

Asymmetric cryptography

Mailing with GnuPG

Moritz Bubek

Outline

- Motivation
- Encryption/Decryption
- Asymmetric En-/Decryption & RSA
- PGP / GnuPG
- Web of Trust

Why talking about this?

- Linux on Simons Computer
- Moritz should help
- long long procedure
- Moritz needs account
- Simon creates account ...
- Simon want to send data, **BUT**



Bundesnachrichtendienst





DIE 007 *ANOTHER* Day

11.22

JAMES
BOND

www.mgm.com
www.jamesbond.com



© 2001-2002 METRO-GOLDWYN-MAYER STUDIOS, INC. ALL RIGHTS RESERVED.
DIE ANOTHER DAY™ TRADEMARK UNITED ARTISTS CORPORATION AND DANJAO, LLC.
JAMES BOND MATERIALS © 1999-2002 UNITED ARTISTS CORPORATION AND DANJAO, LLC.
007 GUN LOGO, JAMES BOND, AND ALL OTHER JAMES BOND RELATED TRADEMARKS™ DANJAO, LLC.





Who and Why

- email is less confidential than a postcard
- mailserver administrator, mails stored on the server !
- hacker penetrate servers
- secret services, espionage
- but also preventing criminality

what to do: encryption

- encrypt your communication
- very old idea (Babylon, Greek, Rome, ...)
- just replace the characters (rot13)
- easy to crack with statistic methods
- better: encryption with a key
(IDEA, DES, Blowfish, ...)
- text XOR key
- only OTP really secure !

Basics

Alice

Text

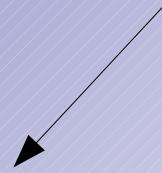
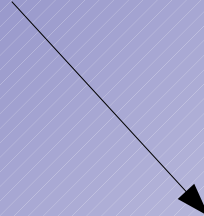
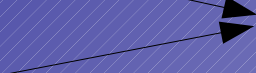
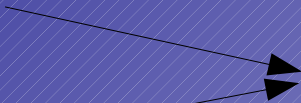
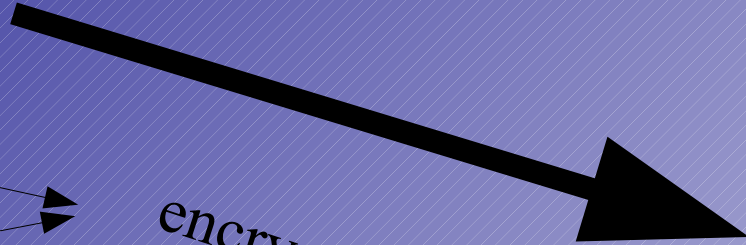
Key

encrypted message

Bob

Key

Text



nevertheless it does not work

- Problem 1: you have to transfer the key through unsecure channels
Eve catches the key and everything is lost
- Problem 2: is my partner really my partner ?
authentication problem
- Problem 3: needs $n(n-1)/2$ keys !

asymmetric Encryption

- pair of two key parts
- public key
- private key
- encrypt with public key
decrypt with private key
- trapdoor-algorithm

example: RSA

- guess two primes p and q (500 digits)
- calculate product $N = p * q$
- Euler Function $\Phi(N) = (p-1)(q-1)$
- guess e with $1 < e < \Phi$, coprime (teilerfremd)
- calculate d , with $e * d \bmod \Phi = 1$ (erw. eukl Alg)
- publicKey: e, N ; privateKey: d, N
- delete the rest !

Use of RSA

- Encrypt

$$C = T^e \bmod N$$

- Decrypt

$$T = C^d \bmod N$$

Is RSA secure?

