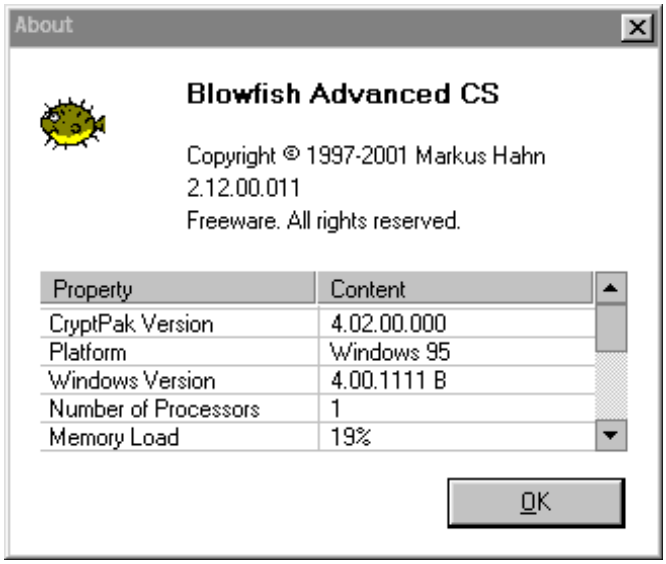
Blowfish Advanced CS

Version 2.12.00.011

Lasse J. Kolb, Mai 2001
http://www.bsn.ch/Lasse

Einleitung



Alle hier beschriebenen Informationen beziehen sich auf die Programmversion 2.12.00.011. Blowfish Advanced CS (bfaCS) wurde von Markus Hahn (*mailto:markus_hahn@gmx.net)* programmiert.
Die Software läuft unter Windows 9x, NT als auch Windows 2000. Linux, MacOS, OS/2 u. a. bleiben außen vor, jedoch gibt es gerade unter Linux viele gute Verschlüsselungsprogramme.
Auch für Windows tummeln sich viele derartiger Programme auf Share- oder Freewareseiten, jedoch stelle ich häufig deren Qualität – insbesondere bezüglich der Sicherheit – in Frage. Unter Unix stellt dieses meist kein Problem dar, weil die Programme normalerweise im Sourcecode verfügbar sind und sich somit auf ihre Sicherheit überprüfen lassen. Bei Windows-Kryptoprogrammen ist meist kein Sourcecode mitgeliefert – anders bei Blowfish Advanced CS: Hier ist der Programm-Sourcecode verfügbar (Open Source Software, kurz: OSS).
Die Programm-Hilfedatei bietet nützliche Hinweise zur Passwortwahl sowie Informationen zur verwendeten Zufallszahlen-Generierung.

Um es kurz zu machen: Blowfish Advanced CS ist eines der besten Verschlüsselungs-Programme, das mir für die Windows-Plattform untergekommen ist.
Deshalb möchte ich hier einen kurzen Überblick über die wichtigsten Funktionen von bfaCS geben.

Seit dem 3. April 2000 ist Blowfish Advanced CS *Freeware*!

Verschlüsseln von Dateien

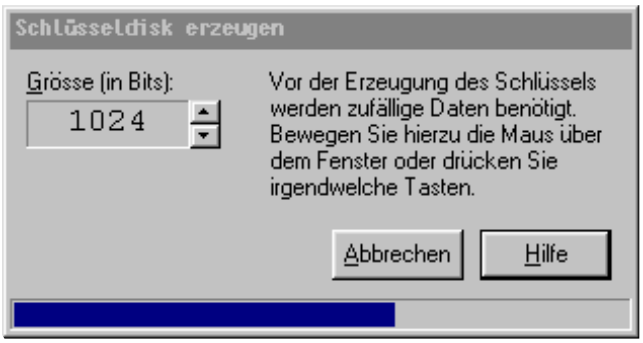
Die wichtigste Funktion von bfaCS ist die Verschlüsselung von Dateien.
Diese können nahezu beliebig groß sein! Selbst Dateien > 4 GB sind kein Problem.
In der aktuellen Version von bfaCS kann zwischen neun verschiedenen Verschlüsselungsalgorithmen gewählt werden (Blowfish, Twofish, IDEA, triple-DES u. a.).
Die Verschlüsselung erfolgt immer symmetrisch. Das bedeutet, es wird mit ein und demselbem Passwort ver- und entschlüsselt (Im Gegensatz zu asymmetrischen Programmen wie z. B. PGP).
Wer Schwierigkeiten hat sich Passwörter zu merken, der kann sich Passwörter generieren lassen, und diese auf einem Datenträger – z. B. einer Diskette – abspeichern, und diese beim Ent- bzw. Verschlüsseln einlegen. bfaCS kann Passwörter mit einer Länge von bis zu 1024 bit erzeugen.
Derartige Passwörter können als sicher angesehen werden, da sie viele Sonderzeichen enthalten, die sich vermutlich kein Mensch merken könnte. Das erschwert das Passwortknacken erheblich, weil das Passwort so nicht zu erraten ist, und praktisch nur durch ausprobieren herauszubekommen ist. Wir gehen von einem 1024 bit-Passwort aus, also 128 Buchstaben lang. Auf der Tastatur sind 256 verschieden Zeichen verfügbar. Wenn wir nun 256^{128} rechnen und davon ausgehen, dass das Passwort nach der Hälfte der möglichen Kombinationen geknackt wird … Nunja, ich glaube es wird deutlich, dass sich Ottonormalverbraucher um seine verschlüsselten Daten keine Sorgen zu machen braucht. Bei Blowfish Advanced CS sind sie in guten Händen.
Weiterhin besteht die Möglichkeit, die Daten beim Verschlüsseln zu komprimieren.
Dies ist weniger zur Platzersparnis gedacht; vielmehr wird dadurch das Dateientschlüsseln weiter erschwert. Als Kompressionsmethode kommt hierbei das LZSS-Verfahren zum Einsatz, welches in etwa der Kompressionsstärke „Super-Fast-Modus“ bei ZIP-Kompression entspricht.
Dateinamen lassen sich während der Verschlüsselung verschleiern. Der richtige Dateiname wird hierbei verschlüsselt im Dateikopf gespeichert und beim Entschlüssen wiederhergestellt. So lässt sich anhand des Dateinamens nicht auf den Inhalt schließen – ein meiner Meinung nach sehr sinnvolles Feature.
Nützlich ist auch die Schlüsselteilung, bei der sich ein Schlüssel bzw. Passwort in mehrere Teile teilen lässt. So kann man einen Teil z. B. auf Diskette speichern, den anderen merkt man sich. Nun können Dateien nur noch entschlüsselt werden, wenn beide Teile des Schlüssels korrekt eingegeben wurden. Man kann so z. B. das Passwort auf mehrere Personen aufteilen, d. h. jeder kennt einen Teil des Schlüssels, und die Daten lassen sich nur entschlüsseln, wenn jede Person anwesend ist. So werden Manipulation von Daten durch einzelne Personen ausgeschlossen, weil niemand alleine die Daten entschlüsseln kann.
Auf die restlichen Features gehe ich hier nicht näher ein, sie stehen in der Anleitung von bfaCS.

Entschlüsseln von Dateien

Wie bereits beschrieben, handelt es sich bei bfaCS nur um symmetrische Verschlüsselungsverfahren.
So gibt es über das Entschlüsseln eigentlich nicht mehr viel zu schreiben, weil es im Prinzip so funktioniert wie das Verschüsseln – nur umgekehrt eben.
Wenn mit mehrfachen Schlüsseln gearbeitet wurde, so müssen diese in der Reihenfolge angegeben werden, in der auch verschlüsselt wurde:
Wenn Sie beispielsweise mit einer Keydisk (Erster Schlüsselteil) und einem Passwort (Zweiter Schlüsselteil) verschlüsselt haben, so muss in genau dieser Reihenfolge auch entschlüsselt werden. Andernfalls gibt bfaCS eine Fehlermeldung aus, dass der Schlüssel (Passwort) nicht stimmt und verweigert die Entschlüsselung. Oder besser, *eine mögliche Entschlüsselung.* Einigen anderen Kryptoprogrammen ist es nämlich egal, welchen Schlüssel man zum Entschlüsseln verwendet; sie entwirren die Daten mit dem ihnen eingegebenen Schlüssel – ob dieser nun stimmt oder nicht. Wenn man nun die vermeintlich korrekt entschlüsselte Datei laden will, funktioniert sie nicht. In einem solchen Fall muss man die Datei mit dem falschen Passwort wieder verschlüsseln (sofern man weiß, an welcher Stelle man sich vertippt hat), um sie dann mit dem korrekten Passwort wieder zu entschlüsseln. Das führt schnell zu Verwirrung, so dass man am Ende oft nicht mehr durchblickt und die Datei nicht mehr korrekt entschlüsseln kann.

Ein solcher Fall kann mit Blowfish Advanced CS zum Glück *nicht* passieren! Don't worry :-)

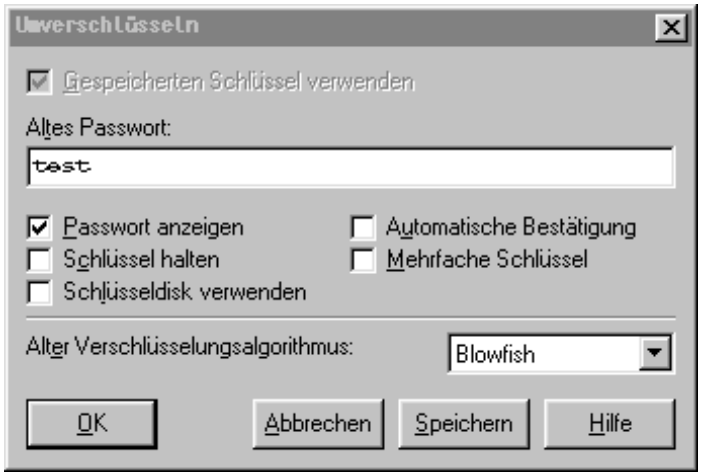
Schlüsseldisk erzeugen

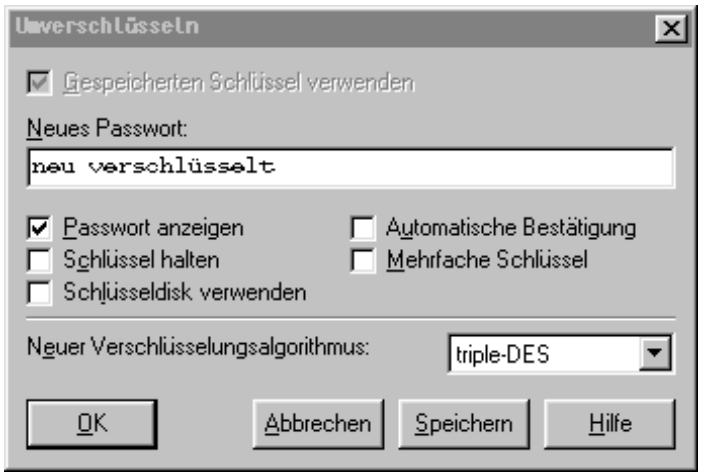


Passwörter lassen sich auf externen Datenträgern speichern.
Als Passwort-Datei kann jede beliebige Datei verwendet werden. Jedoch sollte man bedenken, dass Dateien mehrfach vorhanden sein können. Wählt man also eine Datei, die sowieso Bestandteil von jedem Windows-System ist, so ist diese Datei auch auf anderen Systemen vorhanden, welches das Passwortknacken erleichtern kann. Deshalb ist es sehr wichtig, Dateien zu verwenden, von denen es möglichst wenig gibt, oder die gar einzigartig sind.
Deshalb hat bfaCS einen Generator eingebaut, der genau solche Dateien erzeugt – Zufallsdateien. Doch was ist schon echter Zufall?
Über dieses Thema kann man stundenlang diskutieren, und einige Leute haben dieses u. a. getan. Deshalb verlässt sich bfaCS auf den wohl besten Zufallsgenerator, den es zur Zeit gibt: Yarrow. Wer's nicht glaubt, kann es unter http://www.counterpane.com nachlesen.

Umverschlüsseln von Dateien

Diese Funktion habe ich bisher bei noch keinem anderen Programm gesehen.



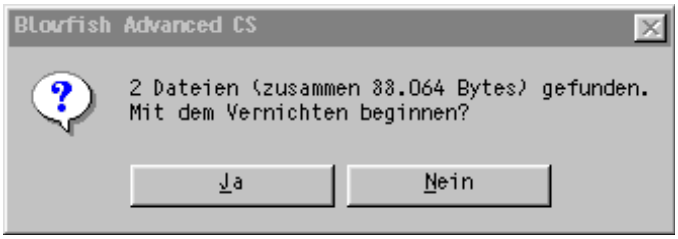


Was ist mit „Umverschlüsseln“ gemeint?

