Kryptographie im Informatikunterricht der Sekundarstufe I

Arno Pasternak
Fritz-Steinhoff-Gesamtschule Hagen

16. April 2006

Inhaltsverzeichnis

Zusammenfassung

Schüler und Schülerinnen wissen heute im Allgemeinen, dass Informationen im Netz nicht sicher sind, sondern eventuell „abgehört“ werden können. Daraus folgt die Forderung, Daten zu verschlüsseln.

Was das bedeutet und welches Verschlüsselungsverfahren für welchen Zweck das Geeignete ist, ist für Schüler und Schülerinnen nicht selbstverständlich klar. Ebenso wird dem Schüler bzw. der Schülerin nicht deutlich, welche Ideen und Ziele sich hinter welchem Verfahren verbirgt, wenn er oder sie in irgendeinem Anwendungsprogrammen irgendwelche „Knöpfe“ drückt.

Erstaunlicherweise ist es nun gar nicht notwendig, dass die Schüler grossartig mathematische Theorien erarbeiten müssen, um die Ideen der Verschlüsselungen zu verstehen. Gerade im Gegenteil: Ohne viel Mathematik kann ein Schüler von Heute lernen, was beim Verschlüsseln mit seinen Daten beim Sender und Empfänger passiert.

In diesem Vortrag zeige ich, wie man im Informatikunterricht in der Sekundarstufe I wesentliche Verfahren der Kryptographie behandeln kann.

Dabei werden der Reihe nach die Verfahren

- Caesar
- Vigenere
- Public Key
- Signieren von Daten

bearbeitet.

Diese Unterrichtssequenz sollte meines Erachtens eine Pflichteinheit im Unterricht von Schülern und Schülerinnen der Sek I darstellen. [1]

1 Einbettung in den Informatik-Unterricht der Sek I

1.1 Voraussetzungen

Sinnvoll ist es, dass die Schülerinnen und Schüler die Verletzlichkeit von Computersystemen in Netzen (im Gegensatz z.B. zu Telefonnetzen) kennen. Je nach Unterrichtsablauf kann dies beispielsweise durch folgende Vorkenntnisse bekannt sein:

- In einer technisch orientierten Einheit haben die Schüler z.B. an Hand einer Ampelsteuerung erfahren, wie diese durch Computer mit Hilfe eines Interfaces funktionieren. Anschliessend wurden zwei Computer gekoppelt, die ihre Steuerungen mit dieser Kopplung syncronisieren. Dann ist es nur noch ein kleiner Schritt, dass statt Syncronisierungsimpulsen Daten zwischen diesen beiden Computern übertragen werden können. Sind erst zwei Computer vernetzt, ist der Übergang zu einem Netz mit mehreren Computern (entweder praktisch oder theoretisch) leicht möglich. Den Schülern wird dabei schnell klar, dass Informationen in einem Netz nicht nur allein dem Empfänger zugänglich sind.
- Die Schüler haben beispielsweise HTML-Seiten erstellt, die dann per FTP auf einen Web-Server übertragen wurden. Dabei sind grundsätzliche Strukturen der Kommunikation in einem Netz gleichartiger Knoten wie z.B. Client-Server-Strukturen besprochen worden. Ebenso ist den Schülern bekannt, dass die Kommunikation zwischen den beteiligten Knoten eine textbasierte Kommunikation ist[2]. Sie haben sinnvollerweise diese Kommunikation bei einer derartigen FTP-Sitzung mit einem textbasierten Clienten durchgeführt. Ebenso ist bekannt, dass der Zugriff auf einen Host über grössere Entfernungen über eine Terminal- oder eine Telnet-Sitzung stattfinden kann. Die Gefährdung des Abhörens dieser FTP- und/oder Telnet-Sitzung in einem Netz ist thematisiert worden. Eventuell sind Alternativen durch verschlüsselte Protokolle wie SSH angesprochen bzw. praktiziert worden.

