DIPLOMARBEIT Sonja Oblak

Öffentliche Chiffrierverfahren für die Schule und in der Praxis

Public-Key-Kryptosystems in school and in practice

Danksagung

Der größte Dank gebührt meiner gesamten Familie, die mich in all den Jahren immer unterstützt hat und hinter mir gestanden ist. Meine Eltern Maria und Walter haben mir das nötige Selbstvertrauen sowie Zielstrebigkeit vermittelt, Eigenschaften mit denen ich mein Studium meistern konnte. Für Auflockerung, Unterhaltung und jugendliche Tipps zur Bewältigung von Situationen aller Art, danke ich meinem Bruder Tobias.

Meinen Studienkollegen ein herzliches Danke, ohne sie hätte ich so manche mathematische Aufgabe nicht gelöst. Für das Korrekturlesen bedanke ich mich bei meinem Freund Markus.

Meinem Betreuer Herrn Univ. Prof. Dr. phil. Dietmar Dorninger danke ich besonders für die schnelle und freundliche Unterstützung. Er ermöglichte mir meine Vorstellungen zu verwirklichen und stand mir mit Rat und Tat zur Seite.

Kurzfassung

Diese Arbeit wendet sich in gleicher Weise an Lehrer von höheren Schulen wie an im Berufsleben stehende Personen, welche Wissen über öffentliche Chiffrierverfahren (Public-Key-Kryptosysteme) benötigen.

Im ersten Teil der Arbeit werden zunächst Grundbegriffe aus der Kryptographie zusammengestellt und die Arbeitsweise von öffentlichen Chiffriersystemen anschaulich erklärt. Dann folgen Ausführungen über Mathematik welche zum Verständnis des Folgenden unbedingt notwendig sind und welche einem Lehrenden geläufig sein müssen, wenn er kryptographische Inhalte in der Schule vermittelt bzw. dem Praktiker den Weg zum weiteren Verständnis weist. Den Primzahlen, die unter anderem gewissermaßen das Fundament der Verschlüsselung bilden, ist ein eigenes Kapitel gewidmet.

Der zweite Teil beschäftigt sich mit ausgewählten Verfahren der Kryptographie und wie diese in der Praxis verwendet werden. Es handelt sich dabei insbesondere um RSA, Zero-Knowledge und das Knapsack-Verfahren. Die Sicherheit und Umsetzung der Algorithmen im Alltag wird diskutiert, und zum besseren Verständnis werden einige Beispiele mit konkreten, einfachen Zahlen durchgerechnet. Weiters werden der Schlüsseltausch nach Pohlig und Hellman und Ansätze der Quantenkryptographie behandelt.

Im abschließenden dritten Teil werden einige Gedanken formuliert, warum kryptographische Inhalte im Schulunterricht vermittelt werden sollten. Eine kurze historische Betrachtung endet mit der Frage: „Wer benötigt Kryptographie heute?“, wobei gleich hier vorweg genommen werden kann: Jeder! Es wird ein kurzer Blick auf die im zweiten Teil beschrieben Verfahren geworfen und versucht, die grundlegenden Aspekte, die für die Schüler wichtig sind, nochmals hervorzuheben.

Anmerkung

Soweit in dieser Arbeit Berufsbezeichnungen, Ämter und Funktionen in der männlichen Form verwendet werden, ist dies geschlechterneutral zu verstehen. Eine bessere Lesbarkeit des Textes soll so erreicht werden.

Inhaltsverzeichnis:

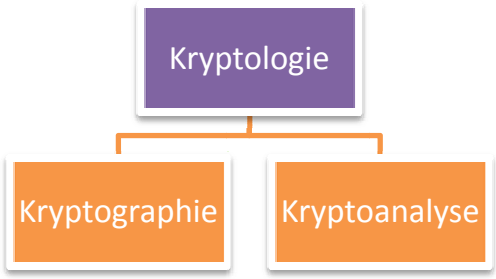
Vorkenntnisse für die Lehrenden

Begriffsbestimmung

Auf den folgenden Seiten werden Begriffe aus der Kryptologie verwendet, welche nachfolgend erklärt werden. Wir folgen dabei der in [b9] angeführten Unterteilung.

Kryptographie wird in [b9] definiert als die Lehre der Absicherung von Nachrichten durch Verschlüsselung.

Kryptoanalyse als die Kunst, Chiffretexte aufzubrechen d.h., den Klartext zu reproduzieren, ohne Kenntnis des Schlüssels.



Kryptologie vereinigt Kryptographie und Kryptoanalyse.

Der lesbare Text einer Nachricht wird **Klartext** genannt. Wir bezeichnen ihn mit (= message).

Der Klartext kann **chiffriert** (= verschlüsselt) und **dechiffriert** (= entschlüsselt) werden. Die Chiffre (encryption) ist eine invertierbare, d.h. eine umkehrbare eindeutige Abbildung, welche aus dem Klartext mit Hilfe eines Schlüssels K den Geheimtext (ciphertext)

erzeugt. Die Umkehrung von e zur Wiederherstellung des Klartextes wird Entschlüsselung genannt und im Folgenden mit d (decryption) bezeichnet.

Aus dieser Definition ergibt sich:

$e(m) = c$ und $\mathrm{d}(c) = m$ d.h. $d(e(m)) = m$

Nach der Entschlüsselung eines Chiffretextes kommt der Klartext zustande.

Kryptographische Verfahren haben die Aufgabe mindestens eine der vier aufgelisteten Eigenschaften zu gewährleisten.

1. Geheimhaltung:

Ziel der Geheimhaltung ist es, das Lesen einer Nachricht für Unbefugte unmöglich bzw. sehr schwer zu gestalten.

2. Authentifizierung:

Dies ist ein Identitätsbeweis des Senders einer Nachricht gegenüber dem Empfänger. Mit anderen Worten: der Empfänger muss sich sicher sein, dass eine bestimmte Nachricht nicht von einem unbefugten Absender stammt.

3. Integrität:

Die Nachricht darf während der Übermittlung nicht von Dritten verändert werden; sie bleibt unverletzt.

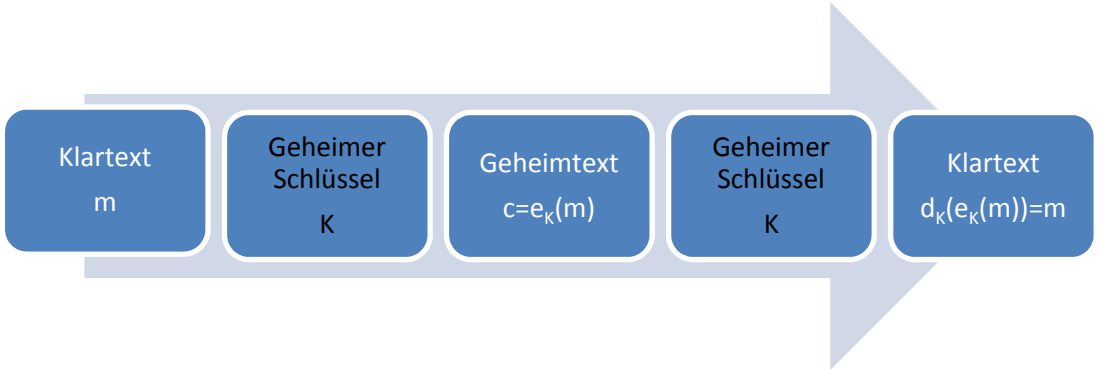
4. Verbindlichkeit:

Der Sender kann später nicht abstreiten eine bestimmte Nachricht übermittelt zu haben.

Kryptographische Algorithmen sind verschiedene Berechnungsvorschriften, d.h. mathematische Funktionen zur Ver- und Entschlüsselung. Mittels einer groben Einteilung unterscheidet man:

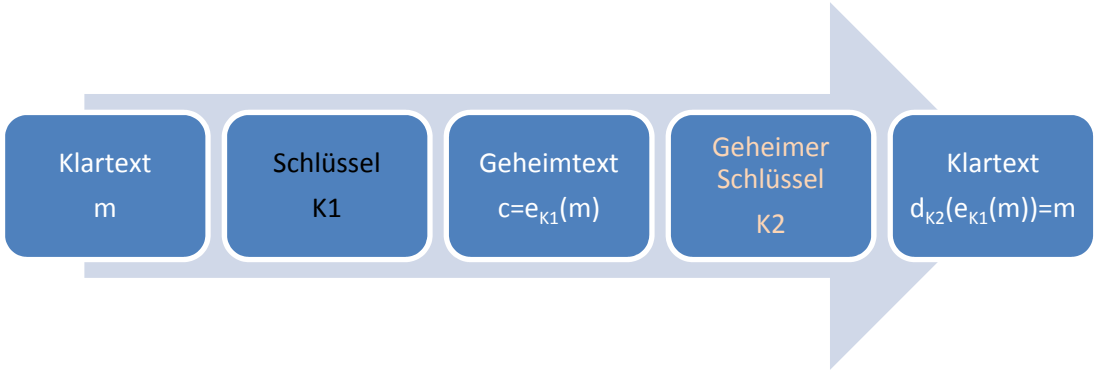
Symmetrische Algorithmen:

Hierbei wird mit dem selben Schüssel chiffriert und dechiffriert. Dieser Schlüssel muss unbedingt geheim bleiben, sonst kann die Sicherheit der Nachricht nicht gewährleistet werden.



Asymmetrische Algorithmen - Public Key:

Hier wird zum Chiffrieren der öffentliche Schlüssel K1 und zum dechiffrieren der geheime Schlüssel K2 verwendet.



Vor 1978 verwendete man ausschließlich Algorithmen deren Sicherheit davon abhing, ob ihre Funktionsweise geheim blieb. Daraus ergaben sich mehrere Nachteile:

- Wenn eine Person ihren Arbeitsplatz wechselte, musste der Algorithmus geändert werden.
- Eine unbefugte Person konnte aus den Maschinenprogrammen die Arbeitsweise des Verfahrens rekonstruieren.
- In diesen Fällen konnte keine Qualitätskontrolle stattfinden. Wäre ein Algorithmus in der Öffentlichkeit bekannt geworden, hätte man ihn in der Praxis nicht mehr sicher einsetzen können.

Aus diesen Gründen forderte man:

Die Sicherheit eines Verschlüsselungsverfahrens darf nur von der Geheimhaltung des Schlüssels abhängen, nicht jedoch von der Geheimhaltung des Algorithmus.

Heute zeigt sich, dass Offenlegung der beste Test für ein neues Verfahren ist. Besteht das Verfahren in der Praxis, wird es als weitgehend sicher angesehen. Sind über einen längeren Zeitraum die Versuche der Experten den Algorithmus zu knacken erfolglos, verstärkt sich das Vertrauen der Benutzer.

Verschlüsselungsverfahren, die in dieser Arbeit beschrieben werden, sind **Public - Key Algorithmen**. Ihre Arbeitsweisen sind vollständig dokumentiert. Die Sicherheit wird dadurch nicht beeinträchtigt.

Grundlegende Eigenschaften von Public–Key-Kryptosystemen

- Jede beteiligte Person hat ein Paar von Schlüsseln, einen öffentlichen und einen geheimen privaten.
- Der **öffentliche Schlüssel**, er wird zu Verschlüsselung verwendet und im Folgenden mit e (encryption) bezeichnet.
- Der **private Schlüssel**, er wird zu Entschlüsselung verwendet und von uns mit d (decryption) bezeichnet.

Das Interessante an diesem Sachverhalt ist, dass man aus der Kenntnis des öffentlichen Schlüssels keine Vermutungen über den privaten Schlüssel ziehen kann, geschweige denn ihn berechnen kann.

Die Tatsache die das ermöglicht, liegt in der verwendeten mathematischen Methode. Zur Verschlüsselung werden sogenannte **Einwegfunktionen** verwendet. [b4]

Definition: Eine Funktion $f: A \rightarrow B$ heißt Einwegfunktion, falls für alle $x \in$ A der Wert $f(x)$ leicht zu berechnen ist, es aber so gut wie unmöglich ist, zu einem gegebenen $y \in B$ ein $x \in A$ zu finden mit $f(x) = y$ (es sei denn, man ist im Besitz einer geheimen Zusatzinformation, welche dies gestattet).

Die Begriffe „leicht“ und „so gut wie unmöglich“ werden in Bezug auf den Rechenaufwand verstanden. Die Rechengeschwindigkeit nimmt zwar schon seit Jahren rapide zu, doch sie scheint nicht unbegrenzt. Hier sei nur erwähnt, dass ein Quantencomputer, welcher heute noch nicht auf kommerzieller Basis existiert, den im Folgenden beschrieben Algorithmen mit Leichtigkeit beikommen könnte. Doch noch ist seine Entwicklung erst am Anfang, weshalb man sich bezüglich der Sicherheit der beschriebenen Verfahren keine allzu großen Sorgen machen sollte. [b12]

Der öffentliche Schlüssel ist für jeden Teilnehmer sichtbar. Man kann sich vorstellen, er scheint wie eine Telefonnummer in einem frei zugänglichen Telefonbuch auf. Will man einer Person eine geheime Nachricht zukommen lassen, wird einfach ihre Nummer gewählt. Mit ihrem privaten Schlüssel gelingt es dann nur der Person der die Nachricht zukommt, diese zu lesen.

Daran lässt sich bereits etwas von der Genialität von Public–Key-Kryptosystemen erkennen.

Das Eingabealphabet

Dieses ist sowohl bei symmetrischen, als auch bei asymmetrischen Verfahren von Bedeutung [b4]. Nachrichten werden als Wörter in einem **Eingabealphabet** A **quellencodiert**. Man führt dazu die Buchstaben der Wörter in Folgen von Symbolen aus A über.

Setzt man voraus, dass alle Wörter in dem neuen Eingabealphabet A die gleiche Länge haben, so ist der Begriff der **Blocklänge** von Bedeutung.

Die Länge der Wörter bezeichnen wir mit n. Gilt für n: $n > 1$, dann verwendet man anstatt des Begriffes Wort auch Block. n wird dabei als Blocklänge bezeichnet. Im Fall $n = 1$ besteht die Nachricht aus einem einzigen Wort.

Quellencodierung

Sie darf nicht mit der Codierung verwechselt werden, welche zum Ziel hat, Fehler zu korrigieren. Dies ist Inhalt der Codierungstheorie, sie wird hier nicht behandelt.

Beispiele:

1. Sei unser Alphabet $A = \{00, 01, 02, \dots, 26\}$ wobei die Paare folgende Zuordnung aufweisen:
   $Leerzeichen \rightarrow 00,\ A \rightarrow 00$, $B \rightarrow 01$, $C \rightarrow 02$, usw.
   Als Wortlänge wählen wir $n = 1$. Für die Nachricht „NICHT SCHUMMELN“ ergibt sich somit: 14 09 03 08 20 00 19 03 08 08 05 12 14

Wählen wir als Blocklänge $n = 5$ erhalten wir: $14090, 30820, 00190, 30808, 05121, 40000$

2. Sei unser Alphabet $A = \{0,1\}$. Die Zahlen z von 0 bis 26 codieren wir als binäre Fünferblöcke. Für $z = b_4 2^4 + b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$ schreiben wir: $b_4 b_3 b_2 b_1 b_0$.

Das Leerzeichen sei wieder 0, A sei 1,…, $Z = 26$. Für die Nachricht „NICHT SCHUMMELN“, mit $n = 5$ als Blocklänge ergibt sich somit:

$$01110, 01001, 00011, 01000, 10100, 00000, 10011, 00011, 01000, 01000,$$

$$00101, 01100, 01110.$$

Chiffrierung:

Die einzelnen Wörter w einer quellencodierten Nachricht werden mit Hilfe einer Verschlüsselungsfunktion v in Wörter über einem Alphabet B, dem Ausgabealphabet, übergeführt. Wir nehmen dabei an, v ist injektiv.

Um mit der Funktion v operieren zu können, bettet man die Alphabete in algebraische Strukturen ein. Diese Strukturen sind auf A^n und B^n, der Länge der Wörter über den Alphabeten fortsetzbar. Algebraische Strukturen werden in den mathematischen Grundlagen behandelt.

Einige weitere Vorteile asymmetrischer Chiffrierverfahren:

Schlüsseltausch

Das Problem des Schlüsseltausches wird unwichtig. Es müssen keine geheimen Schlüssel mehr über unsicheren Strecken transportiert oder gesendet werden. Möchte man mit jemandem Kontakt aufnehmen, lässt man ihm eine Nachricht mit Hilfe seines öffentlichen Schlüssels zukommen.

Neue Teilnehmer

Bei symmetrischen Verschlüsselungsverfahren bedeutete ein neuer Teilnehmer Arbeitsaufwand. Alle am System Beteiligten mussten mit dem Neuling einen eigenen Schlüssel austauschen. Bei asymmetrischen Verfahren fällt diese Arbeit weg.

Zahl der Schlüssel

$\frac{n*(n-1)}{2}$ ist die Anzahl der Schlüssel, die in einem symmetrischen System benötigt werden. Dabei steht n für die Teilnehmeranzahl. Diese Schlüsselanzahl erhöht sich somit quadratisch mit der Anzahl der Beteiligten. Bei einem asymmetrischen Verfahren besitzt jeder Teilnehmer nur zwei Schlüssel, von denen nur einer geheim bleiben muss.

In Zahlen bedeutet das:

Verfahren	Anzahl der Teilnehmer	Anzahl der Schlüssel
asymmetrisches Verfahren	1000	2000
symmetrisches Verfahren	1000	499 500

Kein Vorteil ohne Nachteil:

Die Sicherheit dieser Verfahren ist äquivalent zum Aufwand mit dem Faktorisierungsproblem großer Zahlen. Es gilt somit nicht als bewiesen, dass öffentliche Chiffrierverfahren schwer zu knacken sind. Weiters kommt es durch die großen Zahlen, die man für die Schlüsselerzeugung braucht, zu enormen Rechenzeiten für die Hardware.

Das Schlüsselmanagement für neue Teilnehmer ist bei genauerer Betrachtung nicht ganz so trivial, wie es zu Beginn erscheint. Es könnte sich zum Beispiel ein Betrüger unter einer falschen Nummer im Telefonbuch eintragen. Damit das nicht geschieht, gibt es mehrere ausgeklügelte Systeme. (Zertifizierungsstellen und Web of Trust, werden im Kapitel Zero-Knowledge näher behandelt.)

Mathematische Grundlagen

Für die im nachfolgenden Kapitel verwendeten Rechenoperationen werden einige mathematische Grundlagen benötigt. Es handelt sich hierbei um eine Sammlung von Definitionen und Sätzen, die ein Lehrender im Hinterkopf haben sollte, wenn er kryptographische Inhalte in der Schule vermittelt und welche der interessierte mathematische Laie benötigt, um die Grundlagen der Verfahren zu verstehen. Die Auflistung bietet lediglich einen Überblick. Der interessierte Leser sei auf die Fachliteratur verwiesen, wo er Beweise und Weiterführungen findet.

Das Rechnen mit Restklassenringen und primen Restklassengruppen bildet unter anderem das Fundament kryptographischer Verfahren. So seien hier Gruppen, Ringe und Körper - algebraische Strukturen als erstes erwähnt. Nachfolgend werden verschiedene Begriffe und Sätze der Zahlentheorie beschrieben. Den Primzahlen wurde ein eigenes Kapitel gewidmet. Sätze über Teilbarkeit, der größte gemeinsame Teiler, der euklidische Algorithmus, alles was wichtig erschien, wurde hier zusammengefasst.

Algebraische Strukturen

Gruppe

Ein Paar $(\mathrm{G},*)$ mit einer Menge G und einer zweistelligen Operation $*: \mathrm{G} \times \mathrm{G} \to \mathrm{G}, (a, b) \to a * b$ heißt Gruppe, wenn folgende Axiome erfüllt sind:

- **Assoziativität**: Für alle Gruppenelemente a, b und c gilt: $(a * b) * c = a * (b * c)$
- **Neutrales Element**: Es gibt ein neutrales Element $\mathrm{e} \in \mathrm{G}$, mit dem für alle Gruppenelemente a gilt: $a * e = e * a = a$
- **Inverses Element:** Zu jedem Gruppenelement a existiert ein Element $a^{-1} \in G$ mit $a * a^{-1} = a^{-1} * a = e$

Eine Gruppe $(G,*)$ heißt **abelsch** oder **kommutativ**, wenn für die Verknüpfung $*$ zusätzlich das folgende Axiom erfüllt ist:

Kommutativität: Für alle Gruppenelemente a und b gilt $a * b = b * a$.

Beispiel für eine Gruppe:

Für zwei Permutationen π und ρ von $1, 2, 3$ definieren wir:

$$\begin{pmatrix} 1 & 2 & 3 \\ \pi 1 & \pi 2 & \pi 3 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ \rho 1 & \rho 2 & \rho 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ \pi(\rho(1)) & \pi(\rho(2)) & \pi(\rho(3)) \end{pmatrix}.$$

zum Beispiel:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}.$$

Wir sehen, dass im allgemeinen $\pi * \rho \neq \rho * \pi$ ist. Mit $n = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$ gilt für jede Permutation π, dass $\pi * n = n * \pi = \pi$ ist. Zu jedem $\pi = \begin{pmatrix} 1 & 2 & 3 \\ \pi 1 & \pi 2 & \pi 3 \end{pmatrix}$ ist $\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ \pi^{-1}(1) & \pi^{-1}(2) & \pi^{-1}(3) \end{pmatrix}$ invers.

Allgemein ist einzusehen: Die Permutationen von $1, 2, \dots, k$ mit der Operation $*$, welche analog wie oben im Fall $k = 3$ definiert ist, bilden eine Gruppe. Sie wird die Symmetrische Gruppe S_k vom Grad k genannt.