- if Eve knows Φ she could calculate d easily
- just knowing N --> prime factor segmentation
- key length over 1024 Bit seems to be secure
-

Using RSA in real life: PGP / GPG

- PGP: Zimmermann 1991
- opensource alternative because of patents
- keys on keyserver
- `gpg --gen-key`
- name and emailaddress, protectet by a Mantra
- important: create a revoke key
(`gpg --gen-revoke keyID`)

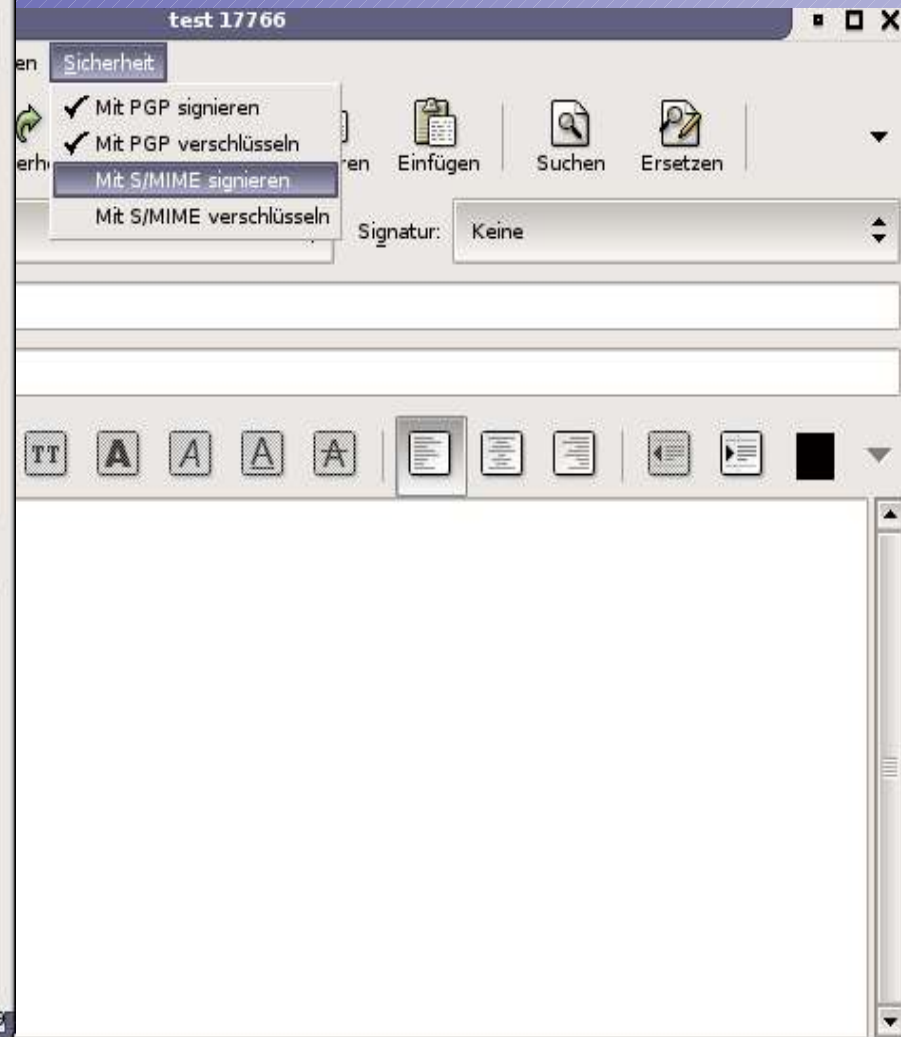
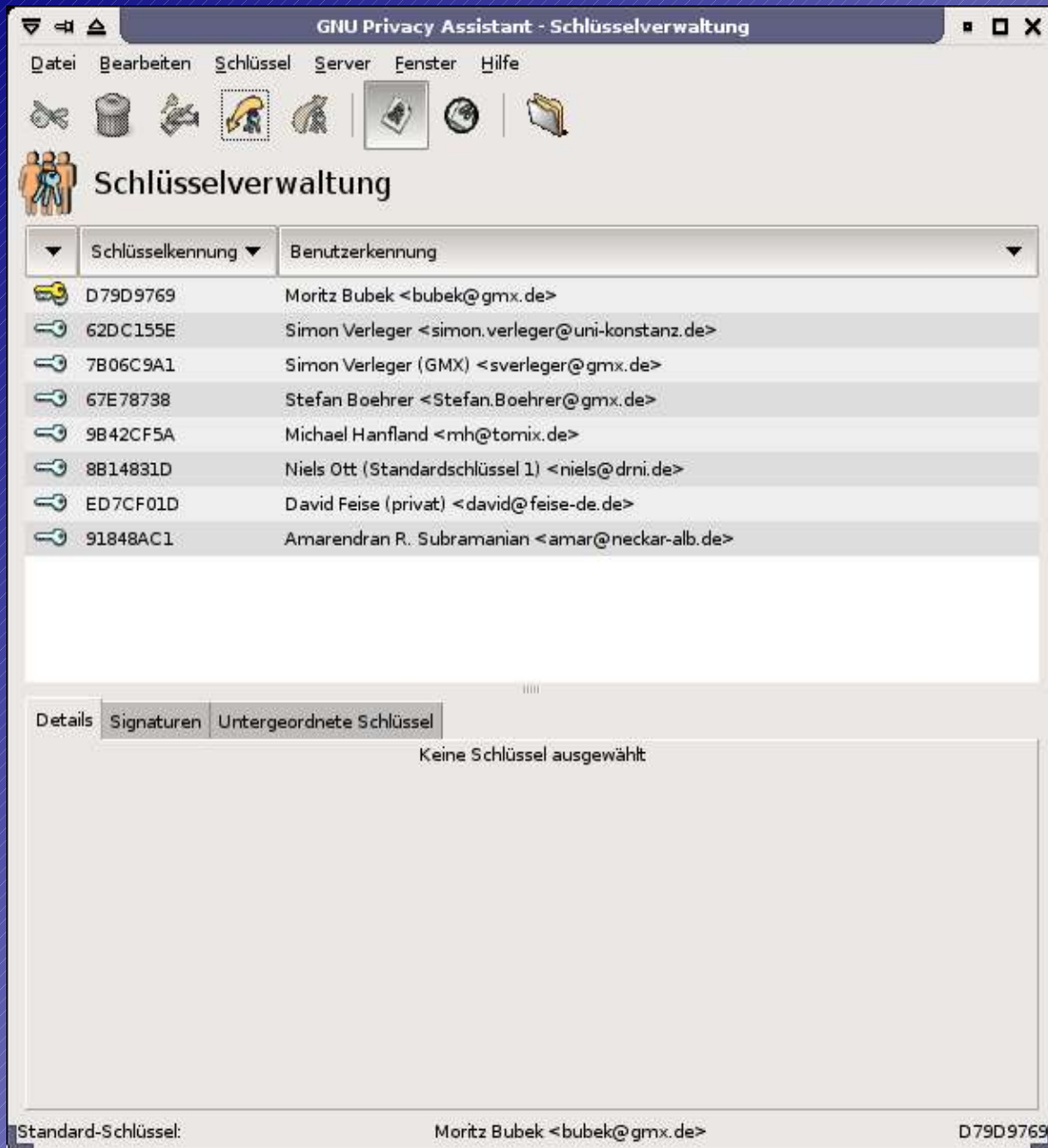
Export/Import