1.2 Ziel der Unterrichtseinheit Kryptographie

Nachdem in den vorherigen Einheiten deutlich geworden ist, dass Verschlüsselung in Computernetzen eine sinnvolle und notwendige Sache ist, soll nun erarbeitet werden, welche verschiedenen Techniken zum Verschlüsseln möglich sind. Dabei geht es nicht darum, dass die Schülerinnen und Schüler ein Verfahren als das allein Seligmachende kennenlernen, sondern dass die Auswahl eines geeigneten Verfahrens von der eingesezten Technik als auch von der Art der Kommnikationspartner abhängig ist. Neben der aktuellen Bedeutung der Verschlüsselung in Netzen soll dabei auch die historische Entwicklung der Kryptographie nicht zu kurz kommen.

1.3 Der Computer als Verschlüsselungsautomat

Der Computer als technische Realisierung des universellen Automatens ist natürlich auch ein wunderbarer Ver- und Entschlüsselungsautomat. Generationen von Chiffrierern und Dechiffrierern wären froh gewesen, eine nur annähernd so gute Maschine wie den Computer gehabt zu

[2] zur Bedeutung der Nutzung von textbasierter Clienten siehe auch [Pasternak 2005].

haben. Die phantastische „Enigma“ ist ein Nichts gegen den Computer als Chiffriermaschine. Meines Erachtens ist es für Schüler wichtig, immer wieder die Mechanismen eines solchen Automaten selbst kennen zu lernen und auszuprobieren. Dazu ist Programmierung unablässig. Auch ein Algorithmus zur Ver- bzw. Entschlüsselung ist erst dann voll verstanden, wenn er automatisiert werden kann. Wenn man ihn aber automatisieren, sprich programmieren kann, dann soll man bzw. der Schüler es auch (zumindest des Öfteren) tun.

In der Fritz-Steinhoff-Gesamtschule Hagen wird im Unterricht der Sekundarstufe I seit den 80-Jahren die Programmiersprache COMAL verwendet. Sie ist eine aus didaktischen Gründen entwickelte Sprache, die B. Christensen aus Dänemark entwickelt hat. Da sie leider nicht weiterentwickelt wurde, enthält sie nicht alle modernen Konstrukte. Für den Anfänger ist sie meines Erachtens auch heute noch eine gute Wahl.
Da in unserer Schule die Informatiker bei der Zuordnung der Computerräume immer den Kürzeren ziehen (bei 5 Unterrichtsräumen mit Computerausstattung und ca. 200 Computern in der Schule!), bleibt uns zumeist keine andere Wahl, da wir fast immer nur im einzigen nicht Linuxfähigen Raum verharren müssen und daher auf DOS und Win 3.1-Programm(-umgebungen) angewiesen sind[3].

2 Die Unterrichtseinheit Kryptographie

Einstieg

Die Schülerinnen und Schüler werden aufgefordert, einen vorgegebenen relativ kurzen Satz wie beispielsweise „Bald beginnt die Fussball-WM“ z.B. in Partnerarbeit nach einem selbst zu findenden Verfahren zu verschlüsseln. Die dabei entstehenden codierten Sätze werden an die Tafel geschrieben. Alle Schüler sind nun aufgefordert, die codierten Sätze ihrer Mitschüler zu dechiffrieren.

Es ist erstaunlich, welche unterschiedlichen und teilweise phantasievollen Verfahren dabei auftreten. Bei der Analyse wird dann den Schülerinnen und Schülern oft relativ schnell klar, dass die Verfahren sich in die Klassen

- Ersetzungsverfahren, bei denen die Zeichen des Klartextes durch andere Zeichen derselben oder einer anderen Zeichenmenge ersetzt werden, sowie die
- Tauschverfahren, bei denen die Reihenfolge der Zeichen des Klartextes verändert wird oder
- eine Kombination dieser beiden Prinzipien

einteilen lassen. Sie stellen auch fest, dass manche ihrer Verfahren zwar eine eindeutige Verschlüsselung zulassen, aber nicht mehr eindeutig entschlüsselt werden können. Eine Verschlüsselung ohne eine Entschlüsselung ist in den meisten Fällen aber nicht erwünscht.

Bei einer weiteren Analyse wird den Schülern deutlich, dass eine maschinelle Bearbeitung von Codierungen erheblich vereinfacht wird, wenn man sich auf Buchstaben und/oder Ziffern bzw. Zahlen als Zeichen beschränkt.

