

Computergestützte Kryptographie

E.R. Sexauer

Dez 2009

Zusammenfassung



Das Bedürfnis, Nachrichten zu verschlüsseln, ist so alt wie die Kommunikation selbst. Die Verfügbarkeit von Computern für jedermann hat es ermöglicht, Verfahren anzuwenden, die auf rechenaufwändiger Mathematik beruhen und ein hohes Maß an Sicherheit gegen Entschlüsselung bieten. Im folgenden werden einige Grundbegriffe und Verfahren der Kryptographie erläutert.



Als Ausgangslage setzt man immer den '*worst case*' voraus, d.h. man nimmt an, daß der Gegner mithören kann und Informationen über das verwendete Verfahren hat.



Das ideale Verfahren verlangt nur einen einmaligen, einfachen und sicheren Austausch eines Schlüssels. Selbst wenn der Gegner das Verfahren kennt und immer mithören kann, sollte er nicht in der Lage sein, Nachrichten zu entschlüsseln.

Inhaltsverzeichnis

1	Klassische Verfahren	4
1.1	Die Glatze des Sklaven	4
1.2	Caesar-Verschlüsselung	5
1.3	Verbesserung, Vigenère-Verschlüsselung	7
1.4	Verbesserung, Rote Kapelle	8
2	Security by Obscurity	9
3	Mechanische Verfahren	10
3.1	Colossus	12
3.2	Abschied von der Mechanik	13
4	Computergestützte Verfahren	14
4.1	Onetimepad, das einzig sichere Verfahren	14
4.1.1	Zufallszahlen aus der Natur	14
4.1.2	Erzeugung von Zufallszahlen per Computer	15
4.1.3	Alternativen ohne Datentransport	15
4.1.4	Der ultimative Generator	16
4.2	Symmetrische Verschlüsselung (DES, AES)	17
5	Hash-Keys	18
5.1	Hashkeys und Zufallszahlen	19
5.2	Rechtslage	19
6	Schlüsselverteilung nach Diffie-Hellmann	20
7	Asymmetrische Verschlüsselung	21
7.1	Signatur mit RSA	21
7.2	Wertevorrat	22
8	Steganographie	23
9	Kryptographie und Rechengeschwindigkeit	24
9.1	GMP (GNU multiple precision arithmetic)	24
9.2	Berechnung von Potenz-Resten	24
9.3	Primzahltest	25
10	Kryptographische Protokolle	26
10.1	Zero-Knowledge-Systeme:	26
10.2	CHAP (Challenge Access Protocol)	26
11	Mann in der Mitte (man in the middle)	28
12	Kryptographie und Öffentlichkeit	29
13	Kryptographie und freie Software	30
14	Kryptographie und Gesetzgebung	31
15	Schlussfolgerung	33

16 Vorhersagen sind schwierig,	34
17 Web of Trust	36

1 Klassische Verfahren

1.1 Die Glatze des Sklaven



Ein Bote wird rasiert und die Nachricht wird auf die tätowiert. Wenn die Haare wieder gewachsen sind, wird der Bote losgeschickt. Der Empfänger kennt das Geheimnis der Glatze, rasiert den Boten und liest die Nachricht.

Nachteile:

- Wenn der Gegner das Verfahren kennt, versagt die Methode.
- Das Nachwachsen der Haare beansprucht viel Zeit.
- In der Regel kann man einen Boten nur einmal verwenden.

1.2 Caesar-Verschlüsselung

Sender und Empfänger besitzen eine Tabelle, in der jedem Zeichen genau ein anderes Zeichen zugeordnet wird - z.B. a->f, d->5, 7->y usf.

Nachteile:

- Die Tabelle muß so übermittelt werden, daß der Gegner nicht mitlesen kann. Wenn man mit vielen Stellen kommuniziert und den Schlüssel häufig wechselt, kann das zum Problem werden.

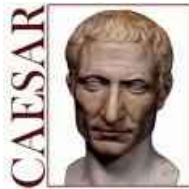


- In jeder Sprache gibt es typische Häufigkeiten für einzelne Zeichen, z.B. ist 'e' in vielen Sprachen das weitaus häufigste Zeichen. Im Deutschen sind die Paare 'ei' und 'ie' sehr häufig, im Englischen ist das 'th' typisch. Wenn der Gegner die Zeichen in der verschlüsselten Nachricht auszählt, kann er den Text relativ leicht entschlüsseln. Ein hübsches Beispiel findet sich der Novelle 'Der Goldkäfer' von E.A. Poe;¹ der Dichter galt übrigens als Experte für Geheimschriften.
- Wenn man die Sprache nicht kennt, können Häufigkeitsprofile einen Hinweis geben.

Trivialversion:

Caesar verwendete eine vereinfachte Version, bei der alle Buchstaben zyklisch um eine bestimmte Zahl verschoben wurden, z.B. mit $z=4$ ergibt sich a->e, b->f usw. Natürlich ist dieses Verfahren noch leichter zu knacken.

Caesar war nicht so schlecht:



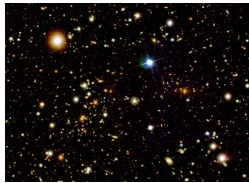
Zu Caesars Zeiten war die Häufigkeitsanalyse noch nicht bekannt; die erste Beschreibung dieses Angriffs stammt aus dem Arabien des 9-ten Jahrhunderts. Außerdem muß gesagt werden, dass es im Lateinischen keine so auffälligen Häufigkeiten gibt. Z.B. sind 'e' und 'i' etwa gleich häufig, dicht gefolgt von 'a', 's', 't' und 'u' die ebenfalls etwa gleich häufig sind². Bei Paaren stehen 'er' und 'um' an der Spitze. Wirklich hervorstechende Tripel gibt gar nicht.

¹Bei Gutenberg.org als Ebook verfügbar

²Gezählt in Caesar Gallischem Krieg

Wertevorrat:

Die Zahl möglicher Vertauschungen ist auch für heutige Computer beachtlich. Bei 40 Zeichen gibt es 10^{47} Möglichkeiten, bei 80 Zeichen sind es bereits 10^{118} .



