Aufgabenstellung SN-Labor Versuch 3: PGP

Vor Beginn des Versuchs richten Sie den Internetzugang und Emailaccount wie beim letzten Versuch ein.

Ab sofort sind alle Protokolle mit dem Schlüssel „Praktikum“ verschlüsselt abzugeben. Bei Veränderungen von Dateien, Texten oder Signaturen überprüfen Sie das Ergebnis. Die Aufgaben 21-25 sind zu Hause zu beantworten.

1. Erstellen Sie sich einen eigenen Schlüssel.
2. Lassen Sie sich von Ihrer Nachbargruppe deren Schlüssel geben und fügen Sie die Schlüssel zu einem Schlüsselbund zusammen.
3. Setzen Sie alle öffentlichen Schlüssel, die Sie bekommen haben, auf „trusted“.
4. Verschlüsseln Sie einen Text mit Ihrem Schlüssel und entschlüsseln Sie diesen anschließend.
5. Verschlüsseln Sie einen Text für Ihren Nachbarn und senden Sie ihn per Email.
6. Verschlüsseln Sie einen Text. Anschließend verschlüsseln Sie den Original-Text noch mal und vergleichen die beiden Chiffrate.
7. Verschlüsseln und Signieren Sie einen Text für Ihren Nachbarn und senden Sie ihn per Email.
8. Signieren Sie einen Text und prüfen die Signatur anschließend.
9. Verschlüsseln Sie eine Datei.
10. Signieren Sie eine Datei.
11. Versuchen Sie eine Datei zu entschlüsseln, für die Sie keinen privaten Schlüssel besitzen.
12. Verändern Sie einen verschlüsselten Text.
13. Verändern Sie einen signierten Text.
14. Verändern Sie die Signatur eines Textes.
15. Exportieren Sie einen Schlüssel, verändern diesen und versuchen ihn zu importieren.
16. Verschlüsseln Sie eine Datei im ASCII-Format und verändern diese.
17. Verändern Sie die Signatur einer Datei
18. Signieren Sie eine Datei und verändern nachträglich die Datei
19. Erstellen Sie eine bootfähige Diskette mit Norton Ghost.
20. Erzeugen Sie ein Image der E-Partition ihres Rechners mit Norton Ghost auf Partition D. (Bitte beachten, dass Norton Ghost bei der Zielpartitionsauswahl nur FAT32 angibt!) Wo liegt das Image jetzt? Vergleichen Sie die Größe der Platte mit der Datei. (Spielen Sie das Image aber nicht wieder auf!)
21. Was passiert, wenn nun Daten auf der Partition E verändert werden und anschließend das Image zurückgespielt wird?
22. Finden Sie heraus, wie viele Kombinationsmöglichkeiten es bei einem Passwort von 4 und 8 Zeichen gibt. (Voraussetzung: verwendete Zeichen: 92 (deutsches Tastaturlayout))
23. Erläutern Sie den Ablauf der digitalen Signatur.
24. Zu welchen Sicherheitsproblemen (Virenscanner, Firewall) kann es kommen, wenn Nachrichten/Daten verschlüsselt gesendet werden?
25. Signieren bedeutet mehr, als eine digitale Unterschrift leisten. Finden Sie heraus, was Signieren noch bedeutet.

Zuerst haben wir, wie in der Aufgabenstellung verlangt, Internet und Email konfiguriert!

1. Erstellen Sie sich einen eigenen Schlüssel.

Um sich in PGP einen eigenen Schlüssel zu erstellen, startet man einfach den Assistenten (Schlüssel oben links) von „PGPkeys“ (s. Abbildung 1).

Abbildung 1 – PGPkeys und Assistent

Um einen öffentlichen Schlüssel zu erstellen, muss man seinen Namen und die Emailadresse eingeben (s. Abbildung 2).

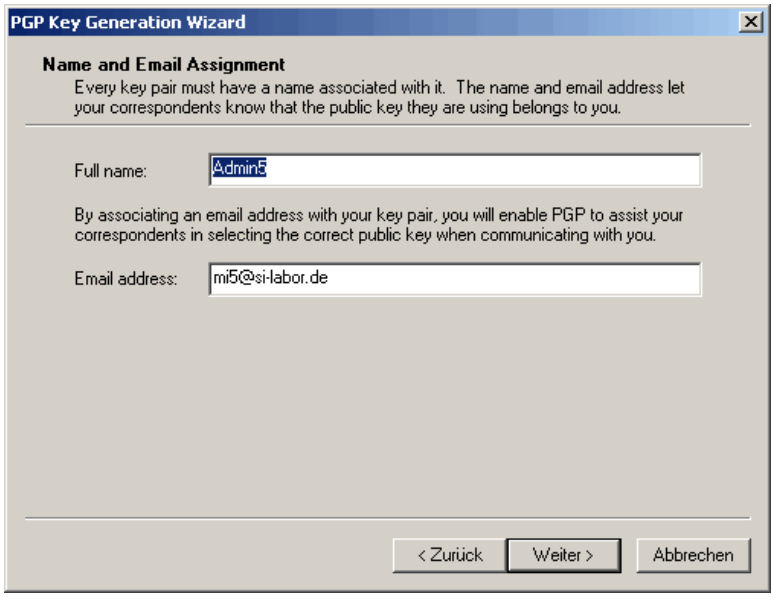


Abbildung 2 – Öffentlicher Schlüssel

Der nächste Schritt ist die Eingabe der „Passphrase“ für die Verschlüsselung unseres privaten Schlüssels (s. Abbildung 3). Die Eingabe des Passworts ist sehr wichtig und sollte nicht aufgeschrieben werden, um eine Entschlüsselung von dritten zu vermeiden.
Zur Verdeutlichung haben wir allerdings für die Screenshots in diesem Protokoll die Option „Hide Typing“ immer ausgeschaltet. Hätten wir die Option standardmässig aktiviert gelassen, so hätte sich zwar der Cursor im Textfeld bei der Eingabe nach rechts bewegt, man hätte allerdings statt Text nur Leerzeichen gesehen.
Auch die Länge des Passwortes ist entscheidend, es sollte ein so langes Passwort gewählt werden, bis der Statusbalken von „Passphrase Quality“ komplett ausgefüllt ist!

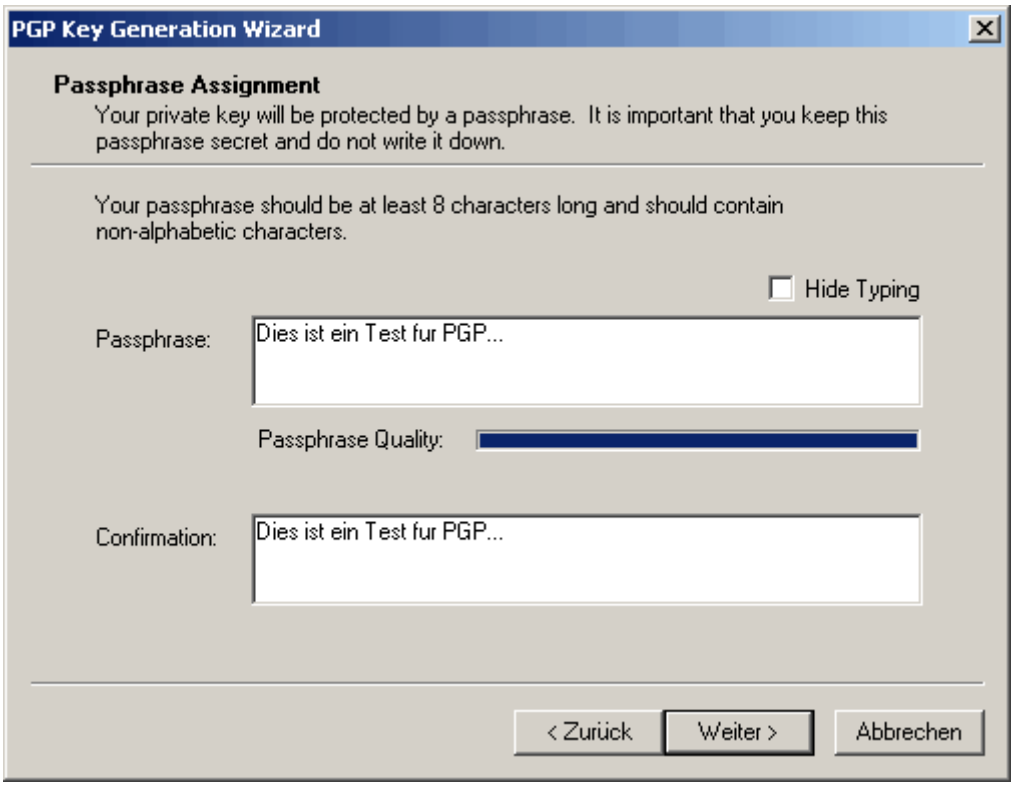
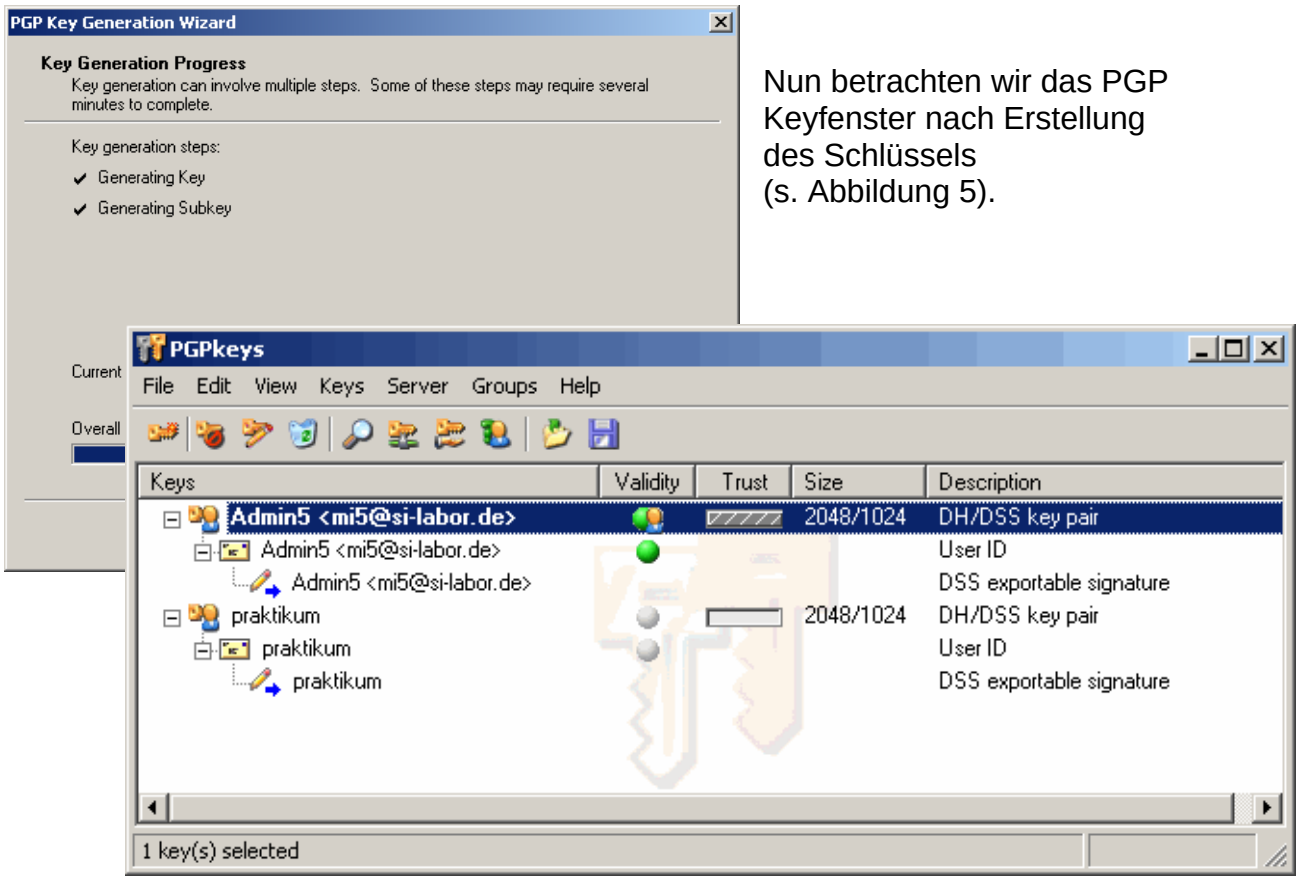


Abbildung 3 – Privater Schlüssel

Anschließend generiert der PC den Schlüssel und man ist am Ende der Schlüsselerstellung angelangt (s. Abbildung 4).

Abbildung 4 – Generierung Schlüssel und Ende Schlüsselerstellung



Nun betrachten wir das PGP Keyfenster nach Erstellung des Schlüssels (s. Abbildung 5).

Abbildung 5 – PGP Keyfenster

2. Lassen Sie sich von Ihrer Nachbargruppe deren Schlüssel geben und fügen Sie die Schlüssel zu einem Schlüsselbund zusammen.

Unsere Nachbargruppe (MI4) hat uns Ihren Schlüssel per Email zugeschickt. Die angehängte Datei wurde von uns geöffnet und in PGP importiert (s. Abbildung 6).

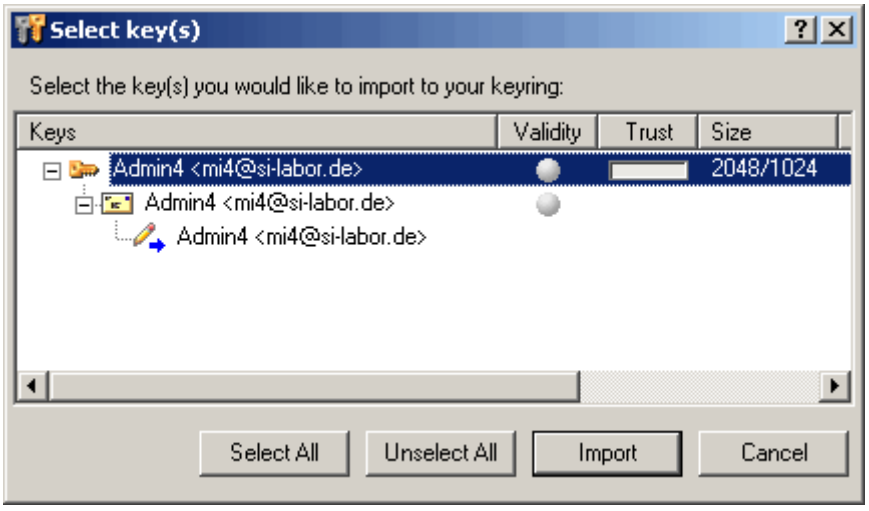


Abbildung 6 – Key importieren

Durch den Import des Schlüssels haben wir unseren Schlüsselbund um den Schlüssel der Gruppe MI4 erweitert (s. Abbildung 7).

