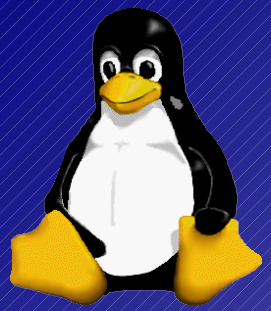
Cryptography Basics

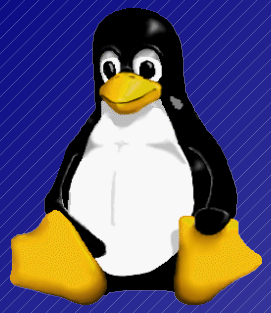
OpenPGP:

PGP & GnuPG



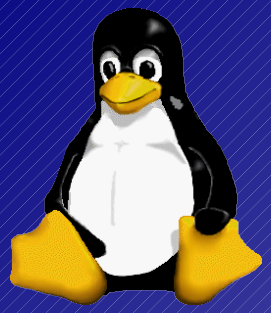
OpenPGP

- open standard for cryptographic formats (mail/file encryption)
  - RFC 2440
- implementations:
  - PGP (original, commercial)
  - GnuPG (free software)
- Protocols:
  - PGP/MIME (RFC 3156)



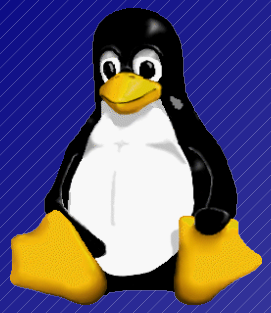
...versus X.509

- X.509: ITU standard for key formats
- used in:
  - SSL, TLS (RFC 3546), eg.:
    - HTTPS (RFC 2818)
    - SMTP-Sec (RFC 2487)
  - S/MIME (RFC 2311)
- Philosophy:
  - hierarchical authentication
  - complicated formats



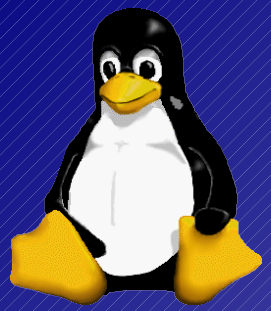
OpenPGP can...

- handle files and mail
- encrypt
  - to one or more recipients using their public keys
  - using one-time password derived symmetric keys
- sign
  - with the users secret key
- check signatures
- handle public keys



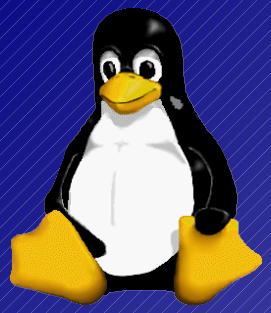
why bother?

- Don't trust the net!
  - Electronic communication can be easily sniffed and analyzed (eg. by criminals)
- Don't trust the government!
  - ...and is done so routinely (eg. Echelon)
- Don't trust the competition!
  - ...eg. for their own countries businesses.
- consider:
  - Why is “snail mail” put into envelopes?
  - Why is cryptography forbidden in all dictatorial states?



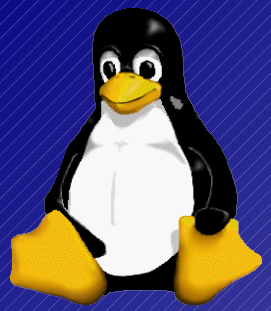
Security Model

- secures communication over an insecure data path (Internet mail)
- secures local data from the casual curiosity and thieves
  - if mail is stored encrypted and files are routinely encrypted
  - if the passphrase is good



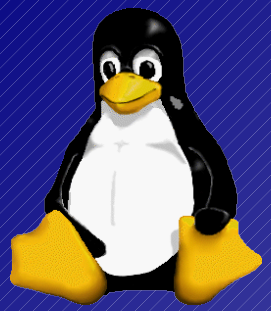
Security Model

- PGP does NOT replace:
  - the need for a firewall
  - antivirus software
  - trust into your admin/family (or anyone else who could sneak a keylogger in)
  - trust into the producers of all the software you installed (including Operating System and PGP)



Keys

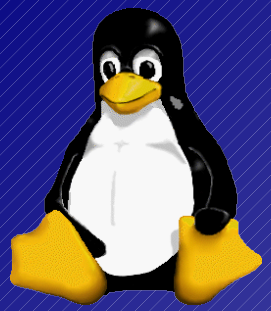
- Based on Public Key algorithms:
  - RSA
  - ElGamal
  - DSA
- Key data contains:
  - Key material
  - UIDs (Name + eMail-address)
  - self-signature(s)
  - signatures



