Version 1.04

Die **Mantra FAQ** wird mit freundlicher Genehmigung von Michael Uplawski Bestandteil der **Deutschen Anleitung zu PGP**
Die hier angesprochenen Punkte sind m. M. nach für eine sichere Anwendung von PGP 5/6 wichtig, besitzen aber darüber hinaus Relevanz für jedes kryptografische Programm, das mit der Eingabe von Passwörtern und Passphrases arbeitet.

Die FAQ geht auf **Randall T. Williams** zurück und wird in englischer Sprache von **Arnoud Engelfriet** als **Passphrase FAQ** im WWW bereitgestellt.

Einleitung

> *Innerhalb der FAQ haben Mantra und Passphrase die gleiche Bedeutung: Eine Abfolge von Zeichen, Ziffern, Text und Sonderzeichen in sinnloser oder sinnhafter Kombination, die Berechtigten Zugriff auf verschlüsselte oder sonstwie gesicherte Daten(bestände) ermöglichen soll und durch ihre Unkenntnis, Unberechtigten diesen Zugriff verweigert. Passphrase anstelle von Passwort soll verdeutlichen, dass mehr als die Anzahl von Zeichen eines Wortes, also eher die Anzahl von Zeichen ganzer Sätze besprochen wird.*
>
> *Es folgt die Übersetzung der Original-Einleitung von Randall T. Williams*

Dies ist die Passphrase FAQ für PGP. Ich habe versucht, alles, was in alt.security.pgp gefragt wurde und ein paar weitere Informationen zu Dingen wie Passworten und verschiedenen Schlüssellängen, zusammenzutragen. Die meisten Menschen, die in einer höheren Schule Algebra hatten, sollten in der Lage sein, die Mathematik zu verstehen. Der Glossar im Abschnitt **8.2** kann bei manchen Begriffen und ihrer Verwendung Hilfestellung geben.

MD5 und IDEA basieren auf Blöcken von 128 Bits. Es ist trivial, sie durch 56 Bit DES-Schlüssel oder Schlüssel anderer Längen zu ersetzen. Passwörter unterscheiden sich von Passphrases durch ihre Länge. Die Analyse Ihrer Passworte und Passphrases macht sich die gleichen Ansätze zunutze.

Die FAQ ist noch im Entstehen begriffen (sind sie das nicht alle?). Änderungen werden ausgeführt, wenn ich darauf hingewiesen werde, darauf stosse oder Zeit dazu finde. "Angewandte Kryptographie" [**1**] lässt mich annehmen, dass über die Forschungen auf diesem Gebiet noch nicht viel veröffentlicht worden ist. Das Knacken von Passwörtern wird zwar abgehandelt, nur wenig wird allerdings über Passphrases ("Mantras") gesagt. Die dabei anzuwendende Mathematik ist normalerweise nicht Teil meiner Tätigkeit. Ich bin ausgebildeter Elektrotechniker. Kryptographie ist bloss ein Hobby. Verzeihen Sie mir bitte alles, was trivial erscheint oder übersehen wurde. Das Meiste habe ich aus dem Kopf

geschrieben. Kommentare, Korrekturen, Hinzufügungen, Aufmunterungen und positive Kritik können (in englischer Sprache) an **ac387@yfn.ysu.edu** gesendet werden, erwarten sie aber bitte nur spärliche Beantwortung. Flames sollten nach /dev/null geschickt werden.

Wo finde ich die Passphrase FAQ?

Das ist eine vertrackte Situation. Da Sie dies lesen, haben Sie sie gefunden. Wenn Sie die FAQ nicht finden, wird Ihnen dieser Abschnitt kaum weiterhelfen. Bis jetzt gibt es nur drei offizielle Bezugsadressen für die Passphrase-FAQ. Dank an Galactus, Don Henson und Patrick Finerty für ihre Web-Seiten. Eine PGP-unterschriebene Ausführung wird darüberhinaus Mitte des Monats nach alt.security.pgp gepostet (meistens, jedenfalls). Die Webseiten:

- **Galactus Seiten (Niederlande)**
- **Don Hensons Seite (Texas, USA)**
- **Patrick Finertys Seite (Utah, USA)**

1.1 Aufbau der FAQ

Zunächst geht es um die Zahlen, die in diesem Dokument benutzt werden. Der grösste Teil der Mathematik wurde auf einem TI-60 Taschenrechner von Texas Instruments erledigt und auf meinem PC durch ein paar Programme überprüft. Die Anzahl signifikanter Stellen (grosser Zahlen) wurde reduziert, mit Ausnahme einiger Textstellen. Eine nahezu 40-stellige Ziffernfolge wird nicht benötigt und liegt wirklich jenseits der Vorstellungskraft vieler Menschen. Zu den exakten Werten einiger der hier verwendeten grossen Zahlen, sehen Sie sich **Abschnitt 6** an. Ich verwende die Notation aus der folgenden Gleichung:

```
3.4E38 = 3.4*10^38 = 2^128
```

Das war leichter zu schreiben und spart ein wenig Platz. Beachten Sie auch, dass hier log(x) [Basis 10] verwendet wird. Da wir nur mit Exponenten arbeiten, machen Sie sich keine Gedanken darüber, ln[x] (Basis 1.718 oder e] zu benutzen, wenn Sie log[x] nicht finden.

Um auf der sicheren Seite zu sein, erfolgen Rundungen zur nächst höheren ganzen Zahl hin, wenn Sicherheitaspekte im Spiel sind. Andere Zahlen, werden einfach auf die nächste Dezimalstelle gerundet. Der Text sollte hier in der Regel eindeutig sein.

Verweise sind nummeriert und in eckigen Klammern eingeschlossen []. Unterverweisen wird ein Kleinbuchstabe hinzugefügt. Wenn das alles klar ist, wenden wir uns jetzt anderen Dingen zu.