2.1 Das Caesar-Verfahren

Es ist naheliegend, dass Militärs schon sehr früh Interesse an der Chiffrierung von Texten hatten. Es ist daher auch nicht sehr verwunderlich, wenn eines der einfachsten Verfahren auf *Caesar* zurückgeführt wird. Ob dies tatsächlich der historischen Wahrheit entspricht, ist mir nicht bekannt. Zumindest wird behauptet, dass Caesar vorgeschlagen hat, einen Buchstaben im Klartext durch den Buchstaben zu ersetzen, der im Alphabet drei Stellen hinter ihm vorkommt.

[3] Es versteht sich von selbst, dass sich jede andere Programmiersprache genauso oder nach Standpunkt des Lesers sogar besser für den Unterricht in der Sekundarstufe I im Allgemeinen und für diese Einheit im Speziellen eignet.

Wenn wir dieses Verfahren mit unserem Automaten - dem Computer - durchführen wollen, ist sofort einleuchtend, dass wir als Zeichenvorrat nicht das „normale“ Alphabet, sondern den ASCII-Zeichensatz verwenden. Falls nicht geschehen, bietet es sich hier an, den Schülern und Schülerinnen diesen Zeichensatz und seine Bedeutung (auch in seiner Bedeutung auf das Erstellen von ASCII-Texten z.B. mit einem Editor und formatierten Texten z.B. mit einer Textverarbeitung) zu vermitteln.

Die Programmierung des Verfahrens ist nicht sehr aufwändig. Auch die Schüler, die teilweise beim Programmieren grosse Schwierigkeiten haben, können zumindest diesen Programmteil nachvollziehen und zumeist auch erklären. Betrachten wir nun den Kern des Programms mit den praktisch identischen Prozeduren zur Ver- und Entschlüsselung nach Caesar[4]:

```
..
0550 PROC caesar_verschlüsseln
0560   stelle:=1
0570   REPEAT
0580     asciizahl:=ORD(satz$(:stelle:))
0590     asciizahl:=asciizahl+verschlüsselungszahl
0600     satz$(:stelle:):=CHR$(asciizahl)
0610     stelle:=stelle+1
0620   UNTIL stelle>LEN(satz$)
0630 ENDPROC caesar_verschlüsseln
0640 //
0650 //
0660 PROC caesar_entschlüsseln
0670   stelle:=1
0680   REPEAT
0690     asciizahl:=ORD(satz$(:stelle:))
0700     asciizahl:=asciizahl-verschlüsselungszahl
0710     satz$(:stelle:):=CHR$(asciizahl)
0720     stelle:=stelle+1
0730   UNTIL stelle>LEN(satz$)
0740 ENDPROC caesar_entschlüsseln
..
```

Diese Prozeduren enthalten schon die Möglichkeit, die Verschiebung der Buchstaben im verschlüsselten Satz nicht nur um drei Positionen, sondern beliebig durchzuführen. Die Tauglichkeit dieses Verfahrens zeigt sich aber nicht im Programmieren und der korrekten Funktionsweise des Programmes bzw. der Prozeduren. Um diese zu testen, erhalten die Schüler von mir einen vorgegeben Satz wie z.B.

```
T]S[XRW/Xbc/^bcTa]=
```

Selbst wenn wie in diesem Fall den Schülern bekannt ist, dass es sich um die Caesar-Verschlüsselung handelt, dauert es für Ungeübte geraume Zeit, diesen Satz zu entschlüsseln. Umso schwieriger wird es, wenn nur Sender und Empfänger der Nachricht wissen, dass es sich um die Caesarverschlüsselung handelt.

Können wir nun den Computer als Entschlüsselungsautomaten einsetzen, sieht die Sache schon ganz anders aus. Durch mehr oder weniger systematisches Probieren mit der Verschlüsselungszahl kommt man sehr schnell darauf, dass es sich um den Satz

```
ENDLICH IST OSTERN.
```

[4]In den hier vorgestellten Programmen bzw. Programmauszügen wird mit globalen Variablen wie z.B. satz$ gearbeitet. Es ist meines Erachtens strittig, ob man auch in der Sekundarstufe I bei der prozeduralen Programmierung völlig auf globale Variabeln verzichten soll. Tendenziell neige ich zu einer frühzeitigen Einführung von Schnittstellen. Eine objektorientierte Schreibweise ermöglicht COMAL nicht.

handelt. Noch deutlicher wird es, wenn die manuelle Veränderung der Verschlüsselungszahl in einer Wiederholungsanweisung automatisch variiert wird und die Lösung praktisch sofort auf dem Bildschirm steht. Das zuvor zumindest scheinbar teilweise geeignete Verfahren wird sofort in seiner Beschränktheit deutlich.