Gibt es in einer Gruppe $(G,*)$ ein Element γ, sodass alle Gruppenelemente Potenzen von γ sind, so heißt γ ein **erzeugendes Element** der Gruppe. Die Gruppe wird zyklisch genannt und ist endlich. Allgemein wird eine algebraische Struktur als endlich bezeichnet, wenn ihre Grundmenge endlich viele Elemente hat. Sei γ ein Element einer endlichen zyklischen Gruppe G mit n Elementen. γ ist genau dann erzeugendes Element der Gruppe, wenn $\gamma^n = 1$ und $\gamma^e \neq 1$ für alle $0 < e < n$.

Bei der Definition der Gruppe, wurde eine Menge mit *einer* Operation versehen. Wir kommen nun zu Mengen auf denen *zwei* Operationen festgelegt sind. Hier sind für uns insbesondere die **Ringe** von Interesse.

Ring

Ein Ring ist ein Tripel $(R, +, *)$, bestehend aus einer Menge R und zwei zweistelligen Operationen $+$ und $*$ in R, für die gelte:

- $(R, +)$ ist eine abelsche Gruppe,
- $(R, *)$ ist assoziativ und $+$ und $*$sind miteinander verknüpft durch die:
- Distributivgesetzte: $\forall a, b, c \in R\ gilt$:
  $a * (b + c) = a * b + a * c$ und $(a + b) * c = a * c + b * c$

Besitzt ein Ring R ein Element 1, sodass $a * 1 = 1 * a = a$ für alle a, so heißt er ein Ring mit 1- Element. Ist $*$ kommutativ, so wird R als **kommutativer Ring** bezeichnet.

Beispiele:

Die Menge der ganzen Zahlen bildet mit den Verknüpfungen $+ und\ *$ einen kommutativen Ring mit Einselement 1. Ein weiteres für unsere Anwendungen wichtiges Beispiel ist der Restklassenring Z_m.

Integritätsring

Ein kommutativer Ring R mit Einselement heißt Integritätsring, falls $0 \neq 1$ ist und der Ring keine Nullteiler besitzt. Dabei heißt $a \neq 0$ ein Nullteiler von R, falls es ein $b \neq 0$ in R gibt mit $a * b = 0$. Ein Integritätsring ist „**nullteilerfrei**“. Früher wurde ein Integritätsring auch als Integritätsbereich bezeichnet.

Euklidischer Ring

Dies ist eine spezielle Klasse von Integritätsringen, welche man für viele Berechnungen in der Kryptographie benötigt.

Ein Integritätsring R heißt ein euklidischer Ring, falls eine Bewertungsfunktion $g: R\backslash 0 \rightarrow N_0$ mit folgenden Eigenschaften existiert:

1.) $\forall a, b \in R\ mit\ b \neq 0\ \exists q, r \in R$ sodass $a = q * b + r$ mit $r = 0\ oder\ g(r) < g(b)$.

2.) Für $a, b \in R\backslash 0$ ist: $g(a) \leq g(a * b)$.

Die Abbildung g heißt dabei euklidische Normfunktion (euklidischer Betrag) des Ringes.

Beispiele:

Die ganzen Zahlen bilden mit der Bewertungsfunktion $g(a) = |a|$ mit $a \neq 0$ einen euklidischen Ring.

Der Polynomring $K[x]$ über einem Körper K wird zu einem euklidischen Ring, wenn man für jedes vom Nullpolynom verschiedene Polynom $p(x) \in K[x]$ definiert: $g(p(x)) = Grad\ p(x)$.

Körper

Ein kommutativer Ring, mit 1- Element und mindestens zwei Elementen heißt ein Körper, wenn in ihm jedes von Null verschiedene Element multiplikativ invertierbar ist, d.h., wenn die Elemente $\neq 0$ gegenüber der Multiplikation eine Gruppe bilden.

Die reellen, rationalen und komplexen Zahlen sind Beispiele von Körpern mit unendlich vielen Elementen. Was die Körper mit endlich vielen Elementen betrifft, so kann man zeigen, dass jeder solche Körper p^t Elemente, p Primzahl, besitzt. Zu jeder Primzahlpotenz p^t gibt es bis auf Bezeichnung genau einen Körper mit p^t Elementen, das „Galoisfeld" $GF(p^t)$. Für $t = 1$ ist dies der Restklassenring Z_p.
Wir haben hervorgehoben, dass die Elemente $\neq 0$ einen Körper per definitionen eine Gruppe bilden. Bei endlichen Körpern ist diese Gruppe, wie man unschwer beweisen kann, zyklisch. Für Galoisfelder $GF(p^t)$ wird diese oft mit $GF(p^t)^*$ bezeichnet. Betrachten wir z.B. das Galoisfeld $GF(4) = GF(2^2)$. Bezeichnen wir seine Elemente mit $0, 1, a, b$ und gehen von nachstehenden beiden Operationstafeln aus.

+	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	a	b
2	0	a	b	1
3	0	b	1	a

Wir sehen, dass sowohl a wie b erzeugende Elemente von $GF(4)^*$ sind: $a^0 = 1$, $a^1 = a$, $a^2 = b$; $b^0 = 1$, $b^1 = b$, $b^2 = a$;

Teilbarkeit in Z

Definition: Man sagt a teilt b, für $a, b \in Z$,wenn es eine ganze Zahl q gibt, sodass $a * q = b$. Man schreibt: $a|b$.

Wir rekapitulieren einige Eigenschaften der Teilbarkeit in Z: $\forall a, b, c, d \in Z\ gilt$:

- $\pm 1, \pm a,$ sind stets Teiler von a die sogenannten trivialen Teiler von a
- $a|0$
- $a|b\ und\ b|c \Rightarrow a|c$
- $a \left| b \Leftrightarrow a * c \right| b * c\ mit\ c \neq 0$
- $a|b\ und\ a|c \Rightarrow a|(x * b + y * c)\ für\ x, y \in Z$
- $a|b\ und\ c|d \Rightarrow a * c|b * d$
- $a|b\ und\ b \neq 0 \Rightarrow |a| \leq |b$
- $a|b\ und\ b|a \Rightarrow a = \pm b$
- Da Z ein euklidischer Ring ist, existiert zu $a \in Z$ mit $b \in Z, b > 0$, genau ein Paar von ganzen Zahlen (q, r), sodass $0 \leq r \leq b\ und\ a = b * q + r$ gilt.
- Die Zahl $d \in Z$ heißt ein gemeinsamer Teiler von a und b, wenn $c|a$ und $c|b$.
- Die größte ganze Zahl, die gemeinsamer Teiler von a und b ist, nennt man größter gemeinsamer Teiler von a und b und schreibt $ggT\ (a, b)$.
- Ganze Zahlen a und b, mit $ggT\ (a, b) = 1$ heißen teilerfremd oder relativ prim.

Der euklidische Algorithmus in Z

Dieser Algorithmus gilt in jedem euklidischen Ring. Wir beschränken uns hier auf Z. Das Verfahren bietet für große Zahlen die einzige Möglichkeit zur Berechnung des ggT. (In der Schule wird dies meist über die Primfaktoren der beiden Zahlen gerechnet, für die Praxis ist dieses Verfahren jedoch ohne Bedeutung.)

Es genügt $0 \leq b \leq a$ anzunehmen weil der $ggT(a, b) = ggT(b, a)$, und $ggT(a, b) = ggT(|a|, |b|)$.

Die Berechnung, des $ggT(a, b)$ mit $0 \leq b \leq a$ verläuft wie folgt:

$a = b * q_1 + r_1 \quad mit\ 0 < r_1 < b$

$b = r_1 * q_2 + r_2 \quad mit\ 0 < r_2 < r_1$

$r_1 = r_2 * q_3 + r_3 \quad mit\ 0 < r_3 < r_2$

So erhalten wir eine monoton abnehmende Folge positiver ganzer Zahlen:

$b > r_1 > r_2 > r_3 > \cdots$

Sie hat endlich viele Glieder, sodass das Verfahren nach endlich vielen Schritten abbricht. Die letzten drei Glieder des von 0 verschiedenen Restes sehen so aus:

$r_{n-3} = r_{n-2} * q_{n-1} + r_{n-1} \quad mit\ 0 < r_{n-1} < r_{n-2}$

$r_{n-2} = r_{n-1} * q_n + r_n \quad mit\ 0 < r_n < r_{n-1}$

$r_{n-1} = r_n * q_{n+1}$

Wir setzen $d = r_n$. Dann gilt: $d \left| r_{n-1} \Rightarrow d \right| r_{n-2} \Rightarrow d \left| r_{n-3} \Rightarrow \ldots \rightarrow d \right| b \Rightarrow d|a.$

Sei t ein weiterer Teiler von a und b. Aus $t|a$ und $t|b \Rightarrow$

$t \left| r_1 \Rightarrow t \right| r_2 \Rightarrow t \left| r_3 \Rightarrow \ldots \Rightarrow t \right| r_n \Rightarrow t|d.$

d ist also die größte natürliche Zahl, die a und b teilt: $d = ggT(a, b)$.

Der erweiterte euklidische Algorithmus

Neben dem größten gemeinsamen Teiler, dem $ggT\ (a, b)$ zweier natürlicher Zahlen a und b, kann man mit Hilfe des euklidischen Algorithmus zwei ganze Zahlen s und t ermitteln, für die folgende Gleichung gilt:

$ggT(a, b) = s * a + t * b.$

Wie das geht werden wir an einem numerischen Beispiel erläutern, aus dem das allgemeine Verfahren ersichtlich ist.

Der euklidische Algorithmus ist auch das Fundament für das Lösen ganzzahliger linearer Gleichungssystemen. Weiters wird er für den chinesischen Restsatz benötigt, der in der Kryptographie immer wieder eine Rolle spielt. Der interessierte Leser sei auf [b4, Seite 43] verwiesen.

Beispiel: Gesucht ist der $ggT(99{,}78)$:

$99 = 1 * 78 + 2$

$78 = 3 * 21 + 15$

$21 = 1 * 15 + 6$

$15 = 2 * 6 + 3$

$6 = 2 * 3 + 0$

3 ist der $ggT(99, 78)$. Beginnt man dieses Gleichungssystem von hinten zu lesen, und die Reste als Linearkombinationen von Dividend und Division darzustellen, erhält man bei der vorletzten Gleichung:

$3 = 15 - 2 * 6$

$= 15 - 2 * (21 - 1 * 15)$ $= 3 * 15 - 2 * 21$

$= 3 * (78 - 3 * 21) - 2 * 21$ $= 3 * 78 - 11 * 21$

$= 3 * 78 - 11 * (99 - 1 * 78) = 14 * 78 - 11 * 99$

Der größte gemeinsame Teiler ist als ganzzahlige Linearkombination der beiden Ausgangszahlen 78 und 99 dargestellt.

Tabellarische Darstellung

Die Zwischenergebnisse lassen sich anschaulich in Tabellen darstellen.

Tabelle 1

a	b	q	s	t
99	78	1		
78	21	3		
21	15	1		
15	6	2		
6	3	2		
3	0			

Tabelle 2

a	b	q	s	t
99	78	1		
78	21	3		
21	15	1		
15	6	2		
6	3	2		
3	0		1	0

Tabelle 3

a	b	q	s	t
99	78	1	-11	14
78	21	3	3	-11
21	15	1	-2	3
15	6	2	1	-2
6	3	2	0	1
3	0		1	0

Rekursive Variante

In Tabelle 1 ist der euklidische Algorithmus dargestellt. Die Division mit Rest hat dabei immer die Form $a = q \cdot b + r$, wobei q und r bestimmt werden. q wird in der gleichen Zeile

eingetragen. (b, r) wird bei (a, b) in der nächsten Zeile eingetragen. Diese Schritte wiederholen wir solange, bis in der Spalte von b eine Null steht. In der linken unteren Ecke kann man jetzt den größten gemeinsamen Teiler ablesen, nämlich 3.

Jetzt beginnt die Berechnung der ganzzahligen Koeffizienten s und t. In jeder Zeile soll dabei $3 = s \cdot a + t \cdot b$ gelten. Für die letzte Zeile ergibt sich: $s = 1$ und $t = 0$. Wir sehen die Ergebnisse in Tabelle 2. Nun arbeitet man von unten nach oben. Das neue s ist das alte t aus der darunter liegenden Zeile. Über s und t der alten Zeile und dem q der aktuellen Zeile berechnet man sich das t der darüber liegenden Zeile.

$t = s_{alt} - q * t_{alt}$

In der vorletzte Zeile erhalten wir so $s = 0$ und $t = 1 - 2 * 0 = 1$.

Darüber $s = 1$ und $t = 0 - 2 * 1 = -2$ usw.

Wir führen diesen Vorgang so oft durch, bis die Tabelle ausgefüllt ist. Das Ergebnis ist in Tabelle 3 ersichtlich. Die Einträge für s und t in der ersten Zeile sind die gesuchten Werte. Der größte gemeinsame Teiler findet sich, wie schon erwähnt, in der unteren linken Ecke. Für das Beispiel gilt damit:

$3 = -11 * 99 + 14 * 78$

Dies wird auch die **Vielfachsummendarstellung** genannt.

Anwendung findet der erweiterte euklidische Algorithmus bei der Berechnung eines multiplikativ inversen Elements in einem Restklassenring Z_m. Auf diesen Ring kommen wir nun zu sprechen.

Restklassenring Z_m

Gegeben sei ein $m > 0$ aus Z. Wir definieren $a \equiv b \; mod \; m$ (gesprochen: a ist kongruent zu b modulo m) falls $m|(a - b)$. Dies ist äquivalent dazu, dass es ein $k \in Z$ gibt mit

$a = k * m + b$.

Die Relation $a \equiv b \; mod \; m$ ist eine **Äquivalenzrelation**, die eine Relation $\sim$ in einer Menge M definiert, für die gilt:

- Reflexivität: $x \sim x$ für alle $x \in M$

- Symmetrie: $x \sim y \Rightarrow y \sim x$ für alle $x, y \in M$
- Transitivität: $x \sim y, y \sim z \Rightarrow x \sim z$ für alle $x, y, z \in M$

Darüber hinaus ist die Relation $a \equiv b \, mod \, m$ in Z eine Kongruenzrelation, d.h., aus $a \equiv b \, mod \, m$ und $c \equiv d \, mod \, m$ folgt $a + c \equiv b + d \, mod \, m$ und $a * c \equiv b * d \, mod \, m$. Jede Äquivalenzrelation entspricht einer Klasseneinteilung. Ist $\bar{a}$ die Klasse von a und $\bar{b}$ die Klasse von $b \, mod \, m$ und definieren wir $\bar{a} + \bar{b} = \overline{a + b}$, $\bar{a} * \bar{b} = \overline{a * b}$, so sind diese Operationen $+$, $*$ unabhängig von der Wahl der Repräsentanten der Klasse und wir erhalten auf diese Weise einen kommutativen Ring mit 1- Element, den wir mit Z_m bezeichnen. Ist $m = p$, p eine Primzahl, so ist Z_p ein Körper.

In Z_m kann man im Allgemeinen eine Gleichung $c * a \equiv c * b \, mod \, m$ nicht kürzen. Es gilt jedoch: $c * a \equiv c * b \, mod \, m \Leftrightarrow a \equiv b \, (mod \frac{m}{ggT(c,m)})$, womit man Z_m verlässt. Nur wenn $ggT(c, m) = 1$ ist und $\bar{c} \neq \bar{0}$ ist folgt daher $a = b \, mod \, m$.

Die Gruppe Z_m^*- die eulersche φ- Funktion

Die Elemente $\bar{a}$ von Z_m, für die gilt $ggT(a, m) = 1$, besitzen alle ein multiplikatives Inverses, denn gemäß dem euklidischen Algorithmus ist der $ggT(a, m)$ darstellbar in der Form $1 = a * x + m * y$ mit $x, y \in Z$, d.h. $a * x = 1 \, mod \, m$, also $x = a^{-1} \, mod \, m$. Die Menge der Elemente $\bar{a}$ von Z_m mit $ggT(a, m) = 1$ bilden bezüglich $*$ eine abelsche Gruppe, welche wir mit Z_m^* bezeichnen. Die Anzahl der Elemente dieser Gruppe wird als Eulersche φ– Funktion $\varphi(m)$ bezeichnet.

Ein Beispiel:

$\varphi(15) = ?$

Wir schreiben alle natürlichen Zahlen an, die zu 15 teilerfremd sind: $1, 2, 4, 7, 8, 11, 13, 14$. Das sind acht Zahlen; also ist $\varphi(15) = 8$. Nun berechnen wir $a^8 mod \, 15 \;\; \forall a < 15$.

a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^8 mod \, 15$	0	1	1	6	1	10	6	1	1	6	10	1	6	1	1

Man erkennt:

$a^8 mod\ m$= 1 stimmt immer dann, wenn a und 15 teilerfremd sind. Also vermuten wir $a^{\varphi(m)} = 1\ mod\ m$. Das gilt tatsächlich stets gemäß dem

Satz von Euler: Seien a und m zwei teilerfremde natürliche Zahlen. Dann gilt:

$a^{\varphi(m)} = 1\ mod\ m.$

Explizit kann für ein $n \in N, n > 1$, die Funktion $\varphi(n)$ wie folgt berechnet werden.

Satz: Sei $n \in N, n > 1$: dann gilt: $\varphi(n) = n * \prod_{p|n}(1 - \frac{1}{p})$. Dabei durchläuft p alle Primzahlen, die Teiler von n sind. Falls n eine Primzahl ist, auch n selbst.

Beispiel:

$$\varphi(10) = 10 * \left(1-\frac{1}{2}\right) * \left(1-\frac{1}{5}\right) = 10 * \frac{1}{2} * \frac{4}{5} = 4$$

Ist p eine Primzahl, dann gilt offensichtlich $\varphi(p) = p - 1$, und für das Produkt von $p * q$, zweier Primzahlen gilt:

$\varphi(p * q) = (p - 1) * (q - 1).$

Für kryptographische Zwecke ist jener Fall besonders wichtig, indem n das Produkt aus zwei verschiedenen Primzahlen p und q ist. Aus dem Satz von Euler können wir folgern: $a^{\varphi(m)} - a^{(p-1)*(q-1)} = 1\ mod\ m$, also

$a * a^{k*\varphi(n)} = a^{k*(p-1)*(q-1)+1} = 1^k * a = a\ mod\ m$ für $k \in N$.

Beispiel:

$5^{\varphi(6)}\ mod\ 6 = 5^2 mod\ 6 = 25\ mod\ 6 = 1$

Diese Rechnung ist unmittelbar einsichtig. Die Folgende könnte man zu einem mathematischen Zaubertrick benützen:

$31^{792}\ mod\ 851 = 31^{22*36} mod\ 851 = 31^{\varphi(23*37)}\ mod\ 851 = 31^{\varphi(851)}\ mod\ 851 = 31$

Der Zauberer wählt einen Freiwilligen aus dem Publikum aus. Dieser soll sich eine natürliche Zahl k und eine natürliche Zahl a wählen. Jetzt soll er sich $a^{k*(p-1)*(q-1)+1}\ mod\ p * q$ berechnen. Ob ihm das gelingt? Nehmen wir, er schafft es und erhält das Ergebnis nach langen Berechnungen. Währenddessen hat sich der Zauberer

entspannt zurückgelehnt, denn er kannte das Ergebnis schon lange im Vorhinein. Es ist die vom Freiwilligen gewählte Zahl a.

a muss nicht notwendigerweise teilerfremd zu n sein. Die Folgerung gilt für alle $a \leq n$.

Wie bereits erwähnt kann man die Elemente von Z_m^* mit Hilfe des euklidischen Algorithmus invertieren. Dazu ein

Beispiel:

Gesucht ist die Inverse von $15\ modulo\ 71$.

Es gilt wegen $1 = \ 19 * 15 + (-4) * 71$.

$19 * 15\ mod\ 71 = \ (1 + 4 * 71)\ mod\ 71 = 1$.

Deshalb ist 19 das zu 15 inverse Element modulo 71.

Ein Spezialfall des Satzes von Euler ist der

„**Kleinen" Satz von Fermat**: Es sei p eine Primzahl und $a \in Z$ eine zu p teilerfremde Zahl. Dann gilt: $a^{p-1} \equiv 1 \bmod p$.

Wir sehen das ein, indem wir im Satz von Euler (siehe früher) die Zahl m durch die Primzahl p ersetzen und beachten, dass $\varphi(p) = p - 1$ ist.

Die Erkenntnisse die wir nun gewonnen haben, spielen vor allem für das Verständnis des RSA-Verfahrens eine große Rolle. Das RSA-Verfahren beruht auf dem folgenden mathematischen

Satz: Ist $m = p * q$, sind p und q verschiedene Primzahlen und ist $e \in N$ mit $ggT\big((p-1) * (q-1), e\big) = 1$, dann ist die Abbildung $f(x) = x^e$ eine Permutation von Z_m. Gilt ferner für ein $d \in N$, dass $e * d = 1\ mod\ (p-1) * (q-1)$, so ist die Abbildung $g(x) = x^d$ die zu f inverse Permutation. d.h. $x^{e*d} = x$ für jedes $x \in Z_m$.

Dass für alle $x \in Z_m$ gilt: $x^{e*d} = x$, heißt:

$f\big(g(x)\big) = (x^d)^e = x^{d*e} = x$ und

$g\big(f(x)\big) = (x^e)^d = x^{e*d} = x$.

Also müssen f und g zwei zueinander inverse bijektive Abbildungen sein, was wir zeigen werden, d.h., wir zeigen, dass $x^{d*e} = x$, für alle $x \in Z_m$.

Für den Fall $x = 0$ ist diese Gleichung trivialerweise erfüllt. Daher nehmen wir an $x \neq 0$. Weiters setzen wir voraus, dass $x \leq m - 1$ ist.