1.2 Wie komme ich an Zufallszahlen?

Zufallszahlen sind schwer zu erzeugen. Ziemlich häufig beeinträchtigen nicht-zufällige Ereignisse die Zufälligkeit von Geräten und Schaltkreisen. Ein Vorschlag wäre 1 bis N Marken zu erzeugen und sie in einen sehr guten Mixer zu tun. Sie können versuchen, eine Münze zu werfen; ist allerdings dabei eine Person im Spiel, kann das einem Wurf bereits eine Tendenz geben, und über längere Zeiträume hinweg, das Ergebnis nach einer Seite hin ausrichten. Die Methode mit Bällen, die einige Lotterien verwenden, nutzt eine gute Zufallsquelle, aber übernehmen Sie nicht die Zahlen aus der Lotterie. Lotterienummern sind ein wenig zu offensichtlich und man kann sie leicht ausprobieren. Pool/Billiard-Bälle, die nur

einzeln aus einer Flasche genommen werden können, sind eine billige und unkomplizierte Quelle für Zufallszahlen. Ich überlasse es Ihnen, wie sie die 16 Bälle so übersetzen, dass sie für eine spezielle Anwendung taugen. Wer sich mit "Dungeons and Dragons" und anderen Rollenspielen auskennt, besitzt vielleicht schon einen Satz verschiedenartig nummerierter Würfel. Ein Punkt, der bei Würfeln Aufmerksamkeit verlangt, ist, dass das Hinzufügen von Würfeln (z. B. 2 sechsseitige Würfel) den Output in Richtung einer mittleren Zahl hin beeinflusst (7 wird einmal in 6 Würfen erziehlt) und die extremen Werte seltener auftreten (2 und 12 werden einmal in 36 Würfen erziehlt); es wird dabei einiges an Zufälligkeit eingebüsst. Achten Sie darauf, zufällige Würfel zu benutzen. Die Qualität ist manchmal nicht sehr gut und kann nicht-zufällige Resultate begünstigen. Mehr Hinweise zur Erzeugung von Zufallszahlen für Sicherheitsanwendungen, sind im RFC1750 enthalten [**6**].

Generatoren für Pseuozufallszahlen

Die Verwendung eines PRNG (engl.: pseudo-random number generator) ist in den allermeisten Fällen kein guter Weg, an Zufallszahlen zu gelangen. Das Problem mit PRNGs ist, dass die Zahlen durch eine Funktion erzeugt werden. Dazu gehöhren die Funktionen RND() aus Basic, rand() aus C oder gleichartige Zufallsfunktionen anderer Sprachen. Programmierer verwenden diese einfachen und relativ schnellen Methoden seit Jahren in Programmen und Spielen. Der Grund dafür liegt in der Arbeitsweise der PRNGs. Ein einfacher PRNG wendet Code wie den folgenden an:
R = (A * R1 + B) mod(C): R1 = R: R = R / C.
Für die Konstanten A, B und C werden normalerweise Primzahlen eingesetzt. Die meisten Sprachen sehen vor, dass ein Samenwert für R1 eingesetzt wird, bevor der PRNG aufgerufen wird, aber das ist nicht notwendig und einige PRNGs kümmern sich nicht um den konstanten Summanden B. Ein PRNG ist dadurch leicht zu umgehen, dass viele die Werte in nur 16 Bits speichern. Das bedeutet, dass ein solcher 16 Bit PRNG Schlüsselraum in 65536 * N Versuchen, durch "Brute Force" (Etwa: "Brachialangriff" - Ausprobieren aller Möglichkeiten) komplett aufgedeckt wird, wobei N die Anzahl der eingesetzten pseudo-zufälligen Elemente darstellt. Wahrscheinlich ist nahezu jedermann in der Lage, einen Standard PRNG-Schlüsselraum an einem Tag zu durchsuchen. Im ungünstigsten Fall dauert die Suche, bei Einsatz eines durchschnittlichen Heimcomputers, voraussichtlich weniger als eine Woche.

Mit Glück und wenn ein guter PRNG zur Verfügung steht, kann der Suchraum 2^32 sein, was nicht sehr viel besser ist. Beachten Sie, dass 40 Bit Schlüssel von einem Einzelnen in etwa einer Woche oder weniger durch Brute Force ermittelt werden, wenn er Zugang zu ausreichend Rechnerkapazität hat und dass der NSA 40 Bit-Schlüssel egal sind. Schauen Sie sich [**1e**] an. Dort gibt es mehr Informationen und weitere Verweise zu Zufallszahlengeneratoren.

Hardwaregeneratoren für Zufallszahlen

Hardwaregeneratoren können dazu gebracht werden, das Rauschen verschiedener Halbleiter PN-Verbindungen auszunutzen. Ein gutes Beispiel ist einfach verstärktes Rauschen einer Zenerdiode. Andere Quellen von Hintergrundrauschen sind hochohmige Widerstände und eine Anzahl kommerzieller Chips, die verschiedene Techniken anwenden, Rauschen zu erzeugen. Hardwarequellen für Zufallsinformationen erfordern Sorgfalt, weil sie durch anderes Rauschen oder nicht-zufällige Signale beeinflusst werden. Die meisten Orte sind gesättigt mit dem 50 oder 60 Hz-Rauschen des elektrischen Strohmes, Zeitsignalen, dem digitalen Rauschen von Computern, Fernsehern und Radios und einer Vielzahl weiterer elektronischer Geräte. Zur Sicherheit, sollten Sie den Output einer

Hardware-Quelle verschlüsseln oder hashen. Eine gute Hash-Funktion oder Verschlüsselung versteckt etwaig unentdeckte Muster. Ein preiswerter Zufallsbitgenerator kann für 10 US-Dollar gebaut werden [7] und ein Massenprodukt für unter 5 US-Dollar.

1.3 Wie schwierig ist es, Idea zu brechen?

PGP verwendet IDEA zur konventionellen Verschlüsselung. Der IDEA-Schlüssel ist 128 Bits lang. Wir können die Ausdehnung des Brute Force Schlüsselraumes mit 2^128 = 3.4E38 berechnen. Eine speziell auf das Ermitteln von IDEA-Schlüsseln ausgelegte Hardware, die pro Sekunde 1 Milliarde (1E9) Schlüssel erprobt, braucht 1.08E22 Jahre, um alle möglichen Schlüssel durchzuprobieren. Sie dürfen davon ausgehen, dass die meisten Schlüssel in etwa der halben Zeit, also 5.39E21 Jahren, gefunden werden. Es wird geschätzt, dass die Sonne in 1E9 Jahren zur Nova wird. Weil der Algorithmus sicher ist, muss sich die Kryptoanalyse anderen Dingen, wie RSA oder Ihrer Passphrase zuwenden. Einen IDEA-Schlüssel durch Brute Force zu brechen, übersteigt im Moment noch unsere technologischen Möglichkeiten.