Key Format

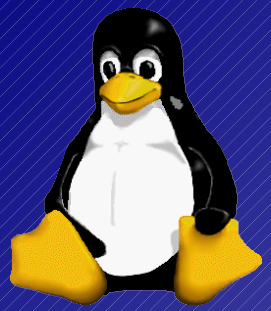
- The key is stored in a hierarchy:

- Key-Material
  - UID 1
    - self-signature
    - signatures
    - revocation signatures
  - UID 2
    - self-signature 2
    - signatures
    - revocation signatures



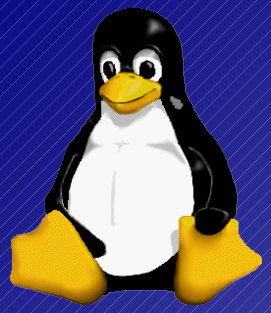
Key example

```
pub  1024D/0F4648C4 2000-05-20 Konrad Rosenbaum <konrad.rosenbaum@gmx.net>
sig         0F4648C4 2001-03-26  Konrad Rosenbaum <konrad.rosenbaum@gmx.net>
sig         98016DC7 2002-06-03  Josef Spillner <dr_maux@users.sourceforge.net>
sig         1242A6F2 2002-06-08  Simon Hausmann <hausmann@kde.org>
sig         28FA388A 2002-06-08  Matthias Kretz <kretz@kde.org>
sig         4456536A 2002-06-11  Holger Freyther (zecke) <freyther@gmx.net>
sig         0485B101 2002-06-11  Nikolas Zimmermann <wildfox@kde.org>
sig         2028C057 2002-06-10  Carsten Niehaus <cniehaus@gmx.de>
sig         30E0B9D8 2002-06-16  Ingo Klöcker <ingo.kloecker@epost.de>
sig         B3B2A12C 2002-06-26  ct magazine CERTIFICATE <pgpCA@ct.heise.de>
sig         13E290A4 2002-06-30  Eva Brucherseifer <eva@rtr.tu-darmstadt.de>
uid                              [revoked] Konrad Rosenbaum <htw6966@htw-dresden.de>
rev         0F4648C4 2002-06-10  Konrad Rosenbaum <konrad.rosenbaum@gmx.net>
sig         0F4648C4 2000-05-20  Konrad Rosenbaum <konrad.rosenbaum@gmx.net>
sig         98016DC7 2002-06-03  Josef Spillner <dr_maux@users.sourceforge.net>
```



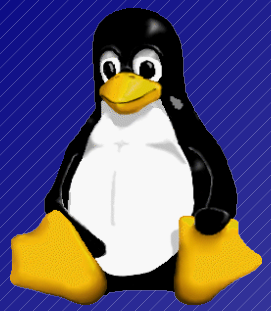
Signatures

- Self-Signature
  - acknowledges that the holder of the key uses this UID
- Signature
  - acknowledges that the holder of the signing key checked that this UID belongs to the key holder



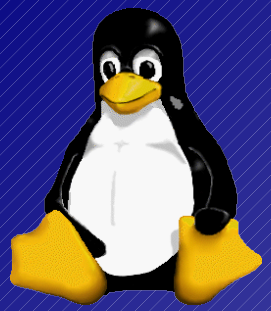
Revocation Signatures

- Self-Signature revocation
  - the holder of the key no longer uses this UID, because:
    - eMail address is no longer valid
    - the key is insecure (all UIDs are revoked then)
    - the key has been compromised (all UIDs)
- Signature revocation
  - the holder of the signing key no longer believes the connection between key holder and UID exists



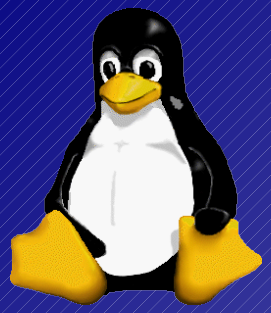
Trust

- Signatures and identities are verified, trust is given.
- Trust is not stored in keys!
- Alice trusts Bob means:
  - Alice trusts that any key/UID signed by Bob is valid without verifying herself.



Key exchange

- direct: Floppy
- indirect:
  - eMail
  - Web-site
  - Key-Servers (eg. pgp.net)
  - !needs key authentication!



Key authentication

- Each key is identified by:
  - KeyID (eg. 0F4648C4)
  - Fingerprint (eg. B333 F8FB 644A D695 F494 7068 9BAA 4EEC 0F46 48C4)
    - MD5 (PGP2.x) or SHA-1 (newer) Hash of the key material
- Authentication must be done securely:
  - key owner identifies himself (eg. passport)
  - key owner announces his fingerprint
  - key recipient compares the fingerprint

Questions?

?