Ein weiteres Problem taucht bei folgendem „Versuch“ auf: Die Schüler sollen sich kurze Mitteilungen (eine Art SMS) zusenden. Diese Mitteilungen werden mit der Tastatur in den Computer eingetippt und als Dateien auf einem gemeinsamen Verzeichnis der Schülergruppe hinterlegt. Dabei wird vereinbart, dass als Dateiname eine Kombination von Empfänger und Absender verwendet wird. Z.B. meint N311N323, dass vom Computer N323 eine Nachricht an den Computter N311 gesendet worden ist. Diese Nachrichten sind nun von allen Schülerinnen und Schülern einsehbar und sollen mit dem Caesarverfahren verschlüsselt werden. Die Schüler erhalten von mir ein Programm, in dem ihre Prozeduren zur Ver- und Entschlüsselung enthalten sind und das um alle weiteren gewünschten Prozeduren[5] ergänzt worden ist.

Wir erhalten damit folgendes sich selbst erklärende Hauptprogramm:

```
0010 //Caesar-Verschluesselungssystem
0020 //
0030 REPEAT
0040   PRINT "Satz eingeben:  1"
0050   PRINT "Satz ausgeben:  2"
0060   PRINT "Satz laden   :  3"
0070   PRINT "Satz speichern: 4"
0080   PRINT "Satz verschlüsseln:   5"
0090   PRINT "Satz entschlüsseln:   6"
0100   PRINT "Meldungen:      7"
0101   PRINT "Verschlüsselungszahl: 8"
0110   PRINT "Ende:           9"
0111   PRINT "Ende:           9"
0120   PRINT
0121   PRINT
0130   PRINT "Eingabe"
0140   INPUT wahl
0150   IF wahl=1 THEN satzeingabe
0160   IF wahl=2 THEN satzausgabe
0170   IF wahl=3 THEN satzladen
0180   IF wahl=4 THEN satzspeichern
0190   IF wahl=5 THEN caesar_verschlüsseln
0200   IF wahl=6 THEN caesar_entschlüsseln
0210   IF wahl=7 THEN meldungen
0211   IF wahl=8 THEN verschlüsselungszahl_eingeben
0220 UNTIL wahl=9
0230 //
0240 //
```

Mit diesem Versuch haben wir eine Struktur, die den Schülern verdeutlicht, welche Probleme auftreten, wenn viele Partner mit vielen anderen Partnern kommunizieren wollen. Es treten Probleme auf, die z.B. Militärs bei ihrer Kommunikation nicht kennen. Nach einer kurzen Arbeitsphase tritt eine erhebliche Unruhe auf. Verschlüsselungszahlen werden gesucht, teilweise durch den Raum gerufen oder durch kurze „Besuche“ bei Kommunikationspartnern mitgeteilt. Natürlich wird die schwache Verschlüsselung zum Mitschnüffeln fremder Nachrichten schnellstens geknackt. Anschliessend bei der Auswertung dieser Phase können leicht alle Probleme einer solchen Kommunikation aufgearbeitet werden.

[5] Der geneigte Leser möge entschuldigen, dass ich hier keine objektorientierte Sprachweise verwende. Mir erscheint sie hier als unnötig aufgesetzt.

Die hier eingeführte Arbeitsweise kann nun ohne Weiteres mit den weiteren Verfahren immer wieder aufgenommen werden, um die Tauglichkeit von Verfahren zu testen.

2.2 Das Vigenere-Verfahren

Wenden wir uns nun der Verbesserung des Caesar-Verfahrens zu, das *Blaise de Vigenere* (1523-1596) zugeschrieben wird. Die Idee ist so einfach wie weitreichend: Statt jedes Zeichen gleichartig zu bearbeiten, werden die Zeichen unterschiedlich durch ein anderes Zeichen ersetzt. Naheliegend ist, entsprechend eines Verschlüsselungswortes- bzw. -Satzes die Zeichen des Klartextes zu verschlüsseln. Dieses Verschlüsselungswort kennen natürlich nur Sender und Empfänger der Nachricht.