$d * e = 1\ mod\ (p-1) * (q-1) \Rightarrow d * e = k * (p-1) * (q-1) + 1$ für ein

passendes $k \in N_0 \Rightarrow x^{d*e} = x^{k*(p-1)*(q-1)} * x$.

Für „p ist kein Teiler von q“ schreiben wir im Folgenden: $p \nmid q$ und unterscheiden für $1 \leq x \leq n-1$ d.h. $1 \leq x < p * q$,die drei Fälle:

1) $p \nmid x$ und $q \nmid x$, **2)** $p \nmid x$ und $q|x$, **3)** $p|x$ und $q \nmid x$.

Ad (1) $p \nmid x \Rightarrow x^{p-1} = 1\ mod\ p$ gemäß dem Kleinen Satz von Fermat. Aus $x^{p-1} = 1\ mod\ p$ folgt aber $x^{d*e} = (x^{(p-1)})^{k*(q-1)} * x = x\ mod\ p$. Das bedeutet: $x^{d*e} = x\ mod\ p$. Genauso ist zu sehen, dass $x^{d*e} = x\ mod\ q$ ist.

Ad (2) Wie bei (1) folgt $x^{d*e} = x\ mod\ p$. Nun aber ist $q|x$, also ist $x = 0\ mod\ q$ und daher $x^{d*e} = x\ mod\ q$. Analoges gilt im Fall (3).

Wir erhalten $x^{d*e} = k_1 * p + x$ und $x^{d*e} = k_2 * q + x$ mit geeigneten $k_1, k_2 \in N_0$. Da p und q also stets verschiedenen Primzahlen sind, folgt $x^{d*e} - x = k_3 * p * q$ für ein $k_3 \in N_0$, d.h. $x^{d*e} = x\ mod\ p * q = x\ mod\ m$.

Also gilt in Z_m: $x^{d*e} = x$. Das RSA-Verfahren ist ein Kryptosystem, bei dem es zu jeder Verschlüsselungsfunktion f auch eine Entschlüsselungsfunktion $g = f^{-1}$ gibt. Solche Abbildungen haben wir im obigen Satz gefunden.

Es existiert also ein d mit $0 \leq d < (p-1) * (q-1)$, welches die Gleichung $e * d = 1\ mod\ (p-1) * (q-1)$ löst und für welches $x^{e*d} = x\ mod\ m$ für jedes $x \in Z_m$. Es gibt allerdings mehr als ein d, welches die Bedingung $e * d = 1\ mod\ (p-1) * (q-1)$ erfüllt.

Satz: Sei $m = p * q$, wo p und q verschiedene Primzahlen sind, und $e \in N$ teilerfremd zu $(p-1) * (q-1)$ ist. Dann gilt $x^{e*d} = x\ mod\ m$ für alle $x \in Z_m \Leftrightarrow e * d = 1\ mod\ kgV(p-1, q-1)$.

Für alle x gilt dann $(x^{e*d-1}-1)*x = v*m = v*p*q \Rightarrow p|x^{e*d-1}-1$ für alle $x \in Z_m$ mit $x \nmid p$. $x^{e*d-1} = 1\, mod\, p$ für alle $x \in Z_m$ mit $x \nmid p \Rightarrow e*d-1 = v*(p-1)$ mit $v \in N$, denn angenommen für ein k, das kein positives Vielfaches von $p-1$ ist, sei $x^k = 1\, mod\, p\, für\, alle\, x \in Z_m$ mit $x \nmid p$, dann müsste für ein zu p teilerfremdes a gemäß dem vorigen Satz gelten $a^k = 1\, mod\, p$, woraus gemäß dem Kleinen Satz von Fermat folgt $k = r*(p-1)$ mit $r \in N$ im Widerspruch zur Annahme, dass k kein Vielfaches von $p-1$ ist.

Dasselbe Argument ergibt für q, dass $e*d-1 = w*(q-1)$ für ein $w \in N$. Mit $e*d-1 = v*(p-1)$ folgt, dass $e*d-1$ ein Vielfaches des $kgV(p-1,q-1)$ ist, d.h. $e*d = 1\, mod\, kgV(p-1,q-1)$. Also muss d diese Gleichung erfüllen, falls $x^{d*e} = x\, mod\, m$. Gilt umgekehrt $e*d = 1\, mod\, V$, wo $V = kgV(p-1,q-1)$, so ist $(p-1,q-1) = s*V$ und daher $e*d\, mod\, (p-1)*(q-1) = e*d\, mod\, s*V = 1$, da $e*d\, mod\, V = 1$ ist.

Primzahlen

Lange Zeit konnte kein großer Nutzen aus der Existenz der Primzahlen gezogen werden. Sie bestachen allein durch ihre besonderen Eigenschaften. Heute spielen sie eine der Hauptrollen in der Chiffrierung von Nachrichten. Sowohl der Algorithmus von RSA, als auch der Fiat - Shamir Algorithmus des Zero-Knowledgeverfahrens, arbeiten mit großen Primzahlen. Aus diesem Grund ist ihnen ein eigenes Kapitel gewidmet. Verschiedene Fragestellungen werden uns zu neuen, vielleicht überraschenden Erkenntnissen führen. Einige fundamentale Überlegungen seien hier vorweg genommen. Gilbert Brads stellt in seinem Buch Verschlüsselungsalgorithmen [b3, Seite 149] unter anderem folgende Fragen:

„**Wie häufig sind Primzahlen?**

Für die Implementierung von Verschlüsselungsverfahren ist es notwendig, innerhalb akzeptabler Zeiten die benötigten Parameter, zu denen Primzahlen gehören, bereitstellen zu können.

Sind die Primzahlen zufällig oder systematisch in der Menge der natürlichen Zahlen verteilt?

Eine systematische Verteilung bedeutet auch eine einfache Ermittelbarkeit von Primzahlen, was wiederum Angriffe auf Verschlüsselungsverfahren erleichtern würde und bei der Abschätzung der Verfahrenssicherheit berücksichtigt werden muss.

Welche Rückschlüsse lassen sich auf spezielle Sorten von Primzahlen ziehen?

In mehrfacher Hinsicht sichere Primzahlen müssen sich in akzeptabler Zeit finden lassen, um in Verfahren Verwendung finden zu können. Auch ist zu klären, ob sich nachweisen lässt, dass die Verwendung von solchen Primzahlen sicher ist (in vielen Protokollen ist der Inhaber dieser Information möglicherweise selbst nicht an der Einhaltung solcher Bedingungen interessiert.)

Ist eine Zahl eine Primzahl?

Eine Primzahl muss sich mit ausreichender Sicherheit in akzeptabler Zeit von einer zusammengesetzten Zahl unterscheiden lassen können."

Was ist eine Primzahl?

Definition: Eine natürliche Zahl wird Primzahl genannt, wenn sie genau zwei natürliche Zahlen als Teiler besitzt: die Zahl 1 und sich selbst.

Daraus resultieren drei Konsequenzen:

- Primzahlen lassen sich nicht als Produkt zweier natürlicher Zahlen darstellen, wenn beide größer als 1 sind.
- Lemma von Euklid:
  Ist ein Produkt zweier natürlicher Zahlen durch eine Primzahl teilbar, so ist bereits einer der Faktoren durch sie teilbar.
- Eindeutigkeit der Primfaktorzerlegung:
  Jede natürliche Zahl lässt sich als Produkt von Primzahlen schreiben. Diese Darstellung ist bis auf die Reihenfolge der Faktoren eindeutig.

Eine natürliche Zahl größer 1 heißt prim, wenn sie eine Primzahl ist, andernfalls heißt sie zusammengesetzt. 0 und 1 sind weder zusammengesetzt noch prim.

Spezielle Eigenschaften der Primzahlen

Alle Primzahlen bis auf die Zahl 2 sind ungerade. Die ungeraden Primzahlen lassen sich daher in der Form $p = 2 * k + 1,\ k \in N, k > 0$, darstellen. Ferner gilt: Jede Primzahl > 2 lässt sich in der Form $p = 4 * k + 3,\ k \in N$ oder aber, für $p > 3$ als $p = 6 * k + 1\ oder\ p = 6 * k - 1,\ k \in N$ schreiben. Nach dem dirichletschen Primzahlsatz gilt, dass es in jeder dieser vier Klassen unendlich viele Primzahlen gibt.

Hat eine natürliche Zahl die Form $n = 4 * m + 3,\ m \in Z^+$, dann enthält sie mindestens einen Primfaktor folgender Bauart: $p = 4 * k + 3$.

Sei $p > 2$ eine Primzahl. Man kann p als $p = a^2 + b^2$ schreiben, wenn p die Form $p = 4 * k + 1$ hat. Diese Darstellung von p ist bis auf die Reihenfolge der Vorzeichen, welche a und b haben eindeutig.

Besondere Primzahlen

Die Primzahlen für sich sind schon etwas Besonderes. Unter ihnen gibt es jedoch einige, die durch auffällige Eigenschaften hervor stechen.

Mersenne Primzahlen

Eine Mersenne Primzahl weist die Form $M_n = 2^n - 1$ auf. Allgemein wird die Zahl $2^n - 1$ die n-te Mersenne Zahl genannt. Ihren Namen verdanken diese Zahlen einem französischen Mönch, Marin Mersenne, der im 16. Jahrhundert auf sie stieß.

Die größten bis jetzt bekannten Primzahlen sind Mersenne Zahlen. Prinzipiell ist nicht jede Mersenne Zahl eine Primzahl. Trotzdem gibt es einige Tricks, mit denen man Mersenne Primzahlen relativ leicht finden kann.

Es gelten die folgenden Aussagen:

- Wenn $M_p = 2^p - 1$ eine Primzahl ist, dann muss auch p eine Primzahl sein. Dies grenzt die Suche ein, da nur mehr solche Zahlen getestet werden müssen, deren Exponent eine Primzahl ist.
- Es wird angenommen, dass die Wahrscheinlichkeit unter den Mersenne-Zahlen eine Primzahl zu finden viel größer ist, als bei einer zufällig ausgewählten ungeraden Zahl mit der gleichen Größenordnung.
- Der im nachfolgenden Kapitel behandelte Lucas-Lehmer Test bietet eine verhältnismäßig einfache Möglichkeit, zu testen, ob eine Zahl eine Primzahl ist oder nicht.

Fermatsche Primzahlen

Eine Fermatsche Primzahl weist die Form: $\mathrm{F_m} = 2^{2^{\mathrm{m}}} + 1$ auf. F_m heißt die m-te Fermatsche Zahl. Ob es sich dabei wirklich um eine Primzahl handelt, findet man für $m = 0, 1, 2, 3, 4$ einfach heraus:

Ergebnis: $3, 5, 17, 257, 65537$; es handelt sich um Primzahlen

Für $m = 5 - 32$ weiß man, dass sie keine Primzahlen sind. Allerdings kennt man von F_{14}, F_{20}, F_{22} und F_{24} keinen Faktor. Euler und Lucas zeigten, dass die Primfaktoren p einer Fermatschen Zahl immer die Bauart: $p = k * 2^{n+2} + 1$ besitzen.

Von insgesamt 213 der Fermatschen Zahlen weiß man, dass sie in Primfaktoren zerlegt werden können. Die größte dieser bekannten Zahlen ist die $F_{2145451}$. Fermatsche Primzahlen sind bis heute keine weiteren bekannt.

Die Carmichael Zahlen

Es handelt sich dabei um sogenannte eulersche Pseudoprimzahlen, das sind große Zahlen, von denen schwer zu erkennen ist, ob sie Primzahlen sind.

Generieren von Primzahlen und Primzahltests

Die großen Primzahlen welche bei kryptographischen Verfahren benötigt werden, müssen erst ausgewählt und gefunden werden. Aus Sicherheitsgründen existiert keine Liste in der sämtliche für private Zwecke geeignete Primzahlen gespeichert sind. In der Praxis wird deshalb so vorgegangen:

Ein Algorithmus erzeugt eine beliebige, möglichst zufällige Zahl. Mit verschiedensten Testverfahren wird geprüft, ob diese Zahl eine Primzahl ist, oder nicht. Da diese Überprüfung selbst mit Hochleistungsrechnern sehr lange dauern würde, geben die Programme die verwendet werden keine eindeutige Antwort. Sie stellen lediglich mit hoher Wahrscheinlichkeit fest, ob es sich um eine Primzahl handelt. Algorithmen, die so arbeiten, nennt man Monte-Carlo-Algorithmen.

Bekannte Primzahltests

Die Probedivision

Hierbei wird der nichttriviale Teiler (triviale Teiler: die Zahl selbst und 1) einer ganzen Zahl ermittelt, vorausgesetzt er existiert. Wird keiner gefunden, dann ist die getestete Zahl eine Primzahl. Dieses Verfahren ist jedoch viel zu zeit- und rechenintensiv, als dass es in der Praxis als Primzahltest bestehen könnte. Als Faktorisierungsverfahren, um Primfaktoren bis zu einer gewissen Schranke zu finden, eignet es sich sehr wohl. Man spricht dann von unvollständiger Probedivision.

Das Sieb des Eratosthenes

Mit dem Sieb des Eratosthenes wird eine Liste aller Primzahlen kleiner oder gleich einer bestimmten Zahl ermittelt. Genau genommen, ist dies kein echter Primzahltest. Es handelt sich nur um die Betrachtung einer Liste von Primzahlen bei der geprüft wird, ob die betreffende Zahl enthalten ist, oder nicht.

Die Idee sei hier nur kurz skizziert. Ausführlicheres findet sich beispielsweise auf: http://de.wikipedia.org/wiki/Sieb_des_Eratosthenes. Dieses Verfahren lässt sich besonders anschaulich in der Schule umsetzen lässt.

Der fermatsche Primzahltest

Der fermatsche Test ist eine probabilistische Überprüfung. Er liefert bei kleinen Testintervallen Ergebnisse mit kleiner Irrtumswahrscheinlichkeit. Sei P die Menge aller Primzahlen. Laut dem kleinen fermatschen Satz gilt:

$a^{n-1} \equiv 1\ mod\ n,\ \ f\ddot{u}r\ n \in P\ , a < p$. (Denn in Z_n^* gilt $a^{\varphi(n)} = 1\ mod n$, siehe früher)

a sei die Basis, zu der getestet wird. Damit lässt sich jedoch nicht mit Sicherheit schließen, dass die getestete Zahl prim ist. Wir können jedoch folgern:

$a^{n-1} \neq 1\ mod\ n\ \ \Rightarrow\ n$ ist keine Primzahl.

Jene zusammengesetzten Zahlen, die den kleinen Fermattest zur Basis a besteht, heißen Pseudoprimzahlen zur Basis a.

Beispiel:

Wir fragen uns, ob eine große Zahl n eine Primzahl ist. Dazu starten wir den fermatschen Primzahltest mit $a = 2$.

Wir betrachten die Tabelle:

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\mathbf{2^{(n-1)} mod\ n}$	1	0	1	2	1	0	4	2	1	8	1	2	4	0

Unter jeder Primzahl finden wir eine 1. In dieser Tabelle finden sich bei 1 ausschließlich Primzahlen. Fährt man jedoch weiter fort, erhält man bei 341 eine 1. 341 ist jedoch keine Primzahl, denn $341 = 11 * 31$, es handelt sich daher um eine Pseudoprimzahl. Die Anzahl

dieser Zahlen beträgt bis 2000 genau 303. Zur Basis a minimiert sich diese auf 7 Zahlen. Man wählt dann einfach eine andere Basis. Leider gibt es also Zahlen, die zusammengesetzt sind, und von diesem Test trotzdem als Primzahlen ausgewiesen werden. Paradebeispiel hierfür ist die Zahl 561.

Sie kann zerlegt werden in: $561 = 3 * 11 * 17$.

Sie besteht den Test für alle Basen bei denen gilt: $ggT(a, n) = 1$

Zahlen mit dieser speziellen Eigenschaft heißen nach ihrem Entdecker: Carmichael Zahlen.

Für sie gilt: $\forall a < n\ ,\ ggT(a, n) = 1 :\ \ a^{n-1} \equiv 1\ mod\ n$

Ihre Zerlegung besitzt mindestens drei Primfaktoren:

$n = p_1 * p_2 * \ldots * p_{k,} \quad k \geq 3, \qquad p_i \neq p_j$

Eine Verbesserung des fermatschen Primzahltest liefert der Lucas-Test.

Der Lucas-Test

Hierbei wird angenommen, N sein eine Primzahl. Gelingt es, ein a mit $1 < a < N$ zu finden, sodass die folgenden beiden Bedingungen erfüllt sind:

$a^{N-1} mod\ N \equiv 1$ und $a^m\ mod\ N \not\equiv 1$,

Für alle $m < N - 1,$ welche $N - 1$ teilen ist N tatsächlich prim.

Hierbei werden wenige Rechenschritte durchgeführt. Es müssen solche m getestet werden, für die gilt: $m|N - 1$

Einziger Schönheitsfehler dieses Verfahrens ist die Tatsache, dass man die Primfaktorzerlegung von $N - 1$ kennen muss. Diese erhält man meist nur durch eine rechenintensive Faktorisierung. Für Zahlen bestimmter Bauart eignet sich der Test jedoch sehr gut.

Der Lucas-Lehmer-Test für spezielle Zahlen

Getestet werden Mersenne Zahlen ab M_3. Der Grundgedanke des Tests beruht darauf, dass Mersenne Zahlen im Dualsystem aus einer Folge von Einsen bestehen. Die Funktionsweise sei kurz skizziert:

p sei ungerade und prim. Man definiert eine Folge $S(k+1)$ rekursiv durch

$S(1) = 4, \qquad S(k+1) = S(k)^2 - 2$

Es gilt dann:

$M_p = 2^p - 1$ ist genau dann eine Primzahl, wenn $S(p-1)$ durch M_p teilbar ist.

Um große Zahlen $S(k)$ zu vermeiden, werden alle Zwischenschritte modulo M_p berechnet.

$Sei\ S(1) = 4,\ S(k+1) = S(k)^2 - 2\ mod\ M_p$, dann ist $S(p-1) = 0 \Rightarrow M_p$ ist eine Primzahl.

Der Miller-Rabin-Test

Wie zu Beginn erwähnt, handelt es sich bei vielen Primzahltests um probabilistische Verfahren. Der Miller-Rabin-Test ist ein solcher. Er bestätigt mit einer gewissen Wahrscheinlichkeit die Aussage $n \in P$. Es gibt jedoch Zahlen, die den Rabin-Miller-Test bestehen, obwohl es sich bei ihnen nicht um echte Primzahlen handelt. Man nennt sie starke Pseudoprimzahlen.

Die Arbeitsweise

n sei eine ungerade Zahl ≥ 3. Es ist festzustellen, ob n prim ist oder nicht. Dazu wählt man, ähnlich dem fermatschen Primzahltest, Basen a aus. Sie sind frei wählbar aus der Menge $\{2,3,\dots,n-1\}$.

$n-1$ wird in seine geraden, und ungeraden Anteile zerlegt: $n - 1 = d * 2^j$

n ist entweder prim oder eine starke Pseudoprimzahl, wenn für ein r mit $0 \leq r \leq j-1$ gilt:

$a^d \equiv 1\ mod\ n$ oder $a^{d*2^r} \equiv -1\ mod\ n$

Der Test berechnet für die Basen $a\ mod\ n$ die Zahlen: $a^d, a^{2*d}, a^{4*d}, \dots, a^{2^{j-1}*d}, a^{2^j*d}$

wobei $2^j * d = n - 1$ ist.

Jedes Element dieser Folge von Zahlen ist das Quadrat des Vorherigen. Angenommen n ist prim, dann gilt nach dem kleinen fermatschen Satz:

$a^{2^j*d} \equiv a^{n-1} \equiv 1\ mod\ n$ und wir erhalten die Zahlen $a^d, a^{2*d}, a^{4*d}, \dots, a^{2^{j-1}*d}, 1$

Ist n eine Primzahl, dann hat die Kongruenz $x^2 \equiv 1\ mod\ n$ nur die beiden Lösungen $x = +1$ oder $x = -1\ mod\ n$, denn $x^2 \equiv 1\ mod\ n$ bedeutet $n|x^2 - 1$

$\Leftrightarrow n|(x-1) * (x+1)$

$\Leftrightarrow n|(x-1) oder\ n|(x+1)$

$\Leftrightarrow x - 1 \equiv 0\ mod\ n\ oder\ x + 1 \equiv 0\ mod\ n$

$\Leftrightarrow x \equiv 1\ modn\ oder\ x \equiv -1\ mod\ n$

Daraus ergibt sich, dass in der oben berechneten Folge der Vorgänger von 1 immer 1 oder -1 ist. Ist n eine Primzahl so erhält man 1, 1,…, 1 oder aber die Folge

$\Delta_1, \Delta_2, \dots, \Delta_r, -1, 1, \dots, 1$, wobei die Δ_i, $i = 1,2, \dots, r$ für beliebige Zahlen stehen.

Sei jetzt n die zu testende Zahl. Beginnt die Folge nicht mit 1 und enthält auch keine -1, dann kann n nicht prim sein. Unter Umständen könnte sie eine starke Pseudoprimzahl sein. Da für ein nicht primes $n \geq 3$ höchstens $\frac{n-3}{4}$ Elemente a mit $ggT(a,n) = 1$ in der Menge $\{2,3, \dots, n-2\}$ vorkommen, ist die Wahrscheinlichkeit, dass man ein a erwischt, welches nicht erkennen lässt, dass n zusammengesetzt ist, kleiner als $\frac{1}{4}$. (Ist $ggT(a,n) > 1$ oder $a^{d*2^r} = 0$ modifizieren wir $r < s$, so ist sofort zu sehen, dass n zusammengesetzt ist). Wiederholt man den Test für k verschiedene $a \in \{1, \dots, n-2\}$ so ist die Fehlerwahrscheinlichkeit $< \frac{1}{4^k}$.