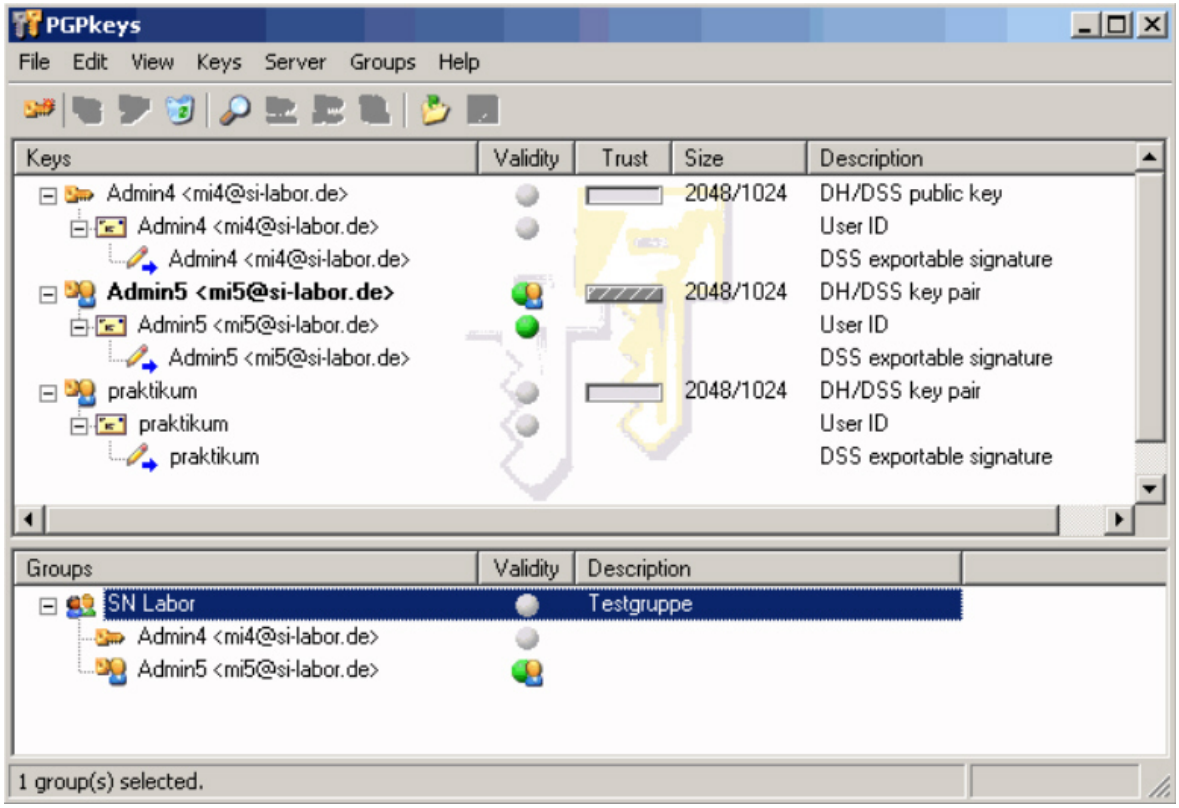


Abbildung 7 – Verfügbare Schlüssel (Schlüsselbund)

Anschließend haben auch wir unseren Schlüssel an die Nachbargruppe per Email versendet. Was durch einen Rechtsklick auf unseren Schlüssel über das Kontextmenü geschieht (s. Abbildung 8).

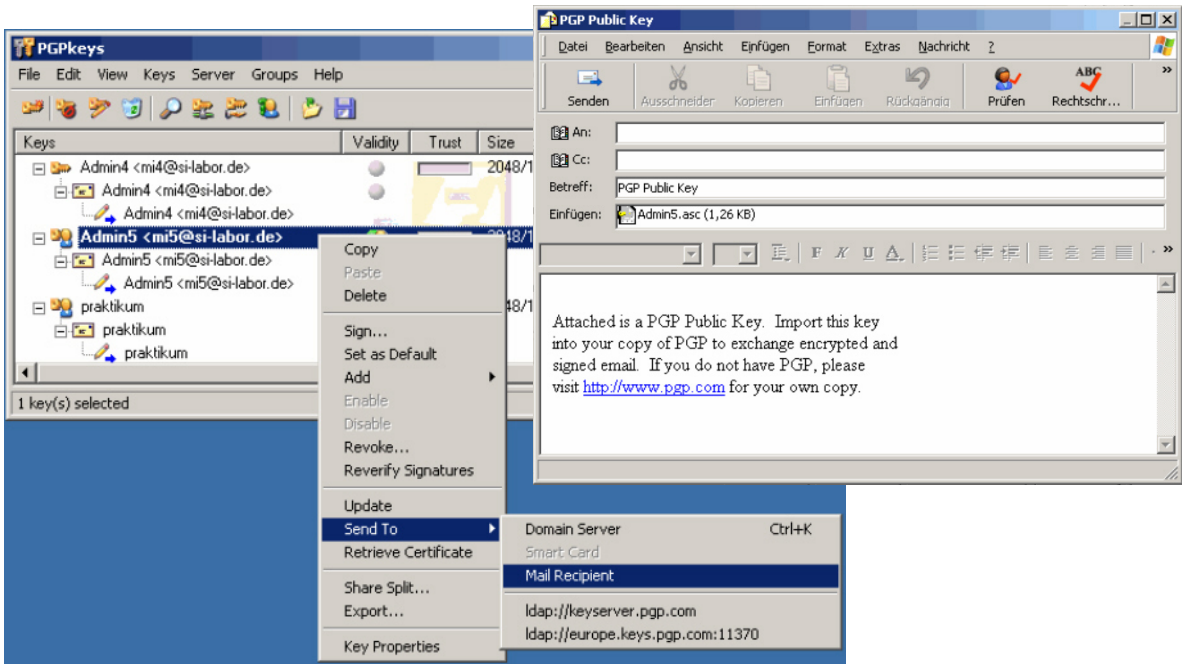


Abbildung 8 – Schlüssel per Email versenden

3. Setzen Sie alle öffentlichen Schlüssel, die Sie bekommen haben, auf „trusted“.

Durch einen Rechtsklick auf den Schlüssel wählt man im Kontextmenü den Punkt „Key Properties“ aus, um den Key auf „trusted“ setzen zu können (s. Abbildung 9).

Dies war allerdings nicht möglich!

Abbildung 9a – Fehlermeldung

Wir müssen zunächst den Schlüssel signieren, durch den Eintrag „Sign“ aus dem Kontextmenü, worauf auch eine Eingabe unseres Schlüssel notwendig war.

Dieser Regler, hat folgenden Hintergrund:

Wie sehr man den Unterschriften eines bestimmten Benutzers vertraut, kann man in den Eigenschaften seines Schlüssels im Schlüsselbund einstellen. Der Trusted-Schieber ist dafür da.

Wenn man diesen Schieber auf „Trusted“ stellt und dann von irgendwoher einen Schlüssel erhält, der mit diesem „getrusteten“ Schlüssel unterschrieben wurde, dann wird der neue Schlüssel auch ohne Unterschrift auf „valid“ gesetzt – es gibt ja eine glaubwürdige Bestätigung der Echtheit, nämlich die Unterschrift mit dem Schlüssel, dem man vertraut!

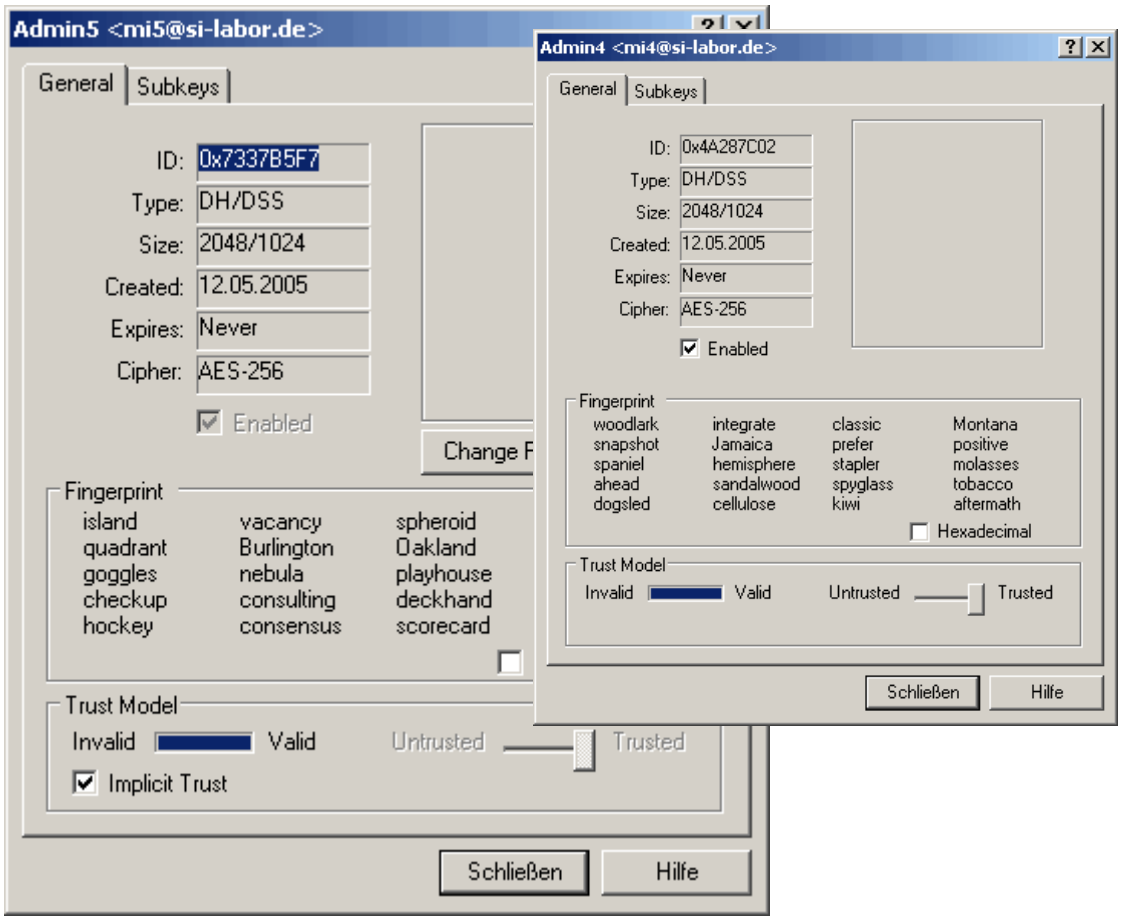


Abbildung 9b – Schlüssel „vertrauen“

4. Verschlüsseln Sie einen Text mit Ihrem Schlüssel und entschlüsseln Sie diesen anschließend.

Der mit dem Editor erstellte Text wird mit einem Rechtsklick, über das Kontextmenü PGP > Encrypt, verschlüsselt (s. Abbildung 10).

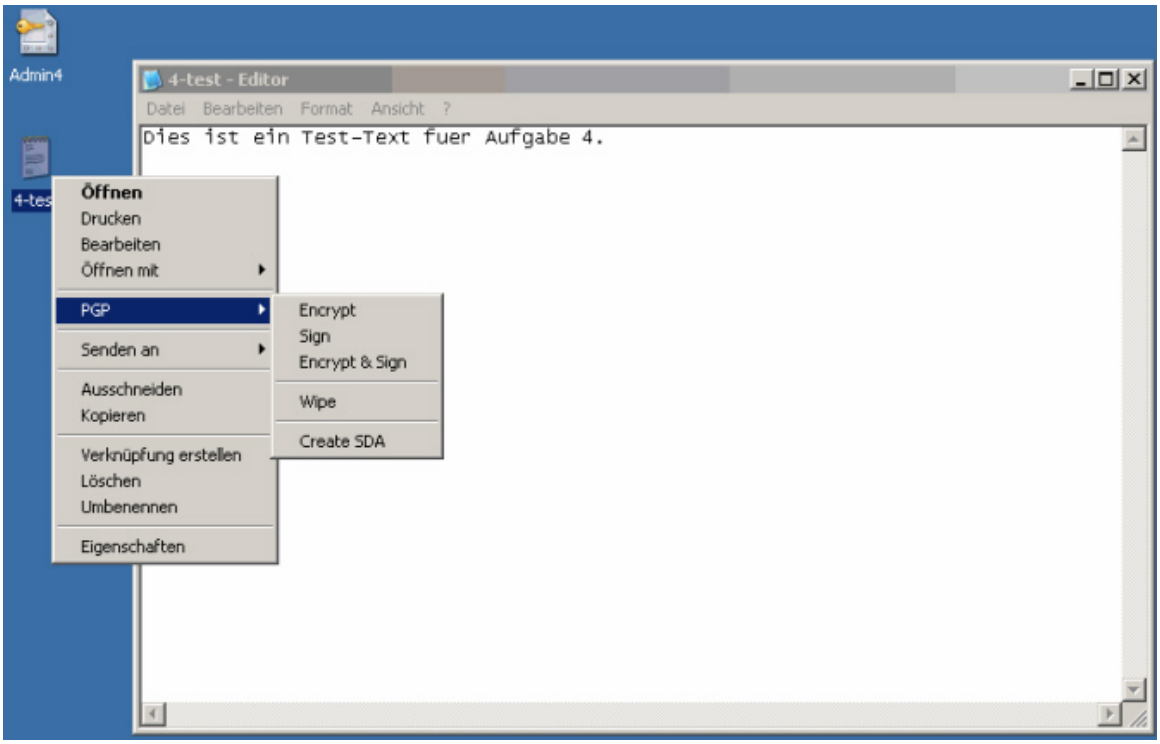


Abbildung 10 – Text verschlüsseln

Nun gelangt man zu der Schlüsselabfrage zum verschlüsseln. Hier wählt man die „Recipients“-Liste der Empfänger aus. In diesem Fall ist es Admin5, wir sind selbst der Empfänger. Auch sollten die beiden Häkchen gesetzt werden, wie im Bild zu sehen, da wir eine Text-Datei verschlüsseln.
Aus dem nun erscheinenden Fenster zur Wahl der zu verwendenden Schlüssel wählen wir nur unseren eigenen aus (s. Abbildung 11).

