aus!
   b. Wie sicher ist die Verschlüsselung nach dem Caesar-Verfahren? Begründe deine Antwort! *Hinweis:* Wie viele verschiedene Geheimtextalphabete gibt es mit diesem Verfahren?
3. Knobelaufgabe für Schnelle: Angenommen, die Buchstaben des Geheimtextalphabets werden in beliebiger Reihenfolge zugeordnet, z.B. folgendermaßen:

KTA	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
GTA	G	Q	H	C	D	U	K	O	X	A	L	P	F	S	J	B	T	Z	R	I	E	N	V	Y	W	M

   Wie würde sich diese Veränderung auf die Sicherheit der Verschlüsselung auswirken? *Hinweis:* Wie viele verschiedene Geheimtextalphabete kann es bei diesem Verfahren geben?

Das Geheimnis des römischen Kaisers Caesar - Lösung

2. *Georg der Gangster* hat im Internet folgende vermutlich mit dem Caesar-Verfahren verschlüsselte Nachricht abgefangen: KEJPWVBGECGUCT

   a. Versuche, den Code auch ohne Kenntnis des Abstands zwischen KTA und GTA zu knacken!

      Durch systematisches Ausprobieren sämtlicher möglichen Abstände mit den ersten Zeichen der Nachricht führt beim Abstand 4 zum Erfolg, die Nachricht beginnt mit ICH …

      Die entschlüsselte Nachricht lautet: ICH NUTZE CAESAR

   b. Wie sicher ist die Verschlüsselung nach dem Caesar-Verfahren? Begründe deine Antwort!
      Hinweis: Wie viele verschiedene Geheimtextalphabete gibt es mit diesem Verfahren?

      Das Verfahren ist sehr unsicher. Durch Ausprobieren aller 25 möglichen Schlüssel mit den ersten Zeichen der Nachricht lässt sich eine mit dem Caesar-Verfahren verschlüsselte Nachricht mit vertretbarem Aufwand entschlüsseln.

3. Angenommen, die Buchstaben des Geheimtextalphabets werden in beliebiger Reihenfolge zugeordnet, z.B. :

KTA	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
GTA	G	Q	H	C	D	U	K	O	X	A	L	P	F	S	J	B	T	Z	R	I	E	N	V	Y	W	M

   Wie würde sich diese Veränderung auf die Sicherheit der Verschlüsselung auswirken?
   Hinweis: Wie viele verschiedene Geheimtextalphabete gibt es mit diesem Verfahren?

   Das Verfahren ist wesentlich sicherer. Durch die beliebige Reihenfolge der Zeichen gibt es viel mehr mögliche Geheimtextalphabete. Die genaue Anzahl möglicher Geheimtextalphabete beträgt 26! („26 Fakultät“): Für die Wahl des ersten Zeichens stehen 26 Zeichen zur Verfügung. Für die Wahl des nächsten Zeichens verbleiben 25 mögliche Zeichen, allein für die Festlegung der ersten beiden Zeichen gibt es also 26*25 mögliche Kombinationen. Für das nächste Zeichen können die 26*25 Kombinationen mit einem von 24 weiteren Zeichen kombiniert werden usw. Für 26 Zeichen entstehen so 26 * 25 * 24 * ... * 3 * 2 * 1 = 26! = 403.291.461.126.605.635.584.000.000 Kombinationen. Ein systematisches Ausprobieren aller Schlüssel ist so nicht mehr möglich!

Informationen zur Häufigkeit verschiedener Zeichen(-kombinationen)

Durchschnittliche Häufigkeit der Zeichen:

Zeichen	*englisch*	*deutsch*
a	8,04 %	6,47 %
b	1,54 %	1,93 %
c	3,06 %	2,68 %
d	3,99 %	4,83 %
e	12,51 %	17,48 %
f	2,30 %	1,65 %
g	1,96 %	3,06 %
h	5,49 %	4,23 %
i	7,26 %	7,73 %
j	0,16 %	0,27 %
k	0,67 %	1,46 %
l	4,14 %	3,49 %
m	2,53 %	2,58 %

Zeichen	*englisch*	*deutsch*
n	7,09 %	9,84 %
o	7,60 %	2,98 %
p	2,00 %	0,96 %
q	0,11 %	0,02 %
r	6,12 %	7,54 %
s	6,54 %	6,83 %
t	9,25 %	6,13 %
u	2,71 %	4,17 %
v	0,99 %	0,94 %
w	1,92 %	1,48 %
x	0,19 %	0,04 %
y	1,73 %	0,08 %
z	0,09 %	1,14 %

Häufige Bigramme:

deutsch	*Häufigkeit*
en	3,88 %
er	3,75 %
ch	2,75 %
te	2,26 %
de	2,00 %
nd	1,99 %
ei	1,88 %
ie	1,79 %
in	1,67 %
es	1,52 %

englisch	*Häufigkeit*
th	3,15 %
he	2,51 %
an	1,72 %
in	1,69 %
er	1,54 %
re	1,48 %
on	1,45 %
es	1,45 %
ti	1,28 %
at	1,24 %

Häufige Trigrammen:

deutsch	*Häufigkeit*
ein	1,22 %
ich	1,11 %
nde	0,89 %
die	0,87 %
und	0,87 %
der	0,86 %
che	0,75 %

englisch	*Häufigkeit*
the	3,53 %
ing	1,11 %
and	1,02 %
ion	0,75 %
tio	0,75 %
ent	0,73 %
ere	0,69 %

Häufige Viergramme im Deutschen:

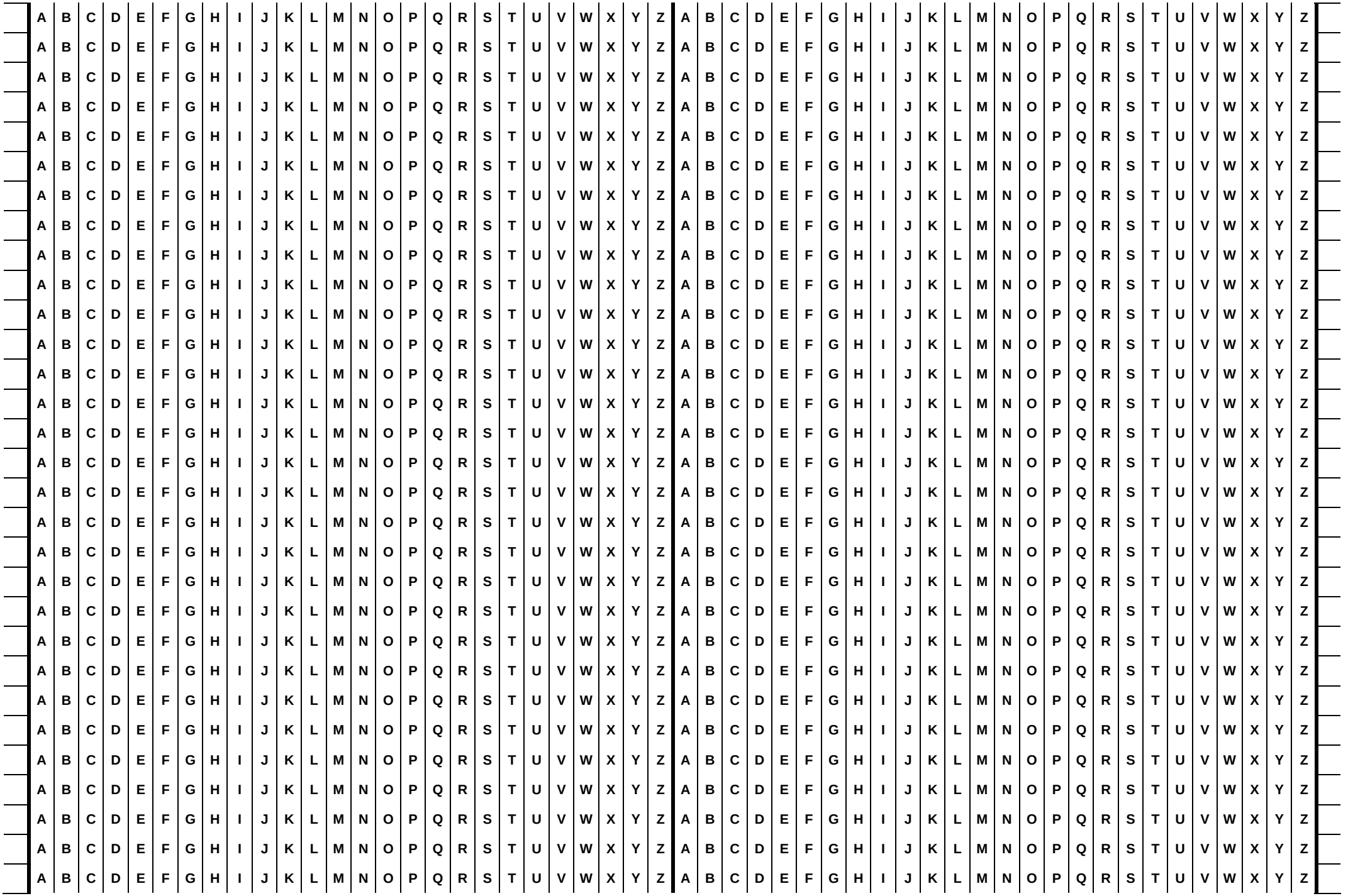
icht, keit, heit, chon, chen, cher, urch, eich

Mittlere Wortlänge und die zehn häufigsten Wörter in verschiedenen Sprachen:

Sprache	*mittlere Wortlänge*	*häufigste Wörter*
deutsch	5,9	**die, der, und, den, am, in, zu, ist, dass, es**
englisch	4,5	**the, of, and, to, a, in, that, it, is, I**
französisch	4,4	**de, il, le, et, que, je, la, ne, on, les**
italienisch	4,5	**la, di, che, il, non, si, le, una, lo, in**
spanisch	4,4	**de, la, el, que, en, no, con, un, se, sa**
russisch	6,3	**и, в, не, он, на, я, что, тот, быть, с**

Quelle: F. L. Bauer: Entzifferte Geheimnisse – Methoden und Maximen der Kryptologie. Springer 1995, S. 223.

30 Doppel-Alphabetstreifen für 15 Arbeitsgruppen



LOESEFOLGENDEKRYPTOGRAMME
ALLESINDCAESARVERSCHLUESSELT

EPPIVERJERKMWXWGLAIV

XZCRPYDEFYOSLEMWPTTXSTYEPCY

XEBDYDOPSCMROCMRGSWWOXWSDNOWCDBYW

JNYGUREENGURANHBOREFPUHYR

HMENQLZSHJLZBGSROZRR

BDASDMYYUQDQZUEFEOTIQD

FKGGTFGKUVGKPGUEJGKDG

STGBDCSXHIPJHZPTHT

MGLOEQMGLWELMGLWMIKXI

WBKUSAMKDISXTKRYIJABQIIU

Häufigkeitsanalyse mit Vigenère verhindern

Blaise de Vigenère

1. Wie wir festgestellt haben, sind monoalphabetische Substitutionsverfahren bei steigender Textlänge anfällig gegenüber einer Häufigkeitsanalyse. Überlege, wie man eine Häufigkeitsanalyse erschweren könnte!
2. Der französische Kaufmann *Blaise de Vigenère* hat das Problem im 16. Jh. gelöst. Versuche anhand einer Demonstration die Idee des Verfahrens nachzuvollziehen:
   a. Öffne das Programm „Krypto“ und gibt einen Satz in das Klartext-Feld ein!
   b. Wähle im Menü Demos den Menüpunkt Vigenère-Verschlüsselung aus!
   c. Klicke nun wiederholt auf den Knopf „Buchstaben verschlüsseln“!
   d. Beschreibe die Vorgehensweise des Verfahrens!
3. Vereinbare mit deinem Partner ein Geheimwort und sendet euch mit dem Vigenère-Verfahren verschlüssle E-Mails zu!
4. Überprüfe dein Postfach! Die E-Mail mit dem Betreff „Haben Sie Erkenntnisse über diese Person?“ ist offenbar nicht an dich gerichtet und nur aus versehen falsch adressiert worden. Aber ein Verschlüsselter Text birgt (fast) immer ein Geheimnis … Schade nur, dass du das Schlüsselwort nicht kennst!
   a. Öffne das Programm *Cryptool*, kopiere den Inhalt der E-Mail in ein neues Dokument und wähle das Menü *Analyse >> Symmetrische Verschlüsselung (klassisch) >> Chyphertext Only >> Vigenère*!
   b. Überlege: Wie lässt sich eine mit dem Vigenère-Verfahren verschlüsselte Chiffre auch ohne Kenntnis des Schlüssels knacken, wenn die Länge des Schlüsselwortes bekannt ist?
   c. Welche Bedingung müsste erfüllt sein, um einen solchen Angriff unmöglich zu machen?
5. Recherchiere: Was besagt das *Kerckhoffs’sche Prinzip*?