Zum Vergleich: Man schätzt, dass es im Universum etwa 10^{80} Atome gibt³.

³Bei einem Zeichensatz von n Zeichen beträgt die Anzahl möglicher Permutationen etwa $\sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

1.3 Verbesserung, Vigenère-Verschlüsselung

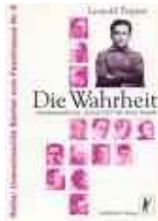
Bei diesem Verfahren werden mehrere unterschiedliche Tabellen verwendet; die zyklische Reihenfolge des Wechsels wird durch ein Schlüsselwort gesteuert. Die Methode kam im 16-ten Jahrhundert in Gebrauch und galt bis zum 19-ten Jahrhundert als sicher. In der einfachsten Version bestimmt jeder Buchstabe des Kennworts die Verschiebung im Alphabet. Komplizierte Versionen ordnen jedem möglichen Buchstaben des Kennworts eine komplette Permutation zu.



Der Mathematiker Babbage⁴, der den ersten mechanischen Computer konzipierte, entdeckte den Schwachpunkt des Verfahrens: Die Länge 'n' des Passworts bestimmt die Periode des Wechsels. Wenn es anhand von Wiederholungen, die z.B. durch Verschlüsselung des häufigen Worts 'der' entstehen können, gelingt, die Länge des Schlüssels zu erraten, zerfällt der verschlüsselte Text in 'n' Caesar-verschlüsselte Teile, auf die dann Häufigkeitsanalysen angewendet werden können.

⁴Babbage publizierte seine Methode nicht, daher wird sie meist nach Kasiski benannt, der wenige Jahre später auf die gleiche Idee kam.

1.4 Verbesserung, Rote Kapelle



Im zweiten Weltkrieg wurde die Nachricht gemeinsam mit dem Text eines Buches verschlüsselt, wobei jeweils das x -te Zeichen der Nachricht und das x -te Zeichen des Buchtexts ein Paar bildeten, welches dann über eine Tabelle (wie bei Cäsar) zu einem neuen Paar verschlüsselt wurde. Die Textzeichen des Buches wurden über eine längere Periode hochgezählt, d.h. wenn die zehnte Nachricht beim Zeichen 2001 endete, begann die Verschlüsselung der nächsten Nachricht mit dem Zeichen 2002.

Probleme:

- Beide Seiten mußten genau Buch führen. Wenn eine Nachricht verloren ging oder verstümmelt wurde, brach die Kommunikation zusammen.
- Auch bei kombinierten Texten gibt es typische Häufigkeiten; 'ee' kommt z.B. häufiger vor als 'xy'. Auf Dauer kann der Gegner durch Auszählen Anhaltspunkte zur Entschlüsselung gewinnen.
- Wenn der Gegner das Buch identifiziert, kann er alle vergangenen Nachrichten entziffern (das ist es im zweiten Weltkrieg tatsächlich passiert).

2 Security by Obscurity



Im zweiten Weltkrieg haben die Amerikaner ein Team von Navajos zur zeitnahen Übermittlung von Funksprüchen eingesetzt⁵. Sie verließen sich darauf, dass deren Sprache von den Japanern weder erkannt noch verstanden wurde.

Natürlich hätte ein dummer Zufall - z.B. ein Japaner, der eine Navajo geheiratet hatte - das Verfahren scheitern lassen.

Verfahren, die auf diesem Prinzip beruhen, sind grundsätzlich unsicher.

⁵Da es in der Sprache der Navajos keine Wörter für moderne Waffen gab, mußte die Sprache künstlich erweitert werden.

3 Mechanische Verfahren



Schon für Cäsar- und Vigenèreverschlüsselung gab es mechanische Hilfsmittel in Form von Chiffrierscheiben, bei den Tabellen durch handliche Geräte ersetzt wurden.

Nachdem es diese Scheiben einmal gab, lag es nahe, sie auf Zahnräder zu montieren und während der Verschlüsselung nach einem bestimmten Schema zu bewegen.

Die bekannteste dieser Maschinen ist die Enigma, die im zweiten Weltkrieg von den Deutschen verwendet wurden. Sie besteht aus mehreren (bis zu 5) Rotoren, von denen jeder einzelne eine Cäsar-Verschlüsselung bewirkt. Die Gesamtverschlüsselung ist die Folge jeder einzelnen Verschlüsselung. Die Anfangsstellung wird durch ein Schlüsselwort bestimmt. Der erste Rotor bewegt sich mit jedem Zeichen eine Position vorwärts. Die anderen Rotoren haben einen oder mehrere Mitnehmer, die für die Bewegung der folgenden Rotoren sorgen.

Dadurch wird eine Periode in der Größenordnung $10^{*}5$ erzeugt, die das System gegen Häufigkeitsanalysen immun macht. Der Schlüsselraum entspricht über 70 Bits - für damalige Verhältnisse eine astronomische Zahl.



Die Enigma wurde trotzdem geknackt. Dafür gab es mehrere Gründe:

- Das Grundprinzip war bekannt, da es patentiert war und kommerzielle Versionen verkauft wurden. Einzelheiten der militärischen Versionen wurden durch Spionage bekannt.