Beispiel:

Wir untersuchen, ob die Zahl 13 eine Primzahl ist. Die Fehlerwahrscheinlichkeit soll dabei $< \frac{1}{100}$ sein.

Wir testen: $n = 13$ mit 4 verschiedenen Basen a:

$13 - 1 = 12 = 3 * 2^2$, d.h. $\mathrm{d} = 3$ und $\mathrm{j} = 2$.

Sei **a = 5, 8, 10 und 11.** Dann erhalten wir

$(5^3, 5^6) = (8, -1) mod\ 13$

$(8^3, 8^6) = (5, -1) mod\ 13$

$10^3 = -1 \, mod \, 13$

$(11^3, 11^6) = (5, -1) mod \, 13$

Die Zahl 13 hat alle Tests bestanden. Da wir vier $n \in \{2, \dots, 11\}$ gewählt haben ist die Fehlerwahrscheinlichkeit kleiner als $\frac{1}{4^4}$= 0,00391.

Public-Key-Kryptosysteme nach Diffie und Hellman

In der Begriffsbestimmung wurde einiges über asymmetrische Chiffrierverfahren erläutert. Auf den folgenden Seiten wird der Hintergrund detailierter beschrieben.

Bei Public-Key-Kryptosystemen wird zum Chiffrieren und zum Dechiffrieren nicht derselbe Schlüssel verwendet. Zur Verschlüsselung werden, wie in den mathematischen Grundlagen erwähnt, Einwegfunktionen verwendet.

Die Details:

Jeder Teilnehmer eines Nachrichtenaustausches veröffentlicht eine Verschlüsselungsfunktion S_X. Seine Entschlüsselungsfunktion T_X bleibt geheim. Die beiden Funktionen haben folgende Eigenschaften:

1. $S_X\big(T_X(m)\big) = m$. Die beiden Funktionen sind zueinander invers. (m= message, alle Nachrichtenwörter)
2. S_X und T_X sind am Rechner beide leicht auszuführen.
3. Es ist nahezu unmöglich aus der Kenntnis von S_X ein geheimes T_X zu berechnen. S_X ist eine Einwegfunktion.

Übermittelung von Nachrichten

A möchte nun eine Nachricht an B senden. Wir schreiben kurz: $A \overset{m}{\rightarrow} B$. Dazu wendet A den öffentlichen Schlüssel S_B auf die Nachricht an. A sendet $S_B(m)$ an B. B wendet seine Entschlüsselungsfunktion T_B auf $S_B(m)$ an:

$T_B\big(S_B(m)\big) = m$ hieraus erhält B die Nachricht m.

Signieren von Nachrichten

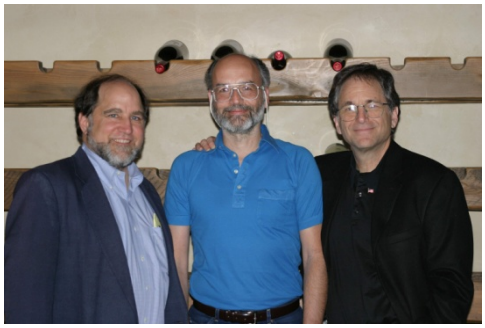
A möchte eine Nachricht an B signieren, um B davon zu überzeugen, dass A der echte Absender der Nachricht ist. Dazu sendet A an B: $S_B(T_A(m))$. B dechiffriert, indem er zuerst seinen geheimen Schlüssel T_B anwendet, und dann den von A öffentlichen Schlüssel S_A. Hieraus erhält er: $S_A[T_B(S_A\{T_A(m)\})] = m$. Da ausschließlich A die Funktion T_A kennt, muss die Nachricht von A kommen. Da nur B als Einziger Zugriff auf T_B hat, kann nur B die Nachricht lesen.

Sollte die Eigenschaft bei Punkt 1 nicht gelten, sprich $(S_X(T_X(m)) \neq T_X(S_X(m))$, so ist das Signieren von Nachrichten nicht möglich. Dies ist beim Knapsack-Verfahren der Fall. Auf den folgenden Seiten werden wir uns mit dem RSA-Verfahren und Zero-Knowledge beschäftigen, sowie dem öffentlichen Schlüsseltausch nach Pohlig und Hellman.

RSA

Das bekannteste und heute noch wichtigste Public-Key-Kryptosystem ist das RSA-Verfahren. Die Grundlagen sowie die Mathematik die hinter dem Algorithmus stehen, finden sich in den Vorkenntnissen für die Lehrenden.

Der Name RSA leitet sich aus den Anfangsbuchstaben seiner drei Erfinder ab. Ronald **R**ivest, Adi **S**hamir und Leonard **A**dleman entdeckten diese Möglichkeit der Verschlüsselung 1977, als sie zu zeigen versuchten, dass ein solches Verfahren nicht existieren kann. Das Gegenteil trat ein. Mit diesen Erkenntnissen produzierten sie das RSA-Kryptosystem. Die Entwicklung dauerte einige Monate. Rivest lieferte die Vorschläge, Adleman führte die Angriffe, und Shamir arbeitete auf beiden Fronten mit. Mit ihrer Arbeit zeigten sie, wie einem klassischen Teil der Zahlentheorie der Einzug in die angewandte Mathematik gelingen konnte. Der Algorithmus besticht insbesondere durch seine Einfachheit, weshalb er sich auch - mit kleinen Zahlen - für den Schulunterricht eignet.



Von links nach rechts, die drei „Väter“ von RSA: Rivest, Shamir und Adleman aus [a1]

Der Algorithmus

Wir erinnern uns an die mathematischen Grundlagen zu Beginn, in denen die Abbildungen f und g erwähnt wurden. Mit $f(x) = x^r$ und $g(x) = x^u$. Für alle $x \in Z_m$ erhielten wir: $x^{r*u} = x$. Der Kernpunkt des ganzen Verschlüsselungsverfahrens ist nun, dass diese Abbildungen Bijektionen in Z_m sind. Zu jeder Verschlüsselungsfunktion ist die Existenz einer Entschlüsselungsfunktion gewährleistet. Der RSA-Algorithmus hat zwei unterschiedliche Anwendungen: Erstens ist es möglich mit seiner Hilfe Nachrichten zu verschlüsseln und zweitens kann man ihn auch als digitale Signatur verwenden. Folgende Phasen 1 und 2 beziehen sich auf die Nachrichtenverschlüsselung, Phase 3 beschreibt das Signaturschema.

Phase 1: Die Schlüsselerzeugung

- Wähle zwei etwa gleich große Primzahlen p und q und bilde deren Produkt $n = p * q$. Die Länge der beiden Zahlen sollte zwischen 384 und 512 Bit betragen. Damit ergibt sich für n eine Länge zwischen 512 und 1024 Bit.
  Die Primzahlen können von ungefähr gleicher Stelligkeit sein, ja sogar von gleicher Stelligkeit, sodass die ersten Hälften der Ziffern übereinstimmen.
- Möglichkeit 1: Wähle eine kleine ungerade natürliche Zahl e, die zu $\varphi(n) = (p-1) * (q-1)$ relativ prim ist, d.h. es gilt: $ggT\big(e, \varphi(n)\big) = 1$.
- Möglichkeit 2: Wähle eine ganze Zahl $e > 1$, die zum $kgV(p-1, q-1)$ teilerfremd ist. Da e und $kgV(p-1, q-1)$ teilerfremd sind, ist die lineare Kongruenz $e * x \equiv 1 \; mod \; kgV(p-1, q-1)$ eindeutig lösbar. Es gilt: $e * d \equiv 1 \; mod \; kgV(p-1, q-1)$.
- Berechne d als die Lösung von $e * d = 1 \; mod \; \varphi(n)$. d existiert, und ist eindeutig bestimmt. Mit Hilfe des erweiterten euklidischen Algorithmus kann d berechnet werden. Das d, welches mit Möglichkeit 1 berechnet wird, ist im Allgemeinen größer als jenes, das mit Möglichkeit 2 berechnet wird. Jene d, die sich gut zum Chiffrieren eignen, unterscheiden sich um Vielfache vom $kgV(p-1, q-1)$. Weiters sollte der $ggT(p-1, q-1)$ möglichst klein sein. Aus den letzten beiden Sätzen erkennen wir, dass der Abstand zwischen zwei d möglichst groß sein sollte, was das Auffinden solcher d erschwert.
- Wir erhalten den öffentlichen Schlüssel als das Paar: $P = (e, n)$ und den geheimen Schlüssel als das Paar: $P = (d, n)$

Resultierend aus den großen Zahlen ergibt sich eine lange Rechenzeit, trotz den mittlerweile sehr leistungsfähigen Computern.

Phase 2: Die Anwendung

Bei der Anwendung unterscheidet man wieder zwei Phasen.

Die Verschlüsselungsphase:

Wir codieren die Nachricht durch eine natürliche Zahl deren Ziffernfolge in Blöcke gleicher Länge $\leq n-1$ zerteilt wird. Sei M die einen Block repräsentierende Zahl mit $0 \leq M \leq n-1$. Die „Nachricht" $m \in Z_n$ wird chiffriert durch $c = m^e \ mod \ n$.

Die Entschlüsselungsphase:

Ein Chiffretext $c \in Z_n$ wird decodiert durch $m = \ c^d \ mod \ n$.

Zum besseren Verständnis der beiden Phasen möchten wir ein Beispiel mit kleinen Zahlen anführen. (Zur Berechnung der entsprechenden Werte wurde DERIVE verwendet). Lisa möchte Bart eine geheime Nachricht senden.

Geheimer Bereich von Bart:

- Bart wählt zwei verschiedene Primzahlen $p = \ 97$ und $q = 131$.
- er bildet $n = p * q = \ 12707$
- er bildet das $kgV(p-1, q-1) = \ 6240$
- er wählt eine zum $kgV(p-1, q-1)$ teilerfremde Zahl $e, e = \ 331$
- er ermittelt aus e und dem $kgV(p-1, q-1)$ die Zahl d, $e * d = 1 \ mod \ \varphi(n)$, $d = \ 5731$= sein geheimer Schlüssel
- Dann gibt Bart die beiden Zahlen $n = 12707$ und $e = 331$ öffentlich bekannt. Hat Bart von Lisa die Nachricht $c = 11453$ erhalten, so entschlüsselt er diese durch
- $m = \ c^{\ d} mod \ n$
- $m = \ 11453^{5731} = \ 9865 \ mod \ n$
- Wie ist die Nachricht, welche Lisa an Bart gesandt hat zustande gekommen?
- Lisa wählte einen Block der Länge $< n$ aus dem als Zahlen codierten Datenstrom, den sie übertragen wollte; sie kam so auf die Zahl: $m = \ 9865$
- Lisa verschlüsselte sie mit der Formel $c = \ m^e \ mod \ n$.
- $c = \ 11453 = \ 9865^{331} \ mod \ 12707$

Phase 3: Signieren mit RSA

Erstellung:

Will Lisa eine Nachricht m signieren, folgt sie dem allgemeinen Signaturschema (siehe oben). Ihr öffentlicher Schlüssel sei (l, a) ihr geheimer sei b. Sie bildet $m_1 = m^b \; mod \; l$, d.h., sie wendet zunächst ihren geheimen Schlüssel an und berechnet dann mit dem öffentlichen Schlüssel von Bart $sig \; = m_1^e mod \; n$.

Verifizierung:

Bart kann die von Lisa erstellte Signatur überprüfen, indem er zunächst auf sig seinen geheimen Schlüssel d anwendet, womit er $m_1 = sig^d \; mod \; l$ erhält. Anschließend überprüft er mit dem öffentlichen Schlüssel von Lisa, ob eine sinnvolle Nachricht hinter m_1 steckt: $m = {m_1}^a \; mod \; n$. Nur wenn die Nachricht wirklich von Lisa gekommen ist, erhält er m. Zu beachten ist, dass m so gewählt wird, dass $m_1 < n$ ist. Die Signatur ist die RSA-Entschlüsselung von m. Das beschriebene Signaturschema ist verbesserungsfähig. Es birgt einige Gefahren in sich, da die Signatur unverschlüsselt gesendet wird. Sollte es einem Angreifer gelingen Bart seinen Schlüssel als den von Lisa zu verkaufen, kann er danach Signaturen erzeugen, von denen Bart fälschlich annimmt sie seien von Lisa. Bart muss sich daher von der Echtheit von Lisas Schlüssel überzeugen. Dies gelingt mit Hilfe des Web of Trust oder den Schlüsselzertifizierungsstellen (weiteres dazu findet sich im Kapitel über Zero-Knowledge). Eine sichere Möglichkeit bietet die Signatur mit Hashwerten. Der interessierte Leser sei auf [b4, Seite 209] verwiesen.

Es ist zu bemerken, dass große Nachrichten nicht gänzlich mit RSA verschlüsselt werden. Dies wäre viel zu rechen- und zeit intensiv. Meist wird die Nachricht mit einem symmetrischen Verfahren chiffriert. Der dafür verwendete Schlüssel wird oft mittels RSA-Verfahren ausgetauscht.

Die Sicherheit

Sie beruht im Wesentlichen darauf, dass es für einen Angreifer nahezu unmöglich ist, aus dem öffentlichen Schlüssel (e, n) den geheimen Schlüssel d zu berechnen. Diese Berechnung ist gleich zu setzen mit der Schwierigkeit n in seine Primfaktoren p und q zu zerlegen. Der Knackpunkt des RSA-Verfahrens reduziert sich auf die Lösung eines Problems allgemeinen mathematischen Interesses: der Faktorisierung einer natürlichen Zahl. In der Zahlentheorie

wird dieses Problem schon seit Jahrhunderten untersucht. Bis jetzt gilt es als schwierig. Sollte sich aber herausstellen, dass es wider Erwarten doch einen Algorithmus gibt, der in „Polynomzeit“ das Ergebnis liefert, dann sind damit auch viele andere mathematische Probleme der gleichen Komplexität in Polynomzeit zu lösen. Die Schwierigkeit der Faktorisierung von n hängt natürlich eng mit der Wahl von p und q zusammen. Deshalb stellt sich die Frage: Wie sieht eine gute Wahl aus?

Die richtige Wahl für p und q

Für eine knifflige Faktorisierung werden p und q möglichst gleich groß gewählt. Ist n von der Länge 1024 Bit, sind die beiden Zahlen 512 Bit groß. Weiters sollten sie so gewählt werden, dass bekannte Faktorisierungsverfahren einen großen Aufwand benötigen, um sie zu errechnen. Dies ist jedoch nicht immer vorherzusagen.

Buchmann schreibt in seinem Buch, Einführung in die Kryptographie: „Nach heutiger Kenntnis sollte n wenigstens 512 Bits lang sein. Aus experimentellen Untersuchungen des Faktorisierungsproblems ergibt sich, dass bei längerfristiger Sicherheit, n eine Länge von 1024 oder sogar von 2048 Bits haben sollte. Solche Empfehlungen sind aber mit Vorsicht zu genießen, weil niemand den Fortschritt im Bereich der Algorithmen oder den Fortschritt bei der Hardwareentwicklung prognostizieren kann.“ [b4]

Die beiden Primzahlen p und q dürfen für ein gutes n auch nicht zu nahe beieinander liegen. (Sie sollten sich bei gleicher Stelligkeit auf alle Fälle mindestens in der zweiten Hälfte ihrer Stellen unterscheiden.)

Ein Beispiel:

Wenn die erste Hälfte der Stellen von p mit der ersten Hälfte der Stellen von q übereinstimmt, dann kann ihre Absolutdifferenz immer noch sehr groß sein. Relativ gesehen, liegen sie aber schon viel zu eng beieinander. Mit dem fermatschen Algorithmus lässt sich ein n, das aus zwei zu eng aneinander liegenden Zahlen p und q besteht unter Umständen leicht berechnen.

Sei $n = p * q,\ p$ und q seien Primzahlen $> 2,$ dann gilt mit $u = \frac{p+q}{2}, v = \frac{p-q}{2}$:

$$n = u^2 - v^2 = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = p * q$$

Durch Umformung der Gleichung erhält man $v^2 = u^2 - n$.

Man testet jetzt für verschiedene Werte $u > \sqrt{n}$ ob $u^2 - n$ eine Quadratzahl ist.

Gestartet wird mit $(\sqrt{n} + 1)$, dann jeweils um 1 erhöht. Findet man ein u, so dass $u^2 - n$ eine Quadratzahl ist, dann hat man die Faktorisierung gefunden. Denn es gilt

$n = (u + v) * (u - v) = p * q.$

Prinzipiell gilt, es ist sehr leicht das Produkt zweier großer Primzahlen zu bilden, jedoch die Faktorisierung dieses in seine Primfaktoren erweist sich als nahezu unmöglich. Diese Art von Problemen steht im engen Zusammenhang mit den sogenannten Einwegfunktionen. Für ein $f: A \to B$ gilt, dass für alle $x \in$ A der Wert $f(x)$ leicht zu berechnen ist. Es ist jedoch nahezu unmöglich zu einem gegebenen $y \in B$ ein x zu finden, mit $f(x) = y$, außer mittels geheimer Zusatzfunktion. Die Funktionen mit denen man im RSA-Verfahren Nachrichten verschlüsselt, sind typische Einwegfunktionen. Eine Richtung (Multiplikation) funktioniert problemlos, die andere (Faktorisierung) ist knifflig.

Die wichtigsten Kriterien für eine gute Wahl von p und q laut [b6] seien hier zusammengefasst:

1. n hat mehr als 200 Stellen.
2. p und q haben beide somit eine Größenordnung von rund 100 Stellen. In der Länge der Dezimaldarstellung sollten sie sich jedoch um einige Stellen unterscheiden.
3. $p - 1$ und $q - 1$ enthalten große Primfaktoren.
4. Der $ggT(p - 1, q - 1)$ sollte möglichst klein sein.

Rechenzeit zur Faktorisierung

Der einfachste Algorithmus zur Faktorisierung einer Zahl ist das Sieb des Eratosthenes. Die Zahl n wird durch alle Zahlen kleiner oder gleich $\sqrt{n}$ dividiert. Ist die Division erfolglos, können sofort alle weiteren Vielfachen aus der Liste der möglichen Primfaktoren gestrichen werden. Dabei wächst die Rechenzeit proportional zu $\sqrt{n}$.

- n ist in Binärdarstellung b Bit lang
- $n \approx 2^b$ und für die Rechenzeit $T(b)$ gilt:

$T(b) = c * \sqrt{n} \approx c * \sqrt{2^b} = c * \sqrt{2}^b$

Mit der Zahl der Bits von n wächst die Rechenzeit exponentiell. Die besten Algorithmen zur Faktorisierung von n arbeiten heuristisch. Sie probieren verschiedene Primzahlen durch.

4 Angriffe auf RSA

Bezüglich der folgenden vier möglichen Angriffe auf die RSA Verschlüsselung siehe [b14].

Angriff 1

Gleiche Primzahlen in verschiedenen Modulen

Jeder Teilnehmer eines Publik-Key-Kryptosystemes hat seinen eigenen öffentlichen Schlüssel (n, e). Er stellt fest, dass sein Modul n nicht teilerfremd zu einem anderen Modul n ist. Das heißt, er kennt einen Faktor des fremden Moduls und kann dieses problemlos faktorisieren. Eines Tages werden Millionen öffentlicher Schlüssel frei im Netz verfügbar sein. Es gibt dann Billionen von möglichen Schlüsselpaaren. Theoretisch könnte mindestens ein Paar einen gemeinsamen Teiler haben.

Praktisch besteht diese Gefahr nicht. Laut dem Primzahlsatz aus der Zahlentheorie gilt:

Die Anzahl aller Primzahlen $\pi(N)$ kleiner als einer Zahl N kann für große N näherungsweise durch $\frac{N}{lnN}$ beschrieben werden. Genauer gesagt gilt:

$$\lim_{N\to\infty} \frac{\pi(N) lnN}{N} = 1$$

Daraus lässt sich folgern: Zwischen 2^{512} und 2^{513} befinden sich ungefähr so viele Primzahlen wie zwischen 1 und 2^{512}. Das sind in etwa $7{,}5 * 10^{151}$. Eine Zahl die in unseren Gedanken und unserer Sprache nicht fassbar scheint. Daher resultiert aus der Anzahl der möglichen Primzahlen kein Risiko.

Viel wahrscheinlicher ist hingegen jener Fall, dass einige Anwender durch die schlechte Wahl der Zufallszahlen gleiche Primzahlen erhalten. Durch sorgfältiges generieren der Zufallszahlen kann dieses Risiko jedoch so gering wie möglich gehalten werden.

Angriff 2

Diebstahl des privaten Schlüssels

Der öffentliche Schlüssel ist für jeden Teilnehmer frei im Netz einsichtig. Der private Schlüssel ist auf dem eigenen Rechner versteckt und wird dadurch geschützt. Oder?

Hier beginnt es bereits problematisch zu werden. Private Keys werden im Allgemeinen selbst verschlüsselt und gespeichert. So sind sie vor ungewollten Anwendern geschützt. Beginnt

man jedoch mit ihnen zu arbeiten, zu entschlüsseln, dann stehen sie als zusammenhängende Ketten im Speicher. Bis jetzt war man der Ansicht, dass sie in den riesigen Datenmengen gut genug verborgen sind. Leider ist dies nicht der Fall. Für die Schlüsselerzeugung werden immer möglichst zufällige Zahlenfolgen verwendet. Durch dieses Merkmal unterscheiden sie sich von allen anderen auf dem PC gespeicherten Bits. Programmierten Texten liegt eine Struktur zu Grunde. Sie weisen ein bestimmtes Muster auf und wiederholen sich. So zeigte Adi Shamir mit einem Kollegen auf einer Tagung im Jahre 99, wie unglaublich leicht ein privater Schlüssel auf einem Rechner zu finden ist.