Häufigkeitsanalyse mit Vigenère verhindern

Blaise de Vigenère

1. Wie wir festgestellt haben, sind monoalphabetische Substitutionsverfahren bei steigender Textlänge anfällig gegenüber einer Häufigkeitsanalyse. Überlege, wie man eine Häufigkeitsanalyse erschweren könnte!
2. Der französische Kaufmann *Blaise de Vigenère* hat das Problem im 16. Jh. gelöst. Versuche anhand einer Demonstration die Idee des Verfahrens nachzuvollziehen:
   a. Öffne das Programm „Krypto“ und gibt einen Satz in das Klartext-Feld ein!
   b. Wähle im Menü Demos den Menüpunkt Vigenère-Verschlüsselung aus!
   c. Klicke nun wiederholt auf den Knopf „Buchstaben verschlüsseln“!
   d. Beschreibe die Vorgehensweise des Verfahrens!
3. Vereinbare mit deinem Partner ein Geheimwort und sendet euch mit dem Vigenère-Verfahren verschlüssle E-Mails zu!
4. Überprüfe dein Postfach! Die E-Mail mit dem Betreff „Haben Sie Erkenntnisse über diese Person?“ ist offenbar nicht an dich gerichtet und nur aus versehen falsch adressiert worden. Aber ein Verschlüsselter Text birgt (fast) immer ein Geheimnis … Schade nur, dass du das Schlüsselwort nicht kennst!
   a. Öffne das Programm *Cryptool*, kopiere den Inhalt der E-Mail in ein neues Dokument und wähle das Menü *Analyse >> Symmetrische Verschlüsselung (klassisch) >> Chyphertext Only >> Vigenère*!
   b. Überlege: Wie lässt sich eine mit dem Vigenère-Verfahren verschlüsselte Chiffre auch ohne Kenntnis des Schlüssels knacken, wenn die Länge des Schlüsselwortes bekannt ist?
   c. Welche Bedingung müsste erfüllt sein, um einen solchen Angriff unmöglich zu machen?
5. Recherchiere: Was besagt das *Kerckhoffs’sche Prinzip*?

E-Mail (nur?) für Dich – eine Unterrichtsreihe des Projekts *Informatik im Kontext*

Vigenère knacken

Die Länge des Schlüsselworts bestimmen

Die polyalphabetische Verschlüsselung nach dem Vigenère-Verfahren hat eine Schwäche: In jeder Sprache gibt es einige kurze Wörter und Buchstabenkombinationen, die recht häufig auftreten. Werden diese mit denselben Buchstaben des Geheimworts verschlüsselt, so wiederholen sich auch dieselben Zeichen im Geheimtext. Über die Analyse des Auftretens solcher Wiederholungen (im Folgenden **Parallelstellen** genannt) lässt sich ein mit dem Vigenère-Verfahren verschlüsselter Text knacken:

1. Finde weitere Parallelstellen in unten stehendem Geheimtext und ergänze die Liste!
2. Berechne jeweils den Abstand zwischen den Parallelstellen!
3. Berechne den größten gemeinsamen Teiler der Abstände der Parallelstellen!
   Der so berechnete Wert ist sehr wahrscheinlich die Länge des Schlüsselworts.

**PWTMYTBADKDGPWPFYWFGUESOTLUPNVYWAPKCSOO
JWWASTLSUZUSJMJBBRSTIMGPYSXOJWWASMMZQLC
HJQWGYDHKOJWWASTMFPADWIPVKLHONZWPDPWRAA
GQPRKNJCNPKGPJJLTHYOWOHPGYJWCUEKUZLGAOW
KHOGPESMZMRWPBKVFVZTQNLAGSFSMVWTDPWRAAG
QPRKNJCNPTGTKEOMSGVLYVCHKBVKLOFOBLGNCIV
XWPLYBZAAEOOWKEWEODZKZOGPWGOMSWMPWTIFFL
CTUTYGUOSLZSILYOHEWEODSRVVYHSFAVVHHWGIP
TGHYHCWJVLERGJWKPDHGJWTUTQNBXGZEUKTWIAZ
PPMOGPWGJQWGYDHKNJCNPSOVWTZPFOMNQUQFGOW
PYTQNBAIVOSXNSNZNVHMSPAHCXBWVDTFJRWFLAS
XAGPHYHCWJVLEOANWKUPTXIYGUFFSQLLHZRKZFG
PYTXIYGUOWKVAEOEAOBBCVOSXVWKUMSGVLYVCHK
BOGYOSTSGGUYSTAAPKYWIPLBBRSRIKULYJUVWKU
PFHMDKLMWMMFRLCGUVKQSWAGVVWYNVLZSILYROM
KKJSBAZSWMOWKHMILSCKZAIRPWZHMGPYSXLWTNC
IVXWPIPNOMZGUSSXIMUIPYUUEGUKICMDEOPFMZM
RWPGOMYGOZSXBOKLGWKTWHYLUKVEWZDAGVEKUOS
YBWPZDHKTDGUFBJEWNJSSLZSILYYUMFPAPAGVKV
LWZKV**

Parallelstelle:	**Abstand:**
OJWWAS	**28, 21**
T..	

Vermutete Länge des Schlüsselworts: ☐

Vigenère knacken

Das Schlüsselwort bestimmen

Überlege: Wie lässt sich bei bekannter Länge des Schlüsselworts (für diesen Text 7) das Schlüsselwort selbst aus dem Geheimtext herausfinden?

Hinweis: Die Verschlüsselung erfolgt für die verschiedenen Buchstaben nach dem Cäsar-Verfahren, das wir bereits geknackt haben …

1234567 1234567 1234567 1234567 1234567

PWTMYTB ADKDGPW PFYWFGU ESOTLUP NVYWAPK
CSOOJWW ASTLSUZ USJMJBB RSTIMGP YSXOJWW
ASMMZQL CHJQWGY DHKOJWW ASTMFPA DWIPVKL
HONZWPD PWRAAGQ PRKNJCN PKGPJJL THYOWOH
PGYJWCU EKUZLGA OWKHOGP ESMZMRW PBKVFVZ
TQNLAGS FSMVWTD PWRAAGQ PRKNJCN PTGTKEO
MSGVLYV CHKBVKL OFOBLGN CIVXWPL YBZAAEO
OWKEWEO DZKZOGP WGOMSWM PWTIFFL CTUTYGU
OSLZSIL YOHEWEO DSRVVYH SFAVVHH WGIPTGH
YHCWJVL ERGJWKP DHGJWTU TQNBXGZ EUKTWIA
ZPPMOGP WGJQWGY DHKNJCN PSOVWTZ PFOMNQU
QFGOWPY TQNBAIV OSXNSNZ NVHMSPA HCXBWVD
TFJRWFL ASXAGPH YHCWJVL EOANWKU PTXIYGU
FFSQLLH ZRKZFGP YTXIYGU OWKVAEO EAOBBCV
OSXVWKU MSGVLYV CHKBOGY OSTSGGU YSTAAPK
YWIPLBB RSRIKUL YJUVWKU PFHMDKL MWMMFRL
CGUVKQS WAGVVWY NVLZSIL YROMKKJ SBAZSWM
OWKHMIL SCKZAIR PWZHMGP YSXLWTN CIVXWPI
PNOMZGU SSXIMUI PYUUEGU KICMDEO PFMZMRW
PGOMYGO ZSXBOKL GWKTWHY LUKVEWZ DAGVEKU
OSYBWPZ DHKTDGU FBJEWNJ SSLZSIL YYUMFPA
PAGVKVL WZKV

Vigenère knacken

Das Schlüsselwort bestimmen

1. Führe mit einem Partner eine Häufigkeitsanalyse für alle mit dem ___. Buchstaben des Schlüsselworts verschlüsselten Zeichen durch und trage die Anzahl der Zeichen in die entsprechende Spalte ein, um so dass häufigste Zeichen zu bestimmen, das sehr wahrscheinlich dem Klartextzeichen **E** entspricht! Damit wäre der Abstand zum Klartextalphabet ermittelt, der Buchstabe 5 Positionen vor dem häufigsten Geheimtextzeichen steht dann für das **A** und ist der Buchstabe des Geheimworts.
2. Tauscht euch mit Paaren aus, die die Häufigkeitsanalyse für die anderen Buchstaben des Schlüsselworts durchgeführt haben, um so gemeinsam das Schlüsselwort zu bestimmen!
3. Ob eure Analyse richtig ist könnt ihr überprüfen, in dem ihr den Geheimtext in *Krypto 1.5* öffnet und mit dem von euch bestimmten Schlüsselwort entschlüsseln lasst!

Geheim-textzeichen	1. Buchstabe	2. Buchstabe	3. Buchstabe	4. Buchstabe	5. Buchstabe	6. Buchstabe	7. Buchstabe
A	𝍸						
B							
C	𝍸 III						
D	𝍸 III						
E	𝍸 II						
F	III						
G	I						
H	II						
I							
J							
K	I						
L	I						
M	III						
N	III						
O	𝍸 𝍸						
P	𝍸 𝍸 𝍸 𝍸						
Q	I						
R	II						
S	𝍸						
T	𝍸						
U	I						
V							
W	𝍸						
X							
Y	𝍸 𝍸 II						
Z	III						
Häufigster Buchstabe:	**P**						
Buchstabe im Schlüsselwort:	**L**						

Schlüsselwort: L ...

Asymmetrisch verschlüsseln ohne Austausch geheimer Informationen

Auch bei unknackbaren Verschlüsselungsverfahren besteht stets die Gefahr, dass jemand das geheime Schlüsselwort erfährt. Bevor Nachrichten verschlüsselt werden können müssen sich die Kommunikationspartner stets auf einen gemeinsamen Schlüssel einigen. Deshalb haben in der zweiten Hälfte des 20. Jahrhunderts einige Wissenschaftler erforscht, ob es nicht ein Verfahren geben kann, bei dem keine geheimen Schlüssel ausgetauscht werden müssen.

Stell dir folgende Situation vor:

Alice und Bob haben je ein Schloss mit Schlüssel.
Alice möchte Bob ein Geheimnis in einer verschlossenen Kiste übermitteln.

Wie kann Alice den Inhalt der Kiste sicher übermitteln, ohne Bob den Schlüssel für ihr Schloss zu geben?

Hinweis: Wie gesagt, *beide* haben je ein Schloss!

Vertraulichkeit mit asymmetrischer Kryptographie herstellen

Eine E-Mail lässt sich bekanntlich nicht mit einem Vorhängeschloss verschließen. Die in der zweiten Hälfte des 20. Jahrhunderts entwickelte **asymmetrische Kryptographie** ermöglicht es, Ver- und Entschlüsseln von Zahlen und Texten mit zwei verschiedenen Schlüsseln zu realisieren, so dass man nicht mehr geheime Schlüssel austauschen muss. Weil dieses Verfahren aufwendiger ist als symmetrische Verschlüsselung, wird es meist benutzt, um Schlüssel für eine symmetrische Verschlüsselung auf einem sicheren Weg auszutauschen. Viele Online-Shops und Online-Banking-Portale nutzen dieses Verfahren, um die Daten ihrer Kunden zu schützen.

Öffne die Animation „*Vertraulichkeit durch asymmetrische Kryptologie herstellen*“, mit der Du das Prinzip der asymmetrischen Kryptographie mit öffentlichen und privaten Schlüsseln kennen lernst!

Wenn die Animation geladen wird bist Du in der Rolle von Bob. Bob hat zwei Schlüssel: den vom Webseiten-Symbol ( ) umgeben **öffentlichen Schlüssel**, den er im Internet veröffentlicht hat, und den vom Tresor-Symbol ( ) umgeben **privaten Schlüssel**, den er sicher verwahrt und niemand anderem mitteilt. Über das Internet kann Bob auch den öffentlichen Schlüssel von Alice lesen, den sie dort veröffentlicht hat.

