Pretty Good Privacy

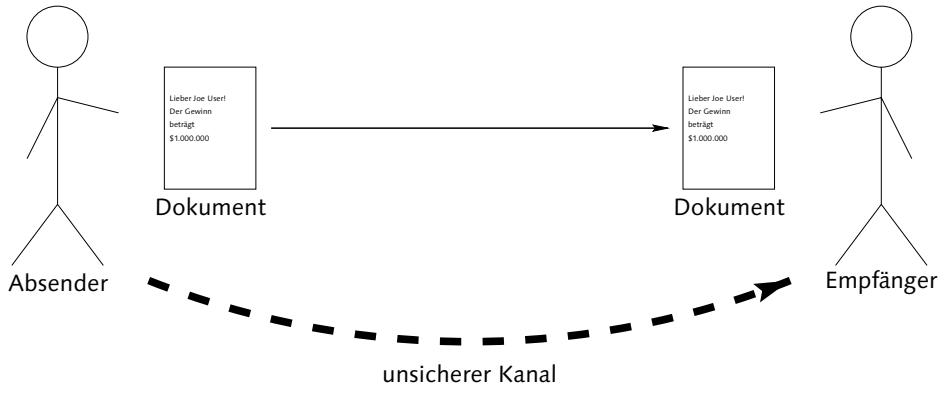
Jörn Clausen
jc@Genetik.Uni-Bielefeld.DE

Übersicht

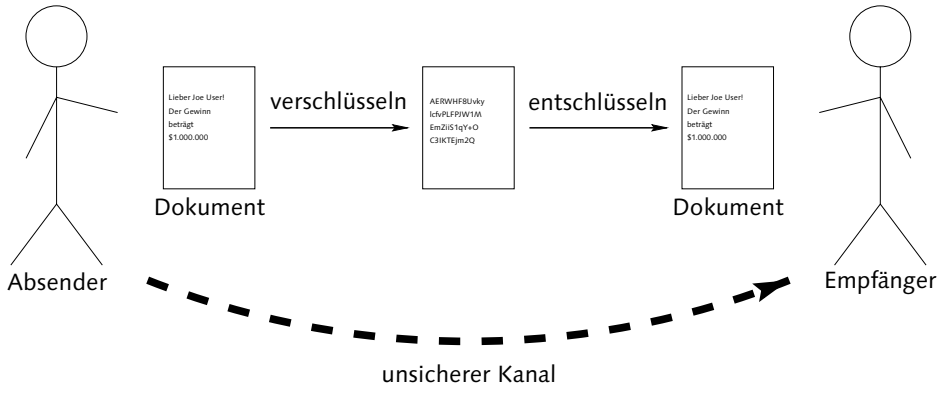
- Was ist Verschlüsselung? Wer braucht das überhaupt?
- Welche Arten von Verschlüsselung gibt es?
- Pretty Good Privacy
- korrekter Umgang mit PGP

Was ist Verschlüsselung?

- fundamentales Problem: Transport einer Nachricht über einen unsicheren Kanal



- Nachricht kann abgehört, abgefangen oder verändert werden
- Lösung: Verschlüsselung

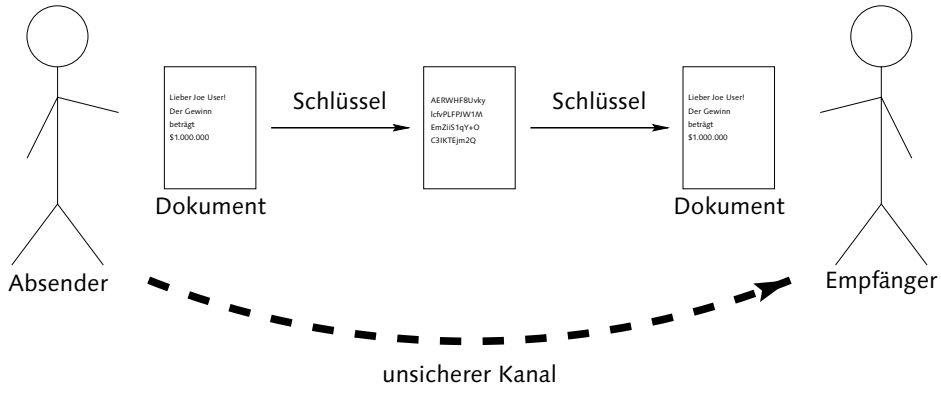


Was geht mich das an?