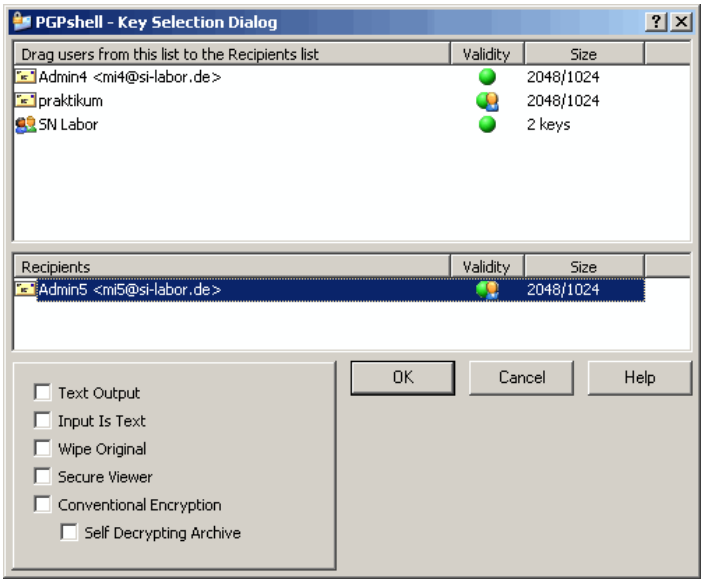


Abbildung 11 – Empfänger Liste und Text

Die Datei wurde verschlüsselt und wie man auf der unteren Abbildung erkennen kann, erscheint es ziemlich schwer den verschlüsselten Text zu knacken (s. Abbildung 12).

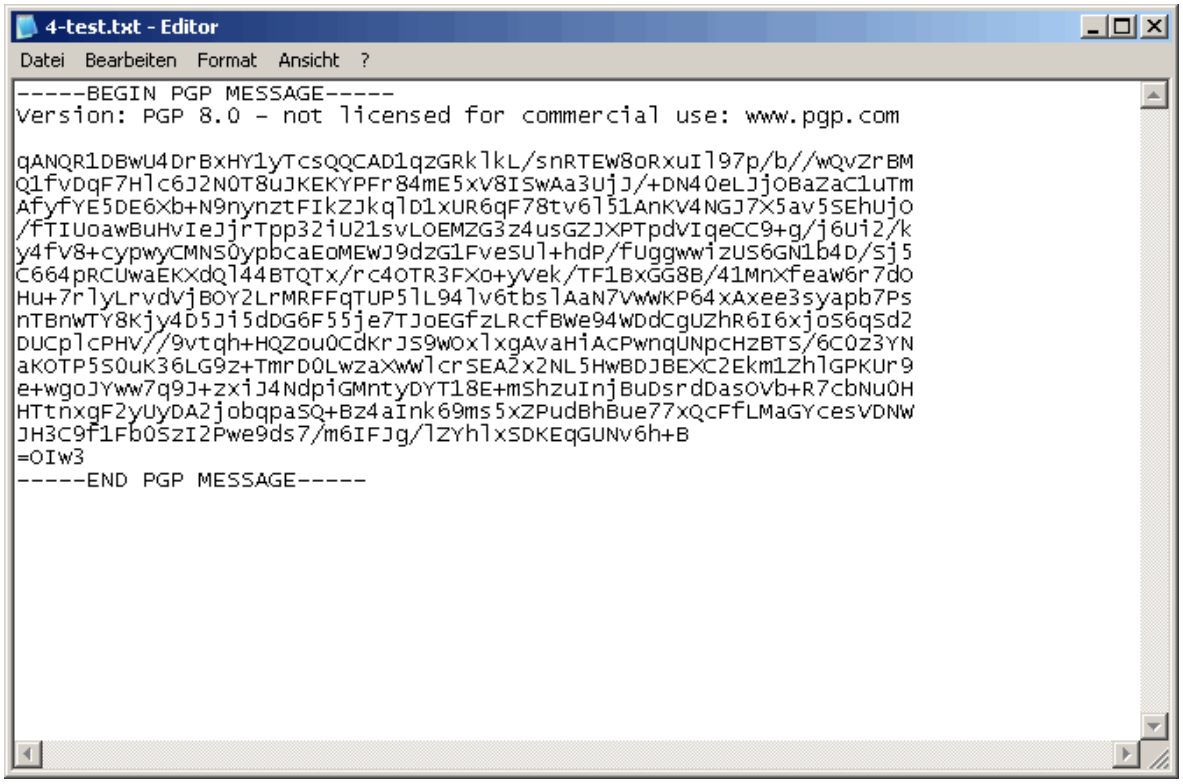


Abbildung 12 – Ansicht verschlüsselte Datei

Um die verschlüsselte Datei wieder zu entschlüsseln, genügt ein Doppelklick auf die Datei und das zuvor eingegebene Passwort für den privaten Schlüssel (s. Abbildung 13).

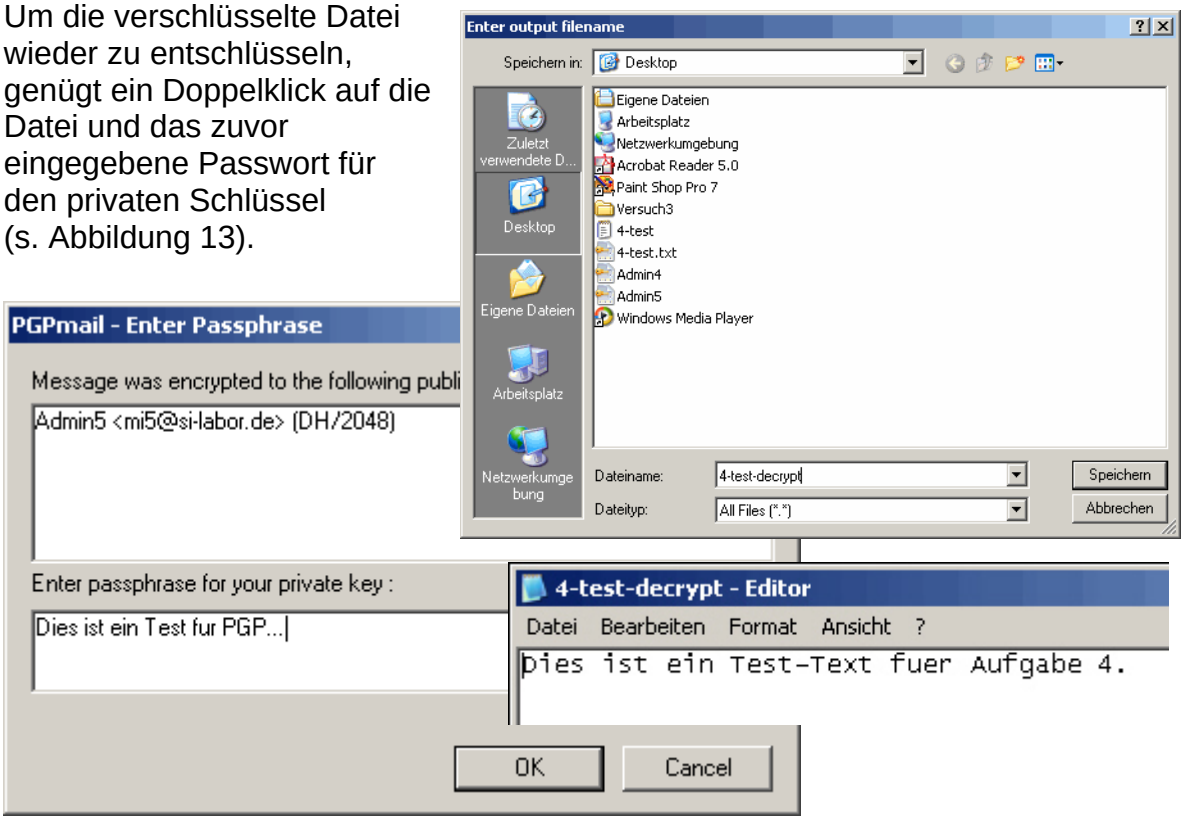


Abbildung 13 – Ansicht verschlüsselte Datei

5. Verschlüsseln Sie einen Text für Ihren Nachbarn und senden Sie ihn per Email.

Der Text für unsere Nachbargruppe wird wie zuvor über das Kontextmenü verschlüsselt. Allerdings muss jetzt bei der Schlüsselabfrage zum Verschlüsseln eine kleine wichtige Änderung vorgenommen werden. In die Empfänger-Liste (Recipients) wird jetzt der öffentliche Schlüssel der Nachbargruppe (Admin4) eingefügt (s. Abbildung 14). Nur so kann unsere Nachbargruppe auch die verschlüsselte Nachricht entschlüsseln.

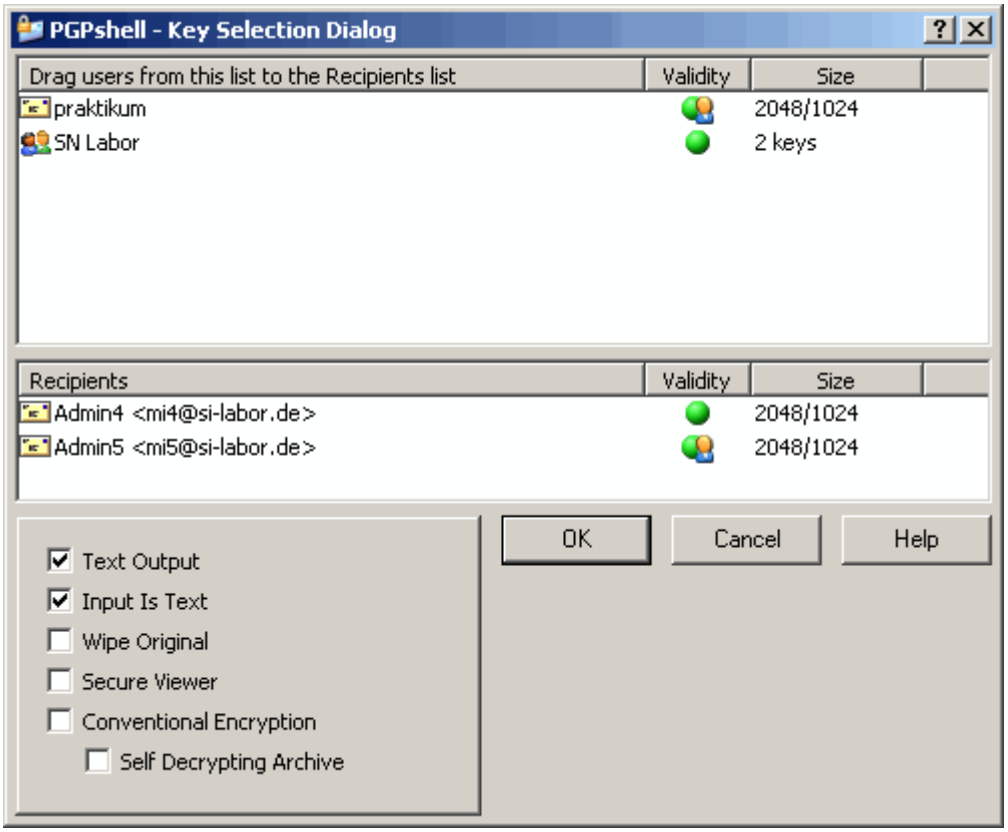


Abbildung 14 – Empfänger Liste mit Nachbargruppe

Anschließend wird die Datei per Email versendet (s. Abbildung 15).

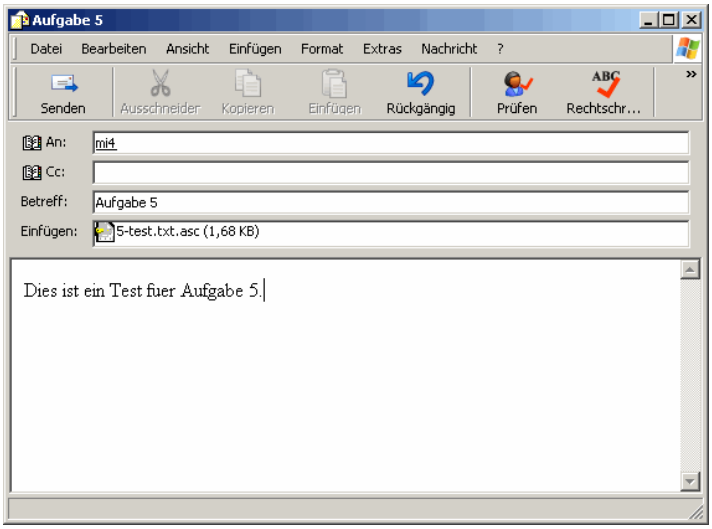


Abbildung 15 – Email versenden

Auch wir haben einen mit unserem öffentlichen Key verschlüsselten Text von der Nachbargruppe per Email erhalten, welchen wir mit PGP wieder entschlüsseln. Wir müssen wieder unser zuvor erstelltes Passwort (den privaten key) eingeben (s. Abbildung 16).

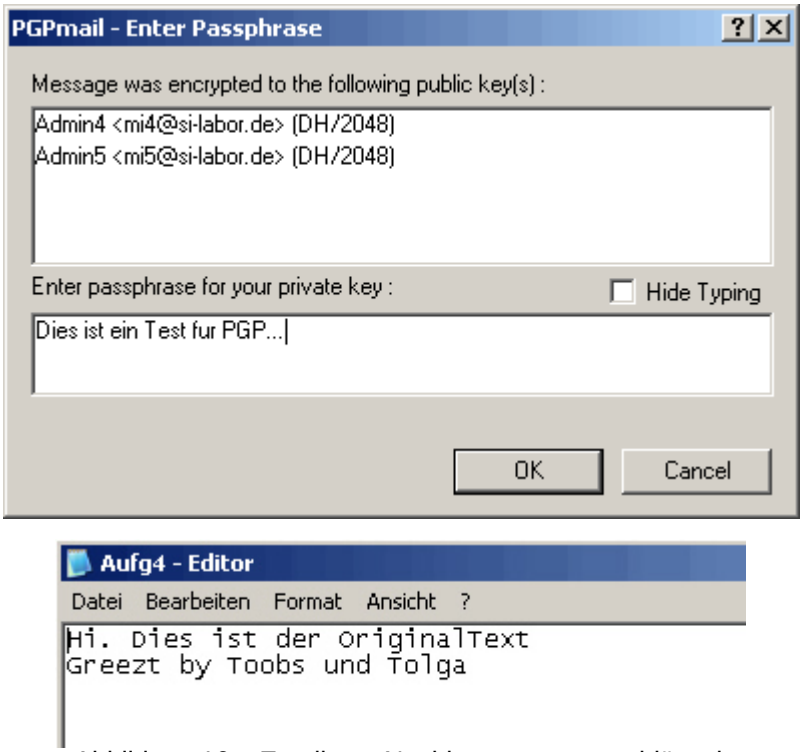


Abbildung 16 – Email von Nachbargruppe entschlüsseln.