Will Bob Alice nun eine geheime Nachricht schreiben, die nur Alice wieder entschlüsseln kann, so sollte er seine Nachricht mit dem öffentlichen Schlüssel von Alice verschlüsseln. Helfe Bob dabei:

- Klicke dazu zunächst auf das Webseiten-Symbol ( ), das den öffentlichen Schlüssel von Alice umgibt, um diesen Schlüssel in die Felder "anzuwendender Schlüssel" auf Bobs Computer zu kopieren!
- Klicke dann im Bereich von Bobs Computer auf den Knopf "Schlüssel auf Nachricht anwenden", um die Nachricht mit dem zuvor kopierten Schlüssel zu verschlüsseln!
- Klicke nun im Bereich von Bobs Computer auf den Knopf "<<", um die verschlüsselte Nachricht über das Internet an Alice zu versenden!

Nun ist Alice am Zug, sie möchte die Nachricht von Bob entschlüsseln:

- Klicke auf den Knopf "Alice" oben links, um in die Rolle von Alice zu wechseln!
- Klicke dann auf das Tresor-Symbol ( ), das den privaten Schlüssel von Alice umgibt, um den privaten Schlüssel in die Felder "anzuwendender Schlüssel" auf Alices Computer zu laden!
- Klicke nun im Bereich von Alices Computer auf den Knopf "Schlüssel auf Nachricht anwenden", um die Nachricht mit dem zuvor kopierten Schlüssel zu entschlüsseln!

Die Nachricht ist ausgetauscht, ohne dass der Klartext im Internet zu lesen war. Im Unterschied zu symmetrischen Verfahren mussten Alice und Bob jedoch niemals ihre geheimen Schlüssel austauschen!

Aufgaben:

1. Alice möchte Bob - und nur Bob und nicht ihren Eltern, deren Computer sie benutzt! - den Namen ihrer neuen Lieblingsband mitteilen. Zeige was Alice und Bob machen, so dass Bob die Nachricht von Alice auf sicherem Wege erfährt!

   Hinweis: Diese Simulation unterstützt nur die Verschlüsselung von Kleinbuchstaben, Punkt und Komma – Zahlen sollten daher ausgeschrieben werden!

2. Verfasse einen Lexikon-Eintrag, der erklärt, wie mit asymmetrischer Kryptographie Ver- und Entschlüsseln kann, ohne vorher geheime Informationen austauschen kann!
3. *für Schnelle:* Könnte es Sinn machen, eine Nachricht mit dem privaten Schlüssel zu verschlüsseln?

Primzahlen finden mit dem Sieb des *Eratosthenes*

Für die asymmetrische Kryptographie benötigen wir mathematische Funktionen, deren Anwendung mit einer Information (dem öffentlichen Schlüssel) sich durch Anwendung mit einer anderen Information (dem privaten Schlüssel) rückgängig machen lässt. Hier spielen **Primzahlen** eine wichtige Rolle, die einige besondere Eigenschaften aufweisen:

Definition:
Primzahlen sind alle natürlichen Zahlen größer als 1, die nur durch 1 und sich selber teilbar sind.
Alle natürlichen Zahlen größer als 1, die keine Primzahlen sind, heißen **zusammengesetzte Zahlen**.
Die 1 ist weder eine Primzahl noch ist sie zusammengesetzt!

Aufgabe 1: Nenne 10 Beispiele für Primzahlen!

Aufgabe 2: Überlege Dir eine Begründung, warum man die anderen Zahlen „zusammengesetzt" nennt! Nenne 10 Beispiele für zusammengesetzte Zahlen!

Wie findet man Primzahlen? Eine sehr effektive Methode ist das **Sieb des *Eratosthenes***.

Aufgabe 3: Informiere Dich unter der Adresse http://www.hbmeyer.de/eratosib.htm über die Funktionsweise des Primzahlsiebs! Bearbeite die auf dieser Seite genannte Aufgabe!

Zur Auswertung dieser Experimente überleg Dir Antworten auf die folgenden Fragen:

- Warum erhält man bereits alle Primzahlen ≤ 400, wenn man nur mit den Primzahlen ≤ 20 „siebt"?
- Wieso kann man sicher sein, dass wirklich nur noch Primzahlen in der Tabelle stehen?
- Warum nannte *Eratosthenes* das Verfahren, das er vermutlich gar nicht selber erfunden hat, „Sieb"?

Wichtig für die moderne Kryptologie im Allgemeinen und das RSA-Verfahren im Besonderen sind die so genannten **Semiprimzahlen**. Das sind natürliche Zahlen ***n***, die genau zwei unterschiedliche Primfaktoren ***p*** und ***q*** haben, so dass $n = p \cdot q$ gilt. Für die asymmetrische Kryptographie ist es wichtig, dass man aus dem öffentlichen Schlüssel ***e*** den privaten Schlüssel ***d*** nicht berechnen kann. Dies wird beim RSA-Verfahren dadurch abgesichert, dass es praktisch unmöglich ist, riesige Semiprimzahlen mit hunderten von Dezimalstellen in ihre beiden Primfaktoren zu zerlegen. Umgekehrt ist es sehr einfach, aus zwei großen Primzahlen durch Multiplikation eine Semiprimzahl zu erzeugen.

Aufgabe 4: Welche der folgenden Zahlen sind Primzahlen, welche sind Semiprimzahlen?
Falls es Semiprimzahlen sind: Gib die Faktoren an!

23, 55, 113, 119, 841, 1829, 3109, 9847, 10807, 13121, 14603, 15551, 16061, 16199,
1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139

Hinweis: Du kannst die Zahlen mit dem Sieb des Eratosthenes faktorisieren, oder mit *CrypTool*:
Einzelverfahren -> RSA-Kryptosystem -> Faktorisieren einer Zahl...
Ist der größte Faktor rot dargestellt, so lässt er sich mit Klick auf den „Weiter"-Knopf in weitere Faktoren zerlegen.
Lässt sich eine Zahl nicht in vernünftiger Zeit faktorisieren, so lässt sich zumindest recht schnell feststellen, ob die Zahl eine Primzahl ist:
Einzelverfahren -> RSA-Kryptosystem -> Primzahltest

Aufgabe 5: Recherchiere: Was hat letzte Zahl mit der *RSA Factoring Challenge* zu tun?
(z.B. auf http://en.wikipedia.org/wiki/RSA_numbers)

Zusatzaufgabe: Recherchiere: Wer war eigentlich *Eratosthenes*?

„Modulares Rechnen“ - Rechnen mit Resten

Folgende Aufgaben sind uns aus dem Alltag bekannt:

a) Heute ist Donnerstag. Welcher Wochentag ist in 47 Tagen?

b) Es ist der Monat März. Welcher Monat ist in 50 Monaten?

c) Es ist jetzt 12 Uhr. Wie spät ist es in 100 Stunden?

Wie lassen sich diese Aufgaben systematisch berechnen?

Eine Woche hat 7 Tage, d.h. der Wochentag wiederholt sich alle 7 Tage. Um herauszufinden, welcher Wochentag der 47. Tag nach einem Donnerstag ist, können wir von den 47 Tagen zunächst die Anzahl aller Tage, die in eine ganze Woche passen abziehen. Doch wie viele ganze Wochen passen in 47 Tage?

Um das zu berechnen dividieren (teilen) wir die Anzahl der Tage (47) durch die Anzahl der Tage, die in eine Woche passen (7) und betrachten die Zahl vor dem Komma (den „Ganzzahlquotienten“):

Das Ergebnis des Taschenrechners lautet: 47 : 7 = **6**,714285...

Nach **6** · 7 = 42 Tagen ist also wieder Donnerstag. Es bleiben 47 – 42 = **5** Tage.
Diesen Wert bezeichnen wir als **Rest** der Division durch 7.
Wir können also sagen: 47 : 7 = **6** + Rest **5**

Hinweis: Bei der schriftlichen Division ist der Rest direkt in dem Moment ablesbar, zu dem das Komme im Ergebnis gesetzt wird:

```
47 : 7 = 6,7...
 42
 → 50
   49
   ...
```

Die Rest-Funktion oder Modulo-Funktion

Allgemein lässt sich das Ergebnis der Division einer Zahl x durch den Divisor **d** als Summe aus dem Produkt des Ganzzahlquotienten mit dem Divisor **d** und dem Rests **r** der Division beschreiben:

$$\frac{x}{d} = q \cdot d + r \qquad \text{Beispiel: } \frac{47}{7} = 6 \cdot 7 + 5$$

Die Rest-Funktion wird auch als Modulo-Funktion bezeichnet, wobei der Divisor auch als „der Modul“ bezeichnet wird. Für die Berechnung des Rests **r** der Division einer Zahl x durch den Divisor **d** nutzen wir fortan die in der Programmierung übliche Schreibweise „mod“:

r = x mod d Beispiel: 47 mod 7 = 5

In der Programmierung werden verschiedene Schreibweisen für die Modulo-Funktion verwendet:

z.B. `r := x MOD d` in *PASCAL* und *DELPHI*,
`r = x % d` in *Java*, *C++*, *PHP* und *Python*.

Notiere unsere obigen Beispiele in der neuen Schreibweise!

a) 47 mod 7 = 5 b) ____ mod ____ = ____ c) ____ mod ____ = ____

Aufgabe: Berechne: r = 6241 mod 85 = ____

„Modulares Rechnen“ – Rechnen mit Resten - Lösung

Im Alltag berechnen wir oft den Rest einer Division:

Heute ist Donnerstag. Welcher Wochentag ist in 47 Tagen?	Donnerstag + 5 = Dienstag
Es ist der Monat März. Welcher Monat ist in 50 Monaten?	März + 2 = Mai
Es ist jetzt 12 Uhr. Wie spät ist es in 100 Stunden?	12 Uhr + 4 = 16 Uhr

Unsere obigen Beispiele in neuer Schreibweise:

47 mod 7 = 5 50 mod 12 = 2 100 mod 24 = 4

Das RSA-Verfahren

Das RSA-Verfahren wurde 1978 von **R**ivest, **S**hamir und **A**dleman entwickelt.

1. Geeignete Schlüssel wählen

- Wähle zwei Primzahlen **p** und **q** !

  p = ____ und **q** = ____

- Berechne das Produkt **N** der beiden gewählten Primzahlen p und q !

  N = p · q = ____ · ____ = ____

- Berechne das folgende Produkt **phi** = (p-1)·(q-1) !

  phi = (___-1)·(___-1) = (___)·(___) = ____

- Bestimme zwei natürliche Zahlen **d** und **e** so, dass gilt: $(d \cdot e) \bmod phi = 1$
  (Für Interessierte: da der Rest der Division (d · e) : phi den Wert 1 ergibt sind die Zahlen d und e „modular invers“ bezüglich der Division durch phi).

  erster Versuch:

  - Die erste Zahl, die als Rest einer Division durch ____ den Rest 1 läst, ist ___:
    ____ mod ____ = 1

Nun sind ein **geheimer** Schlüssel **d**

und ein **öffentlicher** Schlüssel **(e;N)** gefunden.

2. Verschlüsseln

Die Verschlüsselung einer natürlichen Zahl m < N durch einen beliebigen Teilnehmer erfolgt mit Hilfe des öffentlichen Schlüssels (e;N) :

$$c = m^e \bmod N$$

Bsp.: Verschlüsselung der Zahl ____: $c = \_\_^{\_\_} \bmod \_\_ =$ ____

3. Entschlüsseln

Die Entschlüsselung einer verschlüsselten Zahl c durch den Empfänger erfolgt mit Hilfe des geheimen Schlüssels d und dem öffentlichen N:

$$m = c^d \bmod N$$

Bsp.: Entschlüsselung der Zahl ____: $m = \_\_^{\_\_} \bmod \_\_ =$ ____

Das RSA-Verfahren – Lösung für p=5, q=11

Das RSA-Verfahren wurde 1978 von **R**ivest, **S**hamir und **A**dleman entwickelt.

1. Geeignete Schlüssel wählen

- Wähle zwei Primzahlen **p** und **q** !

  p = 5 und **q** = 11

- Berechne das Produkt **N** der beiden gewählten Primzahlen p und q !

  N = p · q = 5 · 11 = 55

- Berechne das folgende Produkt **phi** = (p-1)·(q-1) !

  phi = (5 -1)·(11 -1) = (4)·(10) = 40

- Bestimme zwei natürliche Zahlen **d** und **e** so, dass gilt: $(d \cdot e) \bmod phi = 1$
  (Für Interessierte: da der Rest der Division (d · e) : phi den Wert 1 ergibt sind die Zahlen d und e „modular invers" bezüglich der Division durch phi).