Die Problematik besteht nun darin: Nehmen wir an es gelingt einer außenstehenden Person sich mit unerlaubten Methoden in einen Server zu hacken. Diese Person bringt den Server zum Abstürzen und braucht danach nur mehr den Speicher zu analysieren. Der Vorgang ist so einfach, dass er auch automatisch durchgeführt werden könnte.

Trotz dieser Thematik besteht kein Grund zur übermäßigen Sorge. Die Schlüssel, die mit dieser Methode gefunden werden können, weisen ein Bit Länge von 1024 oder 2048 auf. Für jene mit einer Bit Länge von 128 trifft das nicht zu. Als Ausweg empfiehlt es sich, immer wieder neue Schlüssel zu generieren.

Wie wir sehen, hat das beste Verschlüsselungsverfahren der Welt keinen Sinn, wenn man nicht in allen Bereichen absolute Vorsicht walten lässt.

Angriff 3

Heimliches oder zufälliges Verändern des privaten Schlüssels

RSA Signaturen sind meist sehr zeitaufwendig. Deshalb versucht man so gut es geht, Rechenzeit einzusparen und den Algorithmus trotzdem sicher zu lassen. Dabei verwendet man den chinesischen Restsatz mit dessen Hilfe man ein System von linearen Kongruenzen lösen kann. (Ausführlicheres findet sich in [b4]). Man berechnet dabei nicht mehr Werte modulo $p * q$, sondern entweder modulo p, oder modulo q.

Durch einen Fehler, dabei ist irrelevant ob sich dieser in der Software oder in der Hardware ereignet, wird ein Bit von p oder q geändert. Diese minimale Änderung hat zur Folge, dass man n bereits faktorisieren kann. Somit wäre das Verfahren geknackt, was einer wahren Katastrophe gleichkommen würde.

Abhilfe für dieses Problem bietet die Möglichkeit, Signaturen immer wieder auf ihre Korrektheit zu prüfen. Allerdings ist das wieder mit Rechenzeit verbunden.

Angriff 4

Angriff mit ausgewähltem Geheimtext

Dieser Angriff zielt nicht auf einen verschlüsselten Text ab, sondern auf eine digitale Signatur, welche mit RSA erzeugt werden kann:

Lisa erzeugt eine Unterschrift indem sie eine Text mit ihrem privaten Key verschlüsselt. Bart kann sich davon überzeugen, dass Lisa etwas signiert hat, indem er ihren öffentlichen Key darauf anwendet.

Der Angriff läuft im Detail wie folgt ab: Homer, der Angreifer, fängt einen chiffrierten Schlüssel von Lisa ab. Er kennt also $c = m^e \ mod \ n$ (In diesem Fall ist m der sogenannte Sitzungs-, und (e, n) Lisas öffentlicher Schlüssel.). Um die Nachricht lesen zu können, muss Homer m herausfinden. Dazu verschlüsselt er eine beliebige zu n teilerfremde Zahl mit dem öffentlichen Schlüssel von Lisa (Nichtteilerfremde Zahlen von n sind Primfaktoren von n. Wüsste er diese, wäre er ohnehin fertig.). Anschließend multipliziert er den erhaltenen Geheimtext mit dem abgehörten Geheimtext c:

$y = c * r \ mod \ n$

Homer bittet Lisa, dieses y zu unterzeichnen. Lisa dechiffriert indem sie berechnet:

$z_i = y^d = c^d * r^{e*d} \ mod \ n.$

Dieses Ergebnis sendet sie an Homer zurück. Homer kennt nun den Rest, da:

$m = c^d \ mod \ n$ und $r^{e*d} = r \ mod \ n$

und er mit Hilfe des erweiterten euklidischen Algorithmus die Gleichung $z = \ m * r \ mod \ n$ lösen kann. Dazu multipliziert Homer z mit r^{-1}, welches er aus $r * x = 1 \ mod \ n$ berechnet: $z* r^{-1} = m \ mod \ n.$

Es wäre natürlich viel schneller gewesen, hätte Homer Lisa gleich $c = m^e \ mod \ n$ zum unterschreiben gegeben. Dabei wäre Lisa wahrscheinlich hinter die betrügerischen Absichten gekommen. Dazu hätte sie allerdings alle Sitzungsschlüssel aufbewahren und diese mit c vergleichen müssen.

Bei dem beschriebenen Angriff hat Lisa jedoch keine Chance das Komplott auf zu decken. Da empfiehlt es sich $c = m^e \ mod \ n$ RSA-verschlüsselt an Bart zu senden.

Eine Liste weiterer Angriffe auf RSA findet sich in [b9]:

Angriff	**Erkenntnis**
Chosen-Ciphertext-Angriff gegen eine RSA-Signatur. (Angriff 4)	Niemals beliebige, von Unbekannten vorgelegte, Dokumente unterzeichnen.
Angriff gegen RSA mit gemeinsamen Modul n. (Angriff 1)	Eine Benutzergruppe darf niemals den gleichen Wert für n wählen.
Bei kleinem Verschlüsselungsexponenten e und kurzen Nachrichten ist ein Angriff möglich.	Nachriten vor der Verschlüsselung anfüllen. Blöcke von M sollten etwa gleich lang sein wie n. PGP tut dies automatisch.
Angriff gegen RSA mit kleinem Entschlüsselungsexponenten d.	Großes d wählen. Dies ist bei kleinem e automatisch der Fall.
Angriff gegen Nachrichten die zuerst verschlüsselt und dann signiert wurden.	Immer zuerst die Nachricht signieren und dann verschlüsseln.

Wettbewerbe

Die Zerlegung von n

Wie bereits erwähnt, basiert die Sicherheit des RSA-Verfahrens hauptsächlich darauf, dass es nahezu unmöglich ist große Zahlen zu faktorisieren. Die Spezialisten nehmen an, dass es keinen geeigneten Algorithmus gibt, bewiesen ist es jedoch nicht. Die Faktorisierung von großen n in Einzelfällen gelang immer wieder:

Im Jahr 1995 erregten Arjen K. Lenstra und Mark S. Manasse großes Aufsehen, als sie verkündeten, sie hätten die Fermat Zahl $F_9 = 229 + 1 = \ 2^{512} + 1$ in ihre Primfaktoren zerlegt.

Im August 1999 wurde eine 512 Bit lange RSA Zahl (das entspricht 155 Dezimalstellen) erfolgreich zerlegt.

Ein weiterer Meilensteil gelang im Mai 2005 als das Bundesamt für Sicherheit und Informationstechnik in Bonn zusammen mit der Universität und dem CWI in Amsterdam eine 200-stellige RSA Zahl faktorisierte.

Die Ruhe vor dem Sturm?

Die Ereignisse wurden in der Öffentlichkeit mit unterschiedlichen Gefühlen aufgenommen. Erstaunen lösten sie wahrscheinlich bei allen aus. Es folgten zahlreiche Diskussionen über die Sicherheit von RSA und anderen, ähnlichen Verfahren. Die eine Seite jubelte und sah ihre Ablehnung gegenüber dem Algorithmus bestätig, die andere hielt entgegen, dass so „kleine" Zahlen in keinem Verfahren verwendet werden würden.

Weiters unterscheiden sich die Expertenmeinungen, wenn es um die Bitlänge von n geht. Wie bereits erwähnt gelang bis jetzt „nur" die Faktorisierung von „kleinen" RSA Zahlen. Der aktuelle Rekord liegt laut Beutelspacher [b1] bei 512 Stellen. Deshalb sollte es ausreichen mit einer Bitlänge von 786 zu arbeiten. Kritische Geister argumentieren, dass die Weiterentwicklung in allen Bereichen der Wissenschaft dazu führen sollte, größere RSA Zahlen einzusetzen. Sie fordern eine Bitlänge von 1024 oder im Idealfall sogar von 2048. Dabei handelt es sich um Empfehlungen des BSI (deutsches **B**undesamt für **S**icherheit und **I**nformationstechnik). Die Wichtigkeit der Informationen die damit geschützt bleiben soll liegt im Bereich von 24 Stunde.

Zero-Knowledge

Vorstellung der Idee

Im Alltag sprechen wir oft von Geheimnissen. Das Geheimnis eines guten Vortrags, das Geheimnis der Kochkunst, der Liebe oder Ähnlichem. Hierbei handelt es sich um Kenntnisse und um Fähigkeiten, die nicht allen Menschen zur Verfügung stehen. Sie verlieren ihre Kraft allerdings nicht, wenn sie öffentlich bekannt gemacht werden.

Die Geheimnisse die uns im folgenden Kapitel interessieren sind Zahlen und Bits, es sind wertvolle Informationen. Sie können zum Beispiel eine „Sesam öffne" dich Funktion haben. Dabei denken wir an die EC-Karte, an ein Passwort oder etwas in der Art. Ich überzeuge jemanden von meiner Identität, indem ich mein Geheimnis im kleinen Rahmen offenbare. Wir müssen dieses Passwort eingeben, vorweisen, oder manchmal sogar laut aussprechen. Es scheint einleuchtend, dass diese Methode sehr unsicher ist. Jemand könnte uns beobachten,

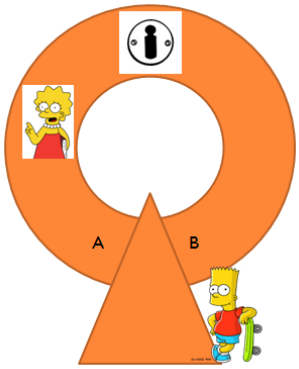
oder belauschen und so in den Besitz unseres Geheimnisses kommen. Das ist gefährlich und kann zu großem Schaden führen.

Der Clou bei Zero-Knowledge ist ein anderer und es ist nahezu paradox.

Ich überzeuge jemanden davon ein bestimmtes Geheimnis zu besitzen, ohne das Geheimnis selbst zu verraten.

Nur dann handelt es sich um ein gutes Geheimnis. Es ist einer einzigen Person bekannt, und wird NIE mit einer anderen Person ausgetauscht. Wie soll man sich das vorstellen? Zur besseren Verständnis betrachten wir ein Spiel:

Bart und Lisa - die Idee von Zero-Knowledge



Die Skizze zeigt die beiden Türen A und B, sowie Lisa (links) und Bart(rechts) [a2]

Bart und Lisa spielen ein Spiel. Lisa der kluge Kopf der beiden (links oben), kennt das Passwort für das Schloss in der Mitte der Grafik. Bart (rechts unten) kennt es nicht. Lisa versucht Bart davon zu überzeugen, dass sie die Tür öffnen kann. Sie verrät dabei aber niemals ihr Geheimnis. Bart kann sich trotzdem davon überzeugen, dass sie ihn nicht belügt.

Der Spielverlauf

Gespielt wird in mehreren Durchgängen, wobei alle nach dem gleichen Schema ablaufen.

Lisa betritt den Vorraum (Dreieck) und wählt, zwischen Tür A und Tür B. Sie geht in eine Tür hinein. Bart betritt den Vorraum und sieht zwei verschlossene Türen. Hinter einer der beiden steht Lisa. Er wählt zufällig eine Tür aus. Dann ruft er laut welche Tür es ist. Kennt Lisa das Geheimnis tatsächlich, wird sie immer hinter der von Bart genannten Tür herauskommen.

- 1. Fall: Lisa steht hinter Tür A. Bart wählt Tür A. Lisa kommt natürlich durch Tür A zu ihm heraus.

- 2. Fall: Lisa steht hinter Tür A. Bart wählt Tür B. Jetzt muss Lisa ihren Schlüssel verwenden und das Schloss aufsperren. Danach kann sie bei Tür B hervorkommen.

Der misstrauische Leser könnte anmerken, dass Bart nur mit einer Wahrscheinlichkeit von 50 Prozent nicht von Lisa betrogen wird. Das stimmt für den Einzelfall. Wie zu Beginn erwähnt wird jedoch in mehreren Durchgängen gespielt und Bart könnte Lisa „bis in alle Ewigkeit" testen um sicher zu gehen, dass sie immer bei der richtigen Tür heraus kommt. Gott sei Dank ist das nicht notwendig, denn Bart lässt sich viel schneller davon überzeugen, dass Lisa tatsächlich den richtigen Schlüssel besitzt und nicht nur Glück hat und zufällig richtig herauskommt.

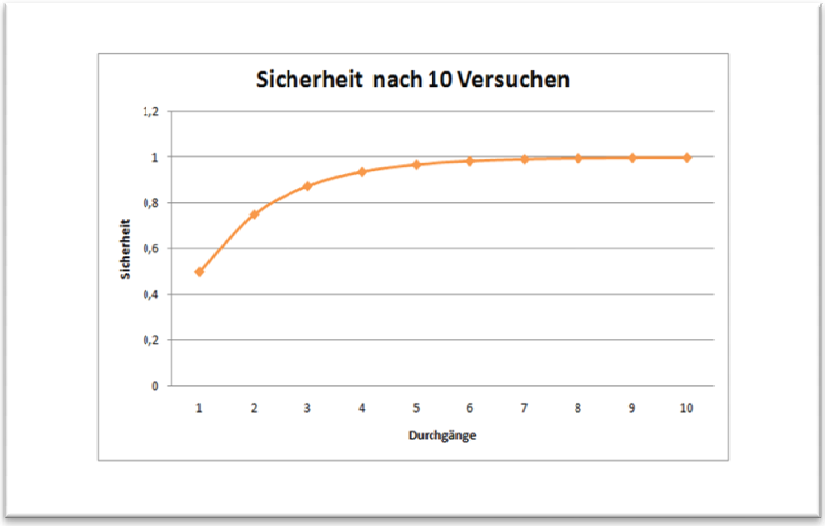
Die schnelle Überzeugung

Die Betrugswahrscheinlichkeit von Lisa beträgt 50 Prozent. Bart und Lisa spielen das Spiel n mal. Die Sicherheit die Bart hat, dass Lisa das Geheimnis kennt beträgt demnach:

$$1 - \left(\frac{1}{2}\right)^n.$$

Die Sicherheit nach mehreren Durchgängen in Prozent:

- 1. Durchgang $1 - \left(\frac{1}{2}\right)^1 = \frac{1}{2}$ 50%
- 2. Durchgang $1 - \left(\frac{1}{2}\right)^2 = 1 - \frac{1}{4}$ 75%
- 3. Durchgang $1 - \left(\frac{1}{2}\right)^3 = 1 - \frac{1}{8}$ 87,5%
- …
- 10. Durchgang $1 - \left(\frac{1}{2}\right)^{10} = 1 - \frac{1}{1024}$ 99,9%



Aus dieser Grafik lässt sich erkennen, dass Bart schon nach 10 Durchgängen mit einer 99,9 prozentigen Sicherheit sagen kann, dass er Lisa glaubt, dass sie das Geheimnis kennt.

Dabei ist nochmals zu erwähnen, dass Lisa ihr Wissen nicht preis gegeben hat. Wie der Name Zero-Knowledge sagt: Null Wissen wird übertragen. Bart ist aber trotzdem davon überzeugt, dass sie dieses Wissen hat.

Die Zero-Knowledge Eigenschaft

Eine Interaktion zwischen Lisa und Bart hat die Zero-Knowledge Eigenschaft ⇔ Wenn es einem Simulator mit Hilfe von Bart gelingt, ohne Kenntnis des Geheimnisses eine Interaktion zu rekonstruieren, die nicht von der Originalinteraktion unterschieden werden kann.

Wie konstruiert man eine Interaktion? Zum Beispiel mit einem Video.

Bart nimmt das Spiel auf Video auf. Darauf kann man erkennen, wie Lisa den Vorraum betritt und sich für Tür A oder Tür B entscheidet. Danach schließt sie die Tür hinter sich. Bart betritt als nächster den leeren Vorraum. Man sieht ihn, wie er vor den beiden geschlossenen Türen steht und entweder A oder B laut ruft. Kurz danach kommt Lisa hinter dieser Tür hervor.

Mit anderen Worten ist die Zero-Knowledge Eigenschaft dann erfüllt, wenn es möglich ist ein Video zu konstruieren das nicht vom Originalvideo zu unterscheiden ist. Wenn es möglich ist die Situation so nachzumachen, ohne genaue Informationen zu investieren, dann kann man aus diesem Video auch keine Informationen herausnehmen. Es wird definitiv kein Wissen übertragen.

Wie kann man ein Video nachmachen das nicht vom Original zu unterscheiden ist, wenn die entscheidende Information fehlt, nämlich die Tür zu öffnen? Das ist ganz einfach, nur ein wenig zeitaufwendiger.

Alles was ein Angreifer über Lisas Geheimnis kennen kann, ist auf dem Video von Bart gespeichert. Bart will Lisa davon überzeugen, dass er keine Ahnung von ihrem Geheimnis hat. Er dreht deshalb mit Meggie ein Video, das von seinem Video mit Lisa nicht zu unterscheiden ist, außer natürlich, dass Meggie mitspielt. Es beginnt gleich. Meggie betritt den Raum und entscheidet sich für Tür A. Bart geht in den Vorraum und ruft A. Meggie kommt durch Tür A wieder heraus. Eine richtige Szene ist im Kasten. Meggie geht wieder in den Vorraum entscheidet sich für Tür A. Bart kommt hinein und ruft dieses mal Tür B. Da Meggie das Geheimnis nicht kennt, kann sie jetzt nicht wie Lisa vorhin den Durchgang benutzten und bei B herauskommen. Es bleibt ihr nichts anderes übrig, als wieder durch Tür A zu kommen. Sie ist traurig, dass sie falsch geraten hat. Doch Bart sagt, das ist kein Problem, wir löschen die Szene einfach und drehen sie noch einmal. Dieses Spiel wiederholen Bart und Meggie, bis sie t gute Szenen aufgenommen haben. Dazu werden sie 2t Versuche benötigen.

Die Idee, welche hinter Zero-Knowledge steckt wurde erläutert. Im Weiteren wird darauf eingegangen, wie das Protokoll in der Praxis aussieht.

Der Fiat-Shamir Algorithmus

Dieser Algorithmus wurde im Jahr 1986 von Amos Fiat und Adi Shamir, zwei israelischen Mathematikern, vorgestellt. Damals bot er eine ganz neue Möglichkeit der Benutzerauthentifikation. Genauer genommen kommunizieren zwei Computer miteinander, wobei der eine Computer die Chipkarte des Benutzers ist. Der Algorithmus wird heute noch in der Praxis verwendet, unter anderem ist er auf Chipkarten implementiert. Er ist effizient berechenbar und der Speicherplatz der dabei benötigt wird, hält sich in annehmbaren Grenzen. Wie bei vielen kryptographischen Protokollen, kommt es im Vorfeld zu einer Schlüsselerzeugung. Diese Schlüssel werden im weiteren Verlauf benötigt.

Der Knackpunkt des Verfahrens besteht darin, dass es nahezu unmöglich ist (in relevanter Zeit) die Quadratwurzel einer sehr großen Zahl zu berechnen. „Wenn man die Primfaktorzerlegung von n nicht kennt, ist es praktisch unmöglich eine Zahl s zu finden mit $s^2 \, mod \, n \, = \, v$. Es wird empfohlen, n so groß zu wählen, dass n etwa 200 Dezimalstellen

hat. Das bedeutet, das n in normaler Schriftgröße eine Zahl ist, die etwa 1 m lang ist.“ [b1, Seite 81]

Beschreibung	**Mathematik**	**Lisa und Bart**
Lisa wählt zwei große Primzahlen p und q und bildet deren Produkt.	$p, q \in P$ $n = p * q.$	
Lisa wählt eine Zahl s, hält sie geheim und bildet deren Quadrat v. Das Paar (v, n) kann als Public Key in einem Register veröffentlicht werden.	s= geheim $v = s^2 mod\ n$	
Das Protokoll		
Lisa wählt zufällig eine positive ganze Zahl r und bildet x. Jetzt sendet sie x an Bart.	$r \in Z^+$ $x = r^2 mod\ n$ x → Bart	Lisa wählt Tür A oder Tür B und geht bei dieser Tür hinein.
Bart wählt ein zufälliges Bit b und sendet dieses an Lisa.	b → Lisa	Bart ruft Tür A (oder Tür B).
Es gibt zwei Möglichkeiten. b kann 0 oder 1 sein. Je nachdem reagiert Lisa und sendet y an Bart.	$b = 0$ Lisa sendet $y = r$ an Bart $b = 1$ Lisa sendet $y = r * s\ mod\ n$	Lisa kommt bei der gerufenen Tür heraus.
Bart überprüft nun die Antworten von Lisa bei $b = 0$ und $b = 1$.	$b = 0$ $y^2 mod\ n = x$ $b = 1$ $y^2 mod\ n$ $= x * v\ mod\ n$	Bart sieht, dass Lisa die richtige Tür erwischt hat.

Die Vorteile des Verfahrens

Beide Computer, oder Bart und Lisa, brauchen nur sehr wenige Rechnungen $mod\ n$ auszuführen. Lisa berechnet die Zufallszahl r zum Quadrat. In der Hälfte der Fälle (wenn $b = 1$ ist) muss sie auch noch r mal s berechnen. Bart muss y zum Quadrat nehmen und in der Hälfte aller Fälle (wenn $b = 1$) $x\ mit\ v$ multiplizieren. Der Authentifikationsrechner (Bart) erhält seine Ergebnisse nur mittels öffentlich zugänglichen Informationen. Während der Benutzer (Lisa) das Geheimnis s für die Berechnungen benötigt.