- `gpg --export bubek@gmx.de`
- `gpg --search Dietsche`
- signing keys
`gpg --edit-key`
`sign`
- editing, revoking

Encrypt/Decrypt

- `gpg --encrypt --recipient sverleger@gmx.de`
- `gpg --decrypt file`
- directly in the mail program (plugin?!)

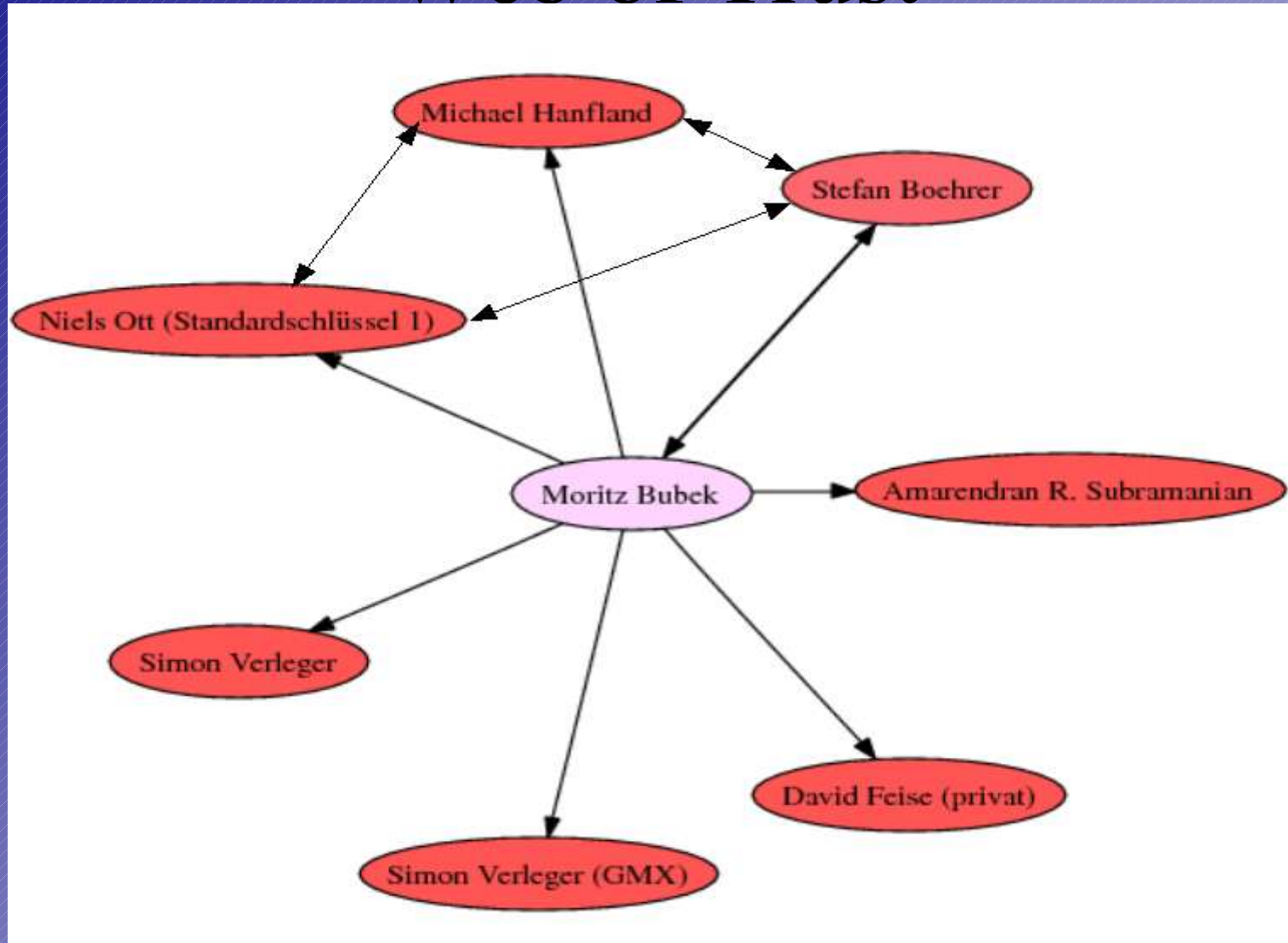
More comfortable?