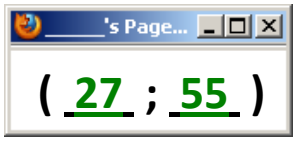
  erster Versuch:

  - Die erste Zahl, die als Rest einer Division durch 40 den Rest 1 läst, ist 41 :
    41 mod 40 = 1 — 41 lässt sich nicht in zwei Faktoren zerlegen, weil sie eine Primzahl ist.
    -> es gibt kein Zahlenpaar (d;e) mit d_e = 41

  zweiter Versuch:

  - Die zweite Zahl, die als Rest einer Division durch 40 den Rest 1 läst, ist 81 :
    81 mod 40 = 1 — 81 lässt sich als 3 _ 27 darstellen. Damit ist ein Zahlenpaar (d;e) mit d_e = 81 gefunden.

Nun sind ein **geheimer Schlüssel d** und ein **öffentlicher Schlüssel (e;N)** gefunden.



2. Verschlüsseln

Die Verschlüsselung einer natürlichen Zahl m < N durch einen beliebigen Teilnehmer erfolgt mit Hilfe des öffentlichen Schlüssels (e;N) :

$$c = m^e \bmod N$$

Bsp.: Verschlüsselung der Zahl 2 : $c = 2^{27} \bmod 55 = 18$

3. Entschlüsseln

Die Entschlüsselung einer verschlüsselten Zahl c durch den Empfänger erfolgt mit Hilfe des geheimen Schlüssels d und dem öffentlichen N:

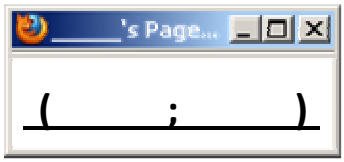
$$m = c^d \bmod N$$

Bsp.: Entschlüsselung der Zahl 18 : $m = 18^3 \bmod 55 = 2$

Mit RSA Daten verschlüsselt austauschen

Aufgaben:

1. **Erstelle** ein eigenes Schlüsselsystem gemäß Anleitung oder wähle angebotenes aus!
2. **Tausche** mit einem Partner die öffentlichen Schlüssel aus!
3. **Verschlüssle** deinen Geburtstag mit dem **öffentlichen** Schlüssel **deines Partners**!
   Verschlüssle dabei zunächst den **Tag**, dann den **Monat**!
4. **Tauscht** nun eure **verschlüsselten** Geburtstage aus!
5. **Entschlüssle** den Geburtstag deines Partners mit **deinem geheimen** Schlüssel!

Mein Schlüsselsystem		**Schlüsselsystem deines Partners**	
mein **geheimer Schlüssel:**	mein **öffentlicher Schlüssel:**	sein **geheimer Schlüssel:**	sein **öffentlicher Schlüssel:**
			

Verschlüsselung meines eigenen Geburtstags:

Entschlüsselung des Geburtstags meines Partners:

Mit RSA Daten verschlüsselt austauschen

Aufgaben:

1. **Wähle** ein Schlüsselsystem aus, bei dem n > 32.000.000 ist.
   Die RSA-Demo von *CrypTool* und folgende Information werden dir dabei behilflich sein:
   $2^{24} = 16.777.216$ $2^{25} = 33.554.432$ $2^{26} = 67.108.864$
2. **Tausche** mit einem Partner die öffentlichen Schlüssel aus!
3. **Fasse** dein Geburtsdatum in einer großen Zahl **zusammen**, die aus den Ziffern deines Geburtsdatums in der Reihenfolge **ttmmjjjj** besteht!
   Beispiel: 23.5.1991 → 23051991 (Die Null nicht vergessen, falls Tag oder Monat < 10!)
4. **Verschlüssle** dein Geburtsdatum mit dem **öffentlichen** Schlüssel **deines Partners**!
   Nutze dazu die RSA-Demo von *CrypTool*!
5. **Tauscht** nun die **verschlüsselten** Geburtstage per E-Mail aus!
6. **Entschlüssle** das Geburtsdatum deines Partners mit **deinem geheimen** Schlüssel!
   Nutze dazu die RSA-Demo von *CrypTool*!

Mein Schlüsselsystem	**Schlüsselsystem deines Partners**
mein **geheimer Schlüssel:**	sein **geheimer Schlüssel:**
	
mein **öffentlicher Schlüssel:**	sein **öffentlicher Schlüssel:**
	

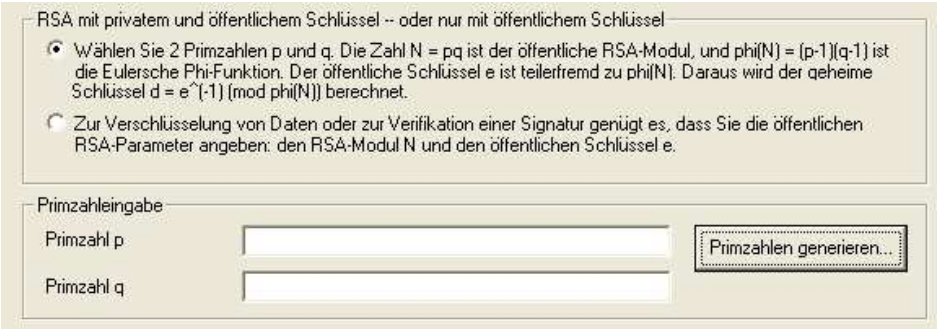
Verschlüsselung meines eigenen Geburtsdatums:

Entschlüsselung des Geburtsdatums meines Partners:

Anleitung: Ver- und Entschlüsseln mit der RSA-Demo von *CrypTool*

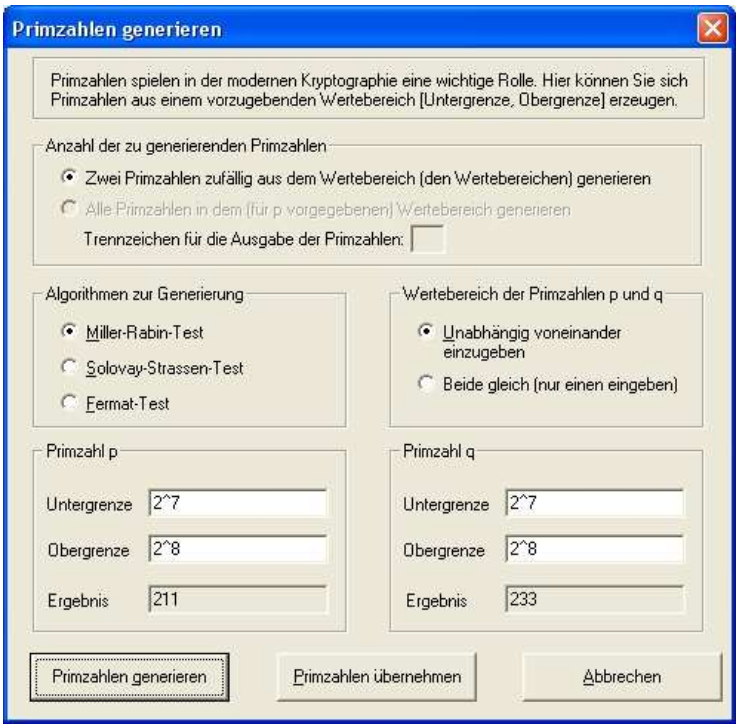
1. RSA-Schlüsselpaar generieren:

Starte *CrypTool* und rufe im Menü *Einzelverfahren* → *RSA-Kryptosystem* → *RSA-Demo...* auf! Es erscheint ein (auf den ersten Blick etwas unübersichtliches) Fenster. Betrachte zunächst nur den oberen Ausschnitt:



Es gibt zwei Möglichkeiten, die Schlüssel **(e; N)** und **d** zu konstruieren:

- zwei Primzahlen in die Felder für p und q eintragen oder
- die Primzahlen von *CrypTool* wie folgt erzeugen lassen: Knopf „Primzahlen generieren“ drücken. Ein neues Fenster erscheint mit den voreingestellten Werten:

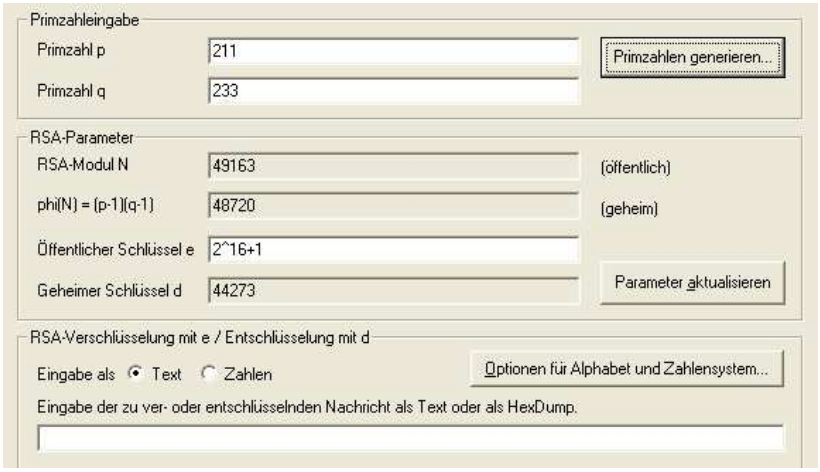


Wir haben also zwei Primzahlen zwischen $2^7 = 128$ und $2^8 = 256$ erhalten. Wenn uns diese Zahlen nicht gefallen sollten, drücken wir ggf. mehrfach „Primzahlen generieren“ und erhalten dann andere Primzahlen aus diesem Bereich, z. B. 227 und 251. *CrypTool* benutzt Pseudozufallszahlen, die stets in der gleichen Reihenfolge auftreten, es macht also Sinn, mehrmals auf den Knopf zu drücken!

Klicke nun auf „Primzahlen übernehmen“. Es erscheint wieder der Ausgangsschirm, aber neben den Primzahlen p und q sind bereits der **RSA-Modul N** und **phi(N) = (p-1)(q-1)** eingetragen. Als öffentlicher Schlüssel e ist immer $2^{16}+1 = 65537$ voreingestellt[1]. Wem diese Zahl nicht gefällt, kann auch hier eine andere eintragen. Diese muss aber teilerfremd zu phi(N) sein!

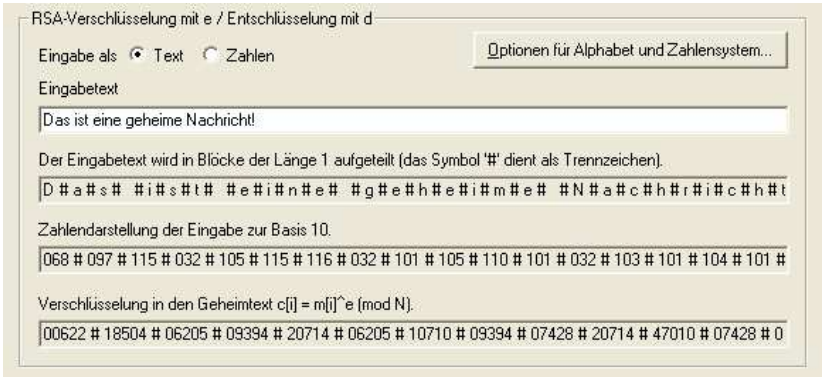
[1] Wer wissen will, warum gerade diese Zahl bevorzugt wird, sollte sich z. B. mit Hilfe des Windows-Taschenrechners ihre Darstellung im Dualsystem anschauen

Der zugehörige geheime Schlüssel wird ebenfalls automatisch erzeugt:



2. Verschlüsseln:

- Gib einen Text in das untere Eingabefeld ein!
- Klicke auf „Verschlüsseln“ um ihn zu **verschlüsseln**!

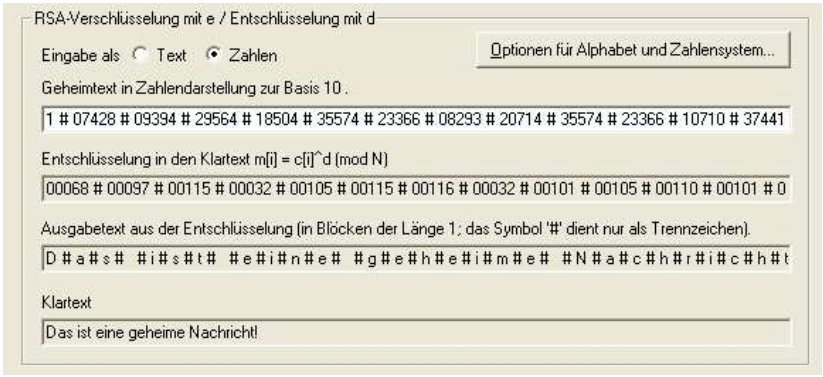


Wegen des relativ kleinen Moduls **N** wird der Text in Blöcke der Länge 1 unterteilt[2] und als Zahlen dargestellt (die entsprechenden ASCII-Nummern). Diese werden dann Block für Block (bei Blocklänge 1 also Zeichen für Zeichen) verschlüsselt.