6. Verschlüsseln Sie einen Text. Anschließend verschlüsseln Sie den Original-Text noch mal und vergleichen die beiden Chiffrate.

Unser Text (siehe Abbildung 17) wird wie schon in den Aufgaben 4 und 5 über das Kontextmenü PGP > Encrypt verschlüsselt. Danach wird derselbe Text ein zweites Mal verschlüsselt. Beim Vergleich der Chiffrate fällt auf, dass die Schlüssel in den ersten 24 Stellen („qANQR1DBwU4DrBxHY1yTcsQQ“, siehe Abbildung 18, blau hinterlegt) des Schlüssels identisch sind.

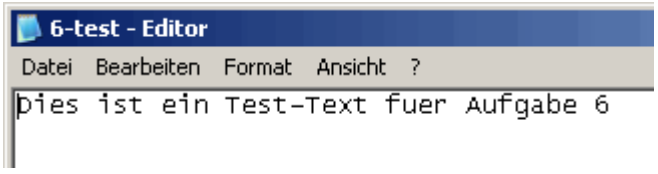


Abbildung 17 – Beispieltext

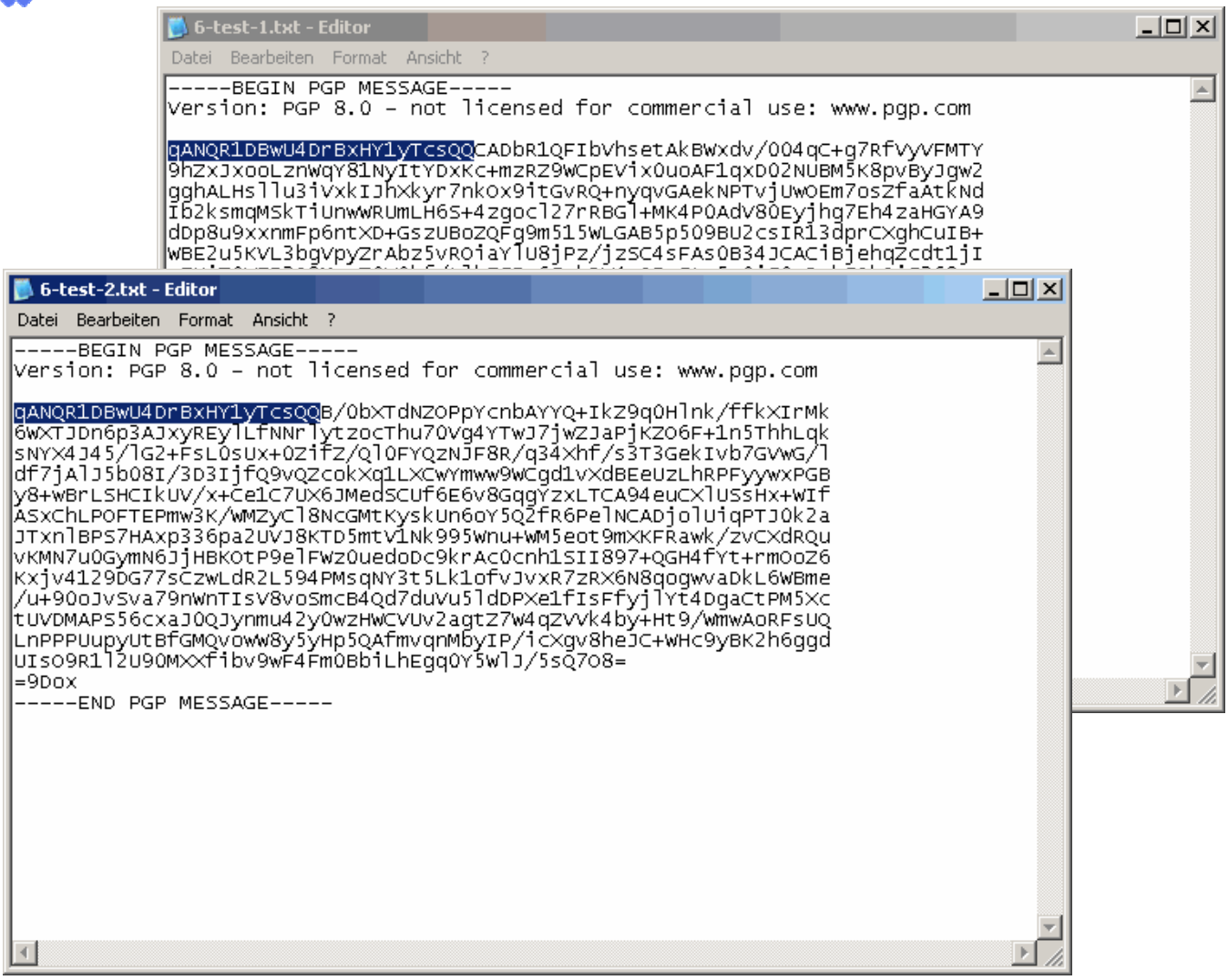


Abbildung 18 – zwei Mal verschlüsselter Beispieltext

7. Verschlüsseln und Signieren Sie einen Text für Ihren Nachbarn und senden Sie ihn per Email.

Über das Kontextmenü (s. Abbildung 20) der Textdatei ist es möglich, eine Datei (s. Abbildung 19) nicht nur zu verschlüsseln sondern noch dazu auch zu signieren (PGP > Encrypt and Sign). Wie schon in Aufgabe 5 fügen wir zum Verschlüsseln den öffentlichen Schlüssel der Nachbargruppe ein (s. Abbildung 21). Zum Signieren muss nun allerdings im Unterschied zum einfachen Verschlüsseln noch die eigene Passphrase (private Key) eingegeben werden (s. Abbildung 22). Im Anschluss wird die Datei per Email an die Nachbargruppe verschickt.

Abbildung 20 – Verschlüsseln und Signieren

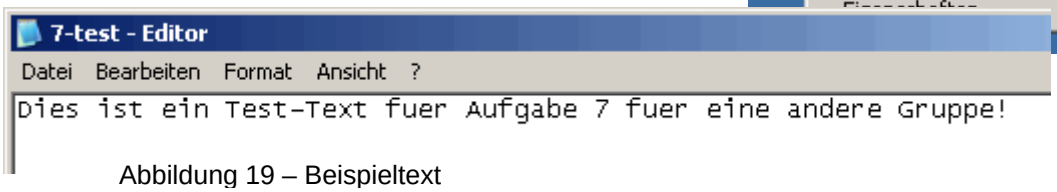


Abbildung 19 – Beispieltext

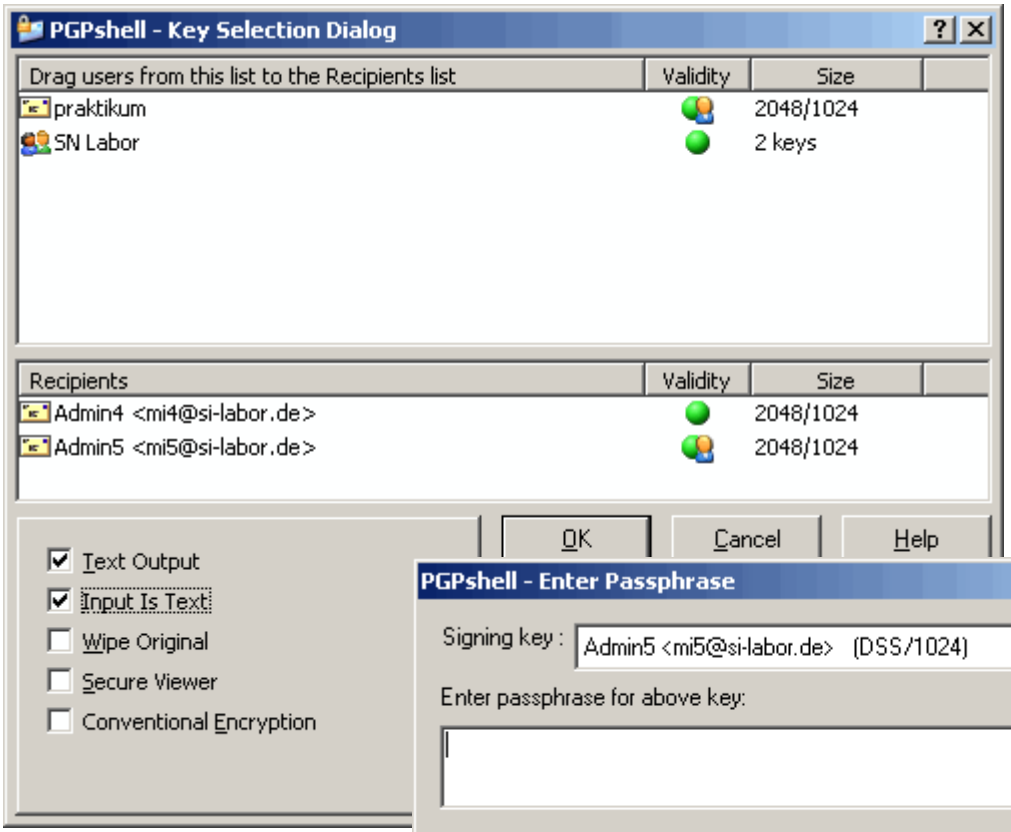


Abbildung 21 – Empfänger Liste mit Nachbargruppe

Abbildung 22 – Eingabe des private Keys zum Signieren

Auch wir haben wieder einen verschlüsselten und signierten Text erhalten, welchen wir nach Eingabe unseres private Keys auch entschlüsseln können (s. Abbildung 23). Da die Datei auf dem Übertragungsweg nicht verändert wurde, gibt der PGPlog keinen Signaturfehler aus (s. Abbildung 24). Die Datei kann somit geöffnet werden (s. Abbildung 25).

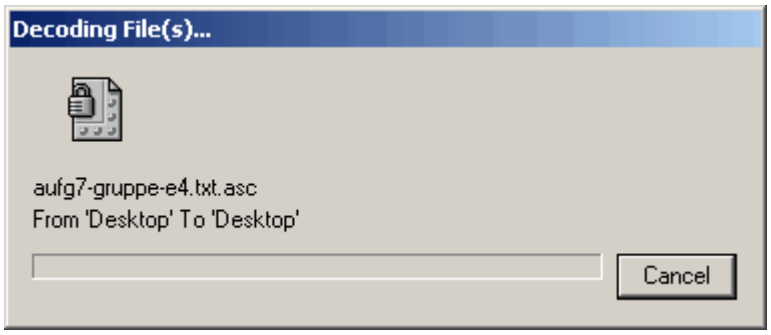


Abbildung 23 – Die Datei der Nachbargruppe wird entschlüsselt

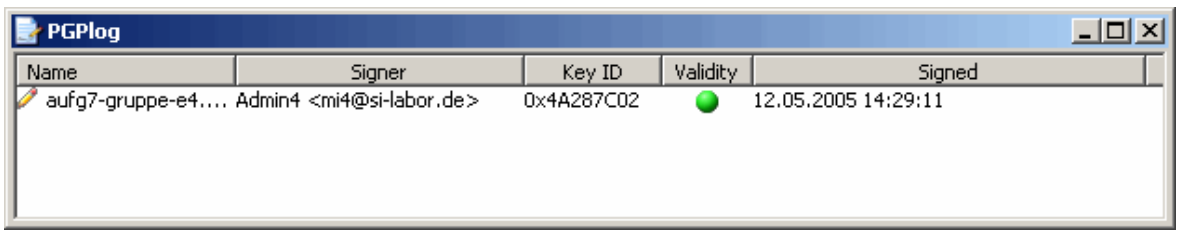


Abbildung 24 – PGPlog erkennt keine Veränderung

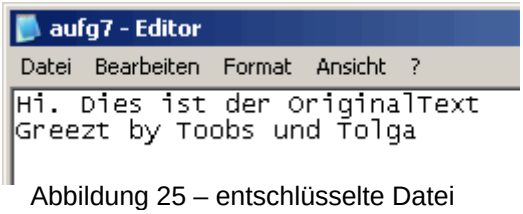


Abbildung 25 – entschlüsselte Datei

8. Signieren Sie einen Text und prüfen die Signatur anschließend.

Über das Kontextmenü wird der Text zunächst signiert. Dazu ist die Eingabe unseres private Keys notwendig. Nachdem die Datei signiert wurde, kann man ebenfalls über das Kontextmenü die signierte Datei überprüfen (PGP > Verify Signature) (s. Abbildung 26). Hierbei tritt kein Fehler auf (s. Abbildung 27).

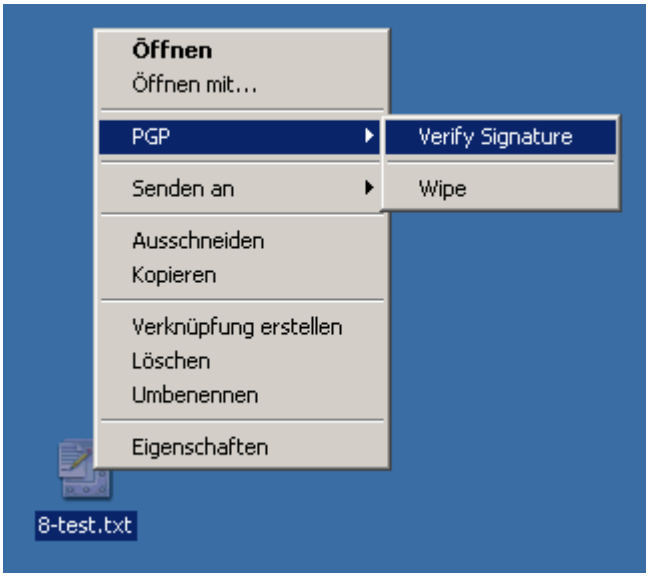


Abbildung 26 – Signatur überprüfen

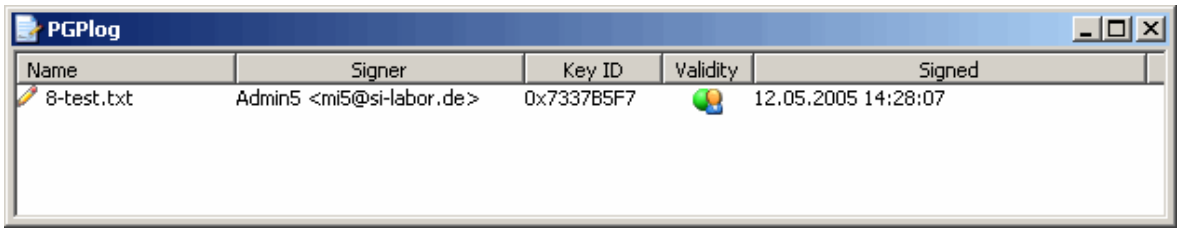


Abbildung 27 – PGPlog erkennt keine Veränderung

9. Verschlüsseln Sie eine Datei.

Wie schon in den Aufgaben 4, 5 und 6 verschlüsseln wir die Datei über das Kontextmenü. Im Gegensatz zu den vorherigen Aufgaben verwenden wir allerdings eine Bilddatei und keine Textdatei.