- Schon Anfang 1930 hatte der polnische Mathematiker Rejewski das damals verwendete Modell der Enigma analysiert und eine Farm von Maschinen gebaut, mit denen er den Schlüssel bestimmen konnte. Seine Erkenntnisse stellten die Polen später den Engländern zur Verfügung.
- Die Maschine war so konstruiert, dass sich ein Zeichen niemals in sich selbst transformierte. Dadurch wurde das Ausprobieren drastisch vereinfacht - man konnte einen Versuch abbrechen, sobald man auf eine Gleichheit stieß.
- Die Engländer entwickelten Maschinen, die das Ausprobieren gewaltig be-



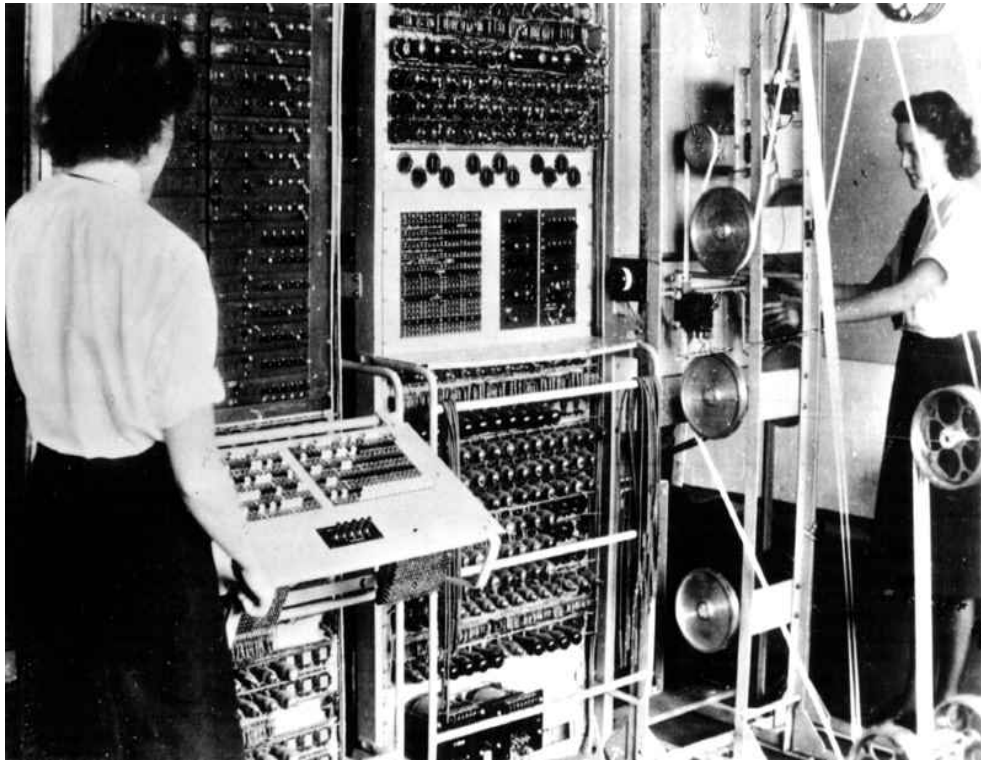
schleunigten. Bekannt wurde die Turing-Bombe⁶.

- Oft waren den Entschlüssern Teile der Texte bekannt; die stereotype Form militärischer Meldungen erwies sich dabei als sehr hilfreich.

⁶Als Turing nach dem Krieg wegen Homosexualität in Schwierigkeiten geriet, ließ die englische Regierung ihn fallen. Erst 2009 entschuldigte sich der Premierminister Brown dafür.

3.1 Colossus

Im Team, das an der Enigma arbeitete, wurde 1943 der erste programmierbare Computer entwickelt. Er hatte 1.500 Röhren. Aus Geheimhaltungsgründen wurde er nach Kriegsende demontiert; seine Pläne wurden vernichtet.



Die amerikanische ENIAC wurde erst 1945 in Betrieb genommen, sie hatte 18.000 Röhren und war zur Berechnung von ballistischen Bahnen entwickelt worden.

3.2 Abschied von der Mechanik

Enigma war die letzte Verschlüsselungsmaschine, die elektromechanisch betrieben wurde. Die Zukunft gehörte elektronischen Geräten. Der Einsatz von Elektronik bedeutete nicht nur einen quantitativen Sprung:

- Durch die hohe Verarbeitungsgeschwindigkeit wurden jetzt Verfahren einsetzbar, die bisher aufgrund des hohen Rechenaufwands indiskutabel waren.
- Ungewollt, aber zwangsläufig wurde mit zunehmender Verbreitung von Computern die Kryptographie von einer Geheimwissenschaft zum Gegenstand öffentlich betriebener Forschung.

4 Computergestützte Verfahren

4.1 Onetimepad, das einzig sichere Verfahren



Wenn beide Seiten über die gleiche Folge von echten Zufallszahlen verfügen, jedes Zeichen der Nachricht mit einem Zeichen der Zufallsfolge verschlüsseln, z.B. durch 'xor', und die verwendeten Zeichen der Folge anschließend löschen, ist das Verfahren mathematisch sicher, da jedes Verschlüsselungsergebnis gleich wahrscheinlich ist. Diese Methode wurde während des Kalten Krieges auf dem 'heißen Draht' zwischen Washington und Moskau verwendet. Wenn die Zufallsfolgen tatsächlich vernichtet wurden, wird man nie mehr rekonstruieren können, was dort kommuniziert wurde.

Nachteile:

- Die Zufallsfolge muß genauso lang sein wie die eigentliche Nachricht. Das kann zu Problemen beim Austausch führen. Für ein Aufklärungsflugzeug ist es beispielsweise kein Problem, beim Start von der Bodenstation einen Datenträger mitzunehmen, für einen Geheimagenten kann es eine unüberwindliche Schwierigkeit darstellen.
- Eine zeichengenaue Buchführung ist zwingend erforderlich.
- Die Erzeugung von Zufallszahlen ist nicht einfach.

4.1.1 Zufallszahlen aus der Natur



Der radioaktive Zerfall ist zwar statistisch konstant, aber im kleinen unvorhersehbar. Ein Stück Radium und ein Geigerzähler liefern ein perfektes zufälliges Geräuschmuster.



Auch die Bewegungen von Blättern im Wind - es nie wirklich windstill - generieren gute Zufallszahlen.