3. Entschlüsseln:

Trotzdem wollen wir uns überzeugen, dass sich dieser „Geheimtext“ wieder korrekt **entschlüsseln** lässt:

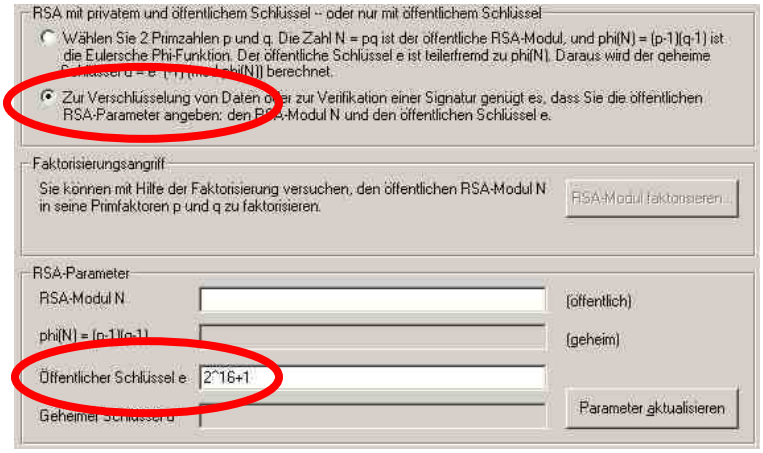
- Kopiere den Geheimtext in die Eingabezeile!
- Klicke auf den Knopf „Zahlen“! (Wenn Du das vergisst, weißt dich *CrypTool* darauf hin.)
- Klicke auf den Knopf „Entschlüsseln“! Ergebnis:



[2] Bei einem so kleinen Modul N handelt es sich um eine schlichte monoalphabetische Verschlüsselung, die ein Knacken per Häufigkeitsanalyse erlaubt. Bei größeren Primzahlen werden jedoch stets mehrere Zeichen in einem Block zusammengefasst.

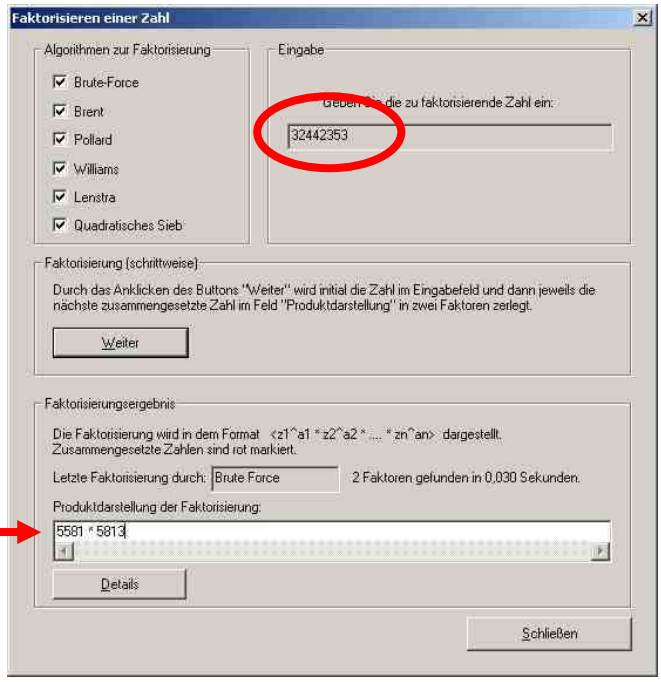
Anleitung: RSA knacken mit *CrypTool*

1. Starte CrypTool und rufe im Menü *Einzelverfahren → RSA-Kryptosystem → RSA-Demo...* auf!
   Wähle diesmal den zweiten Radioknopf „Zur Verschlüsselung von Daten..."!
   In diesem Fall können nur der RSA-Modul N und der öffentliche Schlüssel e eingegeben werden (e ist wieder auf $2^{16}+1$ voreingestellt, das kann aber geändert werden).



2. Gib einen RSA-Modul N ein!

3. Klicke nun den Knopf „RSA-Modul faktorisieren"! Wenn die eingegebene Zahl ein gültiger RSA-Modul (und nicht zu groß) ist, werden die beiden Primfaktoren p und q in dem Faktorisierungsfenster von CrypTool gefunden. Falls N keine Semiprimzahl ist, wird eine entsprechende Fehlermeldung ausgegeben.



Erfolgreiche Faktorisierung des RSA-Moduls N= 32442353.

Nach Schließen dieses Fensters werden die Primfaktoren p und q im normalen Fenster vom RSA-Demo eingetragen, außerdem werden sogleich phi(N) und der geheime Schlüssel d berechnet. Man hat damit wieder ein voll funktionsfähiges RSA-System, obwohl nur der öffentliche Schlüssel bekannt war!

Merke: **<u>Die Sicherheit von RSA wird ganz wesentlich</u>**

<u>von der Schlüssellänge des gewählten RSA-Moduls bestimmt!</u>

Forschungsauftrag

Wieviel Bit muss ein RSA-Modul haben, damit er nicht mehr innerhalb von wenigen Sekunden mit *CrypTool* in der oben beschriebenen Weise geknackt werden kann?

Hinweis: Wie kann ich z. B. einen RSA-Modul mit 64 Bit erzeugen? Dafür werden zwei Primfaktoren p und q mit jeweils 32 Bit Länge benötigt. Man gebe also im Fenster „Primzahlen generieren..." als Untergrenze jeweils 2^{31} und als Obergrenze 2^{32} ein. Nach den Klicks auf „Primzahlen generieren" und danach „Primzahlen übernehmen" hat man zwei Primzahlen p und q mit 32 Bit Länge und einen Modul N = p·q mit 64 Bit Länge erzeugt!

RSA-Schlüssel

öffentlich	privat	öffentlich	privat
(35;17)	5	(15;11)	3
(55;33)	17	(55;23)	7
(55;27)	3	(65;29)	5
(51;11)	3	(91;29)	5
(26;17)	5	(43;77)	7

Integrität und Authentizität mit digitaler Unterschrift sicherstellen

Durch Verschlüsseln mit Alices öffentlichem Schlüssel kann Bob Alice Nachrichten senden, die nur Alice wieder entschlüsseln kann. Allerdings kann JEDE(R) eine solche Nachricht verfassen - die Integrität der Nachricht (Wurde die Nachricht im Nachhinein verändert?) und die Authentizität des Absenders (Stammt die Nachricht wirklich von Bob?) bleiben also ungeklärt. Doch bietet die asymmetrische Kryptographie auch für diese Herausforderung eine elegante Lösung: Eine so genannte "Hash-Funktion" liefert für eine bestimmte Folge von Zeichen immer denselben Wert. Verändert man auch nur ein Zeichen der Folge, fügt man ein Zeichen hinzu oder entfernt man ein Zeichen, so ergibt sich stets ein anderer Hashwert. Du kannst die Entwicklung des Hashwerts beim Verfassen einer Nachricht im Feld Hashwert verfolgen!

Sende ich nun den Hashwert meiner Nachricht zusätzlich zur eigenen Nachricht, so kann der Empfänger den Hashwert der empfangenen Nachricht berechnen und das Ergebnis mit dem der Nachricht beigefügten ursprünglichen Nachricht vergleichen - jegliche Veränderung lässt sich so auf einen Blick feststellen!

Doch könnte jemand, der die Nachricht auf dem Kommunikationsweg verändert auch den Hashwert verändern, so dass der zur veränderten Nachricht passt - die Veränderung bleibt unbemerkt! Auch könnte die Nachricht nach wie vor von jemand ganz anderem verfasst worden sein, der den passenden Hashwert ermittelt und angehängt hat. Um zu zeigen, dass Bob und kein anderer als er genau diese Nachricht verfasst hat, verschlüsselt er den Hashwert mit seinem eigenen privaten Schlüssel. Aus dem Hashwert wird die "digitale Unterschrift" - auch "**digitale Signatur**" genannt. Nun kann jeder mit Bobs öffentlichem Schlüssel den Hashwert der Nachricht wieder entschlüsseln - es ist aber eindeutig belegt, dass die Signatur nur mit Bobs privatem Schlüssel erstellt worden sein kann.

Damit das Vertrauen in die digitale Unterschrift bestehen bleibt, müssen alle Teilnehmer gut auf ihre privaten Schlüssel aufpassen. In Programmen, die solche Schlüssel verwalten, werden die Dateien, in denen ein privater Schlüssel gespeichert ist, deshalb oft selbst mit einer Passworteingabe vor unbefugtem Benutzen des Schlüssels geschützt!

Aufgabe:

Öffne die Animation „Integrität und Authentizität mit digitaler Unterschrift sicherstellen“ und sende eine digital signierte Nachricht an Alice:

- Verschlüssle dazu zunächst die Nachricht mit Alices öffentlichem Schlüssel! (Anleitung siehe „Vertraulichkeit durch asymmetrische Kryptologie herstellen“)
- Kopiere den Hashwert der verschlüsselten Nachricht als Signatur, indem Du im Bereich von Bobs Computer auf den Knopf "V" klickst.
- Verschlüssle nun den Hashwert mit Bobs eigenem privaten Schlüssel: Klicke erst auf das Tresor-Symbol ( ) um Bobs privaten Schlüssel und dann auf den Knopf "Schlüssel auf Signatur anwenden" unterhalb des Signatur-Eingabefelds!
- Versende die Nachricht an Alice (Knopf "<<") und wechsle in die Rolle von Alice!
- Überprüfe in der Rolle von Alice zunächst Integrität und Authentizität der Nachricht, indem Du die erhaltene Signatur mit Bobs öffentlichem Schlüssel entschlüsselst: Klicke erst auf das Webseiten-Symbol ( ) um Bobs öffentlichen Schlüssel und dann auf den Knopf "Schlüssel auf Signatur anwenden" unterhalb des Signatur-Eingabefelds im Bereich von Alices Computer! Stimmt der Hashwert der verschlüsselten Nachricht mit der entschlüsselten Signatur überein? Dann kann die Nachricht in dieser Form nur von Bob sein!
- Entschlüssle nun die eigentliche Nachricht mit Alices privatem Schlüssel! (Anleitung siehe „Vertraulichkeit durch asymmetrische Kryptologie herstellen“)

E-Mails verschlüsseln und digital unterschreiben

Das E-Mail-Client-Programm *Thunderbird* lässt sich mit dem Programm *GnuPG* und dem Add-On[1] *Enigmail* um Funktionen zum Verschlüsseln und digitalen Unterschreiben von E-Mails erweitern.

Schritt 1: Sicherstellen, dass alle Programme installiert sind

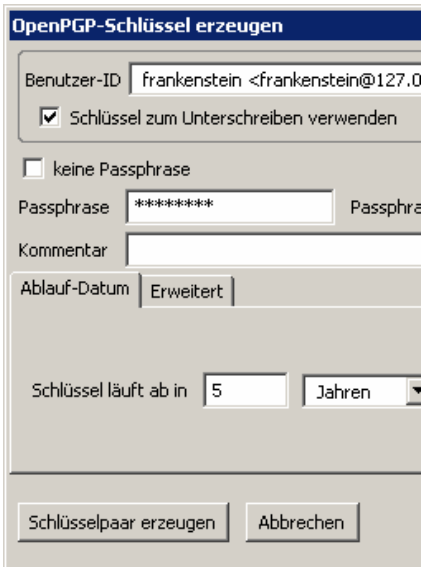
Um E-Mails mit *Thunderbird* verschlüsseln und digital unterschreiben zu können müssen folgende Programme in der angegebenen Reihenfolge auf deinem Computer installiert werden:

A. *Thunderbird* - das Programm kann sich jede(r) lizenzkostenfrei unter http://www.mozillamessaging.com/de/ herunterladen und installieren

B. *Gnu Privacy Guard (GnuPG)*- dieses Programm erledigt die Schlüsselerzeugung, das Ver- und Entschlüsseln, und das Signieren von Nachrichten gemäß dem Open PGP-Standard.
Jede(r) kann sich das Programm für Windows lizenzkostenfrei unter ftp://ftp.gnupg.org/gcrypt/binary/gnupg-w32cli-1.4.10b.exe herunterladen.
(Links zu Versionen für andere Betriebssysteme findest Du unter http://www.gnupg.org/download/index.de.html#auto-ref-2 .)
Ob *GnuPG* auf deinem Computer bereits installiert ist kannst Du herausfinden, indem Du prüfst, ob es im Start-Manü unter Programme bereits einem Ordner „GPG“ gibt!