10. Signieren Sie eine Datei.

Wie schon in Aufgabe 8 signieren wir die Datei unter Eingabe unseres private Keys über das Kontextmenü. Im Gegensatz zu den vorherigen Aufgaben verwenden wir allerdings eine Bilddatei und keine Textdatei.

Nach Auswahl des Menüpunktes erstellt PGP eine Signaturdatei mit der zusätzlichen Endung .sig zum Dateinamen. Diese beinhaltet nur die Signatur, die Originaldatei wurde im Gegensatz zum Vorgehen bei der Textdatei dabei nicht von PGP verändert.

11. Versuchen Sie eine Datei zu entschlüsseln, für die Sie keinen privaten Schlüssel besitzen.

Zur Lösung dieser Aufgabe haben wir eine Datei nur mit dem öffentlichen Schlüssel der Nachbargruppe verschlüsselt. Obwohl wir die Datei selbst verschlüsselt haben, ist es uns im Nachhinein nicht möglich die Datei wieder zu entschlüsseln, da uns der nötige private Schlüssel der Nachbargruppe fehlt (s. Abbildung 28).

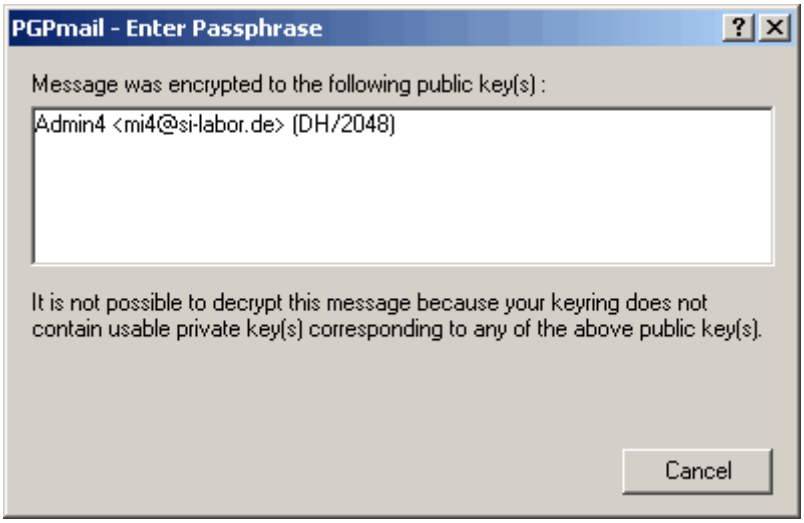


Abbildung 28 – Entschlüsseln ohne private Key nicht möglich

12. Verändern Sie einen verschlüsselten Text.

Zunächst haben wir einen Text in Word erstellt und diesen über das PGP-Icon in der Traybar direkt verschlüsselt (PGP-Icon > Current Window > Encrypt) (s. Abbildung 29). Das Chiffrat wurde sofort in Word sichtbar, welches wir um das Wort „Veränderung“ an einer beliebigen Stelle ergänzten (Rot gekennzeichnet, s. Abbildung 30). Ein anschliessendes Entschlüsseln (PGP-Icon > Current Window > Decrypt and Verify) (s. Abbildung 30) schlug dadurch fehl s. Abbildung 31).

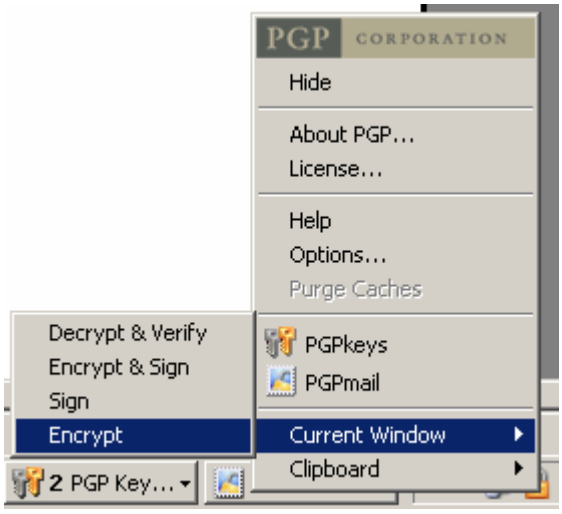


Abbildung 29 – Direktes Verschlüsseln

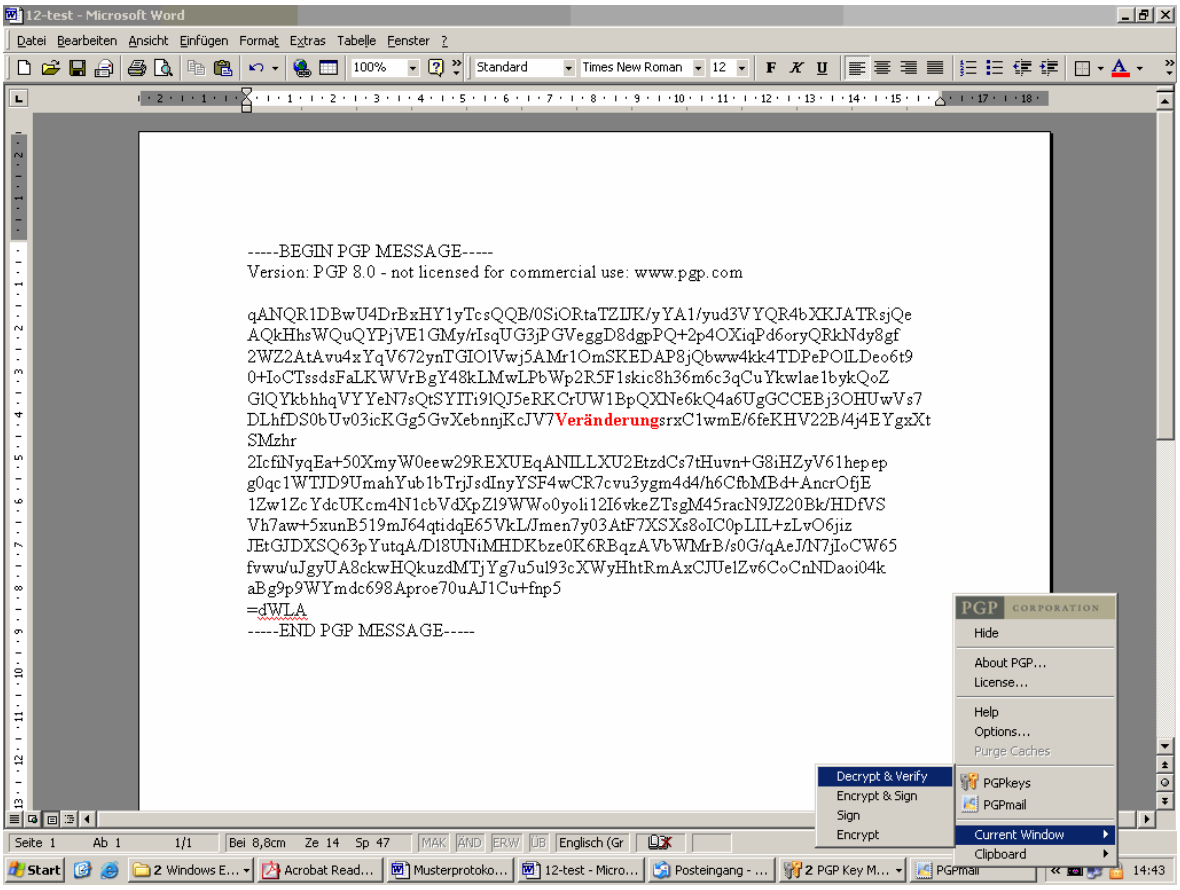


Abbildung 30 – Veränderung der Verschlüsselung und Vorgehen zum Entschlüsseln

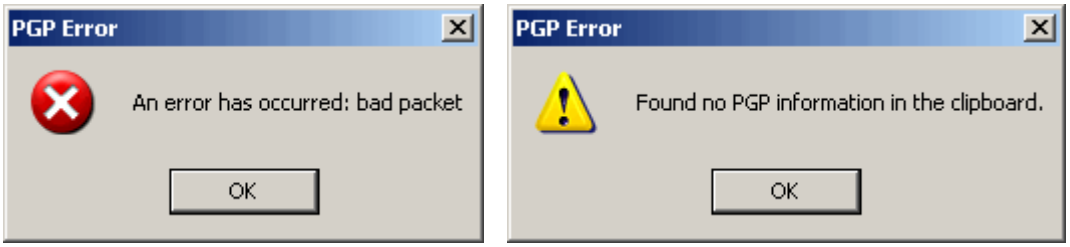


Abbildung 31 – Auftretende Fehlermeldungen nach dem Entschlüsseln

13. Verändern Sie einen signierten Text.

Auch in dieser Aufgabe haben wir zunächst eine Word-Datei erstellt, welche wir danach direkt über das PGP-Icon in der Traybar signiert haben (s. Abbildung 32). Die Signatur wird dabei automatisch direkt an den Originaltext angehängt.

Nach der Veränderung des Textes in der Word-Datei („Teil b“, in rot gekennzeichnet, s. Abbildung 33) und der anschliessenden Überprüfung der Signatur gibt das Protokoll eine Warnung aus: „Signature did not verify. Message has been altered.“ (s. Abbildung 34)