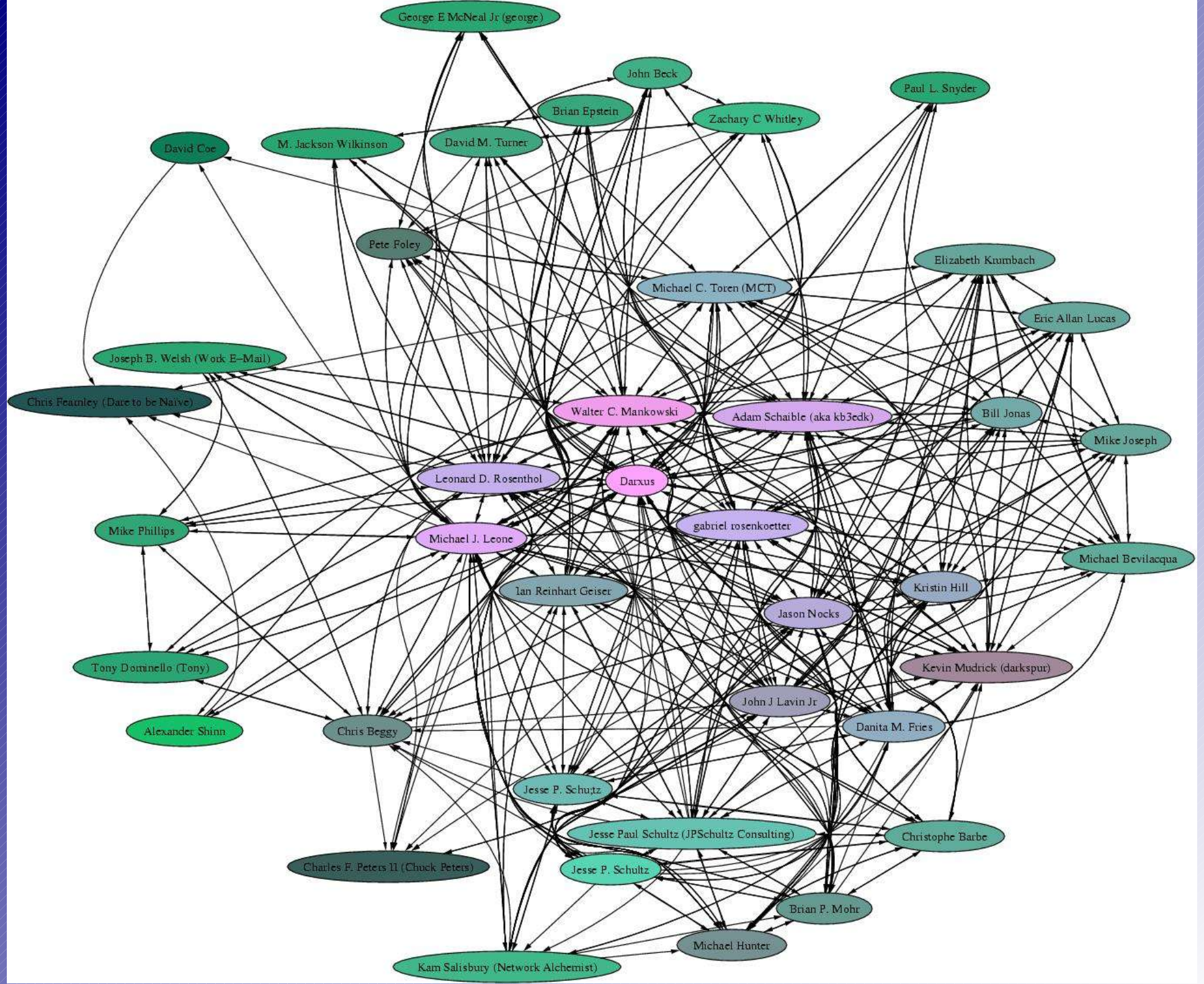


Who is my communication partner?

- problem: authentication
- solution: signature encrypted with the privateKey
(inverse procedure !)
- hash of the text
- receiver is able to decrypt with public key
- if calculated hash matches the decrypted --> ok

Web of Trust





Conclusion

- possibilities available
- nobody uses
- YOU should create your own key !!!
- key signing party
- bubek@gmx.de

8419 5A80 F1C8 098A EFF8 7669 B6EE DB31 D79D 9769

Sources

- www.gnupg.org
- de.wikipedia.de
- www.lysator.liu.se/~ceder/
- www.chaosreigns.com
- myself