- typische Fehleinschätzungen:
  - „Ich habe doch keine Geheimnisse!“
  - „Da wird schon nichts passieren.“
- EMail inheränt unsicher
- Brief vs. Postkarte
- Kooperation mit Wirtschaft:
  - Urheberrecht, Patentierbarkeit
  - Industriespionage
- staatliche Überwachung
- Strafverfolgung

symmetrische Verschlüsselung

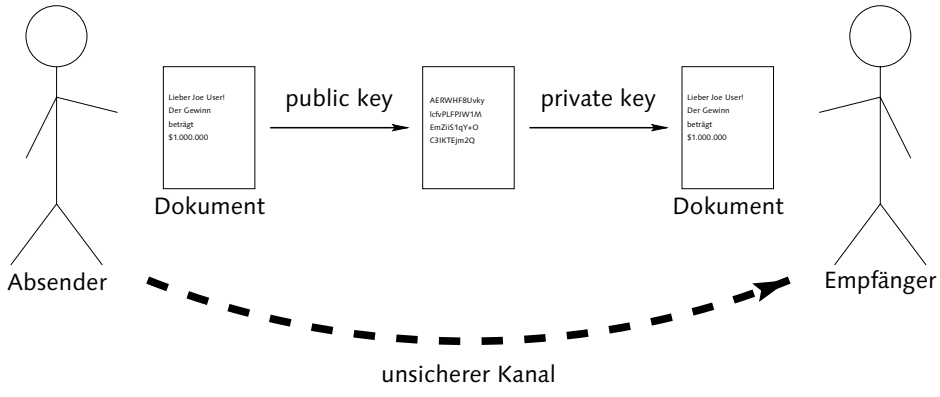
- klassisches Szenario:



- gleicher Schlüssel zum Ver- und Entschlüsseln
- Probleme:
  - Schlüssel muß über unsicheren Kanal
  - alle Teilnehmer brauchen Schlüssel
  - Was, wenn ein Teilnehmer kompromittiert wurde?