C. Das *Thunderbird*-Add-On *Enigmail* - ermöglicht es Dir, die Funktionen von *GnuPG* direkt im E-Mail-Client-Programm *Thunderbird* aufzurufen. Das Add-On lässt sich unter https://addons.mozilla.org/de/thunderbird/addon/71/ herunterladen (Wenn Du eine ältere Version von *Thunderbird* hast, klicke auf der Webseite unten auf den Link „Alle Versionen“ und wähle die Version , die zu deiner *Thunderbird*-Version passt!).
Sind *Thunderbird* und *GnuPG* auf deinem Computer installiert, so starte nun *Thunderbird* und rufe im Menü *Extras* → *Add-ons…* auf!
Es öffnet sich ein Dialogfenster „Add-ons“. Klicke auf den Knopf „Installieren“ unten links und wähle die Datei zuletzt heruntergeladene Datei „enigmail … xpi“ aus!
Nach der Installation des *GnuPG*-Add-Ons gibt es in *Thunderbird* einen zusätzlichen Menü-Eintrag „OpenPGP“.

Schritt 2: Schlüsselpaar erzeugen



Sind alle Programme installiert, so solltest Du dir zu allererst ein Schlüsselpaar erzeugen:

1. Wähle im Menü *OpenPGP* → *Schlüssel verwalten* !
2. Wähle im Dialogfenster „OpenPGP-Schlüssel verwalten“ im Menü *Erzeugen* → *Neues Schlüsselpaar* !
3. Wähle nun im Dialogfenster „OpenPGP-Schlüssel erzeugen“ das Postfach aus, für das Du das Schlüsselpaar verwenden möchtest!
4. Darunter solltest Du ein Passwort zweimal eingeben. Das Passwort stellt sicher, dass nur Du deinen privaten Schlüssel benutzen kannst, selbst wenn sich jemand anders (z.B. mit einem Spionage-Programm) Zugang zu der Datei verschafft, in der dein privater Schlüssel gespeichert ist.
5. Erzeuge das Schlüsselpaar durch einen Klick auf den Knopf „Schlüsselpaar erzeugen“!
Durch Speichern des Widerruf-Zertifikats kannst Du den Schlüssel vor Ablauf der Gültigkeit deaktivieren.

[1] Als ein „Add-On“ bezeichnet man eine Erweiterung für ein Programm, die man sich installieren kann, um das Programm mit zusätzlichen Funktionen auszustatten.

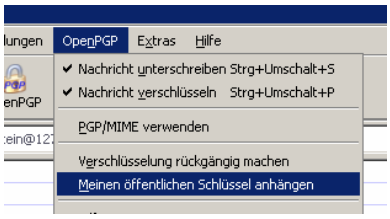
Schritt 3: Öffentliche Schlüssel austauschen

Für den Austausch öffentlicher Schlüssel gibt es zwei Möglichkeiten:

A. Exportiere deinen öffentlichen Schlüssel in eine **Datei**, indem Du
   - im Dialog „OpenPGP-Schlüssel verwalten“ den entsprechenden Schlüssel markierst und
   - im Menü *Datei* → *Exportieren* aufrufst.
   - Natürlich wollen wir den privaten Schlüssel nicht veröffentlichen. Klicke also auf „Nein“, wenn Du danach gefragt wirst. (Die Möglichkeit, den privaten Schlüssel mit zu exportieren ist für den Fall einer Sicherungskopie des Schlüssels gedacht.)
   - Die erzeugte Datei kannst Du nun deinen Kommunikationspartnern übermitteln (z.B. auf einen USB-Stick kopieren, auf deine private Homepage hoch laden oder als Anhang in einer E-Mail versenden (Hier sollte aber aus dem Inhalt der E-Mail hervorgehen, dass sie wirklich von Dir stammt).
   - Bekommst Du den Schlüssel von deinem Kommunikationspartner, so kannst Du ihn im Dialog „OpenPGP-Schlüssel verwalten“ über das Menü *Datei* → *Importieren* in die Liste dir bekannter Schlüssel einfügen - wenn Du dir sicher bist, dass das der richtige Schlüssel ist!

B. Veröffentliche deinen öffentlichen Schlüssel auf einem **Schlüssel-Server**, indem Du
   - im Dialog „OpenPGP-Schlüssel verwalten“ den entsprechenden Schlüssel markierst und
   - im Menü *Schlüssel-Server* → *Schlüssel hochladen* aufrufst.
   - Wähle einen der angegebenen Schlüsselserver aus und klicke auf „OK“!
   - Hat dein Kommunikationspartner seinen öffentlichen Schlüssel ebenfalls auf dem Schlüsselserver veröffentlicht, so kannst Du ihn über das Menü *Schlüssel-Server* → *Schlüssel suchen* durch Eingabe seiner E-Mail-Adresse suchen.
   - Markiere den Eintrag mit der korrekten E-Mail-Adresse und klicke auf „OK“, um den Schlüssel in die Liste dir bekannter Schlüssel einfügen.

Schritt 4: E-Mails verschlüsseln und digital unterschreiben



Haben Absender und Empfänger ihre Schlüssel einmal wie oben beschrieben ausgetauscht, so wird und das eigentliche Verschlüsseln und digitale Unterschreiben der E-Mails von *Enigmail* leicht gemacht:

- Zum Verschlüsseln einer E-Mail klicke nach dem Verfassen der E-Mail im Menü *OpenPGP* auf den Eintrag *Nachricht verschlüsseln*. Der Haken vor diesem Menüeintrag zeigt, dass die Verschlüsselung aktiviert ist. Möchtest Du die E-Mail doch nicht verschlüsseln, so klicke einfach erneut auf den Menüeintrag *OpenPGP* → *Nachricht verschlüsseln* um den Haken zu entfernen.

- Zum digitalen Unterschreiben einer E-Mail klicke nach dem Verfassen der E-Mail im Menü *OpenPGP* auf den Eintrag *Nachricht unterschreiben*. Fortan ist ein Haken vor diesem Menüeintrag gesetzt, das Verschlüsseln ist also aktiviert. Möchtest Du die E-Mail doch nichtunterschreiben? Dann klicke einfach erneut auf den Menüeintrag *OpenPGP* → *Nachricht unterschreiben* um den Haken zu entfernen.
  Über den Menüeintrag *OpenPGP* → Meinen *öffentlichen Schlüssel anhängen*“ kannst Du übrigens bequem deinen öffentlichen Schlüssel der E-Mail als Dateianhang hinzufügen.

Hinweis: Das Absenden einer **verschlüsselten** Nachricht gelingt natürlich nur, wenn der öffentliche Schlüssel zur **E-Mail-Adresse des Empfängers** bekannt ist.
Unterschreiben kann ich dagegen jede E-Mail, es liegt dann am Empfänger, ob er die Unterschrift mit meinem öffentlichen Schlüssel überprüfen möchte oder nicht.

Aufgabe: Starte wie zu Beginn dieser Unterrichtsreihe das Netzwerkanalyseprogramm *Socket Sniff*! Erzeuge dir ein Schlüsselpaar, tausche deinen öffentlichen Schlüssel mit anderen Mitschülern und versendet Euch verschlüsselte und digital unterschriebene E-Mails! Betrachtet den E-Mail-Verkehr in *Socket Sniff*! Wie schlau werden nun neugierige Angreifer, die sich Zugang zu Routern oder dem E-Mail-Server verschaffen aus Euren E-Mails?!

Auftrag an die Stammgruppen

In diesem Gruppenpuzzle werdet ihr euch über das **Echelon-System, De-Mail, PGP** und **Kommunikationsfreiheit** Informieren. Am Ende der nächsten Stunde sollen *alle Schülerinnen und Schüler* in eurer Gruppe folgende Fragen beantworten können:

E 1 Was ist das Echelon-System?
E 2 Wie kann man sich und sein Unternehmen vor Informationsbeschaffung schützen?

D 1 Was ist De-Mail?
D 2 Was sind die Hauptkritikpunkte an dem geplanten „Bürgerportal“?

K 1 Wie und warum können Staaten E-Mail-Verkehr kontrollieren?
K 2 Was bringt Dir die Kommunikationsfreiheit?

P 1 Wie erschafft PGP Vertraulichkeit?
P 2 Wie erschafft PGP Authentizität und Integrität?

Nehmt Euch eine Minute Zeit, um zu sammeln: Was würdet ihr Euch unter den oben genannten Begriffen vorstellen? Betrachtet auch die Bilder im unteren Bereich dieses Arbeitsbogens!

Da ihr für die Beantwortung der Fragen viele Informationen braucht, sollt ihr Euch in vier Gruppen aufteilen, die jeweils zwei Fragen beantworten. Jede Gruppe erhält ein Arbeitsblatt mit wichtigen Informationen zum Thema, sowie Anhaltspunkte für eine weitergehende Recherche.

Am Ende der Stunde sollen alle auf ihrem Gebiet Expertinnen bzw. Experten sein. In der nächsten Stunde sollt ihr Euch dann gegenseitig informieren. Alle müssen sich also vorbereiten, einen kurzen, freien Vortrag zu „ihrem“ Thema zu halten und Nachfragen zu beantworten.

Arbeitsbogen *Echelon*

Aufgaben

1. Lest den Text auf dem Arbeitsbogen gründlich durch. Versucht, Unklarheiten miteinander zu klären. Natürlich könnt ihr auch den Lehrer fragen!
2. Überprüft Euer Wissen anhand der Kontrollfragen auf den nächsten Seiten.
3. Bereitet Euch darauf vor, einen kurzen Vortrag zu den unten aufgeführten Fragen E 1 und E 2 zu halten. Macht Euch Stichpunkte dazu!
4. Überlegt Euch, welche Nachfragen von Euren Klassenkameradinnen und -kameraden kommen könnten. Bereitet Euch darauf vor, diese zu beantworten. Fertigt Euch dazu Sprechkarten an!
5. Recherchiert nach weiteren Informationen zu den unten aufgeführten Fragen, um diese in Euren Vortrag einbauen zu können!

E 1 Was ist das Echelon-System?

Radom der US-Basis Bad Aiblingen (Bayern); [Quelle: en.wikipedia.org]

Der Name Echelon steht für ein globales Abhörsystem der Staaten des UKUSA-Abkommens. Da es sich hierbei um ein geheimdienstliches Projekt handelt, wurde die Existenz des Systems von öffentlicher Seite nie bestätigt. Im Jahr 2001 führte ein Ausschuss der EU jedoch einen Indizienbeweis für die – seitdem als gesichert geltende – Existenz von Echelon.
Echelon funktioniert hauptsächlich über das Abfangen von Satellitenkommunikation mittels Richtantennen (siehe Bild: Jede Kuppel verbirgt eine Richtantenne, so dass die Ausrichtung der Antenne geheim bleibt). Der Anteil des globalen Datenverkehrs über Satelliten beträgt nur etwa 5% - das Abhören von Kabeln ist wesentlich schwieriger, da hierfür eine physikalische Verbindung zum Kabel hergestellt werden muss. Die gewonnen Daten werden anschließend durch Computer nach bestimmten Stichwörtern durchsucht und verdächtige Nachrichten näher analysiert.
Das Abhören von Kommunikation ist per se nicht verboten: Nachrichtendienste dürfen lauschen, solang dies zum Zweck der Gefahrenabwehr, Bekämpfung von Kriminalität und Drogen-, bzw. Waffenhandel geschieht. Es wird jedoch befürchtet, dass Echelon auch zum Zweck der Wirtschaftsspionage eingesetzt wird – Beweise hierfür gibt es jedoch nicht. Grundsätzlich kann man annehmen, dass Wirtschaftsspionage eher auf „klassische“ Methoden zurückgreift: Einschleusung von Informanten, Anwerben von firmeninternen Mitarbeitern, Datenklau bei Geschäftsreisenden und gezieltes Abhören.
Nichtsdestotrotz hat die EU 2001 Empfehlungen an mittelständische und kleine Unternehmen, Behörden und Privatpersonen ausgesprochen, sensibel mit ihren Daten umzugehen. Einen guten Schutz bietet dabei der Einsatz von Verschlüsselung.