4.1.2 Erzeugung von Zufallszahlen per Computer

Ein Computer ist eine deterministische Maschine, auch wenn es vielen Anwendern nicht so vorkommt. Daher ist es nicht trivial, Zufallsfolgen zu erzeugen. Linux verwendet ein zweistufiges Verfahren:

- Das Device `/dev/random` wird durch äußere Ereignisse wie Mausbewegungen oder Tastatureingaben verknüpft mit der Zeit gefüllt. Dies liefert starke Zufallszahlen. Wenn keine Ereignisse vorliegen, wird die Ausgabe geblockt.
- Das Device `/dev/urandom` benutzt `/dev/random`, solange dieses Daten liefert. Anderenfalls wird ein Pseudogenerator benutzt.



Spötter sagen, dass man bei Windows keinen Zufallsgenerator braucht...

4.1.3 Alternativen ohne Datentransport

Wenn Alice und Bob per Onetimepad kommunizieren möchten, könnten sie sich z.B. darauf einigen, eine stark frequentierte Mailingliste oder Newsgroup zu abonnieren, und aus deren Daten per Hashing eine Zahlenfolge zu erzeugen. Solange der Gegner nicht weiß, welche Mailingliste benutzt wird und das Auswahlverfahren nicht genau kennt, kommt diese Methode einem echten Onetimepad sehr nahe.



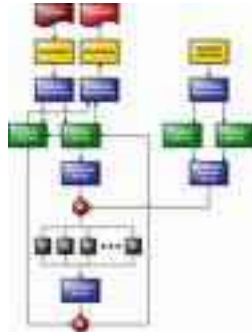
Natürlich kann man auch chaotische Systeme, z.B. die Mandelbrot-Iteration oder das Mehrkörperproblem, verwenden, um ein Pseudo-Onetimepad zu erzeugen. Man muß dabei nur sicherstellen, auf beiden Seiten die gleiche Gleitkommabibliothek zu verwenden, da auf die Hardware nicht unbedingt Verlaß ist.

4.1.4 Der ultimative Generator

Es gibt Generatoren im Handel, die auf quantenphysikalischen Effekten basieren und mehrere Mbit/s echte Zufallsfolgen liefern (<http://qrbg.irb.hr/>).

4.2 Symmetrische Verschlüsselung (DES, AES)

Die zunehmende Verbreitung von Computern erweckte den Bedarf nach Verschlüsselung, bei der das Verfahren bekannt war und überall implementiert werden konnte, und die Sicherheit ausschließlich vom Schlüssel selbst abhing.



Die Nachricht wird durch eine Serie mehrfacher Umordnungen, gesteuert durch einen Schlüssel, der beiden Seiten bekannt sein muß, kodiert. Beim klassischen DES war der Schlüssel 56 Bits lang, bei AES kann der Schlüssel bis zu 256 Bits lang sein. Die dadurch entstehende Zahl der kombinatorischen Möglichkeiten war bzw. ist so groß, daß Entschlüsselung durch Ausprobieren nicht praktikabel war bzw. ist. Zur Ver- und Entschlüsselung wird das gleiche Kennwort verwendet.

BIG BROTHER



IS WATCHING YOU

DES (Data Encryption Standard) war von der amerikanischen Regierung initiiert und von Anfang an für jedermann lizenzfrei zugänglich. Da die NSA an dem von IBM entworfenen Verfahren einige Änderungen vorgenommen hatte, hielt sich hartnäckig das Gerücht, sie hätte Traps eingebaut, um selbst mitlesen zu können.

Anmerkung-1:

Durch steigende Rechenleistung und verbesserte Probiervverfahren kann der Gegner seine Chancen verbessern. Deshalb wurde das 1976 entwickelte DES mit dem damals astronomisch langen Schlüssel im Jahr 2001 durch AES ersetzt.

Anmerkung-2:

Der Hauptaufwand bei der Entschlüsselung liegt nicht im Bereitstellen der Permutationen sondern in der Überprüfung, ob damit sinnvolle Ergebnisse erzielt werden. Dies erfordert inhaltliche Analysen, z.B. Nachschlagen im Wörterbuch.

5 Hash-Keys



Ein Hashkey ist eine Zahl vorgegebener Länge, die aus einer beliebigen Zahlenfolge (Text) gebildet wird. Er wird - ähnlich wie bei der symmetrischen Verschlüsselung durch eine Folge von Umordnungen aus dem Text gebildet. Anders als bei der Verschlüsselung, ist diese Umordnung nicht umkehrbar.

Hashkeys werden als Fingerabdruck eines Texts verwendet. Im Umfeld der Kryptographie spielen solche Signaturen eine wichtige Rolle.

Ein einfaches Beispiel ist die Quersumme: $12345 \rightarrow 15$. Weglassen oder Hinzufügen verändert die Quersumme, ein Dreher hingegen tut das nicht. Zahlen mit der Quersumme 15 gibt es viele.

Im Handel wird die EAN-Prüfziffer verwendet. Sie ist die letzte Dezimalstelle einer gewichteten (1212...) Prüfziffer und entdeckt die typischen einfachen Tippfehler.

An eine gute Hashfunktion werden folgende Anforderungen gestellt:

- Es soll nur durch Probieren möglich sein, aus dem Hashkey den Text zu erzeugen.
- Es soll nur durch Probieren möglich sein, zu einem Hashkey einen Text zu erzeugen.
- Ähnliche Texte sollen unähnliche Hashkeys erzeugen.
- Der Hashkey soll so groß sein, daß Kollisionen extrem unwahrscheinlich sind.

Verfahren, die dies ermöglichen, sind beispielsweise die Varianten von SHA1; das ältere MD5 sollte nicht mehr verwendet werden.

Beispiel:

```
echo '30' | sha1sum 97ea7ec8a6bb8ab9049d86bc39b5be2b0800b14b
```

```
echo '31' | sha1sum 22980f6cd0807e719d7f1b7cf25edc9df659ba1f
```

5.1 Hashkeys und Zufallszahlen

Da Hashkeys auch auf kleinste Veränderungen empfindlich reagieren, eignen sie sich gut zur Erzeugung von Zufallszahlen:

```
ps aux | sha1sum 47dda820711ef5ecf3d2f896bed314ce1fc1373d
```

Da sich der Zustand des Systems ständig ändert, ist der Wert des Keys nicht reproduzierbar.

5.2 Rechtslage

SHA1 gilt als sichere elektronische Signatur.

Anmerkung zur Manipulation:

Gelegentlich gibt es Berichte - besonders zu MD5 - nach denen signierte Dokumente erfolgreich manipuliert wurden. In der Praxis sind solche Manipulationen allerdings sehr schwierig. Nehmen wir an, ein Kaufvertrag besteht aus einem Formular und Eingabefeldern für Datum, Name, Artikel und Preis. Wenn der Manipulator den Preis ändern will, muss er das Dokument an anderer Stelle so abändern, dass sich wieder der gleiche Hashkey ergibt. Wenn das Formular selbst durch eine Hashsignatur abgesichert ist, bleiben ihm dazu nur die Eingabefelder - auch mit MD5 ist dies praktisch unmöglich.

6 Schlüsselverteilung nach Diffie-Hellmann

Eine potentielle Schwachstelle aller Kryptographie ist der Austausch von Schlüsseln. Das Dilemma erscheint unlösbar:

- Seltener Wechsel verschafft dem Gegner umfangreiches Material zum Angriff
- Häufiger Wechsel erhöht die Gefahr der Entdeckung.

Hier liefert die Mathematik eine erstaunliche Lösung: Mittels Restklassenarithmetik können zwei Personen, Alice und Bob, auf einer abgehörten Leitung ein gemeinsames Geheimnis erzeugen. Um in den Besitz dieses Geheimnisses zu kommen, müßte der Lauscher Gleichungen vom Typ $a^{**}x \text{ mod}(p) = b$ lösen, was bei hinreichend großen Zahlen a, b und p die Rechenleistung auch moderner Cluster bei weitem überfordert.

Alice	$153.171^{**}X = 519 \text{ mod } 1137$	Bob
	?????	
		

Ein Beispielprogramm findet sich im Anhang.

7 Asymmetrische Verschlüsselung

Das Diffie-Hellman-Verfahren ermöglicht es zwei Kommunikationspartnern einen gemeinsamen, geheimen Schlüssel auszuhandeln. Es setzt allerdings voraus, dass vor dem Datenaustausch ein Dialog stattfindet, was für eine globale Kommunikation höchst lästig ist. Auch dafür bietet die Mathematik eine Lösung an: Die asymmetrische Verschlüsselung.

Hier gibt es zwei Schlüssel: Einen öffentlichen Schlüssel zum Verschlüsseln und einen geheimen, privaten Schlüssel zum Entschlüsseln. Der Zusammenhang zwischen beiden Schlüsseln beruht auf mathematischen Theorien, die zu schwer berechenbaren Zahlen führen.



Bei dem häufig verwendeten RSA wird der Umstand ausgenützt, daß kein schnelles Verfahren bekannt ist, große Zahlen in ihre Primfaktoren zu zerlegen. Der private Schlüssel besteht aus zwei großen Primzahlen, der öffentliche Schlüssel ist deren Produkt. Nur wer den privaten Schlüssel kennt, kann mit Hilfe eines Satzes von Euler die mit dem öffentlichen Schlüssel erzeugte Nachricht mit geringen Aufwand dekodieren. Als Schlüssellänge verwendet man heute 100 Dezimalstellen und aufwärts.

Nachteile:

- Ver- und Entschlüsselung erfordern auch auf modernen Computern hohen Rechenaufwand. Daher wird das Verfahren meist nur dazu verwendet, Schlüssel für symmetrische Verfahren sicher auszutauschen.
- Es gibt keinen Beweis, daß es kein schnelles Verfahren zur Primfaktorzerlegung gibt. Daher ist es theoretisch denkbar, daß irgendwann ein solches Verfahren entdeckt wird.

RSA und Häufigkeitsanalyse

Natürlich verschlüsselt man mit RSA nicht einzelne Zeichen sondern Bitgruppen - bei großen Zahlen ist damit eine Häufigkeitsanalyse hoffnungslos.

7.1 Signatur mit RSA



Trotz der Bezeichnung ist RSA in einer Beziehung symmetrisch: Nachrichten, die mit dem *privaten* Key verschlüsselt werden, können mit dem *öffentlichen* Key entschlüsselt werden. Damit ist eine elektronische Unterschrift

realisierbar - jeder kann die Unterschrift lesen, aber nur der Besitzer des privaten Keys kann sie erzeugen.

7.2 Wertevorrat

- 100 Dezimalstellen: In diesem Bereich gibt es etwa 10^{97} Primzahlen. Schränkt man den Bereich für die beiden Faktoren auf 10^{45} bis 10^{55} ein, gibt es in diesem Intervall etwa 10^{52} Primzahlen⁷.
- Bei 300 Dezimalstellen lauten die entsprechenden Werte 10^{297} und 10^{152} .

Rechtslage:

Die Mathematik für RSA stammt aus dem 18-ten Jahrhundert⁸. Die Anwendung durch RSA war patentfähig. Das Patent lief im Jahre 2000 aus. Da das Patent nur in den USA galt, war es leicht zu umgehen.

Alternativen:

Es gibt vergleichbare Verfahren, die auf diskreten Logarithmen⁹, Graphen oder elliptischen Kurven¹⁰ beruhen. Auch dort gibt es keinen Beweis, daß es keine einfache Methode gibt, um den privaten Schlüssel aus dem öffentlichen Schlüssel zu erzeugen.

⁷Die Zahl der Primzahlen $< n$ ist etwa $n/\log(n)$

⁸Satz von Euler

⁹diskrete Logarithmen sind ganzzahlige Lösungen von Gleichungen der Form $a^x = b \pmod{q}$

¹⁰elliptische Kurven sind Gleichungen der Form $y^2 = x^3 + a*x + b$

8 Steganographie

Dieses Verfahren ähnelt dem Onetimepad. Es setzt voraus, daß beide Seiten über eine gemeinsame Bildquelle verfügen.



Der Absender kodiert seine Nachricht binär und ändert entsprechend diesem Bitmuster das Urbild so ab, daß dies optisch nicht wahrnehmbar ist. Z.B. könnte er für ein Nachrichtenbit den Farbwert des entsprechenden Pixels um eins erhöhen. Bei 16-Mio Farben ist das nicht zu entdecken. Der Empfänger vergleicht das so modifizierte Bild mit dem Original und dekodiert so die Nachricht.

Bei der Fülle von Bildern, die im Internet angeboten werden, ist es nicht schwer für beide Seiten, sich das Bildmaterial zu verschaffen. Sie müssen sich nur auf ein Auswahlverfahren einigen, das natürlich einem eventuellen Angreifer nicht bekannt sein darf.



Hier leisten beispielsweise Pornonewsgroups nützliche Dienste, da dort täglich Millionen von Bildern über die Leitungen gehen...

9 Kryptographie und Rechengeschwindigkeit

Der permanente Zuwachs an Rechengeschwindigkeit ist zum Vorteil der Verschlüssler, da z.B. die Erhöhung der Schlüssellänge von 10 auf 100 nur geringe Mehrleistung beim Verschlüsseln erfordert, aber beim Probieren für Entschlüsselung den Aufwand verzehnfacht.

Bei Verlängerung um zehn Dezimalstellen erhöht sich der Entschlüsselungsaufwand bereits um den Faktor 10^{10} (zehn Milliarden).

Beispiel:

1.000.000 Loops: je einmal mult, exp, divmod

1000000 loops, 30 digits, t= 4.7157869339

1000000 loops, 50 digits, t= 5.2382349968 ratio= 1.11078703729

1000000 loops, 70 digits, t= 5.83930492401 ratio= 1.1147466518 1.23824613068

1000000 loops, 120 digits, t= 7.81990194321 ratio= 1.1147466518 1.23824613068
1.65823902836

9.1 GMP (GNU multiple precision arithmetic)

Um mit den für sichere Kryptographie erforderlichen Zahlenungetümen in vertretbarer Zeit rechnen zu können, gibt es die frei verfügbare GMP. Sie ist teils in C, teils in Maschinsprache geschrieben und verwendet ausgefuchste Verfahren. Für alle gängigen Programmiersprachen gibt es Schnittstellen zur GMP.

Da GMP eine Softwarebibliothek ist, erzielt sie auf unterschiedlicher Hardware gleiche Ergebnisse - bei Verwendung von Gleitkommahardware wäre dies heute trotz aller Normierung nicht unbedingt der Fall.

9.2 Berechnung von Potenz-Resten

Bei der Berechnung von Potenzen der Form $173851^{7891} \bmod(56791)$ gibt es einen einfachen Trick:

- $2^{13} = 2^{10} * 2^3 = 1024 * 8 = 8192$
- $2^{13(7)} = 1024(7) * 8(7) = 2 * 1(7) = 2$

Es genügt also, für jeden Zwischenschritt die Reste zu bestimmen und diese zu multiplizieren. Noch einfacher wird die Sache im Binärsystem, mit dem der Computer ja rechnet:

- $10^{1101} = 10^{1000} * 10^{0100} * 10^{0000} * 10^1$
- '1' im Exponenten bewirkt einen Shift nach links
- '0' im Exponenten ergibt 1, Multiplikation mit 1 kann man weglassen

Es genügt also, für die 1-Stellen einen Shift zu machen, die jeweiligen Reste zu bilden und zu multiplizieren. Dadurch wird die Berechnung erheblich beschleunigt. Die GMP enthält eine Funktion, die dieses beschleunigte Verfahren verwendet.

Nur dadurch ist es möglich, die z.B. bei RSA erforderlichen Berechnungen in vertretbarer Zeit durchzuführen.

9.3 Primzahltest

RSA und Diffie-Hellman benötigen große Primzahlen. Die GMP stellt Verfahren bereit, die im Sekundenbereich eine hundert-stellige Zahl testen können.

10 Kryptographische Protokolle

Diese Protokolle dienen dazu, eine Sitzung zu steuern. Insbesondere geht es darum, die Identität des Gegenübers zu verifizieren, ohne dabei Informationen preiszugeben. Sie bilden eine Schicht über der Verschlüsselung. Häufig wird dabei eine dritte, vertrauenswürdige Stelle hinzugezogen.

Bekannte Beispiele sind PGP, SSH und SSL.

Bei Buchverschlüsselung oder Onetimepad muß die Protokollschicht auch die Buchführung regeln und Methoden zur Wiederaufnahme nach Zusammenbruch der Kommunikation zur Verfügung stellen.

10.1 Zero-Knowledge-Systeme:

- Alice kennt ein Geheimnis, das Bob kaufen möchte.
- Bob will nicht bezahlen, bevor er sich davon überzeugt hat, dass Alice das Geheimnis kennt.
- Alice will das Geheimnis nicht verraten, bevor Bob bezahlt hat.

Ein mögliche Strategie besteht darin, eine Folge von Fragen (Challenges) an Alice zu stellen, deren richtige Beantwortung extrem unwahrscheinlich ist, wenn sie das Geheimnis nicht kennt.

Ein anschauliches Beispiel, wie soetwas möglich ist, findet sich unter http://en.wikipedia.org/wiki/Zero-knowledge_proof.

10.2 CHAP (Challenge Access Protocol)

Bob will sich bei seinem Provider Alice anmelden. Beide kennen das Passwort, aber Bob will das Passwort nicht eingeben, bevor er weiß, dass Alice wirklich Alice ist.

Lösung:

Nehmen wir an, das Passwort ist 1001

1. Bob schickt Alice einen Challenge, die Zahl 17. Der Provider muss den Rest $1001(17)=15$, zurückschicken. Wenn sein Gegenüber das Passwort nicht kennt, ist die Wahrscheinlichkeit, richtig zu raten $1/17$.
2. Bob schickt 3 weitere Zahlen, z.B. 13, 21 und 12.

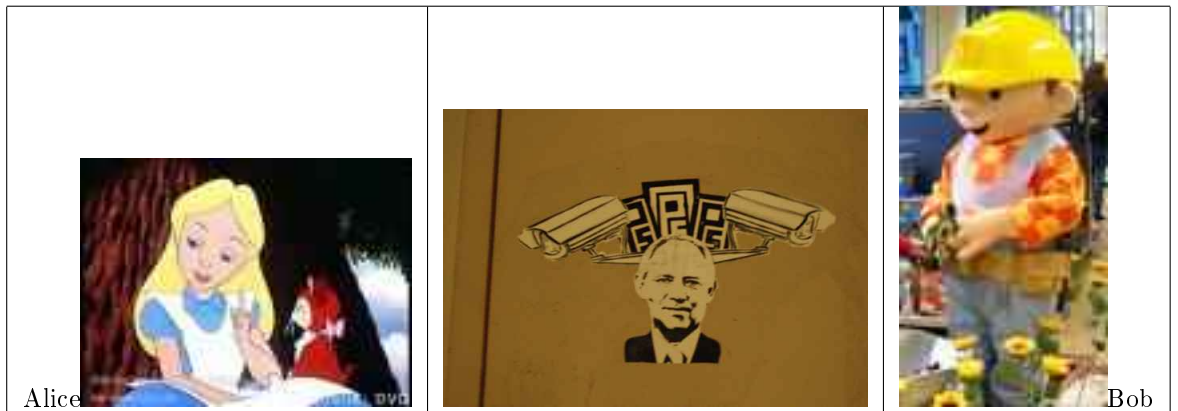
3. Die Wahrscheinlichkeit, 4-mal richtig zu raten ist rund $1/10.000$ - bei größeren Zahlen ist sie natürlich entsprechend kleiner.

Nur bei vier Richtigen glaubt Bob, dass Alice es wirklich ist. Natürlich reagiert er nur auf das Gesamtergebnis, nicht aber auf Zwischenergebnisse.

Dieses einfache Verfahren gibt einem Lauscher Informationen, aus denen er relativ schnell das Passwort berechnen kann. Es gibt aber bessere Verfahren, bei denen der Lauscher keine Chancen hat.

11 Mann in der Mitte (man in the middle)

Bei diesem Angriff hat der Gegner Zugriff auf die Datenleitung und schaltet sich zwischen Sender und Empfänger. Beiden spiegelt er vor, der jeweils andere zu sein.



Wenn die Datenleitung die einzige Kommunikationsmöglichkeit mit der Welt ist, gibt es gegen diesen Angriff keine Abwehr.

SSH und SSL können diesen Angriff erkennen, wenn einmalig über Fingerabdruck oder Zertifikat die Identität des Gegenübers bestätigt wurde. Dafür wird allerdings ein sicherer Datenaustausch benötigt.

Wenn Alice und Bob ein gemeinsames Geheimnis haben - z.B. den Spitznamen ihres Lateinlehrers - können sie daraus eine Identifikation herleiten, ohne das Geheimnis übertragen zu müssen.

Anmerkung zu Zertifikaten:

Kommerzielle Zertifizierer unterliegen den jeweiligen Gesetzen ihres Standorts. Sie genießen keinen Vertraulichkeitsstatus wie Anwälte, Ärzte oder Priester. Auch eine Community wie CAcert bietet keinen Schutz gegen eingeschleuste Agenten.

In diesem Sinne sind selbst generierte Zertifikate am sichersten.

12 Kryptographie und Öffentlichkeit

Während sich Kryptographie früher im Dunkeln der Geheimdienste abspielte, sind moderne Verfahren öffentlich bekannt und werden von Wissenschaftlern aus aller Welt analysiert. Für die Sicherheit der Verfahren ist dies von großem Vorteil. Die Regierungen müssen sich damit abfinden - meist unter Murren - daß sie verschlüsselte Kommunikation nicht mithören können.

13 Kryptographie und freie Software

Auch im Bereich Kryptographie gibt es Softwarepatente und Streit um Lizenzrechte. Glücklicherweise sind die wichtigsten Verfahren als freie Software verfügbar:

- Openssh
- Openssl (benötigt für TLS und HTTPS)
- Openpgp (Mail)
- SHA-x
- verschiedene symmetrische Verfahren, z.B. Blowfish

Diese Software ist Bestandteil der meisten Linux-Distributionen und kann unbedenklich für private und kommerzielle Anwendungen verwendet werden.

14 Kryptographie und Gesetzgebung



Manchen Behörden ist es ein Dorn im Auge, dass jeder Nachrichten senden kann, die sie nicht entschlüsseln können oder - anders ausgedrückt - dass die öffentlich zugängige Kryptographie ebenso gut ist wie ihre eigenen Verfahren.

Tatsächlich hat es Anläufe gegeben, dies zu verhindern:



- In den USA fällt Kryptographie unter das Waffengesetz. Eigentlich sollte man meinen, dass damit in diesem schießwütigen Land ein Freibrief für den Gebrauch von Kryptographie ausgestellt sei. Als jedoch P. Zimmermann PGP publiziert hat, wollte die NSA 1993 eine Klage gegen ihn anstrengen. Heftige Proteste aus der wissenschaftlichen Welt haben dies schlußendlich verhindert. Immerhin hat es 3 Jahre gedauert, bis das Verfahren offiziell eingestellt wurde.
- In USA war der Export von starken Verfahren bis 2001 verboten; nur US-Bürger durften beispielsweise Netscape mit 128-Bit-Schlüssel laden. In nder Praxis war dieses Verbot allerdings unschwer umgehbar.
- US-Bürger waren zeitweise von der Mitarbeit an OpenSSL ausgeschlossen, um potentielle Konflikte zu umgehen.