public key-Verfahren

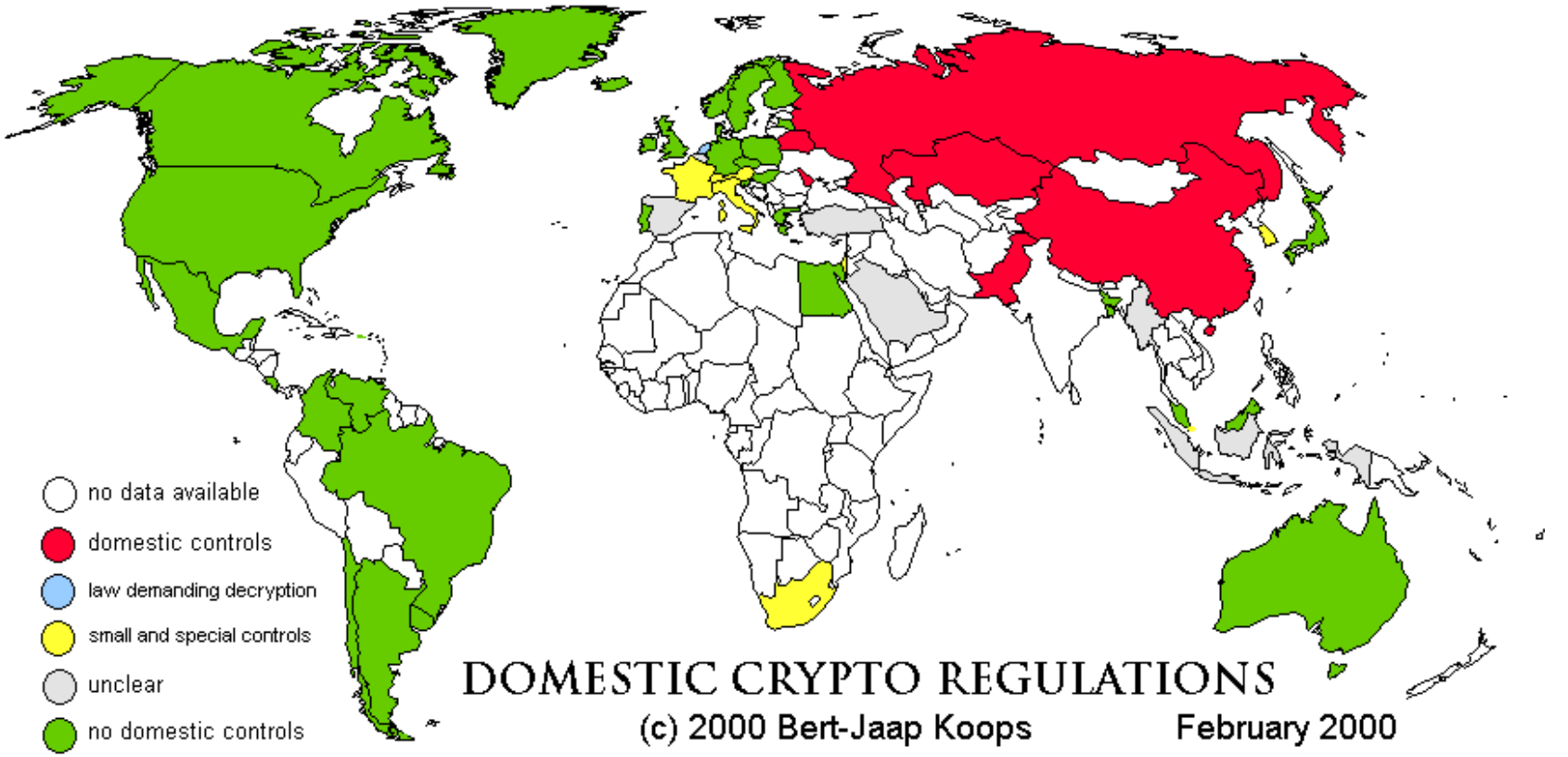
- asymmetrische Verschlüsselung:

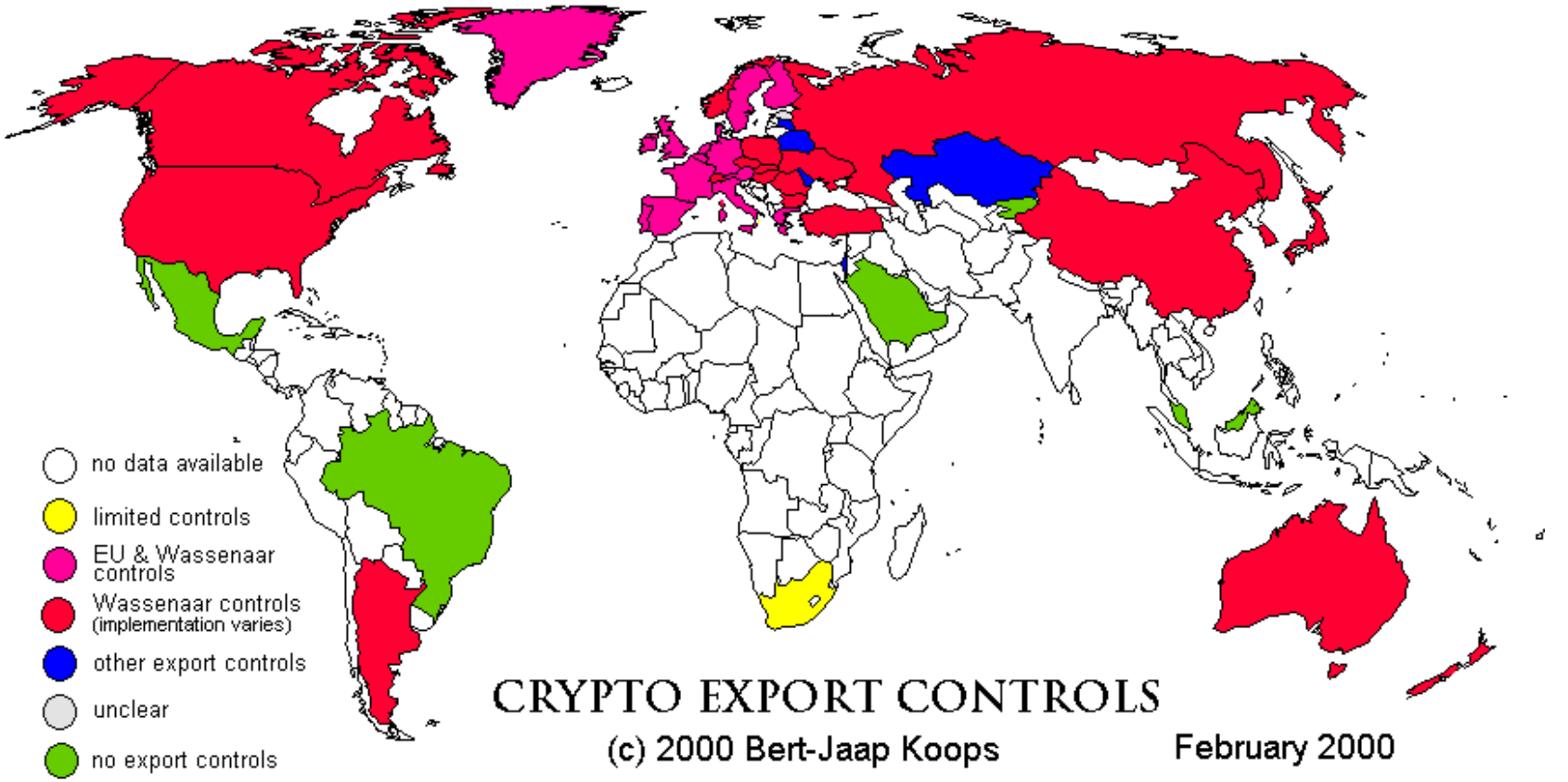


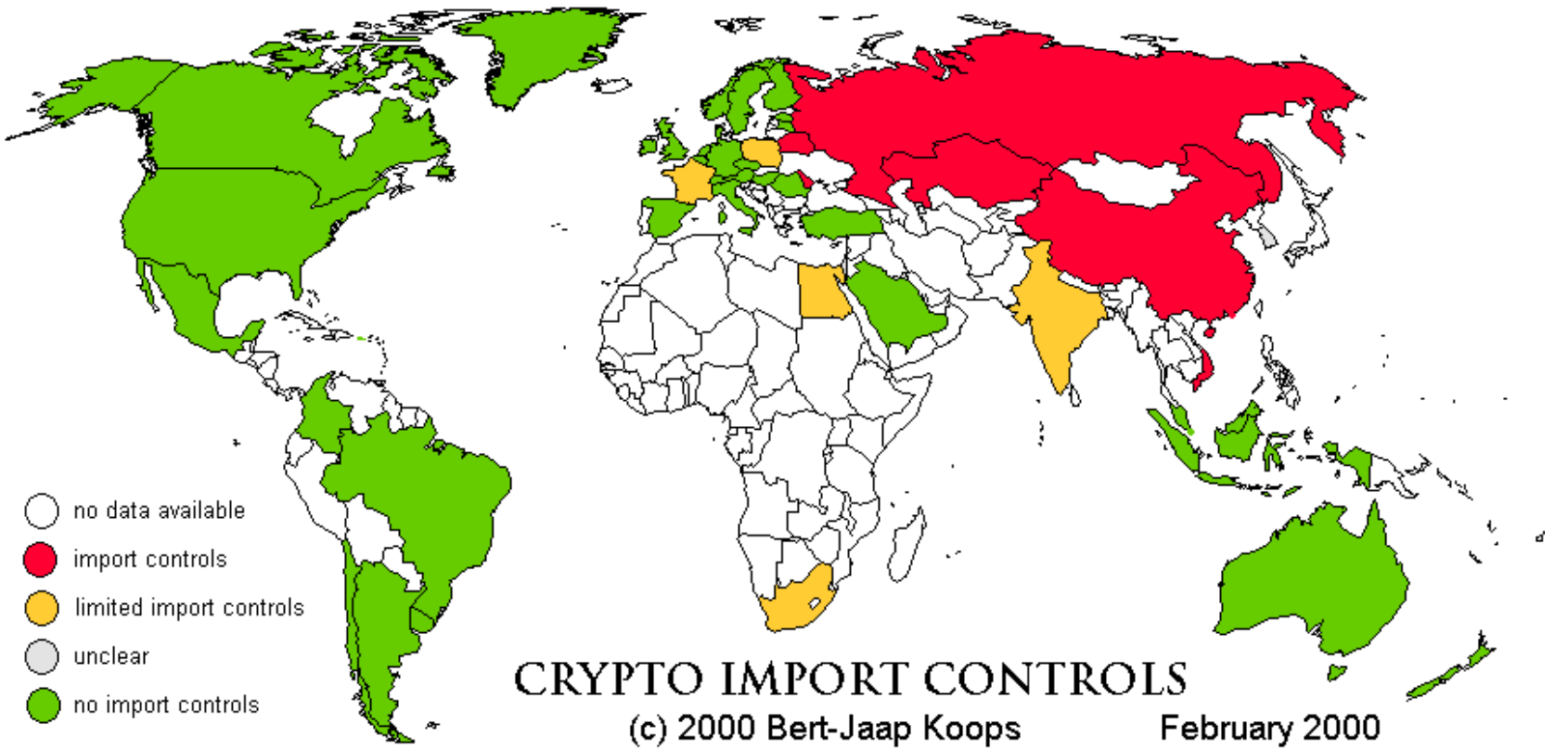
- Schlüssel besteht aus zwei Teilen:
  - *public key* zum Verschlüsseln
  - *private key* zum Entschlüsseln
- Hintergrund: große Primzahlen
- Absender hat public key des Empfängers
- nur Empfänger kann Nachricht entschlüsseln
- private key nicht aus public key berechenbar

Anmerkungen

- symmetrische Verschlüsselung ist schneller
- technische/politische Probleme:
  - Urheberrechte, Patente, Lizenzen
  - gesetzliche Einschränkungen
  - *key escrow*
  - Export-/Import-Beschränkungen
  - Kryptografie häufig Rüstungstechnologie
- „Sicherheit“ abhängig von Länge des Schlüssels
- typische Schlüssellängen:
  - symmetrische Verfahren: 40, 56, 128 bit