Auch bei diesem Verfahren handelt es sich um ein symmetrisches Verfahren, das heisst: Ver- und Entschlüsselung werden nach denselbem Verfahren durchgeführt. Die Qualität der Verschlüsselung hängt im Wesentlichen von der Länge des Verschlüsselungsatzes ab. Es versteht sich von selbst, dass die Sicherheit durch ein häufiges Wechseln des Verschlüsselungswortes erhöht wird. Beispielsweise können die ersten Sätze der Seiten eines Literaturwerkes als Codewort verwendet werden. Verwendet wird die Seite mit der Nummer des entsprechenden Tages im Jahr. Beispielsweise wird am 10. Februar der erste Satz der Seite 41 der Schwarte XYZ verwendet.

In unserem Programm müssen wir nur die Ver- und Entschlüsselungsprozeduren ersetzen und erhalten beispielsweise für die Verschlüsselung:

```
0550 PROC vigenere_verschlüsseln
0560   stelle:=1
0561   codestelle:=1
0570   REPEAT
0580     asciizahl:=ORD(satz$(:stelle:))
0581     verschlüsselungszahl:=ORD(codewort$(:codestelle:))
0590     asciizahl:=asciizahl+verschlüsselungszahl
0600     satz$(:stelle:):=CHR$(asciizahl)
0610     stelle:=stelle+1
0611     codestelle:=codestelle+1
0612     IF codestelle>LEN(codewort$) THEN codestelle:=1
0620   UNTIL stelle>LEN(satz$)
0630 ENDPROC vigenere_verschlüsseln
```

Es ist offensichtlich, dass eine Entschlüsselung ohne Kenntnis des Verschlüsselungwortes nur noch für Experten möglich sein kann. Interessant ist nun, wie die Schüler die Begrenztheit dieses Verfahrens in einer Kommunikationsstruktur von vielen Teilnehmern mit vielen anderen Ad-Hoc-Teilnehmern erkennen. Wie schon beim Caesar-Verfahren erläutert, sollen sich die Schülerinnen und Schüler verschlüsselte Kurzmitteilungen zusenden. Es zeigt sich, dass oft eine noch hektischere „Nebenkommunikation“ als bei der Anwendung des Caesar-Verfahrens stattfindet. Die durch die einfache Caesar-Verschlüsselung codierten Sätze hatten einige Schüler durch systematisches Probieren decodiert und daher keine weiteren Informationen von ihren Partnern verlangt. Dies geht nun nicht mehr und bei unseren 15 Computerarbeitsplätzen im Unterrichtsraum führt dies zu einer intensiven Kommunikation über alle Kanäle.

Bei der Auswertung fällt es den Schülerinnen und Schülern leicht, die Problematik dieses Verfahrens in einer heutigen vernetzten Struktur festzustellen: Dieses Verfahren ist nur sicher anzuwenden, wenn die Codeworte auf einem anderen Kanal als das Computernetz übertragen werden. Zudem muss dieser Kanal noch relativ schnell sein, damit i.A. eine schnelle Kommunikationsaufnahme ermöglicht wird.

Für die Militärs der letzten Jahrhunderte und/oder die Kommunikation zwischen ausgewählten Partnern wie beim „Roten Draht“ zwischen Moskau und Washington ist dies sicher ein ge-

eignetes System[6], jedoch nicht für die heutige Massenkommunikation in einer vernetzten Welt.

2.3 Public-Key - Verfahren

Die oben erwähnten Verfahren finden sich in abgewandelter Form in mancher Literatur. Beim Public-Key-Verfahren versuchen jedoch die meisten Autoren, spezielle Varianten mit hohem mathematischen Aufwand vorzustellen. Die eigentliche Idee dieser Verfahren aus kryptografischer Sicht wird dabei nicht deutlich und sind für Schüler in der Sekundarstufe I nicht nachvollziehbar.