1.4 Wie schwierig ist es, RSA zu brechen?

Faktorisierung ist ein einfacheres Problem als das Durchsuchen des Schlüsselraumes durch Brute Force. Zur Zeit sind die einzigen praktizierbaren Faktorisierungsmethoden für RSA *Quadratische Siebe mit mehreren Polynomen* (MPQS) und ihre Variationen, sowie das *Zahlenfeldsieb* (NFS). Schätzungen zum MPQS lauten um 3.7E9 Jahre für eine Zahl mit 200 Ziffern/664 Bits [**1d**]. Ich sollte hinzufügen, dass niemand weiss, wie lange die Faktorisierung von Zahlen über 130 Ziffern/429 Bits dauert, von einigen Spezialfällen abgesehen. Netzverweise zu bereits faktorisierten Zahlen, betreffen RSA129 und den 384 Bit "Blacknet-Schlüssel". Sie sollten beachten, dass es wesentlich weniger Zeit und Rechnerleistung beansprucht, einen 116-stelligen 384 Bit Schlüssel zu faktorisieren, als einen 129-stelligen 426 Bit Schlüssel. NFS faktorisierte einen 130-stelligen 430 Bit RSA-Schlüssel (RSA130) sogar schneller, als RSA129 faktorisiert worden war. RSA ist wahrscheinlich das schwächste Glied in PGP; allerdings kennt momentan niemand einen guten Ansatz, um ohne spezielle Hardware Zahlen über 155 Ziffern/512 Bits zu faktorisieren.

1.5 Was ist MD5?

MD5 macht aus Ihrer Passphrase einen IDEA-Schlüssel. In der Theorie sollte MD5 für jede denkbare Bit-Kombination einen anderen Output liefern, solange Ihr Schlüsselraum entweder 2^128 oder grösser ist. Der Beweis, dass MD5 alle 2^128 möglichen Ausgaben aus einem gegebenen Schlüsselraum von 2^128 generiert, ist in der Praxis unmöglich. Das entspräche dem Versuch, einen IDEA-Schlüssel durch Brute Force ermitteln zu wollen. Ein interessantes Problem stellt die theoretische Möglichkeit dar, eine äquivalente Passphrase beim Durchsuchen eines Schlüsselraumes von 2^128 oder mehr zu finden.

Vor dem Hintergrund des Angriffs auf MD5, sollten Sie abwarten und die Augen offen halten. Zwar wurde eine Schwachstelle entdeckt, derzeit wird aber die Benutzung des unmodifizierten MD5-Algorithmus noch empfohlen. Die Hinwendung zu SHA oder einer anderen Hash-Funktion für PGP ist zukünftig denkbar.

2.0 Wie lang sollte die Passphrase sein?

Als Daumenregel sollten Sie einen Buchstaben pro benötigtem Bit des Schlüssels vorsehen. Tatsächlich erhalten Sie so 1.2 Bits mit jedem englischen Textzeichen [**1c**]. Setzen wir das in Relation zur Schlüssellänge, bedeutet das, dass 128 / 1.2 = 106.667, oder 107 Buchstaben benötigt werden. Dabei wird normales Englisch angenommen, nur Kleinbuchstaben und Leerzeichen für Passphrases und, dass in Berechnungen alle Leerzeichen der Passphrase ignoriert werden. [**1a**].

Wenige von uns sind allerdings bereit, anderthalb Zeilen Text bei jedem Aufruf von PGP einzutippen. Hier versagt dann die Sicherheit und wir verlassen uns auf schwache Passphrases.

Was ist, wenn ich andere Sprachen verwende?

Die Entscheidung für Ihre Muttersprache liegt wahrscheinlich nahe. In dieser FAQ gelten alle Daten und Statistiken für englischen Text. Die Verwendung einer anderen Sprache oder Kombinationen von Sprachen verändert die Zahlen ein wenig. Ihre Passphrase ist dadurch nicht schwerer zu erraten. Ein Angriff unter Berücksichtigung einer abweichenden- oder auch mehrere Sprachen ist immernoch das gleiche. Der Suchraum entspricht grob dem Ausmass der Sprache oder wächst durch Hinzufügen der durchschnittlichen Ausdehnung des Vokabulars einer weiteren Sprache. Wörterbuchangriffe in anderen Sprachen laufen in der gleichen Weise ab, wie Wörterbuchangriffe in Englisch.

2.1 Was ist, wenn ich Sprichworte oder Zitate verwende?

Die Kurzfassung der Bewertung gewönlicher Aussprüche ist: Benutzen Sie sie niemals. Ein Zitatenschatz kann 40.000 Zitate enthalten [**5**]. Wahrscheinlich könnten Sie einen alten PC XT in eine Ecke stellen und ohne besondere Hardware Sprichworte in relativ kurzer Zeit durchprobieren. Einfache Sätze werden zu allererst überprüft. Wenn Sie ein Star Trek Fan sind, ist "Beam me up Scottie" schlecht als Passphrase zu verwenden. Sollten Sie den Satz in irgendeiner Veröffentlichung finden, benutzen Sie ihn nicht. Eine einfache Hintergrundrecherche eröffnet, für welche Musikrichtgung, Bücher, Fernsehshows, Filme, Spiele, Hobbies und was sonst noch Sie sich entscheiden würden. Bei der Suche nach Ihrer Passphrase, werden Sprichworte ganz zu Begin geprüft. Sie können 40.000 Zitate mit Hilfe einer unmodifizierten PGP-Version in 2,4 Tagen ausprobieren. Siehe Abschnitt "**Was jeder tun kann**".

Was ist, wenn ich Sätze und sinnlose Phrasen kombiniere?

Sätze zu kombinieren, verlangsamt ein wenig die Suche nach der Passphrase. Sinnloser Text bremst auch den Brute Force Angriff. [**4**]. Ein intelligenter Angriff macht sich die normale Satzstellung zu Nutze. Die Anordnung von Substantiven, Verben, Adverben, Adjektiven und aller anderen Satzteile würde zunächst der natürlichen Reihenfolge entsprechen. Ein guter, sinnloser Satz erzeugt im Falle eines Brute Force Angriffes den Anschein der Zufälligkeit, ohne tatsächlich zufällig zu sein.