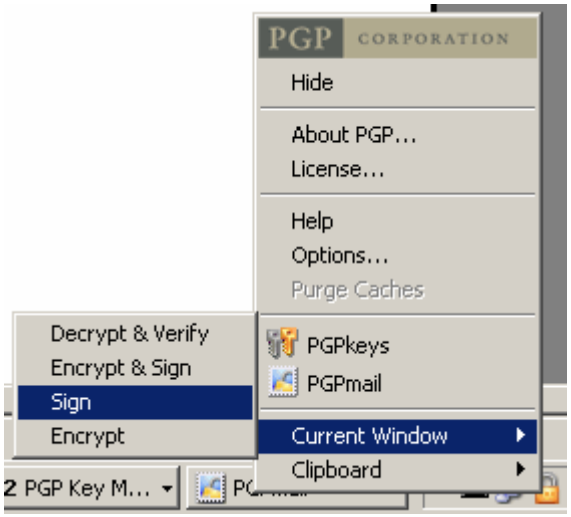


Abbildung 32 – Signieren der Word-Datei

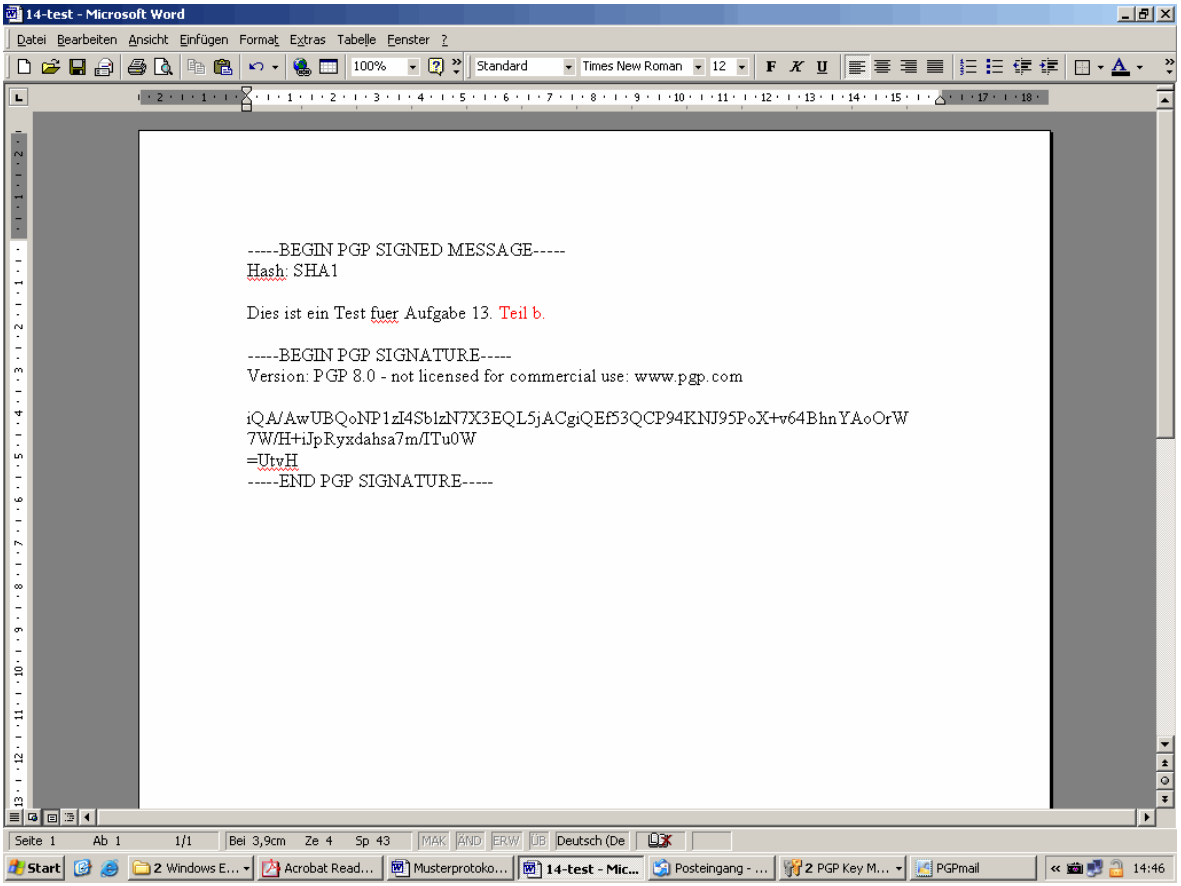


Abbildung 33 – Veränderung des signierten Textes

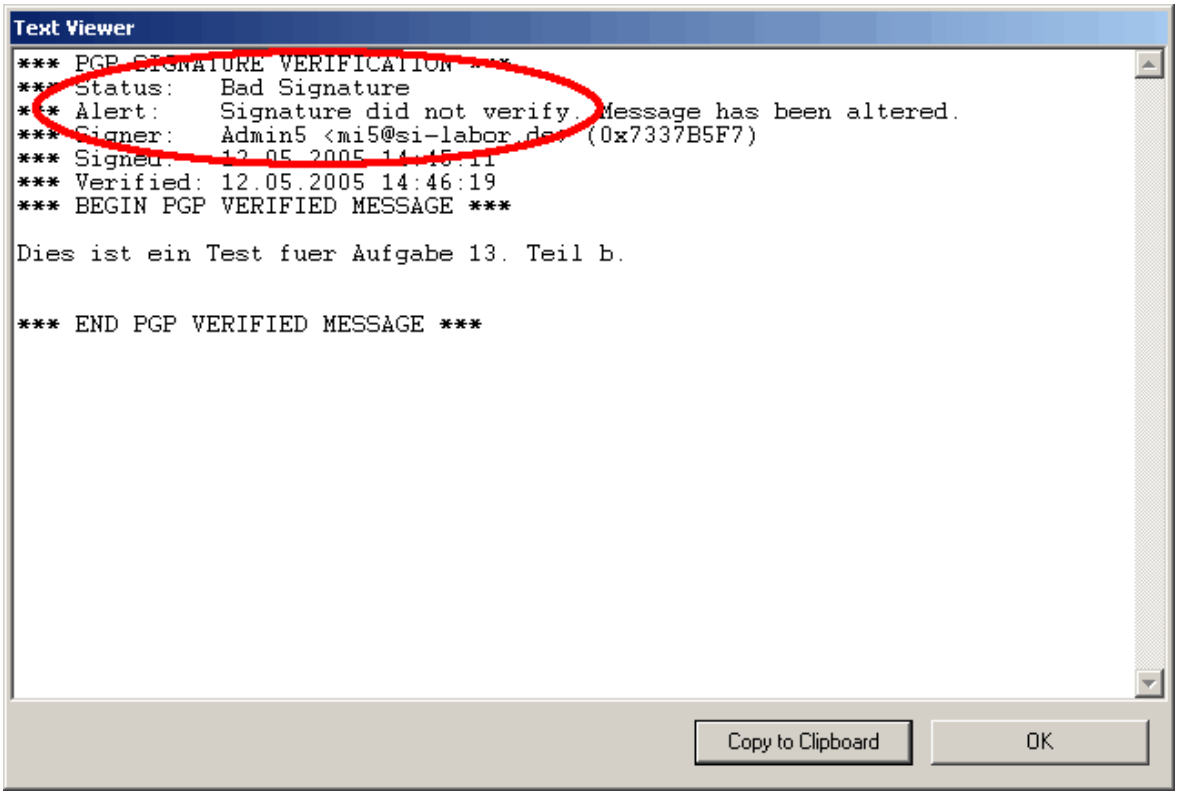


Abbildung 34 – Auftretende Fehlermeldung nach dem Überprüfen der Signatur

14. Verändern Sie die Signatur eines Textes.

Auch in dieser Aufgabe haben wir zunächst eine Word-Datei erstellt, welche wir danach direkt über das PGP-Icon in der Traybar signiert haben. Die Signatur wird dabei automatisch direkt an den Originaltext angehängt (s. Abbildung 35).

Nach der Veränderung der Signatur in der Word-Datei („Veränderung“, in rot gekennzeichnet) und der anschliessenden Überprüfung der Signatur gibt das Programm zunächst einen Fehler aus (s. Abbildung 36).

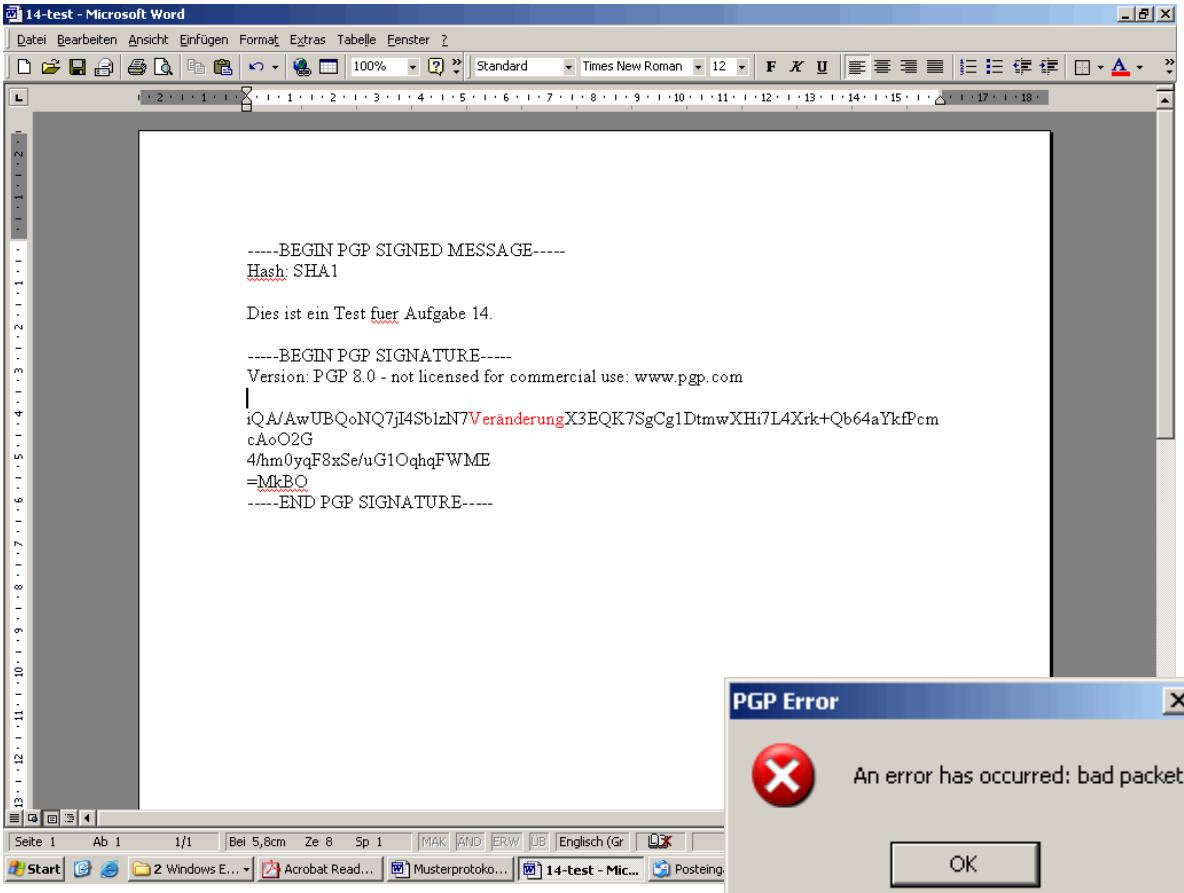


Abbildung 35 – Veränderung der Signatur

Abbildung 36 – Auftretende Fehlermeldung

15. Exportieren Sie einen Schlüssel, verändern diesen und versuchen ihn zu importieren.

Nachdem wir unseren Schlüssel auf den Desktop exportiert haben, verändern wir diesen an einer beliebigen Stelle im Notepad („---VERAENDERUNG---„) (s. Abbildung 37) und importieren ihn danach wieder in PGP.
Dabei tritt allerdings wieder der bekannte „Bad Packet“-Fehler auf, der Schlüssel kann nicht reimportiert werden (s. Abbildung 38).

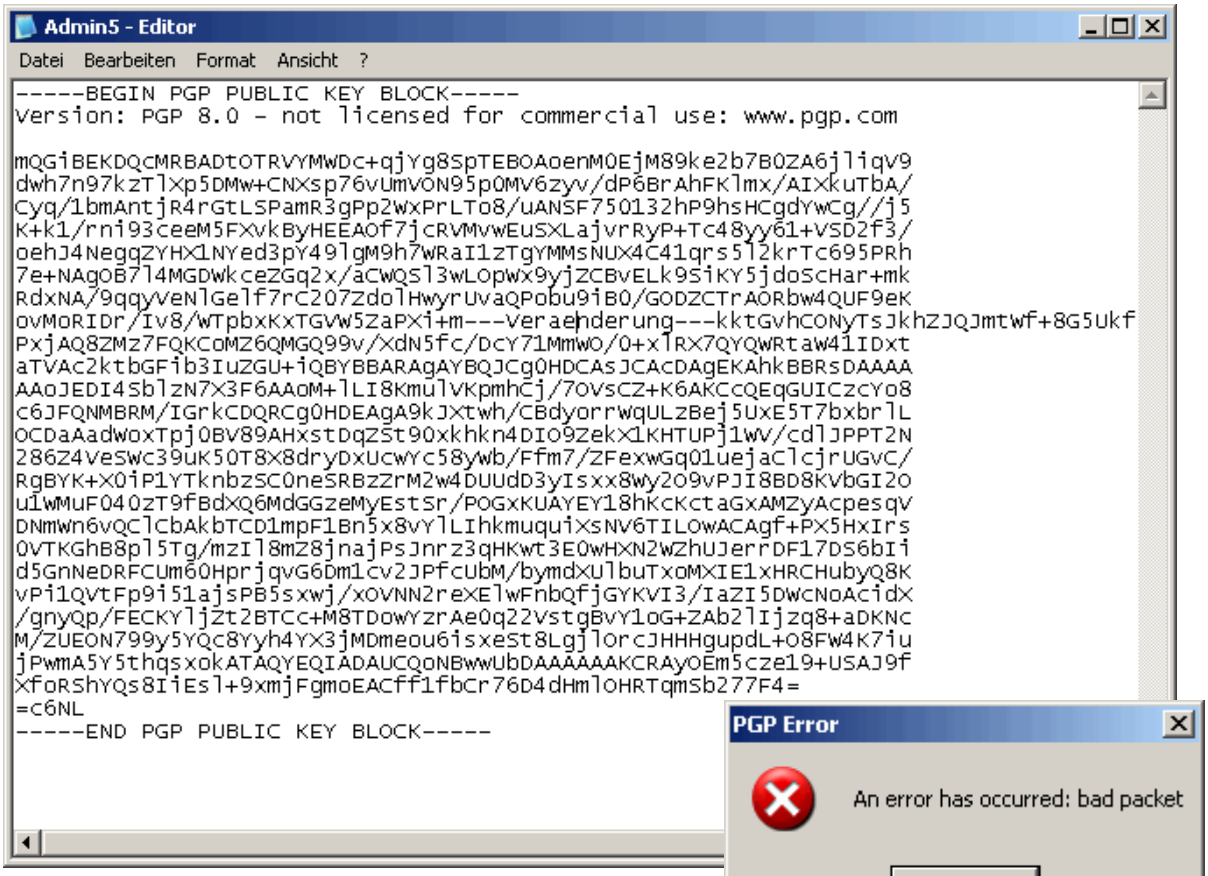


Abbildung 37 – Veränderung des exportierten Schlüssels

Abbildung 38 – Auftretende Fehlermeldung

16. Verschlüsseln Sie eine Datei im ASCII-Format und verändern diese.

Nach dem bekannten Verfahren verschlüsseln wir nun wieder eine Bilddatei. Nach dem Verändern der verschlüsselten Datei in Notepad („VERAENDERUNG“) (s. Abbildung 39) versuchten wir, die Datei wieder zu entschlüsseln. Dies schlug jedoch fehl (s. Abbildung 40).

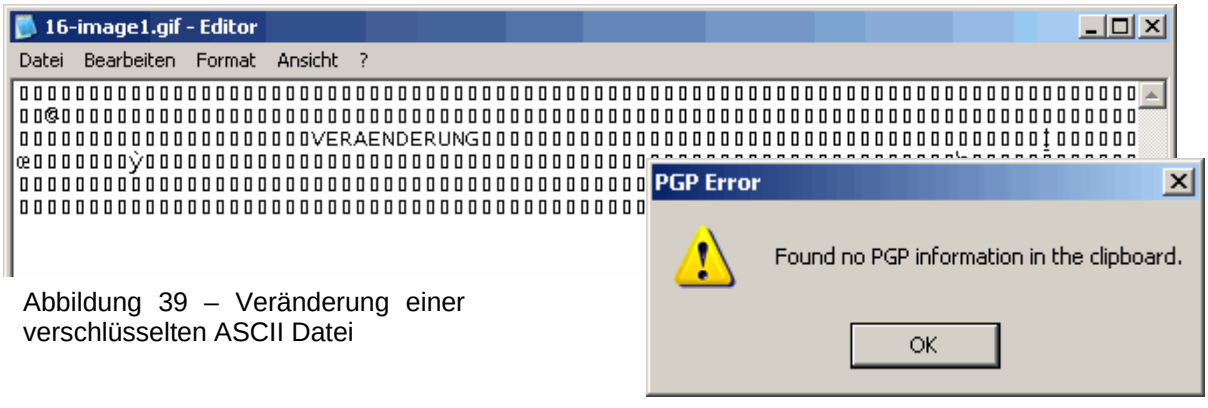


Abbildung 39 – Veränderung einer verschlüsselten ASCII Datei

Abbildung 40 – Auftretende Fehlermeldung

17. Verändern Sie die Signatur einer Datei

Nach dem bekannten Verfahren signieren wir nun eine Bilddatei. Im Gegensatz zu der verschlüsselten Bilddatei bzw. der signierten Word-Datei wird beim Signieren einer Bilddatei eine Signaturdatei erzeugt, welche mitsamt der Bilddatei weitergegeben werden muss, um die Bilddatei zu verifizieren.

Deshalb müssen wir für diese Aufgabe die von PGP erzeugte Datei im Notepad verändern. Hierbei haben wir wieder an einer beliebigen Stelle das Wort „VERAENDERUNG“ eingefügt (s. Abbildung 41). Die anschliessende Überprüfung der Signatur in PGP schlug deshalb wiederum fehl (s. Abbildung 42).