Ich verwende daher ein einfaches Public-Key-Verfahren, damit die Schüler die Prinzipien verstehen können. Dieses Verfahren arbeitet mit Verschlüsselungszahlen. In unserem Programm werden die entsprechenden Prozeduren ersetzt durch:

```
1020 PROC publickeyverschlüsseln
1030   //
1040   internercode$:="528"
1050   stelle:=1
1060   codestelle:=1
1070   internercodestelle:=1
1080   REPEAT
1090     asciizahl:=ORD(satz$(:stelle:))
1100     codezahl:=ORD(code$(:codestelle:))-48
1110     internercodezahl:=ORD(internercode$(:internercodestelle:))-48
1120     summencodezahl:=codezahl+internercodezahl
1130     IF summencodezahl>9 THEN summencodezahl:=summencodezahl-10
1140     asciizahl:=asciizahl+summencodezahl
1150     satz$(:stelle:):=CHR$(asciizahl)
1160     stelle:=stelle+1
1170     IF codestelle=LEN(code$) THEN
1180       codestelle:=1
1190     ELSE
1200       codestelle:=codestelle+1
1210     ENDIF
1220     IF internercodestelle=LEN(internercode$) THEN
1230       internercodestelle:=1
1240     ELSE
1250       internercodestelle:=internercodestelle+1
1260     ENDIF
1270   UNTIL stelle>LEN(satz$)
1280   //
1290 ENDPROC publickeyverschlüsseln
1300
1310
1320 PROC publickeyentschlüsseln
1330   //
1340   stelle:=1
1350   codestelle:=1
1360   REPEAT
1370     asciizahl:=ORD(satz$(:stelle:))
1380     codezahl:=ORD(code$(:codestelle:))-48
1390     asciizahl:=asciizahl-codezahl
```

[6] Die dabei angewandte zweifelsfrei absolut abhörsichere Variante **Vernamchiffre** (1926) kann hierbei angesprochen werden. Sie wird von mir aber im Unterricht nicht gesondert behandelt.

```
1400     satz$(:stelle:):=CHR$(asciizahl)
1410     stelle:=stelle+1
1420     IF codestelle=LEN(code$) THEN
1430       codestelle:=1
1440     ELSE
1450       codestelle:=codestelle+1
1460     ENDIF
1470   UNTIL stelle>LEN(satz$)
1480   //
1490 ENDPROC publickeyentschlüsseln
```

Ich gebe für die Schüler nicht sichtbar einen Satz ein und verschlüssele diesen für die Schüler sichtbar mit einer ihnen dadurch bekannten Verschlüsselungzahl. Dieser chiffrierte Satz wird auf dem gemeinsamen Verzeichnis gespeichert und die Schüler erhalten den Auftrag, diesen Satz zu entschlüsseln. Die Schüler machen sich an die Arbeit und sind zunächst erstaunt, dass eine Entschlüsselung mit der ihnen doch bekannten Verschlüsselungszahl nun so nicht mehr möglich ist. Da das Verfahren bewusst einfach ist, wird es zumeist nach einiger Zeit von einigen pfiffigen Schülern oder Schülerinnen „geknackt“. Die Schülerinnen und Schüler erkennen:

- Es gibt Verfahren, bei denen zwei Schlüsseln notwendig sind. Ein Schlüssel wird zum Verschlüsseln der Nachricht verwendet: der *public key*. Ein zweiter Schlüssel, den sinnvoller Weise nur der Empfänger der Nachricht kennt, wird zum Entschlüsseln benutzt: der *private key*.
- Dieses Verfahren verlangt, dass die Systeme des Senders und des Empfängers die Berechnungsvorschrift kennen müssen. Damit die Kommunikation nicht abgehört werden kann, ist es notwendig, dass die Berechnung sehr aufwändig ist[7].
- Ein Problem stellt die Erstellung der Schlüsselpaare dar. Es darf natürlich nicht möglich sein, dass ein Unberechtigter zu einem öffentlichen Schlüssel den zugehörigen privaten Schlüssel ermitteln kann. Das bedeutet, dass es entweder vertrauenswürdige „Schlüsselerzeugungsstellen“ geben muss oder das verwendete System kann nach einem Zufallsprinzip für die Benutzer Paare bilden, wobei die Wahrscheinlichkeit für ein Auftreten bisher verwendeter Paare praktisch Null sein muss. Hier entwickeln sich oft interessante aktuelle gesellschaftspolitische Diskussionen mit und für die Schüler.

Natürlich wird auch dieses Verfahren praktisch ausgetestet. Die Schüler erhalten dabei von mir eine Liste der öffentlichen Schlüssel ihrer Mitschüler. Jedem Einzelnen teile ich seinen privaten Schlüssel mit. Und siehe da: eine Kommunikation ist sinnvoll möglich.