Diese Funktion bezieht sich auf den Algorithmus, mit dem verschlüsselt wurde. Normalerweise braucht man eine Datei nicht umzuverschlüsseln. Das ist nur dann sinnvoll, wenn eine Methode bekannt wird, diesen oder jenen Algorithmus mit geringem Aufwand zu brechen. Sei es aufgrund einem neu erforschten mathematischen Zusammenhang, oder einer bisher nicht bemerkten Schwachstelle im Algorithmus.
In einem solchen Fall wäre es wünschenswert, wenn man den Algorithmus einfach gegen einen sichereren austauschen könnte. Genau das ist mit bfaCS problemlos realisierbar.
Möglich wird dies durch den schichtweisen Aufbau der verschlüsselten Dateien.
Um Daten umverschlüsseln zu können, benötigt man das Passwort, welches zur Ver/Entschlüsselung verwendet wurde. Mit wenigen Mausklicken wird der alte, unsichere Algorithmus gegen einen neuen ausgetauscht.

Einfacher geht es wirklich nicht.

Vernichten von Dateien

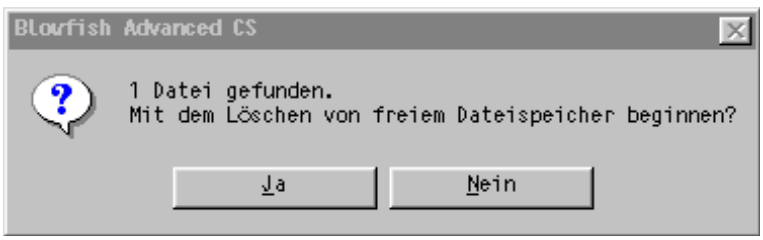


bfaCS beschränkt sich nicht nur auf das Verschlüsseln von Dateien, sondern es kann auch Dateien unwiederuflich löschen.
Dazu werden die Daten mehrfach überschrieben, um ein soft- und hardwaremässiges Wiederherstellen zu erschweren oder gar unmöglich zu machen.
Die Optionen reichen vom einfachen Löschen (welches dem normalen Löschen unter Windows entspricht) bis hin zu 35fachem Überschreiben von Daten. (SFS-Methode)
Die Vernichtungsoption funktioniert mit jedem Dateisystem (FAT12, FAT16, FAT32, NTFS, etc.)
Die hier eingestellte Methode (standardmässig DOD, also 3faches Überschreiben) wird auch verwendet, wenn der Ver- bzw. Entschlüsselungsvorgang von Dateien abgeschlossen ist, um die Originaldaten zu überschreiben. Es bringt nichts, Dateien zu verschlüsseln, wenn man danach die (unverschlüsselte) Originaldatei wiederherstellen kann.

Freien Speicheplatz löschen

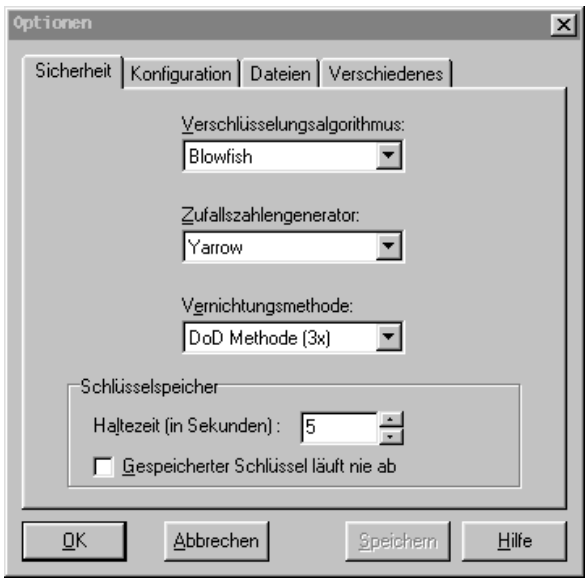
Die Idee, freien Speicherplatz auf der Festplatte oder einem anderen Datenträger zu löschen, scheint zunächst etwas merkwürdig. Denn wo freier Speicherplatz ist, ist ja nichts zum Löschen vorhanden.
Das ist nicht richtig. Da Daten ja nicht zusammenhängend auf der Festplatte gespeichert werden, sondern an vielen verschiedenen Stellen, befinden sich überall Teile einer Datei. Wenn nun eine Datei verschlüsselt wurde, (und die Original-Datei auch noch mehrfach überschrieben wurde) ist das zwar schön und gut, jedoch ist nicht auszuschließen, dass sich die Datei-Teile früher an einer anderen Stelle auf der Festplatte befunden haben.
Deshalb sollte der freie Speicherplatz ebenfalls überschrieben werden, weil sich sensible Dateien von dort eventuell wiederherstellen lassen. Dafür bietet bfaCS verschiedene Optionen an:
Zum einen kann gewählt werden, auf welchen Laufwerken gelöscht werden soll.
Weiterhin gibt es die Möglichkeit, den Windows-eigenen Cache zu umgehen, um den Überschreib-Vorgang zu beschleunigen. (Genauere Informationen stehen in der Programmdokumentation). Neben zwei weiteren Funktionen, auf die ich hier nicht näher eingehe, bleibt noch die Methode, mit der Überschrieben wird. Das geschicht entweder durch Nullen, eine bestimmte Zufallszahl, oder durch kontinuierlich wechselnde Zufallszahlen. (Siehe bfaCS-Hilfe)

Neben dem freiem Festplattenplatz lässt sich zusätzlich noch der freie Dateispeicherplatz, also der Speicherplatz *innerhalb* einer Datei, löschen. Sinn und Zweck ist auch hier, Daten vor dem Wiederherstellen zu schützen.



Programm-Optionen

Da dieser Text nur einen kurzen Überblick über die wichtigsten Programmfunktionen geben soll, werde ich an dieser Stelle nicht auf die Programm-Optionen eingegen. Das würde den Rahmen sprengen. Deshalb verweise ich an dieser Stelle nur auf die Progamm-Hilfe.
Einige Screenshots aus dem Programm habe ich unten verkleinert abgebildet.



Anhang

Markus Hahn / Key ID: 0xAA68473A / Key type: DSS/Diffie-Hellman / mailto:Markus_Hahn@gmx.net
Size: 1024/2048 / Fingerprint: 89A0 69B7 FC5C 0423 16D3 40E0 26F9 9854 AA68 473A
Homepage: http://come.to/Hahn

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 5.0i for non-commercial use

mQGiBDizGAIRBADKIDjc9vBVsEEjjEIrnihjj6scGBmB6Px56IcIVlNZPORNgJMy
A6FpZbWjIATFCJh6xs2Etq8Ity4gK+6lEBHJy+3w9wjOvXm5hne5RxAVjanzzUyi
BdMXY/ed6aMAFsMUNtS9SuRSvV8FqBGudVIxZ3GBAcuBQ4+cvwdJt0QNNwCg/xZi
pXEEZQa44rM8iWngU+Teae8D/0L569TY3fjxpJ12oRfdOlFmo3JGEmsdcjY0jdnM
g2IgXT7E8znNkLJ3AX4Gw+NNxyxV9SQPBsjjCNtCcw/pI7CEg8YbnWwM2ojhwNrK
UDihIVKJE5nyhQK+yAW9gyz/qLaxrPwsw67MEME9hYkwve2O/k1av+WBaUk1KqJG
NGAbA/wNa6nce1VTDl9FVzwXIcfK9nq9f5T+C9IluAt7TGs/yUvjYu8YMTPLHIQw
g2MIS5Lm+XXD61DHHC/RJdaQ1ekcRl7cbx9ldWSXAbhZtCNQlEOG2VssxYngLl6G
S07uvvr+9bRKNOBKN9AcxtbRa5jJBGsEAOYuwyM+iVDogOBIHbQhTWFya3VzIEhh
aG4gPG1hcmt1c19oYWhuQGdteC5uZXQ+iQBLBBARAgALBQI4sxgCBAsDAgEACgkQ
JvmYVKpoRzrkfgCfWqCPrUdWYurcEh8RlQLr9GFRr+oAoP6IqCYKatN8z+cMkJYd
6LvJeaPPuQINBDizGAMQCAD2Qle3CH8IF3KiutapQvMF6PlTETlPtvFuuUs4INoB
p1ajFOmPQFXz0AfGy0OplK33TGSGSfgMg71l6RfUodNQ+PVZX9x2Uk89PY3bzpnh
V5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7HAarTW56NoKVyOtQa8L9GAFgr
5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfIizHHxbLY7288kjwEPwpVsYjY67VYy4
XTjTNP18F1dDox0YbN4zISy1Kv884bEpQBgRjXyEpwpy1obEAxnIByl6ypUM2Zaf
q9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1XpMgs7AAICCACeGNtu3G5dJZmp
z5RFNUtEbwwf2XwP6sQWXtZCFK+5/KeRYHqr34tC5aU8e6JJol8CYQpbiVn9HZVS
LUzjlR4MhgOdFiwGShyd9fqE2FDKzpjhgV0eIhTksZ2jfzEsvmBa+nJ85Gk7WA2l
52WmsM7phAGXnD8pEn8LuklS0s1u02l4wOXBn4A2Xo/7nVM48nq4uC/ljYezt95/
jOqQEyipKfLbgevBT6GGiChuRR8R0JvsGNwwynrp1y84jANEbAmGPGClFGOvEprH
X0GiusJH2YGYIKEMfoxWkFKUVjCYNPzCnEjwR8b8FH8Cgm8z54yRDEfrv/Vrjoma
Fw37Kd74iQBGBBgRAgAGBQI4sxgDAAoJECb5mFSqaEc6DUEAoM/LGC/jY5OI6Y+2
nkHq7ED2Y6fpAJ94ZvS6iG7iyUfGMFQQeAm0xYDfHw==
=NF4S
-----END PGP PUBLIC KEY BLOCK-----
```

Lasse J. Kolb / Key ID: 0x4A1802C9 / Key type: RSA / mailto:Lasse@bsn.ch
Size: 2048 / Fingerprint: 9734 403C 9B59 9BC5 E83C 54FE 4992 576B /
Homepage: http://www.bsn.ch/Lasse (DSS/Diffie-Hellman-Key ebenfalls dort zu finden)

```
Type Bits/KeyID    Date       User ID
pub  2048/4A1802C9 2000/09/19 Lasse J. Kolb <Lasse@bsn.ch>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i

mQENAznHou8AAAEIAMmQDFiLfoE9pvgh9hr7wmmdbPgwyweg7NFp8xU/WCa6UcVM
8nn28OPQEJ5nXh0A2icUpYf/6ncyd/U1QFozR+Rxu0AO5AxySum0DMZnxa0BUPqN
gNWQcwg5/FhIl18qLs926LROpQqkILLlH1SR3MP8OZ1vEaHSzp3au73Z9HXSEyFS
00cB2z4Dt3kk1Sv9ATh4CCEO6QTszJYck6N6Ww/jZrPUzjg6ELvFfw7fuoYxjso0
2TnTglf0Pt8q2Lwnb+HAWAJlIu7f51/JECMk22k6D8ATydrhbPybUFDDnhjmLI3m
/IJYkQKaCfpnl9LI8dXEiyA1dvaeYHSpJUoYAskABRG0HExhc3NlIEouIEtvbGIg
PExhc3NlQGJzbi5jaD6JARUDBRA5x6LvYHSpJUoYAskBATy7B/0YDhWy5kxK8jHB
xmUZcGiraWNfucCQBKTopMmxOexaL9d2Bn47PleToL/tLaqiwifJZQpaBTwWdKof
/HyYd0Mlq0SzWe/mS+PPQR/qLMY7CoxWWbRaNAyp2sa9F7i0c69Dy+dUo4BFcby4
wYCtJNQJzecHF5N6UYJrBr0IkF0BenZwZTZ7r1ixb0TQuMJmDqQLv2B7qGdEdtW/
WXWYMD/PPdlcdHEqQPvNi2WpYD0YC5QTIFWNGb4diQKLim0gKvN4Vazthm1TzLy2
MnBFcIY74eXKIvOYCv4jCjohneLqRIO2f8yklkXJ8ahUBdhP0JeATC9hbON/N731
FgfmlOx9
=NHs0
-----END PGP PUBLIC KEY BLOCK-----
```