  - asymmetrische Verfahren: 512, 768, 1024, 2048, 4096 bit

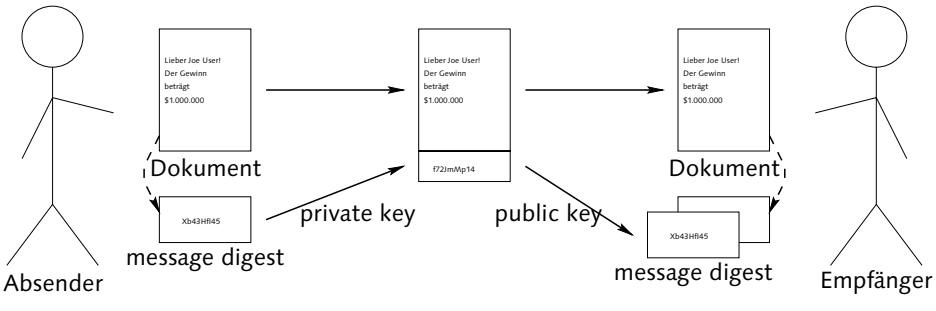

no data available
domestic controls
law demanding decryption
small and special controls
unclear
no domestic controls
DOMESTIC CRYPTO REGULATIONS
(c) 2000 Bert-Jaap Koops
February 2000


no data available
limited controls
EU & Wassenaar controls
Wassenaar controls (implementation varies)
other export controls
unclear
no export controls
CRYPTO EXPORT CONTROLS
(c) 2000 Bert-Jaap Koops
February 2000


no data available
import controls
limited import controls
unclear
no import controls
CRYPTO IMPORT CONTROLS
(c) 2000 Bert-Jaap Koops
February 2000

digitale Signatur

- Geheimhaltung nicht wichtig
- Authentizität des Absenders
- Integrität des Inhalts