2.3 Helfen ungewöhnliche Schreibweise, Zeichensetzung oder Grossschreibung?

"Un9ewönLLi5e 5chrei6wiehsen und GrOssSchrEibUng" verzögert die Suche um zusätzlich etwa 1 Million Versuche [1] pro Wort. Auf angemessene Passphrases bezogen, erreichen Sie so wahrscheinlich mehr als 8 Millionen zusätzliche Versuche (1 Million pro Wort). Zufällige Grossschreibung erzeugt wortlängenabhängige Permutationen. Hinzufügen einer einzelnen Ziffer von 0-9 zu einem Wort, vergrössert das Wörterbuch auf das Zehnfache. Das ist nur ein geringer Zugewinn, kann sich aber manchmal lohnen. Ersetzungen von E durch 3, I durch 1, S durch 5 und Z durch 2 erweitert das mögliche Alphabet um diese Zeichen. Alle Ziffern 0-9 vergrössern das Alphabet auf 36 Zeichen. Buchstabendreher, Buchstaben-Rotationen, Buchstabenverschiebungen und andere Formen der Wortzerstückelung tragen nichts zur Zufälligkeit bei, bremsen aber den Brute Force Angriff einigermassen. Sie nähern sich so einer zufällig wirkenden Passphrase an.

2.4 Was ist, wenn ich Wörter zufällig auswähle?

Ein Wörterbuch [2] soll etwa 74.000 Wörter enthalten. Bei einer Schlüssellänge von 128 Bits, benötigen wir
log(2^128) / log(74.000) = 7.91
zufällige Wörter aus unserem Wörterbuch. Aufgerundet heisst das: 8 zufällige Wörter, um die Passphrase stärker als den IDEA-Schlüssel zu machen. Ein Wörterbuchangriff dauert dann ein wenig länger als ein Brute Force Angriff auf den IDEA-Schlüssel. Das ist eine passable Methode zur Erzeugung von Passphrases, wenn man davon absieht, dass das Ergebnis manchmal schwer zu merken ist. Andererseits fällt die Eingabe recht leicht.

Kann ich ein Schlüsselwörterbuch benutzen?

Ein kleineres Wörterbuch wird viel schneller durchsucht. Alleine, dass eines herumliegt, kann Anreiz bieten, damit zu beginnen. Also stellen Sie lieber sicher, dass Ihr System zur Schlüsselerzeugung wirklich zufällig ist. Programme können kompromittiert werden, schlecht geschrieben sein oder einfach überwacht werden. Versuchen Sie es mit Diceware [8], einem guten System zur Erzeugung von Zufallszahlen. Es ist gleichgültig, ob durch irgendwelche Tricks im Wörterbuch die Anordnung von Wörtern zufälliger erfolgt. Die Grösse des Suchraumes ist alles, was am Ende zählt.
Eine unterstützende Zufallszahlenquelle ist vielleicht nicht zufällig und kann den Suchraum verkleinern. Achten Sie darauf, dass Ihr Schlüsselgenerator wirklich zufällig arbeitet.
Die Wirkung verschiedengrosser Wörterbücher soll hier aufgezeigt werden:
Bei einem Buch mit 10.000 Wörtern (aus Abschnitt 2.7), werden
`log(3.16E13) / log(10,000) = 3.37`
oder ungefähr 4 Wörter benötigt, damit die Passphrase länger als die durchschnittlichen 6 Monate hält. Um mit dem gleichen Wörterbuch eine IDEA-äquivalente Passphrase zu erzeugen, werden dagegen
`log(2^128) / log(10,000) = 9.63`
oder 10 Wörter benötigt. Bei 25.000 Einträgen im Wörterbuch, sind das
`log(2^128) / log(25,000) = 8.76`
oder 9 Wörter. Bei 50.000 Einträgen
`log(2^128) / log(50,000) = 8.20`
oder ebenfalls 9 Wörter.

2.5 Was ist, wenn ich nur zufällig Buchstaben auswähle?

Das normale Alphabet besteht aus 26 Buchstaben. Die Mathematik sagt uns, dass
`log(2^128) / log(26) = 27.23`
zufällige Buchstaben gebraucht werden. Durch Aufrunden bekommen wir 28 Buchstaben, um eine Passphrase stärker als den IDEA-Schlüssel zu machen. Sich an 28 Buchstaben zu erinnern, ist vielleicht schwierig aber nicht unmöglich. Das Eintippen fällt ebenfalls leicht.

2.6 Was ist, wenn ich ganz zufällige Zeichen verwende?

Mit allen erdenklichen ASCII-Zeichen können wir auf 95 mögliche Zeichen zurückgreifen. Führen wir beliebige Tastaturanschläge aus, benötigen wir so
`log(2^128) / log(95) = 19.48`
zufällige Zeichen. Durch Aufrunden gelangen wir auf 20 Zeichen, damit die Passphrase sicherer als der IDEA-Schlüssel ist. 20 zufällige Zeichen sind immer noch schwer zu merken, dabei auch noch schwieriger einzutippen.

2.7 Wie lange dauert ein Angriff auf die Passphrase?

Gehen wir davon aus, dass eine Maschine möglich ist, die 1 Million Schlüssel pro Sekunde ausprobiert.
Ein Pentiumchip führt mit "Piplining" [**3**] pro Zeittakt ca. 1 Prozess aus. Ein kleines Programm, kann bei minimierter Anzahl von Anweisungen auf einem 200 Mhz-Pentium pro Sekunde 1 Million mal ausgeführt werden. Der Cyrix 6x86 arbeitet bei gleicher Taktfrequenz schneller, RISC-Prozessoren erst recht. Das bedeutet, dass ein Desktop-Computer dazu gebracht wird,
`1E6 * 60 * 60 * 24 * 365.25 = 3.15576E13`
Schlüssel pro Jahr auszuprobieren, ohne dabei die heutige Technologie zu überfordern. Ein Schlüssel aus zufälligen Wörtern, muss
`log(3.16E13) / log(74,000) = 2.77`
oder 3 Worte lang sein, um im Durchscnitt länger als 6 Monate zu halten. Alle möglichen Schlüssel sind nach etwa 1 Jahr durchprobiert. Was wir am Ende vereiteln wollen, ist ein einfacher Computer-Angriff. Je intelligenter eine Maschine wird, desto langsamer wird sie auch. Wir können immer herkömmliche Hardware zusammenbauen und sie einzig als Monitor oder zur Kontrolle einsetzen.

Was jeder tun kann

Für ein Experiment wurde eine RAMDisk auf einem 486DX2-66 mit 128K Cache eingerichtet. SmartDrv, eine *unmodifizierte* Ausführung von MIT PGP 2.6.2 und alle anderen, benötigten Dateien wurden geladen. *RAM-shadows* wurden aktiviert, Grafik und BIOS waren "cacheable". Viele andere Einstellungen sorgten für die Beschleunigung aller Prozesse. Es wurde ein Programm in QBasic geschrieben (in DOS 5 und 6.x enthalten), das PGP-Passphrases aus der Umgebungsvariablen "Passphrase" ausprobieren und neue Passphrases an PGP übergeben sollte, um den ausgegebenen Beendigungscode (Errorlevel/Exit-Code) zu überprüfen. PGP wurde mit +batchmode ausgeführt.

Mit dieser Methode ist es möglich, nahezu zwei Passphrases pro Sekunde (tatsächlich 1,8125) zu prüfen. PGP gibt Pieptöne aus und pausiert, wenn Fehler auftreten; einige Einstellungen sorgten jedoch für die Minimierung dieser Ereignisse. Um einen ernstzunehmenden Angriff auszuführen, müsste PGP so modifiziert werden, dass keine Verzögerungen mehr auftreten.

Daraus wird geschlossen, dass jedermann binnen einer Stunde einzelne Wörter aus einem kleinen Wörterbuch ermitteln kann. Weit grössere Wörterbücher wären in weniger als einem Tag durchsucht. Fast jedem stehen die Werkzeuge für einen solchen Angriff zur Verfügung. Das Programm und die nötigen Einstellungen bereiten einigermassen versierten Hackern kaum Schwierigkeiten.

3.0 Wie erzeuge ich sichere Passphrases?

Die Antwort richtet sich danach, wie sicher Ihre Passphrase sein muss. Beginnen Sie mit einem normalen Satz. Verfremden Sie ihn mit Hilfe von Zufallsdaten. Erzeugen Sie einen sinnlosen Satz durch die Veränderung der Wörter. Achten Sie auf zufällig geänderte Satzstrukturen. Fügen Sie zur grösseren Sicherheit ein paar zufällige Wörter oder Buchstaben ein. Das Ziel muss sein, dass Sie sich Ihre Passphrase merken können und dass sie gleichzeitig genauso lange standhält, wie der Brute Force Angriff auf den IDEA-Schlüssel dauern würde.

Der Satz "Meine unüberwindliche super Passphrase ist unschlagbar" ist schwach. Und wenn wir ihn ein wenig verändern? "Leine runde mündliche Puter blassweiss First rutsch Nachbar" ist in Ordnung, nur, dass bei einem Angriff homophone (gleichklingende) Einträge eines Wörterbuches zugrunde gelegt werden können. Andererseits gelangen wir in einem einzigen Schritt zu einem sinnlosen Satz mit abweichender Satzstellung und Wörtern ohne logische Beziehung. Fügen Sie ein paar zufällige Buchstaben hinzu, so dass ein Erraten nur noch durch Brute Force möglich ist, und Sie sind fertig. Der Satz ist recht leicht zu merken, weil Sie ihn aus einem normalen Satz konstruiert haben. Sollten Sie den eigentlichen Satz vergessen, sind Sie wahrscheinlich in der Lage Ihn zu rekonstruieren. Als menschliche Wesen neigen wir dazu, gleiche Tätigkeiten auf vorhersehbare Weise zu erledigen.

Zur grösseren Sicherheit, können Sie auch komplett zufällige Sätze oder Buchstabenfolgen generieren. Das dauert eine Weile und kann die Erinnerung erschweren. Ihr Sicherheitsniveau lässt sich leicht durch die Begrenzung der Schlüssellänge kontrollieren. Eine nahezu idiotensichere Möglichkeit bietet Diceware [8].

4.0 Wie stark ist meine Passphrase?

Mit den Kenntnissen, die wir jetzt über absolute Minima und Maxima von Passphrases haben, können wir eine kleine Formel zur Berechnung der Stärke einer Passphrase entwickeln. Für unsere Zwecke bedeutet dabei "zufällig" richtig zufällig. Pseudo-zufällige Methoden, wie rnd() und linear-kongruente Generatoren zählen hier nicht. Die Konstanten sind auf PGP abgestimmmt. Für Ihre Bedürfnisse müssen Sie sie evtl. anpassen.

PS	Passphrase-Sicherheit
FF	Verfremdungsfaktor Dient dem Versuch, variable Einflüsse, wie sinnlose Sätze, ungewöhnliche Schreibweisen, Zeichensetzung, Grossschreibung und Ziffern, einzubeziehen.
RW	Zufällige Wörter (engl.: random words) (Zählen nicht als sinnloser Satz)
RC	Zufällige Zeichen
RL	Zufällige Buchstaben
OC	Ungewöhnliche Zeichen (Andere als Kleinbuchstaben)

LC Komplette Anzahl von Zeichen

(Buchstaben in ganzen Wörtern und Leerzeichen werden hier ignoriert)(Zählt nicht, wenn eine völlig zufälliges System angewendet wird)

Zur Beachtung: Der Verfremdungsfaktor ändert sich mit dem Aufwand, der betrieben wird.

F1 = 0.5 verknüpfte, sinnlose Sätze

F2 = ? Ungewöhnliche, falsche Schreibweise, Zeichensetzung und Grossschreibung.

Das ist eine Permutation, die von der Anzahl der veränderten Zeichen und der Länge der verwendeten Wörter abhängt. Zur Vereinfachung, verwenden Sie F2 = 4 * OC / LC

F3 = .09 Zufällige Zeichen (Fällt weg, wenn F2 eingesetzt wird.

FF = 1 + F1 + F2 + F3
PS = RW/8 + RC/20 + RL/28 + LC/107 * FF

Die Berechnung der Passphrase-Sicherheit (PS) sollte den meisten Menschen leicht fallen. Eine PS > 1 bedeutet, dass es einfacher ist, anstelle der Passphrae, den IDEA-Schlüssel anzugreifen. Eine PS <1 dagegen heisst, dass der Angriff auf die Passphrase einfacher ist. Wenn Sie schätzen, dass der PS-Wert niedriger als 0,35 liegt, kann die Passphrase in weniger als einem Jahr ermittelt werden. Die Formel ist noch in der Entwicklung und liefert lediglich Richtwerte. Es besteht die Hoffnung, dass etwaige Fehler sich zu Ungunsten der Passphrase-Sicherheit auswirken und es ist wahrscheinlich möglich, die Formel zu überlisten.

4.1 Beispiele für Passphrases

Dies sind Beispiele für Passphrases und zugehöhrige PS-Werte. Wenn Sie hier keine Verständnisschwierigkeiten haben, und auf die gleichen Werte kommen, sind Sie auf dem besten Wege, zu verstehen, wie man Passphrases stark oder schwach gestaltet.

0,855 Sinnloser Satz
`betty was smoking tires in her peace of pipe organs and playing tuna fish.`
In Deutsch z. B.: Betty rauchte Autoreifen in der Stille von Pfeifen Orgeln und spielte Tunfisch

1,05 Eine zufällige Ansammlung von Zeichen
`A6:o@6 Ls+\` uGX%3y[k`

1,34 Ungewöhnliche Zeichensetzung und Grossschreibung
`Web oF thE Trust is BrokEn cAn You Glue it Back ToGether? and give it xRays.`
In Deutsch z. B.: dAs weB oF tRust iSt gerisSeN kaNnst du eS fliCKen? uND rÖnTgen.

0,280 Ein Durchschnittlicher Satz
`There is a sucker born every minute.`
In Deutsch z. B.: Dieser Satz ist schwer zu übersetzen

1,125 Zufällige Wörter
`paper factors difference votes behind chain treaties never group`
In Deutsch z. B.: isomorph Leber ambulant Ort fressen beste zerfasert nehmen Endung

0,761 Satz mit einigen zufälligen Buchstaben
`Ignorance is bliss. spgemxk Education cures ignorance.`
In Deutsch z. B.: Es soll in keinen Sarg sich legen, ssfpoehsaod wer nur will kurz der Ruhe pflegen.

5.0 Wer sollte versuchen, an meine Passphrase zu gelangen, und wie?

Warum sollte irgendjemand Ihre Passphrase wollen? Was die meisten von uns angeht, hat niemand anders ein wirkliches Interesse an dem, was wir verschlüsseln. In der Regel ist der schlimmste "Feind", dem wir begegnen, ein Familienmitglied, das herumstochert wo es eigentlich nichts verloren hat oder vielleicht der Systemadministrator, der unseren Account betreut. Die meisten in der Familie wissen heutzutage kaum, wie ein Angriff auf eine Passphrase zu starten wäre und sogar 256 RSA-Schlüssel wären vor einer computerunerfahrenen Menge sicher. Strafverfolgungsbehörden oder Polizei sind die Angstgegner der wirklich Paranoiden oder der Aussenseiter der Gesellschaft. Wer sich bei dem was er tut auskennt, wird versuchen an die Passphrase heranzukommen, ohne einen Brute Force Angriff auszuführen.

5.1 Strafverfolgung und die Regierung

Für den Fall, dass Sie in ein Untersuchungsverfahren verwickelt werden, erwartet Sie in aller Regel das Folgende. Ihre gesamte Kommunikation wird abgehört. Reicht das so gewonnene Material, werden die Beamten Ihnen einen Durchsuchungsbefehl überreichen und mit Ihrem Computer, Disketten und wahrscheinlich einer Menge anderer Dinge unter dem Arm verschwinden. Kopien Ihrer Klartextkommunikation werden bereits vorliegen. Da man schonmal dabei ist, werden Sie verhört werden. Hat man Ihren Computer, werden Kopien angefertigt und die Festplatte durchsucht. Sollte einiges oder Alles oder verschlüsselt sein, wird die Entschlüsselung inklusive vermeindlich gelöschter Dateien versucht. Sollte sich Ihre Passphrase irgendwo auf der Platte befinden, erlangt man Zugang zu allen an Sie verschlüsselten Dateien. Die Strafverfolgungsbehörden haben ihre eigenen Computer-Experten und rekrutieren bei Bedarf professionelle Hilfskräfte. Ihre individuellen Erfahrungen können, abhängig vom Land in dem Sie leben, variieren.

5.2 Multitasking Systeme

Sie können Windows 3.x, Windows9x/NT, OS/2 oder anderen Betriebssystemen nicht vertrauen, wenn sie Hauptspeicher auf die Festplatte auslagern oder Virtuellen Speicher belegen. Auf Macintosh-Rechnern wird der Inhalt des RAMLaufwerkes vielleicht automatisch auf die Festplatte gespeichert. Einige Windows-Benutzer haben Ihre Passphrase in der Auslagerungsdatei entdeckt. Es sollte sicher sein, PGP unter Windows im DOS-Modus auszuführen, solange Windows inaktiv ist-; anders ausgedrückt: KEINE DOS-Fenster. Windows-Programme, die auf DOS zurückgreifen, schreiben offenbar die Passphrase in die Auslagerungsdatei. Es existieren einige Programme, die gesamte Plattenoberfläche absuchen und das nach kaum mehr als einem Mausklick. Es verursacht auch kaum Aufwand, ein kleines Programm zu schreiben, das Dateien nach Text-Fragmenten absucht. Für ernstere Angriffe und gelöschte Dateien wird einer der vielen Dienstleister in Anspruch genommen, die die Datenrettung von unlesbaren Laufwerken anbieten. Das Hauptproblem bei Multitasking Systemen ist die Kontrolle. Sie können einfach nicht effektiv kontrollieren, was mit den Daten im Speicher geschieht.

5.3 Mehrbenutzer-Systeme

Auf grösseren Mehrbenutzer-Systemen fällt es jemandem mit genügend Zugang leicht, Schnüffel-Programme zu installieren, Dateien zu kopieren und manchmal sogar, einen Benutzer direkt zu überwachen. Beziehen Sie Netzwerk-Rechner ein. In einem Netzwerk können Sie mit der richtigen Software Abläufe fernsteuern. Manche Netzwerk-Software ist sogar bereits mit Programmen zur begrenzten Überwachung ausgestattet. Verwenden Sie Ihren Computer auf der Arbeit, kann das bedeuten, dass Sie Ihre Passphrase einer Anzahl von Leuten eröffnen. Viele versuchen das zu umgehen, indem für das Mehrbenutzer-System ein separater Schlüssel verwendet wird und daheim ein anderer.

5.4 Tempest und elektronische Überwachung

Es ist bekannt, dass das elektronische Rauschen von Computern überwacht und benutzt werden kann. Jedes Kabel agiert als Antenne und strahlt die Signale, die es übermittelt, nach aussen ab. Der verzwickteste Teil ist das Auffinden des richtigen Computers in der Menge vieler identischer anderer. Sollte es nur einen Rechner geben, fällt die Spionagearbeit leicht. Manchmal ist es viel leichter, ein Zimmer abzuschirmen, als speziell abgeschirmte Geräte anzuschaffen. Am Schwierigsten ist es, die Sicherheitslöcher zu finden und zu stopfen. Jedes Kabel, das in einen Raum hineinführt, kann Signale nach draussen befördern, egal, wie die Isolation beschaffen ist. Sie müssen einer mächtigen Regierung oder Firma schon sehr wichtig sein, bevor Sie sich über Tempest-Angriffe Sorgen machen müssen. Tests mit wirklich grundlegender Ausstattung haben ergeben, dass vom Monitor eine Menge Rauschen ausging, von einem Rechner mit Stahlabdeckung dagegen sehr wenig, einiges von der Tastatur. Alle Kabel, die zum Test dienten, waren abgeschirmt, der Computer führte keine Prozesse aus, eine Reihe von Daten wurde auf dem Schirm angezeigt. Die Abhöreinrichtung war nicht sehr sensibel, daher könnte die Abstrahlung intensiver gewesen sein, als tatsächlidh nachgewiesen.

6.0 Wie lege ich meine Passphrase(s) sicher ab?

Die beste Methode ist wahrscheinlich das Aufspalten eines Schlüssels. Sie müssen Teile einer Passpharse verteilen, die alle Ihre regulären Passphrases sichert. Einige Wege führen zum Ziel und schützen Ihre Schlüssel selbst dann, wenn Sie ein paar Freunde verlieren. Zum Beispiel teilen Sie die Passphrase in 3 Teile. Dann übergeben Sie die Teile an 6 verschiedene Freunde. Um Ihre Passphrase zu rekonstruieren, brauchen Sie nur drei Ihrer Freunde und haben Sicherungskopien. Gehen Sie genauso mit der tatsaechlichen Schlüssel-Datei vor. Ihre Freunde können einzeln weder Ihre Passphrase wiederherstellen, noch die Stücke zusammensetzen, ohne dass alle drei kooperieren. Die Sicherheit erhöht sich, wenn Sie mehr Leute einbinden, aber das Wichtigste bleibt, dass Sie Kopien Ihrer Schlüssel so verbreiten, dass nur Sie und niemand anders Sie rekonstruieren kann. Mindestens eine Kopie von PGP und Ihrer Schlüssel sollten Sie ausserhalb Ihres Hauses aufbewahren. Vermindern Sie Ihre Risiken. Siehe [1] für mehr Informationen zum Schlüsselaufspalten u.a.

6.1 Soll ich meine Passphrase niederschreiben?

Ich werde mir jetzt wiedersprechen. Die totale Sicherheit gebietet, dass Sie Ihre Passphrase niemals, nirgends und in keiner Form notieren. Die oben beschriebene Technik der Aufspaltung von Schlüsseln ist nicht unfehlbar.

Das Niederschreiben von Passphrases wirkt der Sicherheit entgegen, wenn nicht sorgsam vorgegangen wird. Gewöhnliches Wegwerfen in verschiedener Form händigt Ihre Passphrase an jeden Beobachter aus. Aufschreiben mit einem gewöhnlichen Stift lässt es zu, Ihre Passphrase von einer Papierunterlage zu lesen, nachdem das oberste Blatt, auf dem geschrieben wurde, entfernt worden ist. Die Kopie Ihrer Passphrase in den Abfall zu werfen, liefert sie der Müllabfuhr aus. Auch der Hausmüll kann ohne grosse Mühen durchsucht werden. Eine Brieftasche ist kein guter Aufbewahrungsort, wenn Sie verletzt oder bestohlen werden. Geschriebenes verursacht i. A. noch viele andere Probleme.

7.0 Wie wende ich den Passphrase-Schalter oder die Passphrase-Variable an?

Es wird empfohlen, beide Methoden nicht anzuwenden. Der Grund ist, dass daraus eine riesige Sicherheitslücke resultiert, wenn Sie nicht extrem aufpassen. Verkehrte Anwendung und gewöhnliche Fehler führen zu einer Gefährdung durch Wörterbuchangriffe oder liefern Ihre Passphrase gleich an den Angreifer aus. Überprüfen Sie PGP sorgfältig im manuellen Testbetrieb, um sicher zu gehen, dass Ihre Batch-Routine richtig arbeitet, bevor Sie sie auf sensible Daten anwenden.

Das System, für das dieser Abschnitt hauptsächlich gedacht ist, ist ein MSDOS PC. UNIX, Mac und andere unterscheiden sich. Es geht hier darum, die möglichen Risiken aufzuzeigen. Sie sollten unbedingt die PGP-Anleitung und die Handbücher zum Betriebssystem lesen, bevor Sie diese Methoden anwenden. Auch das genügt manchmal nicht. Einige Texte sind ziemlich obskur oder enthalten schlicht nicht die nötigen Informationen.

7.1 Setzen der Umgebungsvariablen PGPPASS

Viele Menschen haben einige gute und einige schlechte Methoden erarbeitet, um die Sicherheitsrisiken im Zusammenhang mit PGPPASS zu mindern. Meiner Art, sichere Batch-Routinen auszuführen, dient das Setzen einer Dummy-Passphrase, um in der Autoexec.BAT mehr Umgebungsspeicher zu reservieren, als benötigt wird. Wenn Sie nicht genügend Speicher vorsehen, können Sie eine entsprechende Fehlermeldung erhalten. Im Anschluss verändert das Batchprogramm, gewöhnlich QBASIC, die Umgebungseinstellung anhand von Benutzereingaben. Der Prozess wird ausgeführt und dann wird PGPPASS wieder auf den alten Füllwert zurückgesetzt. Die Sicherheit besteht darin, dass alles im Speicher überschrieben wird. Die Passphrase wird niemals auf die Platte geschrieben.

7.2 Verwendung des -z Schalters

Der Kommandozeilenschalter dient der Bequemlichkeit einiger Benutzer und der Einbindung in Batch-Routinen. Unter MSDOS ist Ihre Kommandozeile auf maximal 128 Zeichen begrenzt. Eine gute Passphrase kann über 80 Zeichen lang sein und schränkt die Nützlichkeit des Verfahrens ein. Enthält sie ausserdem Leerzeichen, bekommen Sie nur das erste Wort bis zum ersten Leerzeichen unter, wenn Sie keine Anführungszeichen

setzen. Viele gelangten zu der Erkenntnis, dass ihre "perfekte" Passphrase komplett nutzlos ist, wenn PGP nur das erste Wort daraus erkennt.

7.3 Passphrases in Batchdateien

Die beste Empfehlung, die man geben kann ist: Lassen sie es sein. Wenn die Batchdatei entdeckt wird, hat man Ihre Passphrase. Das Verfahren abzusichern ist ziemlich kompliziert. Sezten Sie in der Autoexec.BAT eine Dummy-Passphrase. Jetzt fordern Sie den Benutzer durch die Batchdatei dazu auf, die Passphrase einzugeben, setzen Sie die echte Passphrase, führen Sie PGP-Kommandos aus, überschreiben Sie die Passphrase und beenden Sie die Batchdatei. Stellen Sie immer sicher, dass die echte Passphrse überschrieben wird, bevor sie die Batch-Routine abschliessen. Bei Passphrases, die Leerzeichen enthalten, achten Sie auf die Anführungsstriche und testen Sie alles.

8.0 Copyright, Zahlen, Glossar und was sonst noch fehlt

8.1 Die richtig grossen und andere Zahlen

Hier soll deutlich werden, wie gross diese Zahlen tatsächlich sind. Es ist schwierig, mit ihnen zu arbeiten und es besteht dazu kaum ein Anlass, ausser dem Versuch die Massstäbe zu begreifen. Sie brauchen mehr als einen Taschenrechner, um in dieser Form zu arbeiten. Beachten Sie die Länge von 2^128. Das ist die Grössenordnung einer 128 Bit Zahl. Ein 512 Bit Modulus ist etwa viermal so lang.

1 million	=	1.000.000
1 billion	=	1.000.000.000
1 trillion	=	1.000.000.000.000
3.15576E13	=	31.557.600.000.000
2^128	=	340.282.366.920.938.463.463.374.607.431.768.211.456
74.000^8	=	899.194.740.203.776.000.000.000.000.000.000.000.000
95^20	=	3.584.859.224.085.422.343.574.104.404.449.462.890.625
26^28	=	4.161.536.836.220.038.342.098.551.818.958.537.752.576

Das sind die log(x)-Werte, die in der FAQ verwendet wurden. Auch wenn Sie die Bedeutung von log(x) nicht verstehen, sollte es der Vereinfachung der Mathematik dienen.

log(2^128)	=	38.53183945
log(3.16E13)	=	13.49910397
log(74.000)	=	4.86923172
log(50.000)	=	4.69897004

log(25.000)	=	4.397940009
log(10.000)	=	4.0
log(95)	=	1.977723605
log(26)	=	1.414973348

8.2 Glossar

Hier sollen ein paar Dinge klargestellt werden, für den Fall, dass sie aus dem Zusammenhang nicht hervorgehen. Weitere Definitionen werden bei Bedarf aufgenommen.

Angreifer (engl.: Attacker)
: Jeder, der sich Zugriff auf Ihre Passphrase verschaffen will. Es könnte Ihr kleiner Bruder oder die Schwester sein, Gattin, Freunde, ein Hacker aus der Nachbarschaft, Strafverfolgungsbehörden und viele andere.

Brute Force Angriff (dt. etwa: Brachialangriff)
: Das Durchsuchen des gesamten Schlüsselraumes. Jede mögliche Kombination wird nach und nach ausprobiert. Zahlenschlösser mit drei Ziffern haben einen Schlüsselraum von 1.000 und können so überwunden werden (000, 001, 002... 999).

Schlüsselgrösse
: Die tatsächliche Grösse des Schlüssels. IDEA benutzt eine Schlüsselgrösse von 128 Bits

Schlüsselraum (engl.: key space)
: Die Anzahl möglicher Schlüssel. Die Berechnung des Schlüsselraumes kann kompliziert sein, wenn andere Angriffe als das Ausprobieren aller Möglichkeiten denkbar sind. IDEA hat einen Schlüsselraum von 2^128.

Pseudo-zufällig (engl.: pseudo random)
: Eine mathematische Reihe oder andere wiederholbare Abfolge, die den Anschein der Zufälligkeit erweckt.

zufällig (engl.: Random)
: Eine Reihe, die nur mit Hilfe von Aufzeichnungen reproduziert werden kann.

Suchraum (engl.: search space)
: Das Ausmass der Suche, die zum Brechen eines Schlüssels nötig ist. Manchmal haben Schlüssel einen kleineren Suchraum, als es die Schlüsselgrösse vorgibt. Eine feste 40-stellige/130 Bit Zahl (Spielzeug-RSA) schafft einen grösseren Schlüsselraum als der 39-stellige Schlüssel von IDEA, kann aber mit Hilfe einer schnellen Faktorisierungsmethode in ein paar Minuten (oder weniger) ermittelt werden.

9.0 Text-Verweise

Es existieren weitere Bücher, die hier aufgenommen werden könnten, Statistiken und Werke zur Wahrscheinlichkeitsrechnung. Es kann auch sein, dass ich Referenzwerke verpasst habe. Wegen seiner enzyklopädischen Natur habe ich, anstelle einer langen Bibliographie, innerhalb des Dokumentes häufig auf [1] verwiesen.

[1] Bruce Schneier, Applied Cryptography. John Wiley & Sons, 1994

	(erste Taschenbuchausgabe)	(zweite Taschenbuchausgabe)
[1a]	S. 144-5 und S. 190-1	S. 173-5 und S. 234
[1b]	S. 141-3	S. 170-3
[1c]	S. 190 (1.2 Bits Shannon)	S. 234 (1.3 Bits Cover)
[1d]	S. 212	S. 256

[1e] S. 347 Kapitel 15 S. 369 Kapitel 16
[2] The Random House Dictionary. Ballantine Books, 1980
Taschenbuch, etwa 3.8 cm dick mit "über 74.000 Einträgen".
[3] Nick Stam, Inside the Chips. PC Magazine Feb. 21, 1995
S. 190-199
[4] Grady Ward, Creating Passphrases From Shocking Nonsense
[5] The Oxford Dictionary of Quotations. ???
("über 40.000 Zitate" - aus einer Werbeanzeige)
[6] RFC1750 Randomness Recommendations For Security
Eine Quelle ist: **http://www.clark.net/pub/cme/html/ranno.html**
[7] Randall T. Williams, A Simple Random Noise Source, July 01, 1995
Ein Posting in sci.crypt und alt.security.pgp, September und Oktober 1995
[8] Arnold Reinhold, Diceware (A Passphrase generation system)
http://world.std.com/~reinhold/diceware.html

Nürnberg, 03. März 1999 **Michael Uplawski**

Erzeugt in **Arachnophilia** zur Musik von Jim Croce