Verwendet wird Zero-Knowledge bei der Benutzeridentifizierung. Beispiele hierfür wären Zugangsbewilligungen zu PCs, Bankomaten oder aber auch bei Geschäften im Internet. Hinzuzufügen ist, dass es sich hierbei ausschließlich um den Nachweis der eigenen Identität handelt und nicht um die Verschlüsselung von einzelnen Daten. Dafür ist das Verfahren unbrauchbar. Dass sich der Algorithmus trotzdem einer sehr hohen Beliebtheit erfreut, liegt an den Public-Key-Eigenschaften (siehe Begriffsbestimmung und Grundlagen zu RSA). Gegenüber RSA punktet er durch einen weit geringeren Rechenaufwand.

Pay Tv

1991 kamen Zero-Knowledge-Verfahren beim kostenpflichtigen Fernsehen zum Einsatz, heute bekannt unter dem Namen Videocrypt. Im Geschäft erhält man den Decoder, der ohne die richtige Decoderkarte allerdings keinen Nutzen hat. Nach dem Einsetzen dieser Karte, kann man den gewünschten Sender empfangen. Damit nicht von Privatpersonen Karten hergestellt werden können und der Decoder echte von unechten Karten unterscheiden kann, kommt der Fiat-Shamir Algorithmus zum Einsatz.

Die Einfachheit des Algorithmus und die Tatsache, dass er mit nur geringen Anforderungen auskommt führte dazu, dass man ihn in Verbindung mit Chipkarten einzusetzen begann.

Chipkarten

Chipkarten können sich natürlich um vieles größere Codes merken, als das menschliche Gehirn. Die handlichen Plastikkarten in bequemer Brieftaschengröße mit eingebautem Chip sind fähig, Daten zu speichern und kleinere Rechnungen durchzuführen. Sie sind wie kleine Taschenrechner und benötigen dazu keine Batterie. Während ihrer Verwendung wird der Strom von außen eingespeist.

Die erste Funktion dieser Goldkontakte besteht darin, die Karte mit Strom zu versorgen. Die Zweite gewährleistet den Datentransfer von der Karte zum Rechner und wieder zurück. [a5]

Heutzutage sind viele dieser Karten in Umlauf. Ob e-card, Mastercard, oder Simkarte fürs Handy, die Frage lautet: „Warum konnten sie einen so großen Siegeszug in unserem Alltag antreten?“

Mit Hilfe dieser Chipkarten gelang es zum ersten Mal in der Geschichte der Kryptologie eine Möglichkeit zu realisieren, welche für die Bevölkerung und nicht nur für ausgewählte Experten Anwendung fand. In der cleveren Karte verbinden sich zwei Eigenschaften die bisher nur separat auftraten.

1. Chipkarten sind ideal für Kryptologie: Mit ihrer Hilfe können Algorithmen ausgeführt und geheime Schlüssel auf sichere Weise gespeichert werden.
2. Chipkarten sind ideal für Menschen: Sie sind extrem einfach zu benutzen, und handlich im Format. Die einzige Aufgabe für ihren Benutzer besteht darin, sich einen Code zu merken, um sich gegenüber seiner Karte zu authentifizieren.

Die folgenden Ausführungen stammen aus [b1]:

In Österreich bietet mittlerweile die sogenannte Bürgerkarte als Zusatzoption der e-card viele praktische Funktionen. Unter anderem können damit Amtswege rund um die Uhr erledigt werden. Die Basis der Bürgerkartenfunktion ist der Signaturschlüssel, mit dem elektronisch unterschrieben und verschlüsselt werden kann. Die Bürgerkarte vereint die Funktionen „sicherer, persönlicher, elektronischer Ausweis“ und „sichere, persönliche, elektronische Unterschrift“, sie kann im Privat- oder Wirtschaftsbereich verwendet werden.

Die vielen Vorteile der Bürgerkarte finden sich auf der Homepage der e-card [i6]:

Mit der Bürgerkarte erledigt man die Kommunikation mit Behörden und Unternehmen sicher, rasch und bequem. Das bedeutet mehr Komfort, mehr Sicherheit und mehr Tempo:

- Amtswege bequem von zuhause aus erledigen, auch abends und am Wochenende.
- Antragsformulare aus dem Internet herunterladen.
- Einfache Anmeldung auf verschiedenen Websites mit einer Karte, der Bürgerkarte.
- Schluss mit vielen verschiedenen Passwörtern oder Ausweiskarten.
- Dokumente auf höchstem technischem Niveau verschlüsseln und sicher schützen.
- Das elektronische Amt kennt keine Wartezeiten.
- Nutzung elektronischer Zustelldienste für persönliche Sendungen.
- Einfachere Bearbeitung von elektronisch übermittelten Anträgen auch bei der Behörde.

Die Bürgerkarte ist gratis im Internet zu bestellen. Um sie zu verwenden, benötigt man ein Lesegerät, für welches mit Kosten in der Höhe von 20€ zur rechnen ist.

Der Schwachpunkt von Zero-Knowledge

Der Schwachpunkt beim Zero-Knowledge Verfahren wird „Man in the middle attack“ genannt. Der Betrüger befindet sich zwischen den beiden Instanzen. [a3]

In dieser Konstellation kann sich Homer gegenüber Lisa als Bart ausgeben. Wenden wir diesen Sachverhalt auf das Fiat-Shamir Verfahren an, ergibt sich folgendes:

Lisa sendet ihr x an Homer (Sie meint es sei Bart.). Homer leitet x direkt an Bart weiter. Bart wählt ein zufälliges b und sendet dieses an Homer (Er meint es sei Lisa.). Homer sendet das zufällige Bit b weiter an Lisa. Die Antwort von Lisa wird jetzt an Bart weitergeleitet (Vorher passiert sie noch Homer.).

Bart ist jetzt fälschlicherweise davon überzeugt, dass Homer Lisa ist.

Als Ausweg werden in der Praxis exakte Uhren verwendet. Jeder Kommunikationsschritt zwischen Bart und Lisa muss zu einem festgelegten Zeitpunkt stattfinden. Ein Abfangen und Weiterleiten der Nachricht durch Homer stellt eine Verzögerung und somit eine Zeitplanverschiebung dar, welche von beiden Interaktionspartnern registriert wird.

Weiters stellt sich bei Publik-Key-Kryptosysthemen die Frage nach der Echtheit des öffentlichen Schlüssels. Wie kann man sicher sein, dass dieser Schlüssel wirklich der Person gehört, mit der ich kommunizieren möchte? Zwei mögliche Lösungen abschließend beschrieben.

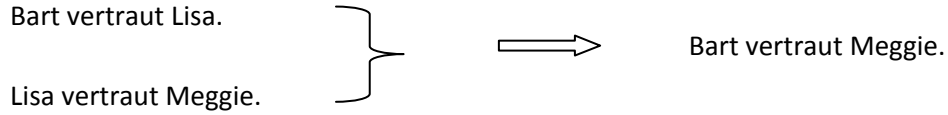
Möglichkeit 1: „Web of Trust“

Die Echtheit von digitalen Schlüsseln wird durch mehrfache Bestätigung gewährleistet. Beim Web of Trust besteht ein Zertifikat aus der digitalen Signatur, die eine andere Person, die

auch am Web of Trust teilnimmt, auf einen Schlüssel abgibt. Meistens geschieht das, nachdem diese Person sich der Identität des Schlüsselinhabers versichert hat (typischerweise bei einer persönlichen Begegnung). Diese Zertifikate werden mit dem öffentlichen Schlüssel verbunden und mit einer Wertung versehen. Einsichtig für alle Beteiligten kann das Ergebnis auf einem Server betrachtet werden.

Die Details

Zwischen den einzelnen Kommunikationspartnern herrscht, vereinfacht gesehen eine transitive Vertrauensbeziehung.



In Worten: Ich vertraue jedem, dem jemand vertraut, dem ich vertraue.

Und umgekehrt: Jeder, der jemandem vertraut, der mir vertraut, vertraut auch mir.

„In der Sprache der Objektorientierung: Jedes Objekt O, das eine Vertrauensassoziation zu anderen Objekten hat, vererbt diese Assoziation an alle Objekte, die eine ebensolche Assoziation zum Objekt O haben.“ (siehe [i9])

Einerseits ermöglicht diese Art der Schlüsselzertifizierung den Beteiligten eine persönliche Auswahl darüber, wer als vertrauenswürdig eingestuft werden kann und wer nicht. Andererseits wird sehr viel Backgroundwissen benötigt, um sich zurecht zu finden. Weiters wird die Löschung einer Signatur nicht immer sofort aktualisiert wodurch weitere Probleme entstehen können. Das populärste Beispiel für die Umsetzung des Web of Trust ist das Programm PGP, Pretty Good Privacy. Sein Erschaffer ist Phil Zimmermann. [i7]

Möglichkeit 2: Zertifizierungsstellen

Die Aufgabe einer Zertifizierungsstelle ist es digitale Signaturen zu überprüfen und für korrekt zu erklären. Ein solches Zertifikat ist wie ein Personalausweis. Er weist einen bestimmten öffentlichen Schlüssel einer Person oder Organisation zu. Mit diesem Zertifikat soll erreicht werden, dass sich hinter einem öffentlichen Schlüssel auch tatsächlich jene Person verbirgt, welche es vorgibt zu sein.

Das Prinzip der Funktionsweise:

Jeder Benutzer kennt den Public Key der Zertifizierungsstelle, jedoch nicht den passenden Private Key. Diesen besitzt nur die Zertifizierungsstelle selbst. Sie produziert nun eine Nachricht deren Inhalt der öffentliche Schlüssel einer bestimmten Person ist. Diese verschlüsselt sie mit ihrem geheimen Schlüssel. Die Nachricht kann von allen gelesen werden. Der öffentliche Schlüssel der Zertifizierungsstelle ist allgemein bekannt. Alle wissen, von wem diese Nachricht stammt. So wurde die Nachricht mit dem privaten Schlüssel der Stelle signiert. Sofern man der Zertifizierungsstelle vertrauen kann, kann man auch sicher sein, dass ein bestimmter öffentlicher Schlüssel einer gewissen Person gehört. [i8]

Das Knapsack-Verfahren

Der Begriff Knapsack kommt aus dem Englischen und hat sein Pendant im Deutschen Wort Knappsack, das heute nicht mehr verwendet wird. Knappsack bedeutete Rucksack oder Ranzen. Wir fragen uns, welche Rolle Knappsäcke in der Chiffrierung spielen. Die meisten asymmetrischen Verfahren beruhen auf dem Faktorisierungsproblem großer Zahlen (RSA), oder der Berechnung diskreter Logarithmen (Schlüsseltausch nach Pohlig und Hellman). Beim Knapsackverfahren geht es um folgendes, das wir zunächst heuristisch erklären:

In einen Rucksack werden Gegenstände aus einem Fundus von Gegenständen vom Gewicht $a_1, a_2, \dots, a_n$ eingepackt; es wird aber nicht bekannt gegeben, welche Gegenstände sich im Rucksack befinden. Nur das Gesamtgewicht s sei bekannt. Die Frage ist: Welche Gegenstände sind eingepackt worden?

Die mathematische Formulierung:

Stelle eine Zahl als Summe dar, wobei die Summanden einer vorgegebenen Menge von Summanden zu entnehmen sind.

Diese Aufgabe ist nicht immer lösbar und wenn, dann nicht immer eindeutig. Das Wichtige ist, dass die Lösung im Allgemeinen für große n nicht in vernünftiger Zeit durch zu führen ist.

Gegeben sind $a = (a_1, a_2, \dots, a_n) \in N^n$ und $s \in N_0$.

Gesucht ist ein $x = (x_1, x_2, \dots x_n) \in \{0,1\}^n$, sodass

$$s = a * x = \sum_{i=1}^{n} a_i * x_i.$$

Dieses Problem nennt sich $(0,1)$-Knapsack-Problem und wird ab jetzt mit $K(a,s)$ bezeichnet. Jedes $x \in \{0,1\}^n$ mit $a * x = s$ wird als Lösung von $K(a,s)$ bezeichnet. $a = (a_1, a_2, \dots, a_n)$ ist ein Vektor. In Verbindung mit einem Problem $K(a,s)$ wird er als ein Knapsack bezeichnet.

Die Lösung jedes Knapsack-Problems $K(a,s)$, erhält man, in dem man alle 2^n möglichen Auswahlen der Menge $\{a_1, a_2, \dots, a_n\}$ betrachtet. Nehmen wir an, n sein größer als 100. Wir sehen ein, dass dies rechnerisch unmöglich ist. In besonderen Fällen hingegen erfordert die Lösung weit weniger Aufwand.

Extra simple Knapsacks

Ein Knapsack wird als extra simpel bezeichnet, wenn in der aufsteigenden geordneten Folge $(a_1, a_2, \dots, a_n)$ jeder Summand größer ist, als die Summe der vorherigen Summanden, d.h.

$a_j > \sum_{i=1}^{j-1} a_i \ f\ddot{u}r\ j = 2,3, \dots, n$. So eine Folge werden wir im Folgenden als „superincreasing“ bezeichnen. Wie wir unten zeigen, ist die Lösung des Problems einfach.

Beispiel:

$a = (a_1, a_2, \dots, a_n)$ mit $a_i = 2^{i-1}\ f\ddot{u}r\ i = 1, 2, \dots, n.$

Für diesen Knapsack gilt auf Grund von $\sum_{i=1}^{j-1} 2^{i-1} = 2^{j-1} - 1$, dass $a_j = \sum_{i=1}^{j-1} a_i + 1.$

Satz: Ist $a = (a_1, a_2, \dots, a_n)$ extra simpel und hat das Knapsack-Problem $K(a,s)$ eine Lösung $x = (x, x_2, \dots, x_n)$, so ist diese Lösung eindeutig, und es gilt für $i = 1,2, \dots, n$:

$$x_i = 1 \Leftrightarrow s - \sum_{j=i+1}^{n} a_j * x_j \geq a_i.$$

Unter Einbeziehung dieses Satzes kann man für ein extra simples a das $K(a,s)$ wie folgt lösen:

Für $i = n$ überprüft man, ob der schwerste Gegenstand eingepackt worden ist, d.h.

$$x_n = 1 \qquad \text{falls } a_n \leq s_n$$
$$x_n = 0 \qquad \text{falls } a_n > s_n,$$

also $x_n = 1 \Leftrightarrow s \geq a_n$. Für $i = n - 1$ erhält man aus $x_{n-1} = 1 \Leftrightarrow s - a_n * x_n \geq a_{n-1}$ den Wert x_{n-1},

für $i = n - 2$ erhält man aus $x_{n-2} = 1 \Leftrightarrow s - a_n * x_n - a_{n-1} * x_{n-1} \geq a_{n-2}$ den Wert x_{n-2}, … bis man bei x_1 angekommen ist.

Beispiel:

Sei $a = (1, 3, 5, 12, 23)$ und $s = 16$. a ist also extra simpel.

$n = 5$: $16 < a_5 = 23 \rightarrow x_5 = 0$

$n = 4$: $16 - 0 * 23 = 16 > a_4 = 12 \rightarrow x_4 = 1$

$n = 3$: $16 - 1 * 12 = 4 < a_3 = 5 \rightarrow x_3 = 0$

$n = 2$: $4 - 0 * 5 = 4 > a_2 = 3 \rightarrow x_2 = 1$

$n = 1$: $4 - 1 * 3 = 1 \geq 1 \rightarrow x_1 = 1$

Als Lösung erhalten wir $x = (1, 1, 0, 1, 0)$

Beim chiffrieren geht man so vor, dass man den extra simplen Knapsack in einen normalen Knapsack „versteckt“.

Das Verfahren

Die Schlüsselerzeugung

- Wähle eine n Glieder lange superincreasing- Zahlenfolge $a^{\sim} = (a_1^{\sim}, a_2^{\sim}, \dots, a_n^{\sim})$. n sollte dabei mindestens gleich 200 sein, sodass $a^{\sim}$ in der Größenordnung 10^{100} liegt.
- Wähle ein geheimes k mit $k > \sum_{i=1}^{n} a_i^{\sim}$. Wähle weiters ein zu k teilerfremdes w.
- Multipliziere alle Elemente von $a^{\sim}$ mit $w \; mod \; k$:
- $a_i = a_i^{\sim} * w \; mod \; k \; mit \; i = 1, \dots, n$
- Die Folge $(a_i)_{i=1,\dots,n}$ bildet den öffentlichen Schlüssel.
- Die Folge $a^{\sim}$ mit w^{-1}, k bildet den privaten Schlüssel.

Chiffrierung:

- Die Nachrichten werden zuerst quellencodiert und in Blöcke der Länge n eingeteilt. $x \in \{0,1\}^n$ ist ein Nachrichtenwort.
- x soll an den Besitzer von $a = (a_1, a_2, \dots, a_n)$ gesandt werden. x wird verschlüsselt durch $s(x) = a * x = \sum_{i=1}^{n} a_i * x_i$

Dechiffrierung:

- Mit Hilfe des euklidischen Algorithmus berechnet man aus $w * v = 1 \, mod \, k$ das unbekannte $v = w^{-1}$. (Da w und k teilerfremd sind, ist das möglich.)
- Dann bildet man $s^{\sim} = s(x) * w^{-1} \, mod \, k$ und löst das Knapsack-Problem $K(a^{\sim}, k^{\sim})$.
- Dieses ist extrasimpel und kann in der vorher beschrieben Weise gelöst werden.

Beim Knapsack-Verfahren wird die Zahlentheorie bis auf die Ermittlung von $w^{\sim}$ und die Multiplikation $modulo \, k$ nicht benötigt. Dieses Verfahren eignet sich deshalb ausgezeichnet zur Illustrierung eines Public-Key-Kryptosystems im Schulunterricht.

In der Praxis wird es heute jedoch fast nicht mehr verwendet. Schon 1982 gelang es Len Adleman auf einer Konferenz für Kryptoanalytiker zu zeigen, dass das Verfahren außer in speziellen Fällen zu brechen ist. Obgleich immer wieder Verbesserungen am Verfahren vorgenommen wurden, konnten die meisten Varianten geknackt werden. Einige ungebrochene Varianten gibt es noch - es stellt sich die Frage, wie lange noch.

Ein Problem beim Knapsackverfahren ist überdies, dass man damit Nachrichten nicht signieren kann, denn die Forderung (1) im Diffie-Hellman Public-Key-Kryptosystem ist nicht erfüllt: Für das Knapsackverfahren gilt nicht: $(S_X(T_X(m)) = T_X(S_X(m))$.

Der Schlüsseltausch nach Pohlig und Hellmann

Asymmetrische Verschlüsselungsverfahren sind sehr rechenintensiv. Das bedeutet, dass einige Zeit in Anspruch genommen werden muss, um einen Text zu chiffrieren. In der Praxis geht man deshalb häufig so vor: Man verschlüsselt einen Text mit einem guten symmetrischen Verfahren. Den Schlüssel, den man dazu verwendet hat, übermittelt man

aber mittels asymmetrischer Verschlüsselung, wodurch sich das Schlüsseltransportproblem erübrigt.

Das Verfahren

Der Vorteil des Schlüsseltausches nach Pohlig und Hellman [b6] liegt darin, dass er Teil eines **Hybridsystems** ist. Dabei werden, wie erwähnt, die Vorteile von symmetrischen und asymmetrischen Verschlüsselungsverfahren gleichermaßen genützt.

q sei eine Primzahlpotenz $p^t \; mit \; p^t > 2^{100}$.

(Wenn $t = 1$ ist, dann soll p keine Fermatsche Zahl sein. Wir wissen aber aus den mathematischen Grundlagen, dass es bis heute nur fünf bekannte Fermatsche Primzahlen gibt, weshalb diese Forderung eher von theoretischem Interesse ist.)

γ sei ein primitives Element des Körpers $GF(q)$, das ist ein erzeugendes Element der zyklischen Gruppe $GF(q)^*$ (siehe mathematische Grundlagen).

Für $x \in GF(q)$ und $v \in GF(q)$ gelte: $v = \gamma^x$.

Aus der Kenntnis von q, γ und v ist es ein völlig aussichtsloses Unterfangen x zu berechnen. Es ist daher unmöglich den diskreten Logarithmus zur Basis γ zu bestimmen. Die Abbildung $f(x) = \gamma^x$ ist eine Einwegfunktion (Zur Definition der Einwegfunktion siehe Begriffsbestimmung.).

Jeder Teilnehmer X wählt sich mit Zuhilfenahme eines Zufallsgenerators ein

- $t_X \in GF(q)$ mit $2 \leq t_X \leq q - 2$
- er bildet in $GF(q)$ ein $s_X = \gamma^{t_X}$
- er gibt s_X öffentlich bekannt
- t_X bleibt geheim

Der gemeinsame Schlüssel s_{XV} für die Teilnehmer X und V scheint nirgendwo auf. Er ist durch $s_{XV} = \gamma^{t_X t_V} = {s_X}^{t_V} = {s_V}^{t_X}$ festgelegt.

Nachrichtenübermittlung

V will an X eine Nachricht senden. V sucht aus dem öffentlichen Verzeichnis $s_X$und bildet mit Hilfe seines geheimen Schlüssels t_V durch ${s_X}^{t_V}$den gemeinsamen Schlüssel s_{XV} .

Wir haben bereits hervorgehoben, dass es nahezu unmöglich ist, aus dem Verzeichnis der öffentlichen Schlüssel s_X, die zur Bildung der gemeinsamen Schlüssel notwendigen, geheimen Schlüsseln t_X zu berechnen (Das wäre der Logarithmus von s_X zur Basis γ in $GF(q)$.) .

Besonders bei kurzen Schlüsseln ist dieses Verfahren effektiv, schnell und sehr sicher.

Quantenkryptographie

Um über Quantenkryptographie sprechen zu können, müssen wir im Vorfeld einige Dinge klären. Wir erinnern uns, dass der Knackpunkt des RSA-Verfahrens darauf beruhte, große Zahlen zu faktorisieren. Peter Shore gelang es 1994, ein Programm zu schreiben [b12, Seite 398], mit dem der Algorithmus geknackt werden könnte. Allerdings benötigt man zur Ausführung des Programmes einen Quantencomputer. Nach unserem heutigen Wissenstand existiert ein solcher erst im Versuchsstadium. Quantenkryptographie kommt jedoch ohne Quantencomputer aus. Die Grundlagen finden sich in der Quantentheorie, welche im Folgenden kurz erläutert werden. Es sei noch erwähnt, dass durch die Möglichkeit dieser Verschlüsselung die perfekte, unangreifbare Geheimhaltung erzielt werden könnte.

Die Idee, die Grundlagen der Quantentheorie für Sicherheitszwecke zu nutzten, fand sich erstmals in den 60-er Jahren des letzten Jahrhunderts. Stephen Wiesner wollte Banknoten herstellen, die kein Fälscher der Welt reproduzieren hätte können. Er nannte seine Währung Quantengeld [b12]. Leider war er seiner Zeit zu weit voraus, weshalb seine Überlegungen nur ein Gedankenexperiment blieben. Wiesner schaffte es jedoch, das Interesse anderer Quantenkryptographen zu wecken. In den folgenden Jahren wurden die Grundlagen der Quantenkryptographie entwickelt. Um ansatzweise zu verstehen wie die Nachrichten-übertragung funktioniert, benötigen wir einige physikalische Grundlagen.

Physikalische Grundlagen

Zum Verständnis der Quantenkryptographie seien im Folgenden die später verwendeten Begriffe nur anschaulich erklärt. Dabei werden die Erklärungen so beschrieben, dass man sie einem Schüler oder einem Laien in wenigen Worten zugänglich machen könnten.

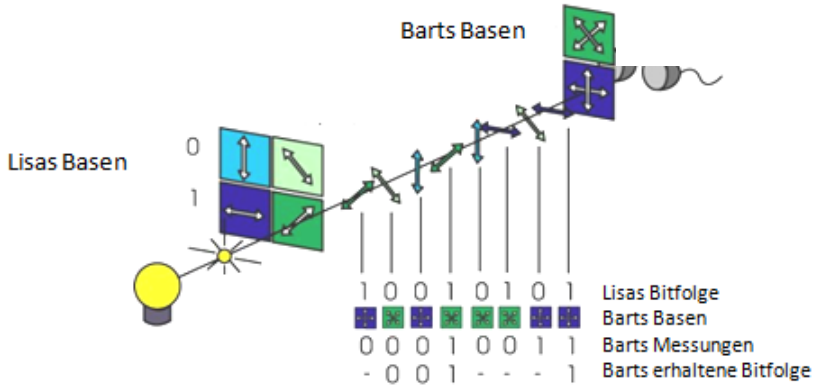
„Ein **Quant** ist ein Objekt, das durch einen Zustandswechsel in einem System mit diskreten Werten einer physikalischen Größe erzeugt wird [i1].“ Ein Photon kann beispielsweise als Quant eines elektromagnetischen Feldes auftreten. **Photonen** sind die Bausteine der elektromagnetischen Strahlung. Wir können sie uns als Lichtteilchen vorstellen, dürfen dabei jedoch nicht vergessen, dass sie auch Welleneigenschaften besitzen. Unter **polarisiertem Licht** verstehen wir ein Bündel von Lichtstrahlen mit gleich ausgerichteten Wellen. Durchdringt das Licht einen **Filter** - im Zusammenhang mit der Übertragung von Nachrichten werden wir dafür den Ausdruck **Basis** verwenden - dann werden die Polarisationsrichtungen des Lichts dabei herausgefiltert. Man sagt, das Licht wird **polarisiert**. Eine der Grundlagen der Quantenkryptographie ist die **heisenbergsche Unschärferelation.** Sie besagt, dass zwei Messgrößen eines Teilchens nicht immer gleichzeitig genau zu bestimmen sind. Man kann entweder das eine oder das andere exakt messen. Stellen wir uns einen Stapel Karten vor. Wir ziehen eine Karte, stellvertretend für ein Teilchen. Wir können uns entscheiden, ob wir die Farbe (Herz, Kreuz,…) oder den Wert (Ass, König, Sieben,…) bestimmen möchten. Beides zu gleich funktioniert in der Quantenmechanik nicht. Weiters verursacht jede Messung in einem System eine Veränderung in diesem. Für die Quantenkryptographie ergibt sich daraus der Vorteil, dass die Kommunikation zwischen Lisa und Bart nicht belauscht werden kann, ohne dass die beiden entdecken, dass ein Lauschangriff erfolgt ist.

Kommunikation

Bart und Lisa wollen sich eine Nachricht senden. Diese Nachricht besteht aus Nullen und Einsen. Sie werden von Photonen mit verschiedener Polarisation dargestellt. Dafür benötigen sie verschiedene Basen. Mit welchen Photonen sowie Basen die Nullen und Einsen übertragen werden, ist aus der Tabelle ersichtlich.

rektilineare Basis ⊕	↕	1	↔	0
horizontale Basis ⊗	⤢	1	⤡	0

Die Übertragung einer Nachricht wird nun in drei Schritte gegliedert. Die Abbildung zeigt plakativ, wie die Photonen von Lisa durch eine Basis gehen, an Bart übertragen und von ihm gemessen werden. [a4]



1. Schritt:

   Lisa sendet Bart eine Zufallsfolge aus Nullen und Einsen. Dazu verwendet sie ebenfalls zufällig, entweder rektilineare oder horizontale Basen.

2. Schritt:

   Bart misst die Polarisation der gesendeten Photonen. Da er nicht weiß, welche Basen Lisa verwendet hat, misst er zufällig mit seinen beiden Basen. Dabei gelingt es ihm manchmal die gleiche Basis wie Lisa zu verwenden, und manchmal nicht. Erwischt er die falsche, wird es passieren, dass er ein gesendetes Photon missdeutet. In der nachfolgenden Tabelle sind die möglichen Messausgänge beschrieben, die eintreten können.

3. Schritt:

   Lisa hat nun ihre Folge übermittelt. Bart hat einige Photonen richtig, andere falsch gemessen. Lisa berichtet nun über einen nicht notwendigerweise sicheren Kanal, welche Basen sie für welche Photonen verwendet hat. Sie verrät ihm jedoch nicht, wie die einzelnen Photonen polarisiert waren. Bart teilt anschließend Lisa mit, in welchen Fällen er die richtige Basis verwendet hat. Nun werden alle Messungen gestrichen, bei denen Bart die falsche Basis verwendet hat.

   Die Möglichkeiten der Photonenübermittlung die Lisa und Bart haben werden in der nächsten Tabelle übersichtlich dargestellt. [b12]

Lisas Basis	Lisas Bit	Lisa sendet	Barts Basis	Richtige Basis?	Bart misst	Barts Bit	Ist Barts Bit richtig?
⊕	1	↕	⊕	Ja	↕	1	Ja
			⊗	Nein	⤢	1	Ja
					⤡	0	Nein
	0	↔	⊕	Ja	↔	0	Ja
			⊗	Nein	⤢	1	Nein
					⤡	0	Ja
⊗	1	⤢	⊕	Nein	↕	1	Ja
					↔	0	Nein
			⊗	Ja	⤢	1	Ja
	0	⤡	⊕	Nein	↕	1	Nein
					↔	0	Ja
			⊗	Ja	⤡	1	Ja

Zu Beginn wurde von einer Nachrichtenübertragung zwischen Lisa und Bart gesprochen. In Wahrheit wird jedoch zumeist nur ein geheimer Schlüssel generiert, mit welchem die eigentliche Nachricht später chiffriert wird.

Die Quantenkryptographie ermöglicht es zwei Kommunikationspartnern einen gemeinsamen Schlüssel zu vereinbaren, den ein Dritter nicht abfangen kann. Versucht jemand die Kommunikation zwischen Lisa und Bart zu belauschen, d.h., die Photonen zu messen, können beide das feststellen. Denn jedesmal wenn die Photonen nicht direkt von Bart gemessen werden, verändern sie sich bei der Verwendung einer falschen Basis und Bart erhält zu viele falsche Messungen. Um sicher zu gehen, dass Lisa und Bart nicht belauscht werden, führen sie nach der Übertragung einen Test durch, mit dem sie feststellen können, ob ihre Kommunikation sicher war. Wir nehmen an die beiden haben einen Schlüssel mit der Länge von 1075 Binärzahlen übertragen. Lisa wählt 75 dieser Zahlen aus und prüft, ob Bart zu dem gleichen Ergebnis gekommen ist. Die Wahrscheinlichkeit, dass ein Lauscher in der Leitung

war und die Messung dieser 75 Zahlen nicht beeinflusst hat, ist kleiner als eins zu einer Milliarde [b12]. Anschließend an die Kontrolle müssen alle 75 Zahlen vom Schlüssel gestrichen werden. Wir erhalten einen 1000 Zeichen langen Schlüssel.

1988 zeigte Charles Bennet zusammen mit einem Kollegen, dass dies auch in der Realität funktioniert [b12]. Sie bauten einen Apparat, der die oben beschrieben Schritte durchführen konnte. Allerdings betrug die Distanz über die kommuniziert wurde, nur 30 Zentimeter. Bis heute hat sich einiges geändert. Bei den Schweizer Parlamentswahlen wurden im Oktober 2007 die Wahlergebnisse mittels Quantenkryptographie über rund 100 km übertragen [i2]. Weiters gelang im April 2004 in Wien eine Übertragung, bei der ein verschlüsselter Scheck vom Rathaus in eine Bank übermittelt wurde. Der Rekord liegt bei einer Entfernung von 184,6 km und wurde im Jahr 2006 aufgestellt. Bis 2008 konnte er noch nicht gebrochen werden. [i3]

Die beschriebene Übertragung erfolgte bis jetzt in Glasfaserleitungen. Es ist einzusehen, dass es schwierig ist, ein Netz solcher Leitungen für quantenkryptographische Zwecke zur Verfügung zu stellen. Deshalb suchen Forscher nach anderen Medien, in denen sie die Photonen verschicken können. Eine Möglichkeit, die derzeit erforscht wird, ist der Luftweg. Verglichen mit einem Glasfaserkabel ist die Luft jedoch ein unsauberes Medium. Die einzelnen Photonen können leicht von unterschiedlichen Störfaktoren abgelenkt werden. Das Ziel ist in einer Höhe von 300 km einen Empfänger von wenigen Zentimetern Durchmesser um die Erde kreisen zu lassen, der die einzelnen Photonen messen kann. Dabei muss das Photon genau in die Richtung des Senders fliegen und die richtige Frequenz haben, um registriert zu werden. Störphotonen, die nicht zur Kommunikation gehören, müssen ausgeschlossen werden. Luftturbulenzen und Dichteunterschiede in der Atmosphäre erschweren die Übertragung weiters. Trotzdem gelang es einem Team von Wissenschaftlern eine quantenkryptographische Übertragung auf dem Luftweg über 500 m durchzuführen. Verglichen mit den 184 km, die mit Glasfaserkabeln bereits möglich sind, klingt das nicht überzeugend. Gelingt jedoch eine Luftübertragung über 2 km, dann sind vermutlich auch 300 km nicht mehr all zu schwierig zu überwinden. Das liegt daran, dass die Dichte der Luft sowie die Luftturbulenzen nach oben hin stark abnehmen.

Resümee

Wenn die Quantenkryptograhie in nächster Zukunft kommerziell genützt werden kann, besteht für eventuelle Lauscher keine Möglichkeit mehr unentdeckt zu bleiben. Das

Jahrhundert lange Hin und Her zwischen Ver- und Entschlüsslern hätte in der bisherigen Form ein Ende. Die Verschlüssler hätten gewonnen. Diese Behauptung wurde in der Geschichte der Kryptographie schon des Öfteren aufgestellt, jedoch immer widerlegt. Die Quantenkryptographie unterscheidet sich von allem bisher Gekanntem, es könnte dieses Mal gelingen. Andere Fragen rücken in den Vordergrund: Wie kann der Gesetzgeber die Kryptographie so regulieren, dass Wirtschaft, Militär und Öffentlichkeit profitieren, Kriminellen jedoch das Handwerk gelegt werden kann? Ein wichtiger Punkt, über den wahrscheinlich noch viel diskutiert werden wird.

Gedanken zu den Verschlüsselungsverfahren im Schulunterricht

Zu Beginn werden einige grundlegende Überlegungen zum Thema Mathematik und Kryptographie in der Schule angestellt. Auf den anschließenden Seiten sollen die im ersten Teil der Arbeit beschriebenen Verfahren in Bezug auf einen möglichen Unterricht betrachtet werden.

Mathematik- eine ungeliebte Wissenschaft?

Die Erinnerung an den Mathematikunterricht ist bei vielen Menschen von negativen Assoziationen begleitet. Die Inhalte, welche im Unterricht vermittelt werden, erscheinen abgesehen vom Unterstufenstoff oft losgelöst von jeglichem späteren Nutzen. Man spricht von „sinnlosen Gedankenexperimenten“. Der Satz: „Wofür bitte braucht man das?“, ist sehr oft von Schülern zu hören. Verständnislosigkeit gefolgt von einer leisen, staunenden Bewunderung wird einem entgegengebracht, „outet“ man sich als Mathematiker. Tatsache ist jedoch, dass unser gesamter Alltag von Mathematik durchdrungen ist. Auf den ersten oder zweiten Blick lässt sich das nicht immer erkennen. Unter anderem ist das auf die Komplexität zurückzuführen, die hinter vielen Prozessen steckt.

Mathematik ist eine Wissenschaft, die sich Schritt für Schritt weiterentwickelt hat. Der Nutzen und die Anwendung der einzelnen Gebiete war nicht immer klar zu erkennen. Exemplarisch erwähnt sei hier die Beschäftigung mit Primzahlen. Als Euklid und andere berühmte Wissenschaftler sie erforschten und Gesetzte darüber formulierten, konnten niemand sich auch nur annähernd vorstellen, welch wichtige Rolle diese besonderen Zahlen Jahre später in der Welt der Geheimdienste und auch der Öffentlichkeit spielen sollten. Hans Magnus Enzensberger widmet sich den „prima Zahlen“, wie er sie in seinem Buch „Der Zahlenteufel“ [b8] nennt. Ein altes Teufelchen will Robert die Angst vor der Mathematik nehmen und besucht ihn deshalb jede Nacht im Traum. Der alte Teufel erzählt ihm von den

„prima Zahlen“: „Du mußt wissen, es gibt da diese hundsgewöhnlichen Zahlen, die sich teilen lassen, und dann gibt es die anderen, bei denen das nicht geht. Die sind mir lieber. Weißt du, warum? Weil sie prima sind. An denen haben sich die Mathematiker schon seit über tausend Jahren die Zähne ausgebissen. Wunderbare Zahlen sind das, die Elf oder die Dreizehn oder die Siebzehn.“ [b8, Seite 55]

Es ist die Aufgabe eines jeden Lehrers sein Fach als lebendiges Wissensgebiet seinen Schülern näher zu bringen. Dabei kann aufgezeigt werden, wie die Zahlentheorie, ein Gebiet der „reinen Mathematik“, die nur sich selbst zum Inhalt hat, Einzug in die Praxis finden konnte. Ziel soll es sein, die Neugierde und das Interesse der Schüler zu wecken und sie zum selbständigen Weiterlernen zu motivieren. Bewunderung für die Leistungen der Mathematiker sollen das verständnislose Kopfschütteln über die Mathematik, das oft anzutreffen ist, ablösen. Heutzutage gibt es viel populärwissenschaftliche Literatur, die sich manchmal fast wie ein spannender Krimi liest und jedem einen Einblick, in die Welt der Mathematik ermöglicht. (Beispielhaft erwähnt sei hier Simon Singhs Buch „Fermats letzter Satz- Die abenteuerliche Geschichte eines mathematischen Rätsels“ [b13]. Als erstes mathematisches Buch schaffte es den Einzug in die Bestsellerlisten.)

Grundlegende Überlegungen

Historische Betrachtung

„Er kam sah und chiffrierte.“ Einer der ältesten Kryptographen dürfte wohl Cäsar gewesen sein. Um mit seinen Feldherren an den verschiedenen Fronten zu kommunizieren benutzte er eine monoalphabetische Substitution. Gab er einen schriftlichen Befehl, wurde jeder Buchstabe durch einen anderen des Alphabets ersetzt. Aus VENI VIDI VIZI wurde YHQL YLGL YLCL. Für damalige Verhältnisse war diese Art der Verschlüsselung völlig ausreichend. Als weiteres historisches Beispiel wäre die Enigma, eine Chiffriermaschine, welche im 2. Weltkrieg eine entscheidende Rolle gespielt hat, zu nennen. Ihre Funktionsweise wird im Technischen Museum in Wien beschrieben.

Zusammenfassend ist zu bemerken, dass die Kryptographie früher vor allem im Militär und während Kriegszeiten Anwendung fand. Kryptographie wurde fast ausschließlich von Geheimdiensten benötigt und erforscht. Im Laufe der Jahre hat sie sich dabei von einer

„Kunst“ zu einer Wissenschaft entwickelt. Die in ihr geltenden Aussagen sind mittels mathematischer Beweise verifizierbar.

Waren es bis zur Entwicklung leistungsfähiger Computer vor allem symmetrische Verfahren, welche in der Kryptographie verwendet wurden, so gewinnen mit zunehmender Computerleistung immer mehr die asymmetrischen Verfahren an Bedeutung. Im Jahre 1976 veröffentlichten Diffie und Hellman einen Artikel mit dem Titel „New Directions in Cryptography“. Darin wurden erstmalig asymmetrische Kryptographische Verfahren für die Öffentlichkeit beschrieben. Weiters wurde der Schlüsseltausch nach Diffie und Hellman präsentiert. Es entwickelte sich ein Wettlauf um das erste praxistaugliche asymmetrische Kryptosystem. Der Sieger der daraus hervorging war RSA. Im Jahr 1985 entwickelte der Amerikaner El Gamal das nach ihm benannte Verfahren, welches als Basis das Diffie-Hellman- Verfahren hat. Der gleiche Gedanke liegt den kryptographischen Verfahren mit Hilfe von elliptischen Kurven zugrunde.

Wer benötigt Kryptographie heute?

In unserer Gesellschaft, in Zeiten von Internet, Handy und Bankomatkarten, benötigt und verwendet geradezu jeder Mensch Kryptographie. Es geht um die Sicherheit und Privatsphäre jedes Einzelnen. Moderne Verschlüsselungsmethoden können mit dem Briefgeheimnis von früher verglichen werden. Es war verboten unautorisiert einen Briefumschlag zu öffnen. Wer eine Postkarte verschickte konnte nicht verlangen, dass niemand die Zeilen darauf las. Kryptographie ist wie ein perfekter Briefumschlag. Er sichert die Geheimhaltung des Inhalts einer Nachricht. Einige Kryptographen fordern, dass jeder Mensch seine Nachrichten verschlüsseln soll. Denn wenn nur jene Menschen verschlüsseln, die etwas zu verbergen haben, würde dies große Aufmerksamkeit erregen

Es gilt, die Schüler dafür zu sensibilisieren, welche Informationen sie der Öffentlichkeit zugänglich machen sollen, und welche nicht. In Zeiten, in denen viele Jugendliche, aber auch Erwachsene ihr gesamtes Privatleben mit Fotos und Texten freiwillig im Internet dokumentieren (genannt seien hier StudiVZ oder Facebook) ist es daher von großer Bedeutung die dadurch entstehenden Gefahren auch im Unterricht zu besprechen. Immer wieder gibt es bereits Berichte in den Medien, dass Arbeitssuchende nicht aufgenommen wurden, da die Jobscouter ein nicht zum Unternehmen passenden Profil auf einer Internetplattform gefunden haben.

Kryptographie wird jedoch nicht nur von den „Guten“ verwendet. Schwierig wird es, wenn kriminelle Organisationen sich dieser allgemein bekannten Verschlüsselungsmethoden bedienen. In diesem Fall kann Kryptographie mit einer High- Tech Waffe verglichen werden. Es stellt sich die Frage: „Wer darf solche Waffen besitzen?“ Dies wird sicher noch einige Diskussion mit sich bringen, und kann durchaus auch im Schulunterricht angesprochen werden.

Simon Singh schreibt dazu in [15, Seite 371]: Ron Rivest, einer der Erfinder von RSA, ist der Meinung, dass die Beschränkung des Gebrauchs der Kryptographie allerdings ein Eigentor wäre: „Jeder amerikanische Bürger kann ohne weiteres ein Paar Handschuhe kaufen, obwohl ein Einbrecher damit ein Haus ausräumen könnte, ohne Fingerabdrücke zu hinterlassen.“

Betrachtung der Verfahren - Bezug zu einem möglichen Schulunterricht

Prinzipiell geht es darum, den Schülern die Genialität, welche hinter den einzelnen Verfahren steckt, näher zu bringen. Sie sollen das großartige Gedankengut, hinter einer mathematischen Idee erkennen und schätzen lernen. Wie viele Verfahren sind auch die heute gängigen Algorithmen den langen mühsamen Denkarbeiten vieler kluger Köpfe zu verdanken.

Im Sinne eines fächerübergreifenden Unterrichts sollen einige Aspekte der Kryptographie eingehender betrachtet werden. Wer braucht Kryptographie? Warum kann sie jemandem Schaden? Wie lange gibt es diese Wissenschaft schon? Was hat sie mit Physik zu tun? Kann man selbst einfache Protokolle schreiben, und damit Nachrichten verschlüsseln? Über diese und andere Fragen sollen die Schüler angeregt durch ihre Lehrer nachdenken und durch ihre Beschäftigung mit dem Thema zu Antworten finden. In vielen Fächern bietet sich die Möglichkeit über Kryptographie nachzudenken. Im Deutschunterricht könnte man dieses Thema erörtern, in Geschichte die Bedeutung für Militär und Wirtschaft diskutieren. Im Physikunterricht wäre es möglich die Grundlagen zur Quantenkryptographie zu erarbeiten. Der Informatikunterricht eignet sich dazu, einige Verfahren zu implementieren.

Einige Aspekte der Chiffrierverfahren lassen sich eventuell im Regelunterricht oder in einer Freistunde mit den Schülern erarbeiten. Dabei kann das Interesse für dieses Gebiet der Mathematik geweckt werden. Tiefere Einblicke in die Algorithmen und die den Algorithmen zugrundeliegende Mathematik lassen sich wahrscheinlich nur im Rahmen eines

Wahlpflichtfaches tätigen. Die didaktischen Methoden, mit denen der Inhalt vermittelt werden kann, reichen von Frontalunterricht über Gruppenarbeiten bis hin zu großen Projekten. Grundlegende Literatur kann vom Lehrenden zur Verfügung gestellt werden. Welche Themen von den Schülern unbedingt behandelt werden sollten, ist vom Lehrenden klarzustellen.

1. RSA

Als einem der wichtigsten Verschlüsselungsverfahren gebührt dem RSA-Kryptosystem die größte Aufmerksamkeit. Es eignet sich hervorragend für den Unterricht in der Schule. Mit kleinen, fassbaren Zahlen kann eine Nachricht ver- und wieder entschlüsselt werden. Dabei ist natürlich darauf einzugehen, dass in der Praxis andere Bit Dimensionen herrschen. Die Schüler sollen verstehen, dass es ganz einfach ist, ein n aus zwei Primzahlen zu bilden, jedoch fast unmöglich, dieses n wieder zu faktorisieren. Es handelt sich dabei salopp ausgedrückt um eine „Einwegfunktionen mit Hintertür“. Die Umkehrfunktion kann nur mit Hilfe einer Hintertür leicht berechnet werden, ansonsten ist es schwierig. Der öffentliche Schlüssel beim RSA- Verfahren ist die Einwegfunktion. Der private Schlüssel bildet die Hintertür, mit der man wieder zur Nachricht gelangen kann.

Für den Schulunterricht würde sich folgende Formulierung aus [b10] anbieten:

Eine Funktion $f: X \to Y$ heißt Einwegfunktion wenn $y = f(x)$ leicht mit einem effizienten Algorithmus E zu berechnen ist, und $x = f^{-1}(y)$ leicht mit einem effizienten Algorithmus D zur berechnen ist. Die Bestimmung von D aus E ist jedoch ohne eine geheim zu haltende Zusatzinformation (Hintertür) schwer. Hierbei ist im Unterricht darauf einzugehen, dass die Umkehrabbildung f^{-1} nur bildbar ist, falls f eine Bijektion ist.

Ein Vorzug des Verfahrens besteht darin, dass neben dem Verschlüsseln von Nachrichten auch eine Benutzerauthentifikation möglich ist. Ich kann meinen „Gesprächspartner“ durch eine elektronische Unterschrift davon überzeugen wirklich mit mir zu kommunizieren.

Die Sicherheit von RSA sollte diskutiert werden. Je nachdem, wieviel Zeit für solche Überlegungen aufgewendet wird, können verschiedene Aspekte betrachtet werden. Zum einen eine gute Wahl für die einzelnen Parameter, zum anderen die Rechenleistung der am Markt vorhandenen Computer. Mögliche Schwachstellen bei schlechter Wahl der Variablen lassen sich durch die Erwähnungen konkreter Angriffe aufzeigen. Im Zuge eines technischen Ausblicks kann es auch zu Überlegungen bezüglich eines Quantencomputers kommen. Peter

Shore gelang es einen Algorithmus zu schreiben, der auf einem Quantencomputer RSA brechen könnte. [b12]

Die Vor- und Nachteile von RSA seinen hier nochmals kurz zusammengefasst:

Vorteile $+$

- $n = p * q$ lässt sich nicht bei entsprechender Wahl der Stelligkeit von p und q in vernünftiger Zeit faktorisieren. Das heißt chiffrierte Nachrichten können nicht von Unbekannten gelesen werden.
- Es gibt unendlich viele Primzahlen. Wählt man p und q von immer größerer Stelligkeit, wirkt man der Rechenleistung der Computer entgegen.
- Der Algorithmus gilt als sicher und er ist vollständig publiziert.
- Alle Schwachstellen lassen sich durch gute Implementierung vermeiden.

Nachteile $-$

- RSA ist für die direkte Verschlüsselung von Nachrichten zu langsam, sodass das Verfahren oft nur für die Chiffrierung von Schlüsseln eines anderen Verfahrens verwendet wird.
- Fällt der private Schlüssel in die falschen Hände, können sämtliche gesendeten Nachrichten im Nachhinein gelesen werden.
- Die Geschwindigkeit der Prozessoren steigt rapide an. Es ist immer mehr möglich, größere Zahlen zu faktorisieren.

2. Zero–Knowledge Proofs

Die Idee, die hinter Zero-Knowledge steckt lässt sich in der Schule an Hand von mehreren Beispielen anschaulich erklären. Ob man Alibabas Höhle oder ein Spiel zwischen Bart und Lisa beschreibt, ist Geschmacksache. Wichtig ist den Schülern das Paradoxon näher zu bringen, jemanden davon zu überzeugen ein Geheimnis zu kennen, ohne auch nur einen Bruchteil davon zu verraten. Diesen Sachverhalt kann man schon mit ganz Kleinen besprechen.

Ältere Schüler sollen selbst kritisch über den Spielverlauf nachdenken und nicht bloß die Tatsachen, die ihnen ein Lehrender erzählt akzeptieren. Dabei wird ihnen auffallen, dass Lisa Bart nur mit einer Wahrscheinlichkeit von 50 Prozent überzeugen kann, dass sie den geheimen Schlüssel wirklich kennt. Allerdings gilt dies jeweils nur für einen Spielverlauf. Nach mehreren Runden, die gespielt werden, und Lisa bei der richtigen Tür herauskommt, sinkt die

Betrugswahrscheinlichkeit rapide. Bart kann sich wie die Grafik (siehe Zero-Knowledge) zeigt schon nach zehn Durchgängen mit 99,9 Prozentiger Wahrscheinlichkeit sicher sein, dass Lisa ehrlich ist.

Der konkrete Ablauf des Fiat-Shamir Algorithmus kann mit den Schülern im Groben durchgedacht und mit konkreten Zahlen verdeutlicht werden. Wichtig ist dabei, dass die Schüler begreifen, dass er für die Verschlüsselung von Nachrichten wenig geeignet ist, sondern bei der Benutzerauthentifikation zum Einsatz kommt. Dabei kann diskutiert werden, wo eine solche erforderlich ist (z.B.: Bankomaten, Telebanking, Pay- Tv,…) und was die Risiken dabei sind.

Vorteile $+$

- Der Algorithmus eignet sich zur Benutzeridentifikation, da nur wenige Rechnungen $mod\ n$ ausgeführt werden müssen.
- Die Berechnungen erfolgen viel schneller, als vergleichsweise beim RSA- Verfahren.
- Er eignet sich zur Implementierung auf Chipkarten. (z.B.: Bürgerkarte)
- Der Algorithmus gilt als sicher und er ist vollständig publiziert.

Nachteile $-$

- Bei der Nachrichtenverschlüsselung findet er keinen Einsatz.
- Es sind relativ viele einzelne Kommunikationsschritte nötig.
- „Man in the middle attack“ (siehe Zero-Knowledge)

3. Bemerkungen zur Verschlüsselung großer Zahlen

Primzahlen

Sowohl bei RSA, als auch bei Zero-Knowledge benötigen wir große Primzahlen. Wie groß ist eine solche Primzahl? Einige Überlegungen:

Die größte derzeit bekannte Primzahl ist eine Mersenne Primzahl. (Siehe die mathematischen Grundlagen.) Sie hat 12 Millionen Stellen. Das klingt viel, wie soll man sich das vorstellen?

- Wieviele Kästchen hat ein kariertes Blatt Papier?
- Wieviele Ziffern der Primzahlen passen auf eine Seite?
- Wieviele Seiten sind das, wenn du die längste Primzahl aufschreiben würdest?
- Wie hoch ist der Stapel an Papier, der dabei entsteht?

- Wie lange braucht man, würde man sich diese Zahl Blatt für Blatt ansehen wollen?
- …

Eine mögliche Lösung:

- Die Anzahl der Kästchen am Papier geschätzt beträgt: $42 * 58 = 2428$
- Dividiert man die Primzahl durch die Anzahl der Kästchen, erhalten wir Anzahl der Seiten.
- $12\,000\,000 : 2428 = 5000$
- Nimmt für die Höhe einer Seite 1 mm an, kommt man auf eine Höhe von 5000 mm. Das entspricht einer Höhe von 5 m. Das entspricht beispielsweise der Höhe eines Sprungturmes im Freibad.
- Wir nehmen an, dass wir in 1 Minute ca. 40 Seiten umblättern können. Für 5000 Seiten ergibt sich dann eine Zeit von 125 Minuten. Wir benötigen mehr als zwei Stunden, wenn wir uns die ganze Zahl nur ansehen möchten.

4. Das Knapsack-Verfahren

Verglichen mit anderen Chiffrierverfahren kann das Knapsack-Verfahren heute in vielen Belangen nicht mehr den Ansprüchen der Verschlüssler genügen. Der Algorithmus wurde zwar im Laufe seiner Geschichte immer wieder verbessert und adaptiert, jedoch konnten auch diese Veränderungen größtenteils geknackt werden. Trotz dieser Tatsache lohnt es sich, ihn in der Schule zu thematisieren. Alleine der Name fordert auf, eine Geschichte zu erzählen, ein Rätsel im Zusammenhang mit dem Einpacken eines Rucksacks zu formulieren.

Die Vorkenntnisse, welche benötigt werden um das Verfahren zu verstehen sind viel geringer, als beispielsweise bei RSA. Abgesehen von der Ermittlung von w^{-1} und einigen Rechnungen $modulo\ k$ kommt man ganz ohne Zahlentheorie aus, wie das folgende Beispiel zeigen soll:

Beispiel:

Die Schlüsselerzeugung:

Lisa erzeugt einen simplen Knapsack, indem sie $(1,4,7,?,?)$ durch die jeweils kleinst möglichen Zahlen ergänzt. Sie berechnet: $1 + 4 + 7 + 1 = 13$ und $1 + 4 + 7 + 13 + 1 = 26$. Daraus erhält sie den superincreasing Knapsack $a^{\sim}$: $a^{\sim} = 1, 4, 7, 13, 26$. Anschließend bestimmt sie eine natürliche Zahl k mit $k > \sum_{i=1}^{n} a_i^{\sim}$. Sie erhält: $k > 1 + 4 + 7 + 13 + 26 =$

51 und wählt für $k = 53$. Jetzt benötigt sie noch ein zu k teilerfremdes w. Sie wählt dazu $w = 29$. Somit hat sie ihren vollständigen privaten Schlüssel: $(a^{\sim}; k;\ w) = (1, 4, 7, 13, 26; 53; 29)$. Für den öffentlichen Schlüssel a_i berechnet sie $a_i = a_i^{\sim} * w\ mod\ k\ mit\ i = 1, \dots, n$ und erhält den Vektor

$a = \begin{pmatrix} 1 \\ 4 \\ 7 \\ 13 \\ 26 \end{pmatrix} * 29 = \begin{pmatrix} 29 \\ 116 \\ 203 \\ 377 \\ 754 \end{pmatrix} mod\ 53 = \begin{pmatrix} 29 \\ 10 \\ 44 \\ 6 \\ 12 \end{pmatrix}$ als öffentlichen Schlüssel.

Verschlüsselung:

Will Bart den Buchstaben Y an Lisa senden muss er Y erst quellencodieren. Legt man die Stellung des Alphabets zu Grunde, so erhält man $Y = 25$. Binär dargestellt ergibt sich damit $Y = 1\ 1\ 0\ 0\ 1$. Y ist also die Nachricht x. Er verschlüsselt, indem er $s(x) = a * x = \sum_{i=1}^{n} a_i * x_i$ bildet:

$s(x) = \begin{pmatrix} 29 \\ 10 \\ 44 \\ 6 \\ 12 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 29 + 10 + 12 = 51$. Jetzt sendet er $s(x) = 51$ an Lisa.

Entschlüsselung:

Lisa kann mit Hilfe ihres privaten Schlüssels $(a^{\sim}; k;\ w) = (1, 4, 7, 13, 26; 53; 29)$ die Nachricht $s(x) = 51$ entschlüsseln. Zuerst berechnet sie mit Hilfe des erweiterten euklidischen Algorithmus (siehe die mathematischen Grundlagen) w^{-1} aus $1 = ggT(k, w) = ggT(53{,}29)$ und kommt auf $w^{-1} = 11$. Hiernach berechnet sie $s^{\sim} = s(x) * w^{-1}\ mod\ k = 51 * 11\ mod\ 53 = 31$. Nun gilt es das Knapsack-Problem $a^{\sim} * x = 31$ zu lösen.

Lisa bestimmt die größte Komponente mit Index i, für die gilt: $a_i \leq s^{\sim}$: $s^{\sim} = 31 > 26$.

An der Stelle x_5 wird 1 gesetzt und wir erhalten: $\begin{pmatrix} 1 \\ 4 \\ 7 \\ 13 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 31 - 26 = 5 \Rightarrow x_5 = 1.$

Nun wird analog weitergerechnet: $4 < 5 \Rightarrow x_2 = 1$ und $\begin{pmatrix} 1 \\ 7 \\ 13 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_3 \\ x_4 \end{pmatrix} = 5 - 4 = 1$

$1 < 4 \Rightarrow x_1 = 1$ und $\begin{pmatrix} 7 \\ 13 \end{pmatrix} * \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = 1 - 1 = 0$

$\Rightarrow x_3 = 0$ und $x_4 = 0 \Rightarrow x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. Dies ist die Nachricht, welche Bart an Lisa gesandt hat.

Merkle und Hellman konstruierten 1978 das MH-Verfahren. Der Trick bestand darin, einen simplen Knapsack in einem mit derselben Lösung zu verstecken. Dies ist gleichzeitig als Vor-, sowie als Nachteil zu sehen, da Kryptoanalytiker genau bei diesem Punkt ansetzen. Die Schwäche beruht darauf, dass ein Angreifer nicht ein bestimmtes Paar (k, w^{-1}) finden muss. Im Allgemeinen existieren mehrere solcher Paare.

Anders als RSA und Zero-Knowledge kann das Knapsack-Verfahren nicht zur Signatur von Nachrichten verwendet werden. Die Bedingung $S_X\big(T_X(m)\big) = m$ wo T_X der geheimen und S_X der öffentliche Schlüssel ist, ist nicht erfüllt. T_X und S_X sind nicht zueinander inverse Funktionen. Wir erhalten somit nicht: $(S_X\big(T_X(m)\big) = T_X(S_X(m))$. (Siehe auch die Details zu Public- Key Kryptosysthemen nach Diffie und Hellman.)

5. Quantenkryptographie – Der Siegeszug einer neuen Verschlüsselungsmethode?

Alles was mit Quanten beginnt lässt manchmal Laien oder auch Wissenschaftler ungläubig den Kopf schütteln. Allein das Wort stellt unsere Weltansichten in Frage. Ähnlich verhält es sich mit der Quantenkryptographie. Für die Schüler stellt sich hier ein ganz anderer Sachverhalt dar. Ging es bei den bisherigen Verfahren darum, Nachrichten möglichst gut zu verschlüsseln und die verschlüsselte Nachricht sicher zu übermitteln, verzichtet die Quantenkryptograhie auf die Chiffrierung. Wird die Nachricht abgefangen, so wird sie dadurch zerstört. Es ist also egal, ob unverschlüsselte Nachrichten abgefangen werden können. Die einzige Frage, die von Bedeutung ist lautet:

„Werden wir bei der Kommunikation belauscht?“

Dieser grundlegende Unterschied muss den Schülern klar gemacht werden. Aufgrund der Nähe zur Physik eignet sich hier die Möglichkeit eines fächerübergreifenden Unterrichts besonders gut. Der Mathematiklehrer kann zeigen, wie Mathematik und Quantenphysik zusammenwirken, ohne große Ansprüche an Kenntnisse aus der Quantenmechanik zu stellen. Die einzelnen Fachvokabeln, die benötigt werden können je nach Zeit mehr oder

weniger exakt eingeführt werden. Für ein erstes Verständnis genügen die Ausführungen im Teil Physikalische Grundlagen.

Die Kommunikationsanordnung kann Schritt für Schritt mit den Schülern durchgedacht werden. Dabei sollen sie verstehen, dass nur die Hälfte der Messungen, die Bart von Lisa empfängt, richtig sind. Im Falle eines Lauschers werden noch weniger korrekte Bits empfangen, wodurch der Lauscher letztlich entdeckt wird. Verglichen mit andern Verschlüsselungsmethoden steht die Quantenkryptographie noch am Anfang.

Vorteile +

- Jeder Versuch die Kommunikation zu belauschen, wird entdeckt.

Nachteile −

- Bis jetzt liegt die Distanz, über die gesendet werden kann, unter 200 km.
- Neue Kommunikationspartner müssen erst eine Glasfaserleitung verlegen lassen, um mit andern in Kontakt treten zu können.

Allein die Aktualität dieses Verfahrens rechtfertigt aber eine eingehendere Auseinandersetzung damit. Im Internet finden sich einige Meldungen, wie Inhalte der Quantenkryptographie im Unterricht Einzug finden. Erwähnt sei an dieser Stelle das Albert-Schweizer Gymnasium in Erlangen. Die vollständigen Unterrichtseinheiten finden sich kostenlos zum Download auf: http://www.didaktik.physik.uni-erlangen.de/quantumlab/.

Literaturverzeichnis

[b1] Beutelspacher A. Kryptologie- Eine Einführung in die Wissenschaft vom Verschlüsseln und Verheimlichen. Friedr. Vieweg & Sohn Verlag, Wiesbaden: 2007

[b2] Bouwmeester D., Ekert A., Zeilinger A. The Physics of Quantum Information. Springer-Verlag Berlin Heidelberg New York: 2001

[b3] Brands G. Verschlüsselungsalgorithmen- Angewandte Zahlentheorie rund um Sicherheitsprotokolle. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/ Wiesbaden: 2002

[b4] Buchmann J. Einführung in die Kryptographie. Springer- Verlag Berlin Heidelberg: 2008

[b5] Czakler K. Diplomarbeit- Zahlentheorie im Schulunterricht- Möglichkeiten und Grenzen: 2007

[b6] Dorninger D. Chiffrierung und Datensicherheit. Unterlagen zu einem Lehrefortbildungsseminar: 1992

[b7] Eigenthaler G. Begleitmaterial zu Vorlesung Algebra: 2005

[b8] Enzensberger H.M. Der Zahlenteufel- Ein Kopfkissenbuch für alle, die Angst vor der Mathematik haben. Deutscher Taschenbuchverlag GmbH & Co. KG, München: 2006

[b9] Ertel W. Angewandte Kryptographie. Carl Hanser Verlag München Wien: 2001

[b10] Grath K. Diplomarbeit- Kryptologische Methoden und deren Anwendung im Schulunterricht: 2005

[b11] Schulz R. H. Praxis und Methodik- RSA & Co. in der Schule. Heft Nr. 152. Berlin: 2008

[b12] Singh S. Geheime Botschaften. Carl Hanser Verlag München Wien: 2000

[b13] Singh S. Fermats letzter Satz- Die abenteuerliche Geschichte eines mathematischen Rätsels. dtv, München: 2000

[b14] Wobst R. Abenteuer Kryptologie- Methoden, Risiken und Nutzen der Datenverschlüsselung. Addison- Wesley Verlag, München: 2001

Webseiten im Internet

[i1] http://de.wikipedia.org/wiki/Quant

[i2] http://www.spiegel.de/netzwelt/tech/0,1518,511087,00.html

[i3] http://wapedia.mobi/de/Quantenkryptografie

[i4]http://www.johannes-bauer.com/thi/millerrabin.php

[i5] http://de.wikipedia.org/wiki/Erweiterter_euklidischer_Algorithmus

[i6]
http://www.chipkarte.at/portal/index.html?ctrl:cmd=render&ctrl:window=ecardportal.chan
nel_ content.cmsWindow&p_menuid=51924&p_tabid=4

[i7]http://de.wikipedia.org/wiki/Web_of_Trust

[i8] http://www.bsi-fuer-buerger.de/schuetzen/07_0301.htm

[i9] http://de.wikipedia.org/wiki/Web_of_Trust

Weitere interessante Seiten

Quantenkryptographie in der Schule:

http://www.didaktik.physik.uni-erlangen.de/quantumlab/Unterricht/index.html

Die Bild zur Quantenkryptographie:

http://www.whitehat.ch/cissp/krypto/Kryptographie-Dateien/image001.png

Ein Artikel zur Quantenkryptographie:

http://www.pro-physik.de/Phy/pdfs/ger_tittel.pdf quantenkryptographie

Der erweiterte euklidische Algorithmus:

http://www.iti.fh-flensburg.de/lang/krypto/grund/gruppezn.htm#section3

Abbildungen:

[a1]: RSA Väter: http://www.usc.edu/dept/molecular-science/pictures/RSA-2003.jpg

[a2]: Lisa und Bart: http://www.tobias-rahlf.de/LisaSimpson11.gif; http://tr.wikipedia.org/wiki/Bart_Simpson

[a3]:Homer: http://media.photobucket.com/image/homer20simpson.gif

[a4]: Quantenkryptographie: http://www.whitehat.ch/cissp/krypto/Kryptographie-Dateien/image001.png

[a5]: e-card: http://www.svb.at/mediaDB/MMDB90265_e-card%20Vorderseite-%20gro%C3%9F.JPG