- *message digest*: „Quersumme“ des Dokuments
- Änderung des Dokuments bewirkt anderen message digest

Pretty Good Privacy

- 1991 von Phil Zimmermann veröffentlicht
- verwendet alle gezeigten Verfahren
- unterschiedliche, bedingt kompatible Versionen
  - veraltete Version: 2.6.3, ein Schlüsselpaar zum Verschlüsseln und Signieren
  - aktuelle Versionen: 5.x, 6.5.x, zwei unabhängige Schlüsselpaare
- zur nichtprivaten Nutzung Lizenz erforderlich
- alternative Implementierungen:
  - OpenPGP (RFC 2440)
  - Gnu Privacy Guard (GnuPG, GPG)

verwendete Verfahren

	symmetrisch	asymmetrisch	message digest
PGP 2.6.3	IDEA, 3DES	RSA	MD5
PGP 5.x+	IDEA, 3DES, CAST	RSA, DH/DSS	MD5, SHA-1
GnuPG	3DES, CAST, Blowfish	DH/DSS	MD5, SHA-1

- IDEA: muß lizensiert werden
- RSA: in den USA patentiert bis 20.9.2000
- MD5: gilt als bedenklich
- PGP 2.6.3 sollte nicht mehr verwendet werden

Funktionsweise von PGP

- Verschlüsselung in mehreren Schritten:
  1. Erzeugung eines symmetrischen *session keys*
  2. Verschlüsselung des Dokuments mit session key
  3. Verschlüsselung des session keys mit public key des Empfängers
  4. gemeinsamer Versand an Empfänger
- pro Empfänger ein verschlüsselter session key

die Vertrauensfrage

- Wem gehört welcher Schlüssel?
- Schlüssel können zertifiziert werden
- vertraue ich einem Zertifikat, kann ich eventuell dem Schlüssel trauen
- Zertifizierungsmodelle:
  - zentrale Behörde: *Certification Authority* (CA)
  - hierarchische Struktur: *introducers*
  - *Web of Trust*: jeder kann jeden Schlüssel zertifizieren

Web of Trust

- jeder kann jeden Schlüssel zertifizieren
- jeder kann als introducer fungieren
- Voraussetzung: gesicherte Erkenntnis über Eigentümer
- indirektes Vertrauen
- *level of trust*
- EMail-Absender **keine** Garantie
- fingerprint

Benutzung von PGP

- Erzeugung von zwei Schlüsselpaaren
- Import und Export von Schlüsseln
- Dateien verschlüsseln
- Dateien entschlüsseln
- Dateien signieren

Schutz der eigenen Schlüssel

- PGP faßt Schlüssel zu *key rings* zusammen
- Dateien im home directory, `$HOME/.pgp/`
- `.pgp`-Verzeichnis privat: `rwx------`
- Dateien darin: `rw-------`
- private keys durch *pass phrase* gesichert
- pass phrase schützen:
  - nicht aufschreiben
  - nicht über ein Netzwerk eintippen

key generation

```
juser@jake pgp -kg
Pretty Good Privacy(tm) Version 6.5.1i
(c) 1999 Network Associates Inc.

Export of this software may be restricted by the U.S. government.

Choose the public-key algorithm to use with your new key
1) DSS/DH (a.k.a. DSA/ElGamal) (default)
2) RSA
Choose 1 or 2: 1
Choose the type of key you want to generate
1) Generate a new signing key (default)
2) Generate an encryption key for an existing signing key
Choose 1 or 2: 1
Pick your DSS ‘‘master key’’ size:
1)  1024 bits- Maximum size (Recommended)
Choose 1 or enter desired number of bits: 1024
Generating a 1024-bit DSS key.
```