2.4 Signieren von Daten

Nachdem die Schülerinnen und Schüler das Prinzip der Public-Key-Verfahren verstanden und auch praktiziert haben, kann die *Authenzität* von Nachrichten thematisiert werden. Jemand kann unter Täuschung über seine echte Identität mit meinem öffentlichen Schlüssel mir zwar eine verschlüsselte Nachrichten zukommen lassen, die nur ich lesen kann, aber woher weiss ich, dass mein Gegenüber auch der ist, als der er sich auszugeben versucht. Die Schülerinnen und Schüler sind es zwar gewohnt, z.B. in Chaträumen eine derartige Anonymität zu geniessen, sehen aber natürlich ein, dass eine Kommunikation in einer vernetzten Gesellschaft natürlich auch die Authenzitätfrage lösen muss.

Naheliegend ist nun, das der Klartext zweimal verschlüsselt wird: Zuerst wird der private Schlüssel des Absenders verwendet, dann der öffentliche Schlüssel des Empfängers. Der

[7] An dieser Stelle kann ein kurzer Hinweis auf konkrete Realisierungen wie z.B. RSA erfolgen. Eine genauere Analyse verbietet sich meines Erachtens auf jeden Fall im Unterricht in der Sek I, aber eigentlich auch in der Sek II.

Empfänger decodiert nun den Chiffriertext mit seinem privaten Schlüssel und dann mit dem öffentlichen Schlüssel des Absenders und kann dann die Nachricht lesen. Mit diesem Verfahren ist der Text vom Absender also signiert worden[8].

Das von uns verwendete Verfahren hat eine besondere Eigenart. Es ist nicht symmetrisch, das heisst: Der zum Signieren zu verwendende private Schlüssel ist nicht identisch mit dem privaten Schlüssel zum Decodieren von Nachrichten. In der Praxis verwendete Verfahren sind verständlicherweise symmetrisch. Die Schülerinnen und Schüler können sich nun signierte verschlüsselte Nachrichten zusenden. Das Handling mit den verschiedenen Schlüsseln verlangt aber ein genaues konzentriertes Arbeiten und zeigt daher, in wie weit die Schüler die Verfahren auch tatsächlich verstanden haben.

3 Folgende Unterrichtseinheiten

3.1 E-Mail

Das von mir im vorigen Kapitel vorgestellte Programmsystem kann als eine einfache Form eines E-Mail-Systems verstanden werden. Es bietet sich daher an, anschliessend das heutige real verwendete E-Mail-System im Internet zu analysieren. Dabei können z.B. problematische nicht verschlüsselte Protokoll-Mechanismen beim Senden und Empfangen von E-Mail untersucht werden. Ebenso kann das Verschlüsseln von E-Mails mit Systemem wie PGP und das quelloffene GnuPG besprochen werden.

Anschliessend sollte die Problematik von Anhängen erarbeitet werden. Vor allem die Möglichkeit der Einbettung von ausführbaren Programmen in vom Benutzer nicht lesbaren (Text-)Dokumenten (vor allem in den Dokumenten eines sachlich nicht gerechtfertigten weit verbreiteten Software-Systems) im Gegensatz zu analysierbaren Dokumenten kann besprochen werden.

3.2 Betriebssyteme und Netze

Eine weitere Möglichkeit ist die Untersuchung von (aktuellen) Betriebssystemen hinsichtlich der Nutzung von kryptografischen Methoden zur Benutzer- und Datenverwaltung. Unter anderem kann analysiert werden, welche Protokolle für Fernzugriffsmethoden auf diese Systeme und/oder Netze (z.B. SSL, VPN, SHTTP) verwendet werden und wie sicher diese sind. Auch die Gefährdung der Computersysteme beim Online-Banking kann hierbei thematisiert werden. Darauf gehe ich aber hier nicht weiter ein.

4 Schlussbemerkungen

Meines Erachtens muss eine Schülerin bzw. ein Schüler nach Abschluss der Sekundarstufe I ein prinzipielles Verständnis über Verschlüsselung im Allgemeinen und den alternativen Prinzipien besitzen. Das bezieht sich nicht nur auf die Schülerinnen und Schüler, die im Augenblick z.B. das Wahlpflichtfach Informatik in der Sekundarstufe I belegen. Es gehört einfach in einer vernetzten Welt zur Allgemeinbildung dazu.

Ich bezweifle, dass die dazu notwendigen Kenntnisse und Kompetenzen in einem der heutigen üblichen Pflichtfächer vermittelt werden können. Wenn dem aber nicht so ist, dann ergibt sich aus dieser Tatsache bereits der Zwang zu einem Pflichtfach Informatik in der Sekundarstufe I. An diesem Beispiel wird deutlich, dass das u.a. von Ulrike Wilkens in ihrer Dissertation korrekt beschriebene „allmähliche Verschwinden der informationstechnischen Grundbildung“ nicht dazu führt, dass „für die allgemeinbildenden Inhalte eines Faches Informatik ein Aufgehen in andere Fächer und Lernbereiche als Alternative immer mehr in Betracht zu kommen“

[8] Für das Verständnis ist dabei unerheblich, ob in der Praxis der gesamte Text, nur Teile davon oder Prüfsummen signiert werden.

scheint[9]. Die Kryptographie ist vielmehr ein Schulfach begründendes Gebiet der Informatik , also eine „fundamentale Idee der Informatik“[10] und keine „Modeerscheinung“[11].

5 Literaturverzeichnis

Friedrich Bauer, Rupert Gnatz und Ursula Hill Informatik - Aufgaben und Lösungen, Erster Teil. Springer Verlag, Berlin, 1975

Rüdeger Baumann Informationssicherheit durch kryptologische Verfahren. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Rüdeger Baumann Digitale Unterschrift. in LOG IN 2 99, LOG IN-Verlag, Berlin, 1999

Peter Batzer Die Enigma. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Klaus-Cl. Becker und Albrecht Beutelspacher Datenverschlüsselung. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Bernd Hafenbrack(Hrsg.) Comal im Unterricht. Deutscher Studienverlag, Weinheim, 1988

Hanns-Wilhelm Heibey, Andreas Pfitzmann und Ulrich Sandl Kryptographie. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Werner Heine Die Hacker. Rowolth Taschenbuch Verlag, Hamburg, 1985

Werner Hülsmann IT-Sicherheit - eine kurze Einführung. in FIFF-Kommunikation 2 2003, FIFF, Bremen, 2003

Klaus Köhler Sicherheit durch Kryptographie. in FIFF-Kommunikation 2 2003, FIFF, Bremen, 2003

Gerhard Krüger, Dietrich Reschke Lehr- und Übungsbuch Telematik. Fachbuchverlag Leipzig, 2000

Arno Pasternak Was wir vergessen haben. `http://pasternak.in.hagen.de/if/koenigstein/vergessen.htm`, 2005

Michael Portz Sicherheit in Netzen. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Hubert Rüger Information Security. in FIFF-Kommunikation 2 2003, FIFF, Bremen, 2003

Sigrid Schubert Basismechanismen der Informationssicherheit. in LOG IN 5/6 96, LOG IN-Verlag, Berlin, 1996

Andreas Schwill Fundamentale Ideen der Informatik. in Zentralblatt für Didaktik der Mathematik, Heft 1 93, Karlsruhe, 1993

Monika Seiffert Verschlüsselungsmethoden. in LOG IN 3/4 94, LOG IN -Verlag, Berlin, 1994

Paul Strathern Turing & der Computer. Fischer Taschenbuch Verlag, Frankfurt am Main, 1998

Andrew S. Tanenbaum Computer-Netzwerke. Wolfram’s Fachverlag, 1. Auflage, Attenkirchen, 1990

Ulrike Wilkens Das allmähliche Verschwinden der informationstechnischen Grundbildung. Shaker Verlag, Aachen, 2000

Helmut Witten Codierungstheorie. in LOG IN 5/6 94, LOG IN -Verlag, Berlin, 1994

Helmut Witten, Irmgard Letzner und Ralph-Hardo Schulz RSA & Co. in der Schule. in LOG IN Hefte 3/4 98, 5/98, 2 99; LOG IN-Verlag, Berlin, 1998/1999

[9] in [Wilkens 2000], S. 63

[10] siehe [Schwill 1993]

[11] Dies wird allein schon durch die Jahreszahlen der in der Literaturliste aufgeführten Veröffentlichungen deutlich.