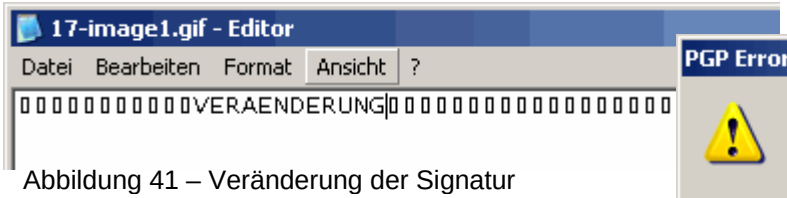


Abbildung 41 – Veränderung der Signatur

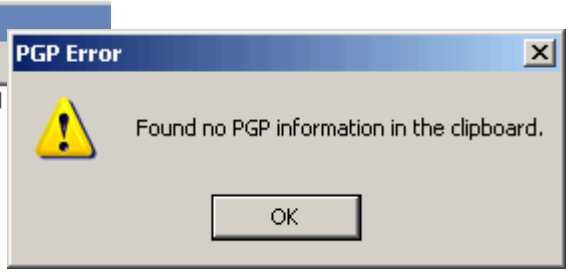


Abbildung 42 – Auftretende Fehlermeldung

18. Signieren Sie eine Datei und verändern nachträglich die Datei

Da bei Nicht-Textdateien die Signatur als eigenständige Datei angehängt wird, ist es auch nachträglich möglich die Originaldatei (in unserem Fall wieder eine Bilddatei) im Notepad zu verändern.

Hierfür fügen wir mal wieder an beliebiger Stelle den Text „---VERAENDERUNG---„ ein und speichern diese ab (s. Abbildung 43).

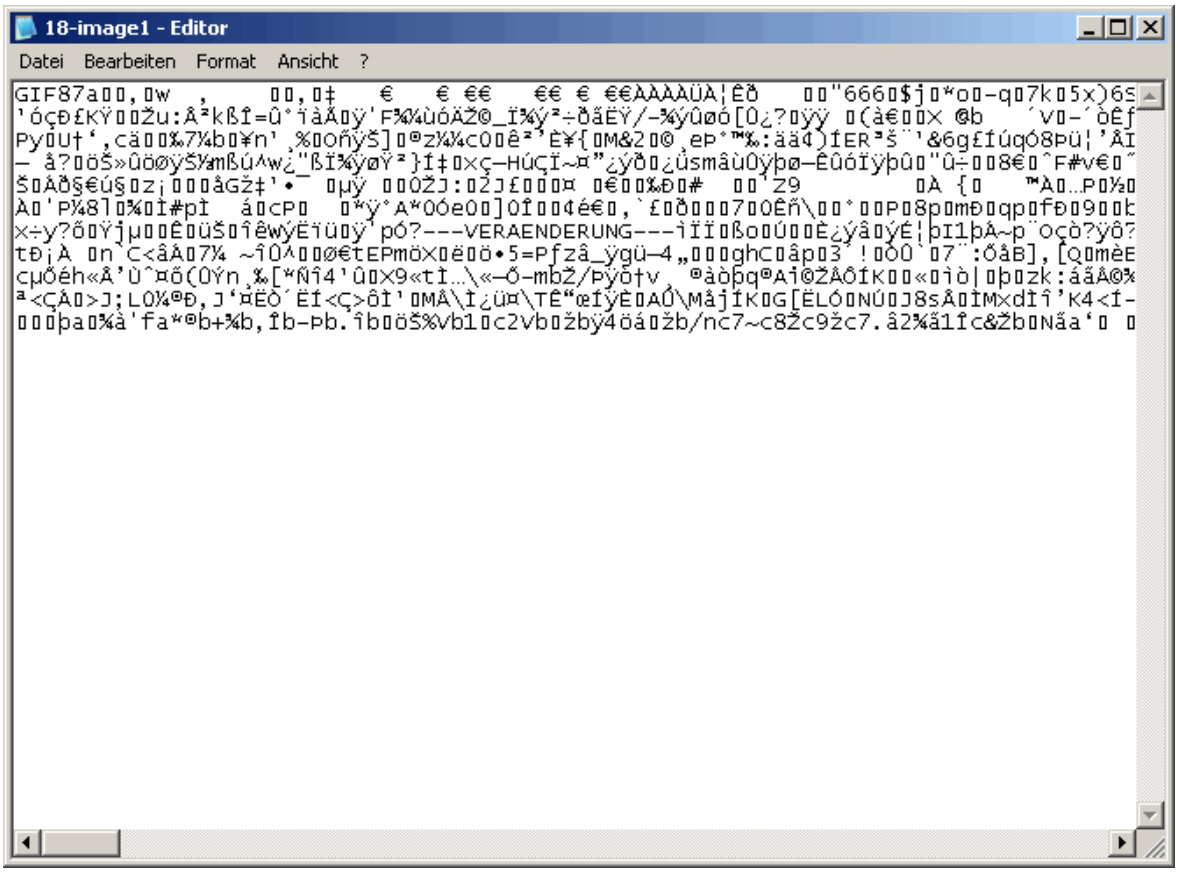


Abbildung 43 – Veränderung der Datei

Diese veränderte Bilddatei ist allerdings nicht mehr lesbar in einem Bildbetrachtungsprogramm.

Auch die Überprüfung der mitgelieferten Signatur gibt nun einen Fehler aus.

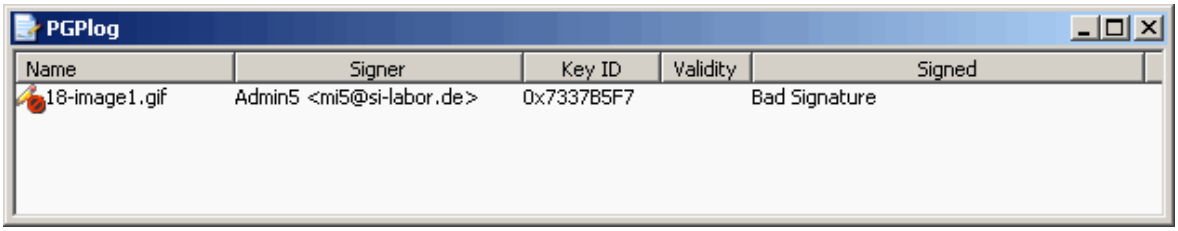


Abbildung 44 – Ausgegebene Fehlermeldung

19. Erstellen Sie eine bootfähige Diskette mit Norton Ghost.

Hierzu starten wir Norton Ghost und wählen den Assistenten für die Erstellung einer „Ghost-Standard-Boot-Diskette“ aus (s. Abbildung 45). Die Standardeinstellungen können weitesgehend übernommen werden.
Man sollte auf folgende Dinge achten:
USB-Erkennung aktivieren, wenn man eine USB-Festplatte das Image speichern möchte (s. Abbildung 46).
Laufwerk der zu erstellenden Boot-Disketten auswählen (s. Abbildung 47).

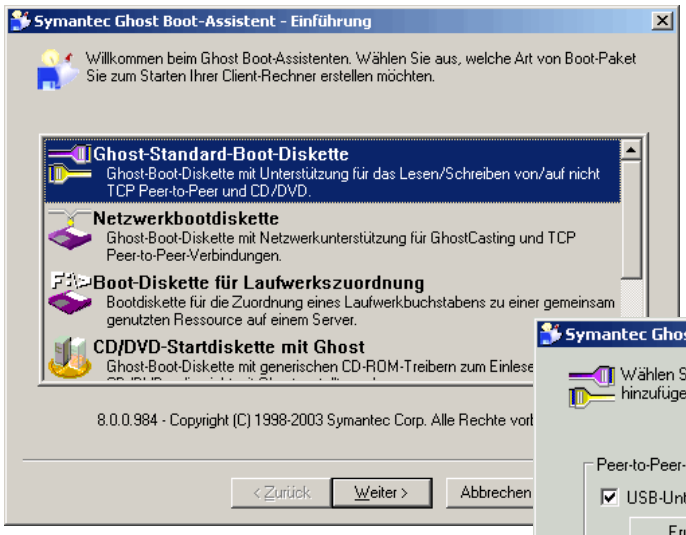


Abbildung 45 – Assistent Boot-Diskette

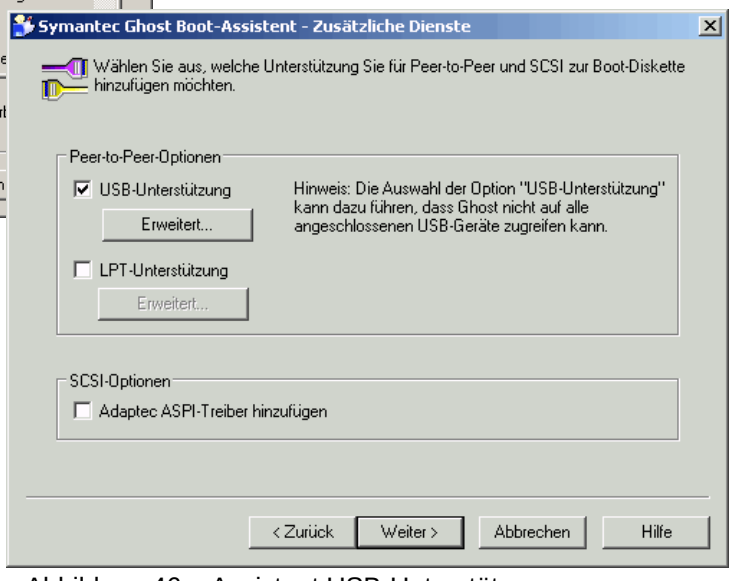


Abbildung 46 – Assistent USB-Unterstützung

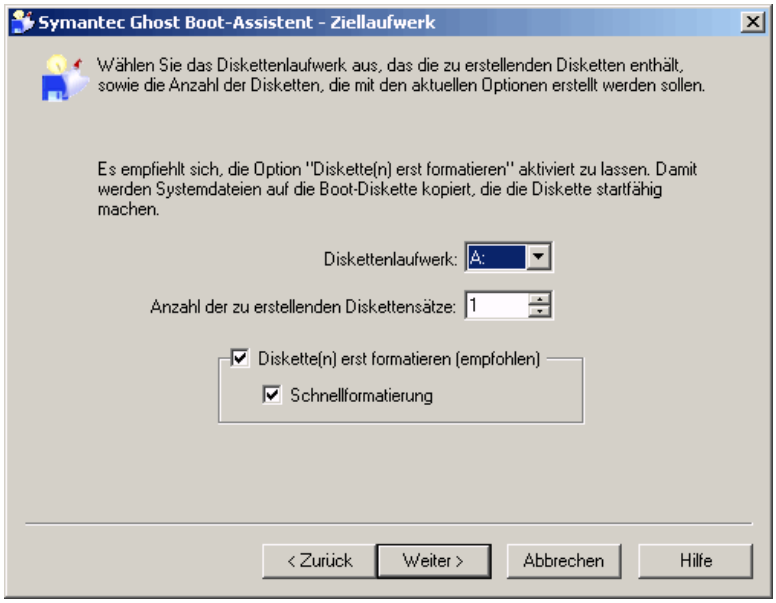


Abbildung 47 – Assistent Laufwerk Auswahl

Norton Ghost erstellt nun 2 Disketten, mit denen man den PC neu starten kann und Images erstellen/wiederherstellen kann (s. Abbildung 48).

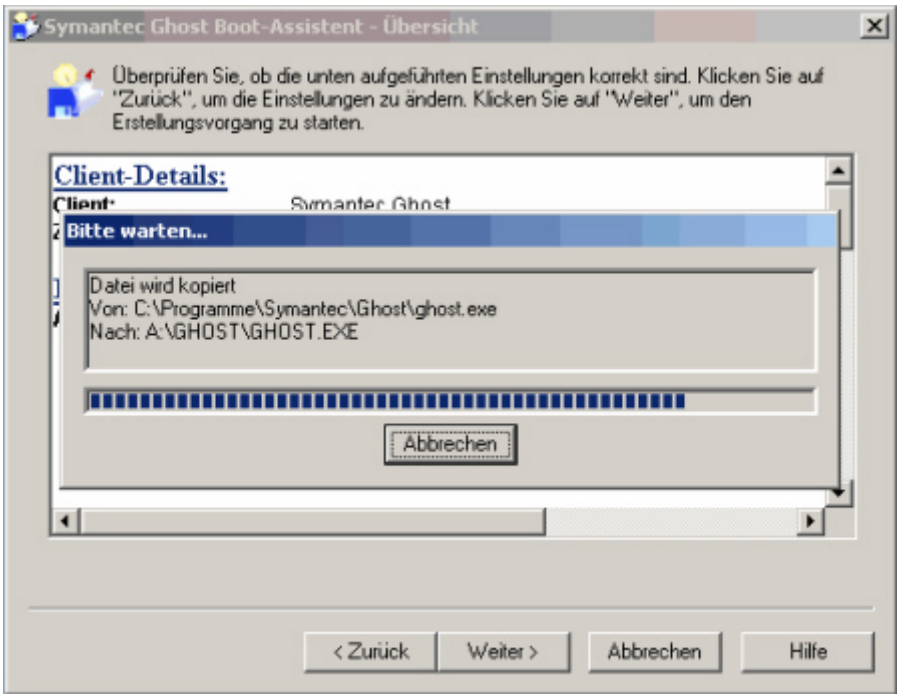


Abbildung 48 – Assistent Boot-Disketten werden erstellt.

20. Erzeugen Sie ein Image der E-Partition ihres Rechners mit Norton Ghost auf Partition D. Wo liegt das Image jetzt? Vergleichen Sie die Größe der Platte mit der Datei.

Der Rechner wird mit den Startdisketten neu gestartet (gebootet)

(s. Abbildung 49). Auswahl 1 des „Startup Menu“ auswählen (Dieser Computer wird von startfähigen Disketten gestartet).

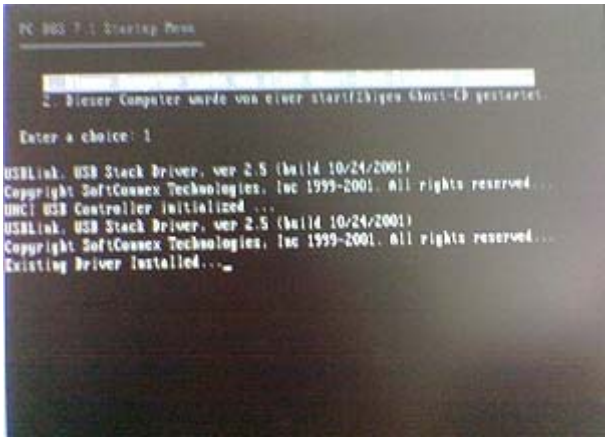


Abbildung 49 – Auswahl Startoptionen

Um ein Image erstellen zu können, wählen wir im Menü von Norton Ghost folgenden Punkte aus:

Lokal > Partition > Auf Image (s. Abbildung 50)

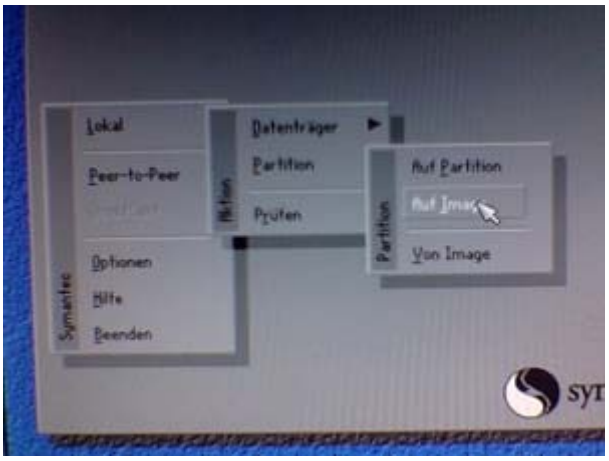


Abbildung 50 – Auswahl „Auf Image“

Anschließend wird das Laufwerk ausgewählt, von dem ein Image erstellt werden soll (s. Abbildung 51), in unserem Fall Laufwerk E:.

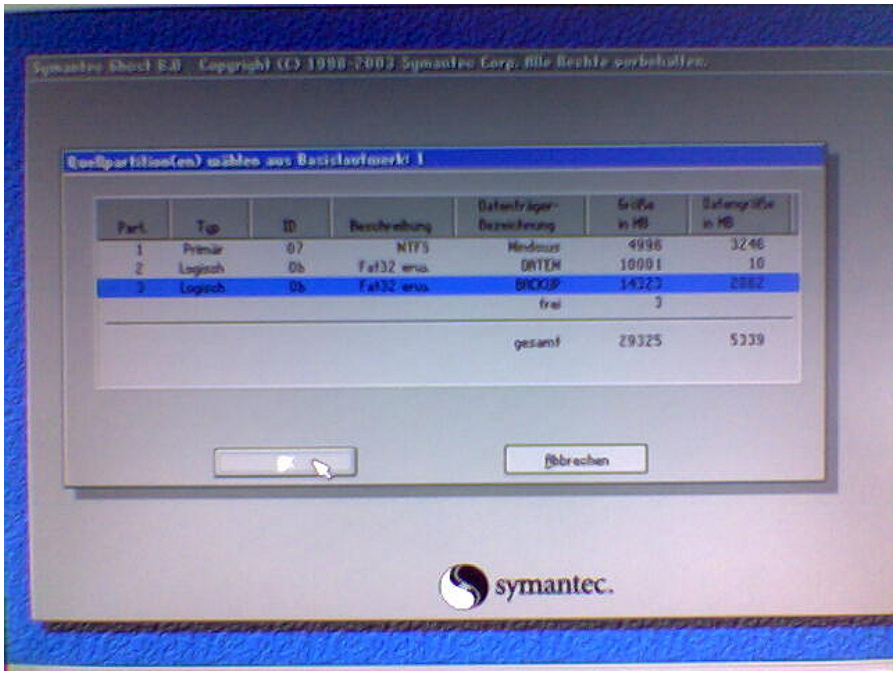


Abbildung 51 – Quellpartition auswählen

Nach der Auswahl der Quellpartition, muss das Ziellaufwerk ausgewählt werden.

Hier ist zu beachten, dass Norton Ghost bei der Zielpartitionsauswahl nur FAT32 unterstützt. So wird aus dem Ziellaufwerk D: -> C: (s. Abbildung 52).

Bei dem Komprimierungsfaktor wählen wir die Stufe „schnelle“ aus.

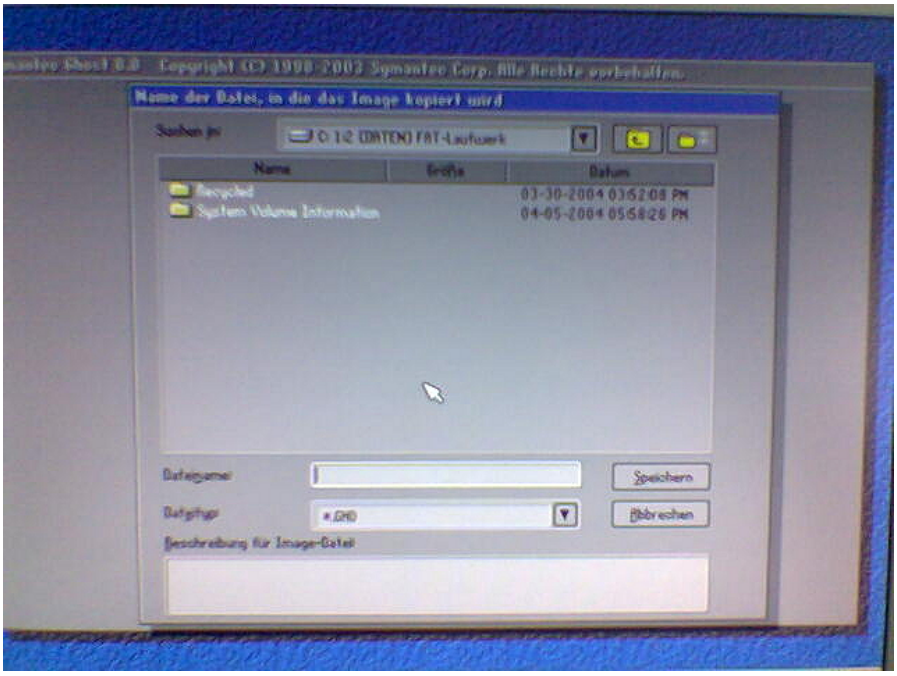


Abbildung 52 – Zielpartition auswählen

Jetzt erstellt Norton Ghost ein Image vom Laufwerk E: auf dem Laufwerk C:.
Das Programm wird beendet und der Rechner neu gestartet (Disketten aus dem Laufwerk nehmen).

Das Image liegt unter Windows auf dem Laufwerk D: und nicht auf C:, da C: eine NTFS Partition ist (s. Abbildung 53).

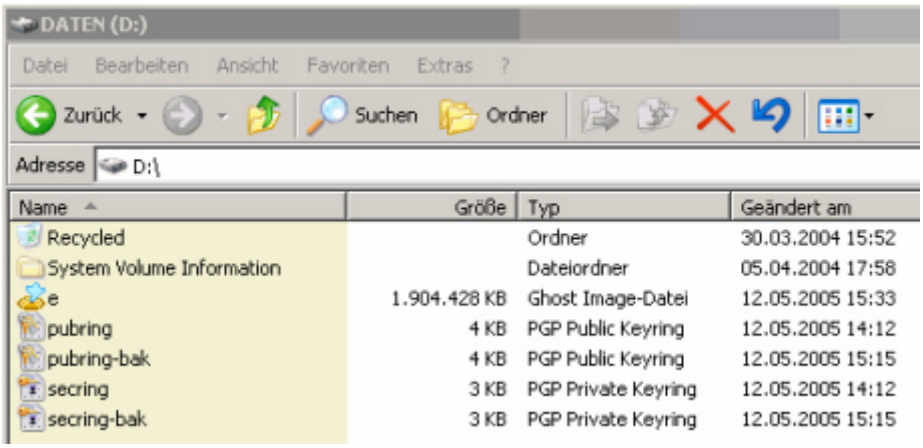


Abbildung 53 – Image auf Laufwerk D:

Die Größe des Ghost Image ist etwas kleiner, wie die Größe vom Laufwerk. Das hängt damit zusammen, dass Norton Ghost das Image (auch auf Komprimierungsstufe „schnell“) minimal komprimiert hat (s. Abbildung 54).

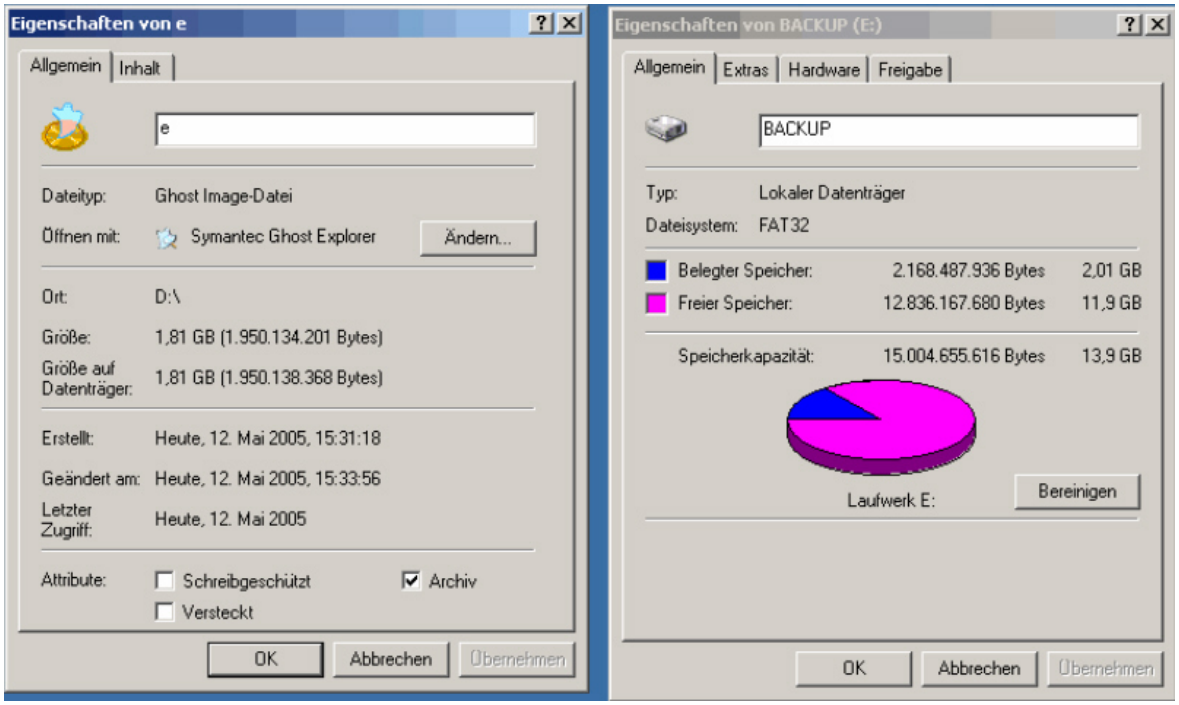


Abbildung 54 – Größe Image und Laufwerk

21. Was passiert, wenn nun Daten auf der Partition E verändert werden und anschließend das Image zurückgespielt wird?
Ändert man Daten auf der Partition E, von der zuvor ein Image erstellt wurde, sind die Änderungen nach der Wiedereinspielung des Images nicht berücksichtigt.

22. Finden Sie heraus, wie viele Kombinationsmöglichkeiten es bei einem Passwort von 4 und 8 Zeichen gibt. (Voraussetzung: verwendete Zeichen: 92 (deutsches Tastaturlayout))

Formel: n^k

n: Kombinationsmöglichkeiten
k: Anzahl

92^4 = 71639296 Möglichkeiten
92^8 = 5132188731375616 Möglichkeiten

23. Erläutern Sie den Ablauf der digitalen Signatur.
Wenn man eine Nachricht im Klartext schreibt, damit sie jeder lesen kann, aber sicherstellen will, dass keiner sie ändert bzw. jeder Empfänger prüfen kann, ob die Nachricht wirklich von ihm stammt, kann man sie mit dem private key unterschreiben (wie beim Verschlüsseln, nur jetzt Sign statt Encrypt wählen) – das erzeugt lesbaren Klartext mit einer digitalen Signatur darunter, die wie ein kleines Kryptogramm aussieht. Jeder, der den public key hat, kann die Unterschrift prüfen (das geht genauso wie Entschlüsseln – Decrypt & Verify).

Beispiel aus dem CrypTool:
Du kannst mit PGP auch Nachrichten unterschreiben oder signieren. Dazu unterschreibst du die Nachricht mit deinem privaten Schlüssel. Nun kann der Empfänger mit deinem öffentlichem Schlüssel überprüfen, ob die Nachricht wirklich von dir stammt.

Zum Beispiel:
Alice will von ihrer Hausbank, der Bob-Bank of Littletown, eine Überweisung tätigen. Dazu schreibt sie die Überweisung, unterschreibt sie anschließend mit ihrem privaten Schlüssel und sendet sie an die Bob-Bank. Nun nimmt Bob den öffentlichen Schlüssel von Alice und überprüft die Signatur. Wenn diese stimmt und Alice genug Geld auf ihrem Konto hat wird Bob die Überweisung erledigen. Man kann natürlich signieren und verschlüsseln kombinieren. Das hieße hier dass Alice, weil Überweisungen ja niemanden was angehen, die signierte Überweisung anschließend noch verschlüsselt.

24. Zu welchen Sicherheitsproblemen (Virenscanner, Firewall) kann es kommen, wenn Nachrichten/Daten verschlüsselt gesendet werden?
Email Programme blockieren den Anhang, dieser kann somit vom Empfänger nicht geöffnet werden. Die verschlüsselte Datei besteht aus einer Menge von undurchsichtigen Zeichen. Es kann vorkommen, dass ein Virenscanner die Zeichen als Signatur eines Virus erkennt. Eine Firewall kann die eingehen Nachricht als Eindringversuch wahrnehmen und versuchen, diese Nachricht zu blockieren.

25. Signieren bedeutet mehr, als eine digitale Unterschrift leisten. Finden Sie heraus, was Signieren noch bedeutet.

Der Empfänger einer digital signierten Datei, kann den Versender einwandfrei identifizieren. Signaturen sind somit sehr vielseitig.
z.B. PGP-Signaturen rechtsverbindlich gestalten:
Mittlerweile sind PGP-Signaturen vielerorts legal geworden. Gesellschaften benutzten digitale PGP-Signaturen in Verträgen, um schnelle Abschlüsse über Email zu erzielen, was es erlaubt, die Arbeit aufzunehmen, ohne auf unterschriebene Papiere warten zu müssen