key generation, cont.

```
You need a user ID for your public key.  The desired form for this
user ID is your name, followed by your E-mail address enclosed in
<angle brackets>, if you have an E-mail address.
For example:  John Q. Smith <jqsmith@nai.com>
Enter a user ID for your public key: Joe User <juser@Genetik.Uni-
Bielefeld.DE>

Enter the validity period of your signing key in days from 0 - 10950
0 is forever (the default is 0): 0

You need a pass phrase to protect your DSS secret key.
Your pass phrase can be any sentence or phrase and may have many
words, spaces, punctuation, or any other printable characters.

Enter pass phrase:
Enter same pass phrase again:
```

key generation, cont.

```
PGP will generate a signing key. Do you also require an
encryption key? (Y/n) y
Pick your DH key size:
1)  1024 bits- High commercial grade, secure for many years
2)  2048 bits- "Military" grade, secure for forseeable future
3)  3072 bits- Archival grade, slow, highest security
Choose 1, 2, 3, or enter desired number of bits: 2

Enter the validity period of your encryption key in days from 0 - 10950
0 is forever (the default is 0):


Note that key generation is a lengthy process.
.......******* ..******* .
Make this the default signing key? (Y/n) y

Key generation completed.
```

key management

```
juser@jake pgp -kvv
Pretty Good Privacy(tm) Version 6.5.1i
(c) 1999 Network Associates Inc.

Export of this software may be restricted by the U.S. government.


Type bits      keyID      Date       User ID
DSS  1024      0x0549197E 2000/07/24
 DH  2048      0x0549197E 2000/07/24 *** DEFAULT SIGNING KEY ***
                              Joe User <juser@Genetik.Uni-Bielefeld.DE>
sig            0x0549197E     Joe User <juser@Genetik.Uni-Bielefeld.DE>
1 matching key found.
```

public key hinzufügen

```
juser@jake pgp -ka jc.pgp

Looking for new keys...
DSS  2048/1024 0x1E7AE331 2000/07/24 Joern Clausen <jc@Genetik....
sig?           0x1E7AE331          (Unknown signator, can’t be checked)
RSA  1024      0x3029F2F9 1996/04/03 Joern Clausen <joern@TechFak....
DE>
sig?           0x3029F2F9          (Unknown signator, can’t be checked)
sig?           0x6130F755          (Unknown signator, can’t be checked)
sig?           0x93F38EE1          (Unknown signator, can’t be checked)
sig?           0xEE71F001          (Unknown signator, can’t be checked)

keyfile contains 2 new keys. Add these keys to keyring ’’? (Y/n) y

Keyfile contains:
   2 new key(s)
```

public key signieren

```
juser@jake pgp -ks jc@Genetik.Uni-Bielefeld.DE

Key for user ID: Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>
1024-bit DSS key, Key ID 0x1E7AE331, created 2000/07/24
  Key fingerprint =  7F 9F 99 C2 2F 9B 91 AF  CC 97 AA 30 92 87 1C CC
                     1E 7A E3 31

READ CAREFULLY:  Based on your own direct first-hand knowledge, are
you absolutely certain that you are prepared to solemnly certify that
the above public key actually belongs to the user specified by the
above user ID (y/N)? y

You need a pass phrase to unlock your secret key.
Key for user ID "Joe User <juser@Genetik.Uni-Bielefeld.DE>"

Enter pass phrase:
Passphrase is good
```

trust level festlegen

```
juser@jake pgp -ke ’jc@Genetik.Uni-Bielefeld.DE’

Key for user ID: Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>
1024-bit DSS key, Key ID 0x1E7AE331, created 2000/07/24
No secret key available.  Editing public key trust parameter.
This key/userID association is fully certified.

Current trust for this key’s owner is: untrusted

Make a determination in your own mind whether this key actually
belongs to the person whom you think it belongs to, based on available
evidence.  If you think it does, then based on your estimate of
that person’s integrity and competence in key management, answer
the following question:

Would you trust "Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>"
to act as an introducer and certify other people’s public keys to you?
(1=I don’t know (default). 2=No. 3=Usually. 4=Yes, always.) ? 2
```