Gruppenpuzzle Echelon, DE-Mail, PGP und Kommunikationsfreiheit

E 2 Wie kann man sich und sein Unternehmen vor Informationsbeschaffung schützen?

In dem „Bericht über die Existenz eines globalen Abhörsystems für private und wirtschaftliche Kommunikation (Abhörsystem ECHELON) (2001/2098 (INI)), S. 17-22“ spricht die EU eine Reihe von Empfehlungen aus, unter anderem:
„ […] „in der Erwägung, dass Sicherheit für Unternehmen nur dann erzielt werden kann, wenn das gesamte Arbeitsumfeld abgesichert sowie alle Kommunikationswege geschützt sind, auf denen sensible Informationen übermittelt werden; dass es ausreichend sichere Verschlüsselungssysteme zu erschwinglichen Preisen auf dem europäischen Markt gibt; dass Privaten dringend zur Verschlüsselung von E-Mails geraten werden muss; dass eine unverschlüsselte Mail gleich einem Brief ohne Umschlag ist; dass sich im Internet relativ benutzerfreundliche Systeme finden, die sogar für den Privatgebrauch unentgeltlich zur Verfügung gestellt werden [...]
29. ersucht die Kommission und die Mitgliedstaaten, geeignete Maßnahmen für die Förderung, Entwicklung und Herstellung von europäischer Verschlüsselungstechnologie und -software auszuarbeiten und vor allem Projekte zu unterstützen, die darauf abzielen, benutzerfreundliche Kryptosoftware, deren Quelltext offen gelegt ist, zu entwickeln; [...]
32. appelliert an die europäischen Institutionen sowie an die öffentlichen Verwaltungen der Mitgliedstaaten, Verschlüsselung von E-Mails systematisch einzusetzen, um so langfristig Verschlüsselung zum Normalfall werden zu lassen;[...]“

Kontrollfragen zu Echelon

1. Wie funktioniert das Echelon-System?
2. Was besagt das UKUSA-Abkommen?
3. Zu welchem Zweck betreiben die UKUSA-Staaten eine globales Abhörsystem?
4. Warum soll „open-source“-Verschlüsselungssoftware besonders gefördert werden und wie hängt dies mit dem Kerckhoff'schen-Prinzip zusammen?
5. Überlege Dir, warum besonders kleine und mittelständische Unternehmen das Ziel von Wirtschaftsspionage sind.

Quellen (Links geprüft am 14.06.10)

Nichtständiger Ausschuss des Europäischen Parlaments, Bericht über die Existenz eines globen Abhörsystems für private und wirschaftliche Kommunikation (Abhörsystem ECHELON) (2001/2098 (INI)): http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//NONSGML+REPORT+A5-2001-0264+0+DOC+PDF+V0//DE
http://de.wikipedia.org/wiki/Echelon
Portfolio des Heise-Verlags über Echelon: http://www.heise.de/tp/r4/special/ech.html

Arbeitsbogen *De-Mail*

Aufgaben

1. Lest den Text auf dem Arbeitsbogen gründlich durch. Versucht, Unklarheiten miteinander zu klären. Natürlich könnt ihr auch den Lehrer fragen!
2. Überprüft Euer Wissen anhand der Kontrollfragen auf den nächsten Seiten.
3. Bereitet Euch darauf vor, einen kurzen Vortrag zu den unten aufgeführten Fragen D 1 und D 2 zu halten. Macht Euch Stichpunkte dazu!
4. Überlegt Euch, welche Nachfragen von Euren Klassenkameradinnen und -kameraden kommen könnten. Bereitet Euch darauf vor, diese zu beantworten. Fertigt Euch dazu Sprechkarten an!
5. Recherchiert nach weiteren Informationen zu den unten aufgeführten Fragen, um diese in Euren Vortrag einbauen zu können!

D 1 Was ist De-Mail?

Logo des E-Government Projekts "De-Mail"

2009 beschloss die Bundesregierung die Einrichtung eines so genannten elektronischen „Bürgerportals“. De-Mail ist ein Bestandteil dieses Portals. Zweck des Bürgerportals soll es sein, eine sichere Kommunikationsstruktur zwischen Bürgern, Behördern, Banken, Versicherungen und Firmen zur Verfügung zu stellen. Die Notwendigkeit eines solchen Portals beruht auf folgenden Überlegungen:
Eine offizielle Kommunikation mit Behördern, Versicherungen und Banken ist nur über Briefpost zulässig, da E-Mails bisher nicht die nötigen Anforderungen an eine sichere Kommunikation erfüllen. Da alle genannten Institutionen jedoch mit elektronischen Mitteln arbeiten, müssen schriftliche Eingänge in elektronische Daten und Ausgänge in Briefpost umgewandelt werden. Der dabei entstehende Arbeitsaufwand kostet natürlich Geld. Häufig müssen Briefe an Behörden per Einschreiben (dabei bestätigt der Adressat den Empfang der Post) gesendet werden – ein solches Prinzip gibt es zur Zeit bei der E-Mail-Kommunikation nicht. Firmen wie z.B. Mailprovider können sich beim Bürgerportal akkreditieren lassen und sind dann verpflichtet, die Sicherheitsstandards der Bundesregierung umzusetzen. Einer dieser Standards sieht vor, dass der E-Mail-Verkehr mittels asymmetrischer Verschlüsselungsverfahren chiffriert werden muss. Auch wenn die Einrichtung eines Online-Bürgerportals häufig begrüßt wird, gibt es erhebliche Bedenken von Seiten der Datenschützer über die konkrete Umsetzung des Projekts. Ein System, das sensible Daten von Bürgerinnen und Bürgern verwaltet, empfängt und versendet, muss vollständig gegenüber Missbrauch von außen – aber auch von innen – abgesichert sein. Das ursprüngliche Sicherheitskonzepts des Bürgerportals wurde auf der Konferenz der Datenschutzbeauftragten des Bundes und der Länder 2009 als unzureichend bezeichnet.

D 2 Was sind die Hauptkritikpunkte an dem geplanten „Bürgerportal“?

Am 29. April 2009 veröffentlichten die Datenschützer des Bundes und der Länder eine Stellungnahme zum geplanten Bürgerportal, in der sie Kritik an der Umsetzung äußerten (in Auszügen):

„Der Entwurf sieht vor, dass nur akkreditierte Anbieter Portale betreiben dürfen. Voraussetzung für die Akkreditierung darf nicht allein der Nachweis der technischen und administrativen Sicherheit, sondern muss auch die tatsächliche Einhaltung datenschutzrechtlicher Standards sein. […]
Die Sicherung der Vertraulichkeit, Integrität und Authentizität von Nachrichteninhalten soll lediglich durch eine Verschlüsselung auf dem Transport zwischen den Dienstanbietern und durch die Sicherung des Zugangs zu den Bürgerportalen erfolgen. Es muss jedoch sichergestellt sein werden, dass Nachrichten auch bei den Portalbetreibern nicht durch Dritte gelesen oder verändert werden können. […]
Die nach der Gesetzesbegründung […] mögliche unsichere Anmeldung mit Passwort wird abgelehnt.
Der Nachweis der Absenderin oder des Absenders soll lediglich durch Anmeldung am Bürgerportal erfolgen. […] So könnten Zugangsdaten beschafft und widerrechtlich dazu verwendet werden, De-Mails zu versenden, empfangene De-Mails zu unterdrücken, zu verzögern und zu verändern […]. Deshalb sind zusätzliche Sicherungsmaßnahmen vorzusehen. [...]“

Quelle: Entschließung der Konferenz der Datenschutzbeauftragten des Bundes und der Länder vom 16. April 2009 - Datenschutz beim vorgesehenen Bürgerportal unzureichend,
URL: http://www.datenschutz-berlin.de/attachments/580/Anlage.pdf?1239962678

Kontrollfragen

1. Aus welchen Gründen soll das Bürgerportal eingeführt werden?
2. Das Motto von De-Mail lautet „So einfach wie E-Mail, so sicher wie Briefpost – verschlüsselt, authentisch, nachweisbar“ Wie hängt dieses Motto mit den im Unterricht aufgestellten Anforderungen an sichere Kommunikation zusammen?
3. Für die oben geäußerten Kritikpunkte gibt es Lösungen! Welche fallen Dir ein?

Quellen (Links geprüft am 14.06.10)

http://de.wikipedia.org/wiki/De-Mail
Artikel des Heise-Verlags: http://www.heise.de/security/meldung/Bundeskabinett-verabschiedet-Buergerportalgesetz-205356.html
IT-Sicherheit-Blog: http://itsicherheit.wordpress.com/2009/04/18/datenschutzer-fordern-nachbesserungen-beim-burgerportal-gesetz/
Offizielle Website De-Mail: http://www.de-mail.de/
(dort findest Du auch weiterführende Links zu offiziellen Dokumenten Regierung!)

Arbeitsbogen *Kommunikationsfreiheit*

Aufgaben

1. Lest den Text auf dem Arbeitsbogen gründlich durch. Versucht, Unklarheiten miteinander zu klären. Natürlich könnt ihr auch den Lehrer fragen!
2. Überprüft Euer Wissen anhand der Kontrollfragen auf den nächsten Seiten.
3. Bereitet Euch darauf vor, einen kurzen Vortrag zu den unten aufgeführten Fragen K 1 und K 2 zu halten. Macht Euch Stichpunkte dazu!
4. Überlegt Euch, welche Nachfragen von Euren Klassenkameradinnen und -kameraden kommen könnten. Bereitet Euch darauf vor, diese zu beantworten. Fertigt Euch dazu Sprechkarten an!
5. Recherchiert nach weiteren Informationen zu den unten aufgeführten Fragen, um diese in Euren Vortrag einbauen zu können!

K 1 Wie und warum können Staaten E-Mail-Verkehr kontrollieren?

Karikatur: "Tweet-Hunting in Iran" vom User Rodrigo. [Quelle: toonpool.com]

Im Deutschen Grundgesetz (Art.5 Abs.1) wird jedem Bürger das Recht auf Kommunikationsfreiheit zugeschrieben. Dieses Recht ist die Grundlage der Meinungs-, Informations- und Medienfreiheit. Dieses Recht zu besitzen ist keine Selbstverständlichkeit – in einigen Ländern ist ein solches Recht nicht vorhanden. Wo ein Recht auf Kommunikationsfreiheit fehlt, ist es dem Staat nicht verboten, E-Mail-Verkehr zu kontrollieren.
Die Kontrolle funktioniert dabei wie folgt: Jeglicher E-Mail-Verkehr wird über einen (oder mehrere) Rechner, so genannte Proxies weitergeleitet. Wie Du bereits im Unterricht gesehen hast, kann man dort den Datenverkehr computergesteuert analysieren und somit auch den Inhalt von E-Mails lesen. Darüber hinaus kann man die E-Mails sogar verändern, wie es im Fall der chinesischen Falun-Gong-Sekte passierte: Dateianhänge von Falun Gong-E-Mails wurden mit Spionagesoftware versetzt, so dass die Empfänger der E-Mails beim Öffnen gleichzeitig die ungewollte Software installierten.
Die Verschlüsselung von Nachrichten ist in China keine Möglichkeit, um sichere Kommunikation zu betreiben: Bereits das Verwenden von nicht genehmigter Verschlüsselungssoftware oder -hardware ist verboten und kann bestraft werden. Weiterhin wollen viele Autoren, dass eine Nachricht möglichst viele Menschen erreicht, um die eigene Meinung zu verbreiten. So sind etwa die chinesischen Unterzeichner der Charta 08 bewusst das Risiko eingegangen, sich durch die Veröffentlichung des Dokuments im Internet einer Strafverfolgung auszusetzen. Eine der zentralen Forderungen der Charta 08 ist das Recht auf freie Meinungsäußerung. Bereits vor der Veröffentlichung des Dokuments wurden Unterzeichner, wie etwa der Dissident Liu Xiabo verhaftet.

K 2 Was bringt Dir die Kommunikationsfreiheit?

Folgende Zitate drehen sich rund um das Thema Kommunikationsfreiheit. Nutze die Zitate als Anregung, um mit Deinen Gruppenmitgliedern über den Nutzen von Kommunikationsfreiheit zu diskutieren! Überlegt dabei auch, warum ein Staat ein Interesse an der Kontrolle von Kommunikation haben kann.

"Jeder hat das Recht, seine Meinung in Wort, Schrift und Bild frei zu äußern und zu verbreiten und sich aus allgemein zugänglichen Quellen ungehindert zu unterrichten [...] Eine Zensur findet nicht statt." (Art. 5 Abs. 1 GG)

„Freiheit. Ich verstehe das Wort nicht, weil ich sie nie entbehren musste.“ (Jonas T. Bengtsson, Die Hölle ist, mit sich allein zu sein, in: Wolfgang Klein (Hg.), Young Euro Connect 2006.“)

„China hat eine Mauer gebaut. Heutzutage baut man Mauern nicht mehr nur über Hügel und Felder, sondern auch im Internet. Die Menschen in China leben hinter einer Mauer, und nur was die chinesische Regierung erlaubt, darf durch diese Mauer hindurch ins Land kommen.“ (Statement auf der Webseite des Chaos-Computer-Club, URL: http://chinesewall.ccc.de/index-de.html)

Kontrollfragen

1. Wie kann ein Staat den gesamten E-Mail-Verkehr des eigenen Landes kontrollieren?
2. Was ist die Charta 08?
3. Warum dürfen Staaten den (elektronischen) Postverkehr kontrollieren, warum ist dies in Deutschland nicht der Fall? Kannst Du Dir vorstellen, wann auch der deutsche Staat E-Mails mitlesen darf?
4. Was will der Künstler „Rodrigo“ mit seiner Karikatur aussagen?
5. Warum ist das Verschlüsseln von Nachrichten keine Lösung um mangelnde Kommunikationsfreiheit auszugleichen?

Quellen (Links geprüft am 14.06.10)

http://de.wikipedia.org/wiki/Internetkontrolle_in_der_Volksrepublik_China
http://de.wikipedia.org/wiki/Charta_08
http://chinesewall.ccc.de/index-de.html
(Webseite des Chaos Computer Club über die Chinesische Internetkontrolle)
Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson, Ignoring the Great Firewall of China, URL: http://www.cl.cam.ac.uk/~rnc1/ignoring.pdf
(Ausführliche Informationen in englischer Sprache über die Technik der Chinesischen Internetkontrolle)
http://futurezone.orf.at/stories/255642/
(Blog des österreichischen Rundfunksenders ORF)

Arbeitsbogen *PGP* („Pretty Good Privacy“)

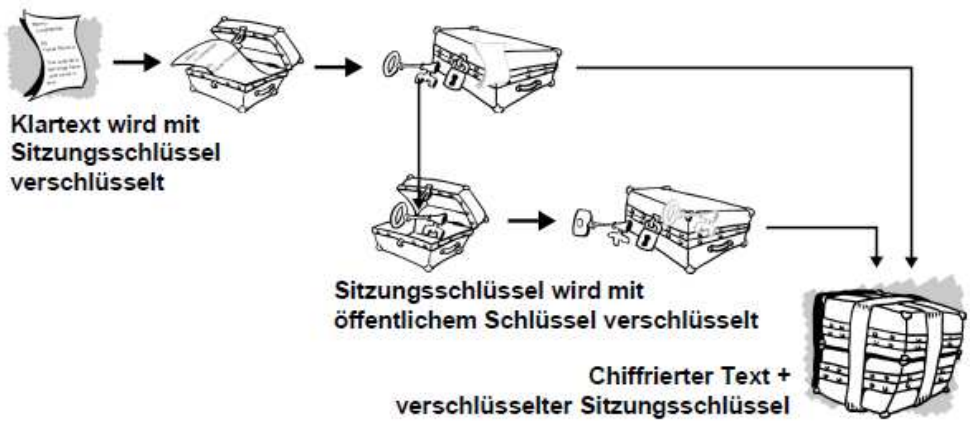
Aufgaben

1. Lest den Text auf dem Arbeitsbogen gründlich durch. Versucht, Unklarheiten miteinander zu klären. Natürlich könnt ihr auch den Lehrer fragen!
2. Überprüft Euer Wissen anhand der Kontrollfragen auf den nächsten Seiten.
3. Bereitet Euch darauf vor, einen kurzen Vortrag zu den unten aufgeführten Fragen P 1 und P 2 zu halten. Macht Euch Stichpunkte dazu!
4. Überlegt Euch, welche Nachfragen von Euren Klassenkameradinnen und -kameraden kommen könnten. Bereitet Euch darauf vor, diese zu beantworten. Fertigt Euch dazu Sprechkarten an!
5. Recherchiert nach weiteren Informationen zu den unten aufgeführten Fragen, um diese in Euren Vortrag einbauen zu können!

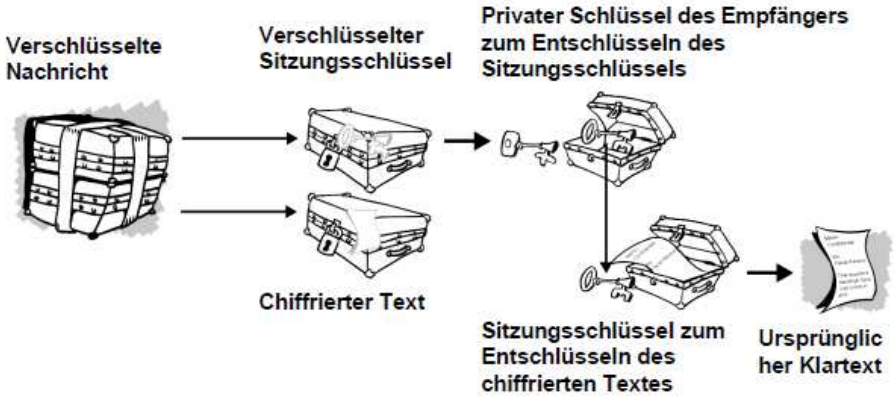
P 1 Wie erschafft PGP Vertraulichkeit?

Logo des des frei verfügbaren „Open-Source“- Projekts GnuPG [Quelle: gnupg.org]

PGP („Pretty Good Privacy“) ist ein hybrides Verschlüsselungssystem, das es ermöglicht, beim Versenden einer Nachricht Authentizität, Integrität und Vertraulichkeit zu gewährleisten. Es wurde 1991 vom US-Amerikaner Phil Zimmermann entwickelt und ist mittlerweile weit verbreitet. PGP ist nutzt ein asymmetrisches Verschlüsselungsverfahren, wie Du es bereits im Unterricht kennen gelernt hast: Jeder Teilnehmer besitzt einen öffentlichen und einen privaten Schlüssel. Möchte man jemanden eine Nachricht schicken, so verschlüsselt man diese mit dessen öffentlichen Schlüssel. Nur der Empfänger besitzt den privaten Schlüssel und somit ist nur er in der Lage, die Nachricht zu entschlüsseln. Auf diese Art und Weise entgeht man der Problematik, den geheimen Schlüssel austauschen zu müssen!
Genau genommen, wird bei PGP die Nachricht nicht mit einem asymmetrischen Verfahren verschlüsselt. Das Verfahren nutzt hier einen kleinen Trick: Der Klartext wird mit einem so genannten Sitzungsschlüssel symmetrisch verschlüsselt. Anschließend wird nur dieser Schlüssel mit einem asymmetrischen Verfahren verschlüsselt. Der Empfänger entschlüsselt anschließend zunächst den Schlüssel und damit dann die eigentliche Nachricht. Klingt kompliziert, ist aber einfach:



Verschlüsseln einer Nachricht mit PGP. [Quelle: Handbuch PGP- Eine Einführung in die Kryptographie, ftp://ftp.pgpi.org/pub/pgp/6.5/docs/german/IntroToCrypto.pdf], S8]

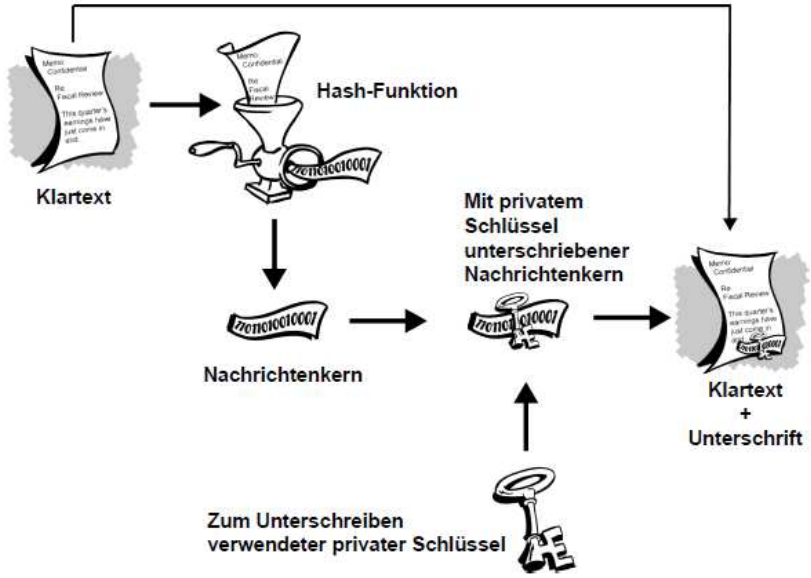


Entschlüsseln einer Nachricht mittels PGP. [Quelle: Handbuch PGP- Eine Einführung in die Kryptographie, ftp://ftp.pgpi.org/pub/pgp/6.5/docs/german/IntroToCrypto.pdf, S8]

Durch diese Methode erspart man sich die sehr zeitintensive Verschlüsselung des gesamten Nachrichtentextes durch ein asymmetrisches Verfahren. Nur den Sitzungsschlüssel asymmetrisch zu verschlüsseln geht hingegen sehr schnell!

P 2 Wie erschafft PGP Authentizität und Integrität?

Neben der Verschlüsselung besitzt PGP noch zwei weitere Verfahren: Die digitale Signatur und ein Hash-Verfahren. Diese dienen dazu, sicher zu stellen, dass 1. die Nachricht auch wirklich von demjenigen stammt, der als Absender genannt ist und 2. dass die Nachricht auf dem Transportweg nicht verändert wurde.Eine Hash-Funktion erzeugt aus einem großen Text einen so genannten Nachrichtenkern, der weitaus kleiner als der eigentliche Text. Dieser Nachrichtenkern wird anschließend mit dem privaten Schlüssel des Senders „unterschrieben“. Verändert man die ursprüngliche Nachricht auf dem Transportweg, so verändert sich auch der Nachrichtenkern. Beim Entschlüsseln fällt dies PGP auf und es meldet die Veränderung. Der unterschriebene Nachrichtenkern kann nur mit dem öffentlichen Schlüssel des Senders wieder entschlüsselt werden. Dadurch wird sichergestellt, dass die Nachricht auch wirklich von dem stammt, der sie gesendet hat.



[Quelle: Handbuch PGP- Eine Einführung in die Kryptographie, ftp://ftp.pgpi.org/pub/pgp/6.5/docs/german/IntroToCrypto.pdf, S12]

Gruppenpuzzle Echelon, DE-Mail, PGP und Kommunikationsfreiheit

Kontrollfragen

1. Warum braucht man bei der asymmetrischen Verschlüsselung einen privaten und einen öffentlichen Schlüssel?
2. Was ist ein hybrides Verschlüsselungsverfahren?
3. Was ist eine Hash-Funktion?
4. Warum heißt PGP „nur“ Pretty Good Privacy? Was ist GnuPG?
5. Warum entwickelte Phil Zimmermann PGP und warum war die US-Regierung damit zunächst nicht einverstanden? (Antwort dazu ist nicht im Text – du musst sie selbst recherchieren!)

Quellen (Links geprüft am 14.06.10)

Handbuch PGP – eine Einführung in die Kryptographie, URL:
ftp://ftp.pgpi.org/pub/pgp/6.5/docs/german/IntroToCrypto.pdf
http://de.wikipedia.org/wiki/Pretty_Good_Privacy
http://www.zdnet.de/news/wirtschaft_sicherheit_security_pgp_erfinder_philip_zimmermann_im_interview_story-39001024-2103010-1.htm
(Interview mit Phil Zimmermann)
Stefan Krempl, „Krieg um Krypto“, Spiegel Online 1998. URL:
http://www.spiegel.de/netzwelt/tech/0,1518,13719,00.html
„Sichere Mailverschlüsselung ohne Umtriebe“, Artikel in der FAZ 12.04.2005, URL:
http://www.faz.net/s/Rub4C34FD0B1A7E46B88B0653D6358499FF/Doc~E357B3B30B1E348128E2FB3B18070F685~ATpl~Ecommon~Scontent.html