- In Frankreich hat es in den 1990-er Jahren (1990-1996) ein Gesetz gegeben, das Kryptographie auf ein bestimmtes Verfahren beschränkt hat und die Bürger verpflichtete, ihren privaten Schlüssel bei einer 'vertrauenswürdigen Behörde' zu deponieren; alle andere Kryptographie war für Privatleute verboten. Das Gesetz wurde zurückgezogen, nachdem ein Journalist, der Gesetzestreue demonstrieren wollte, es nicht geschafft hat, eine solche Behörde zu finden und seine Erlebnisse dabei veröffentlichte.

- Auch in den USA gab es 1991 eine ähnliche Gesetzesvorlage, die jedoch nach Protesten der Elektronikindustrie und Bürgerrechtler fallen gelassen wurde. Welcher dieser beiden Proteste wirksamer war, weiß man nicht so genau...
- Das 1994 in den USA eingesetzte Clipper-System sah die Hinterlegung des Schlüssels bei zwei Bundesbehörden vor. Dieser Standard sollte für Auftragnehmer der öffentlichen Hand zwingend sein, und so die Verbreitung sichern. Das Verfahren setzte sich nicht durch.

Derzeit gibt es in Deutschland keine Versuche, den Gebrauch von Kryptographie einzuschränken. Anbetracht der verbreiteten Nutzung wäre ein solcher Versuch auch kaum durchsetzbar.

Es existieren aber rechtliche Grundlagen, die Leitung abzuhören, ohne dies den Benutzern mitteilen zu müssen.

In Deutschland (und nicht nur dort) ist es Ermittlungsbehörden erlaubt, einen Rechner über Internet anzugreifen, um dort Lauschsoftware zu betreiben (Bundestrojaner), die u.a. auch Schlüssel ausschnüffeln soll.

15 Schlussfolgerung

Jedem Normalanwender stehen heute kryptographische Verfahren zur Verfügung, die ebenso sicher (oder unsicher) sind wie die von Geheimdiensten verwendeten Verfahren. Typische Anwendung sind:

- Mail (TLS)-,
- Web (SSL, HTTPS)-,
- Fernzugriff (SSH, SSHFs) und
- Plattenverschlüsselung (Ecryptfs)

Das größte Risiko liegt - wie so oft - beim Anwender. Gegen leichtsinnigen Umgang mit Kennwörtern hilft kein noch so gutes Verfahren.

16 Vorhersagen sind schwierig,



insbesondere, wenn sie sich auf die Zukunft beziehen...

Schon heute kann man in Clustern einfache Schlüssel knacken - es ist eine Kostenfrage¹¹. Stärkere Schlüssel widerstehen diesen Methoden.

Zu einer wirklichen Herausforderung könnten sich jedoch Quantencomputer entwickeln. Diese Maschinen können viele Zustände in einem Bit, genannt Qbit, darstellen und simultan auswerten. Warum das funktioniert, ist nicht zu veranschaulichen und nur schwer zu verstehen.



Zum Trost:

Feynman, eine GöÙe der Quantenphysik: *Niemand versteht die Quantenphysik*¹².



Falls Quantencomputer in der Zukunft praktisch einsetzbar werden sollten, sind erhebliche Veränderungen zu erwarten. Es gibt den Shor-Algorithmus¹³, mit dem man heutige RSA-Schlüssel in realistischer Zeit

¹¹<http://www.heise.de/newsticker/meldung/Preiswert-Schlüssel-knacken-in-der-Cloud-848574.html>

¹²Seine Studenten haben logischerweise protestiert, warum er das dann von ihnen verlangte.

¹³http://en.wikipedia.org/wiki/Shor's_algorithm

17 Web of Trust



Der Mensch ist ein soziales Lebewesen. Ohne Vertrauen kann er nicht existieren. Auch die beste Kryptographie kann Vertrauen nicht ersetzen. Freie Software und die damit verbundenen Gemeinschaften sind *ein* wesentliches Bollwerk gegen Monopolisierung der Kommunikation und den Überwachungsstaat.



Das wird es nicht geben, solange es das Web of Trust gibt.

Anhang, Diffie-Hellman

```
#!/usr/bin/python
import random
prim=1013 # public, prime
rnd=319 # public, 2<=rnd<=prim-2
# random number for alice and bob each, secret
alice=random.randint(2,prim-1)
bob=random.randint(2,prim-1)
print "public prime, random: ", prim, rnd
print "secret random alice, bob: ", alice, bob
amod=(rnd**alice)%prim
bmod=(rnd**bob)%prim
print "random**secret modulo prime, alicemod, bobmod: ", amod, bmod
print "solve: (", rnd, "** x) modulo", prim, "=", amod
# xchange amod, bmod, public
print "xchange public: ", amod, bmod
almod=(bmod**alice)%prim
blmod=(amod**bob)%prim
print "compute: (bobmod**alice) = (alicemod**bob) modulo prime "
print "alices and bobs common secret: ", almod, blmod
```

Anhang, RSA

```
#!/usr/bin/python
import random
# sample rsa crypt/encrypt
# choose 2 primes, secret
p=113 q=103
pq=p*q # public
# compute phi(p,q), phi=number of coprimes phi=(p-1)*(q-1)
# choose e, coprime to phi
e=59 # public
# solve d*e = 1(phi), d = inverse of e modulo phi # Euler: a**phi(m)=1 mod
m # a**(phi(m)-1)= a**(-1) mod m # d=(e**(phi-1))%phi # private
print "public key: ", pq, e print "private key: ", pq, d
# crypt m with public key m=random.randint(2,1001)
print "crypt with public key, m=", m
c=(m**e)%pq print "c=", m, "**", e, " modulus", pq, "=", c
print "solve: ", c, "= (x **", e, ") modulus", pq
# decrypt c with private key
print "decrypt with private key, c=", c
m=(c**d)%pq
print "m=", c, "**", d, " modulus", pq, "=", m
```