Datei verschlüsseln

```
juser@jake pgp -ea nachricht.txt ’jc@Genetik.Uni-Bielefeld.DE’

Recipients’ public key(s) will be used to encrypt.

Key for user ID: Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>
1024-bit DSS key, Key ID 0x1E7AE331, created 2000/07/24

Transport armor file: nachricht.txt.asc

juser@jake cat nachricht.txt.asc
-----BEGIN PGP MESSAGE-----
Version: PGP 6.5.1i

hQIOA0bHkoh+i6FbEAgA8MsP+wIODVGMSPnXk5J4TUqOY8Zd6Wr0tsvzoASFUfmc
N1RfHLCi4jiFj5ev/AxIfFOwhfhE8Rp6ppI+Nmlbhqs3wBgztX0KgbjdVqVh3mmt
...
j4PTImw7XMiej1YjQzoKy+Li2dENWw==
=nc7h
-----END PGP MESSAGE-----
```

Datei entschlüsseln

```
jc@fasil pgp nachricht.txt.asc

File is encrypted.  Secret key is required to read it.

Key for user ID: Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>
1024-bit DSS key, Key ID 0x1E7AE331, created 2000/07/24
Key can sign.
You need a pass phrase to unlock your secret key.

Enter pass phrase:

Plaintext filename: nachricht.txt

jc@fasil cat nachricht.txt
Hallo!
Das hier ist ganz geheim.
jc@fasil ll nachricht.txt
-rw-------   1 jc       support        33 Jul 25 12:53 nachricht.txt
```

Datei signieren

```
jc@fasil pgp -sta nachricht.txt

A secret key is required to make a signature.
You need a pass phrase to unlock your secret key.
Key for user ID "Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>"

Enter pass phrase:

Passphrase is good

Transport armor file: nachricht.txt.asc

jc@fasil cat nachricht.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hallo!
Das hier ist ganz geheim.

-----BEGIN PGP SIGNATURE-----
Version: PGP 6.5.1i

iQA/AwUBOX6FkZKHHMweeuMxEQLJFACgj22HfsIHAfz0bEvKmPlVjWd8JdAAoLNd
AP+4b6vo5ThWj9ofZWH7ZCwu
=pXn8
-----END PGP SIGNATURE-----
```

Signatur überprüfen

```
juser@jake pgp nachricht.txt.asc
Pretty Good Privacy(tm) Version 6.5.1i
(c) 1999 Network Associates Inc.

Export of this software may be restricted by the U.S. government.

File is signed.  Good signature from user "Joern Clausen ...
Signature made 2000/07/25 13:01 GMT

Plaintext filename: nachricht.txt
```

detached signature

- Dokument und Signatur gemeinsam in einer Datei
- ungünstig, wenn mehrere Signaturen gewünscht
- `pgp -sba nachricht.txt`

```
juser@jake pgp nachricht.txt nachricht.txt.asc

File ’nachricht.txt.asc’ has signature, but with no text.
Text is assumed to be in file ’nachricht.txt’.
Good signature from user "Joern Clausen <jc@Genetik.Uni-Bielefeld.DE>".
Signature made 2000/07/25 13:05 GMT
```

public key verschicken

```
juser@jake pgp -kxa ’juser@Genetik.Uni-Bielefeld.DE’

Extracting from keyring ’/homes/juser/.pgp/pubring.pkr’, userid "ju-
ser@Genetik.Uni-Bielefeld.DE".

Extract the above key(s) into which file? juser

Transport armor file: juser.asc

Key extracted to file ’juser.asc’.

juser@jake cat juser.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 6.5.1i

mQENAzl8PbYAAAEIALtpr16W6W73jvP4bB67VJ7+6IZ1ZfzZwBk0nIalLqZSBQhB
...
=dmhI
-----END PGP PUBLIC KEY BLOCK-----
```

Dokumentation und Referenzen

- http://www-intern.Genetik.Uni-Bielefeld.DE/docs/PGP/
  - Folien
  - *An Introduction to Cryptography*
  - *PGP Command Line Guide*
  - links
- http://www.pgpi.org
- http://www.pgp.com
- http://www.gnupg.org
- http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm