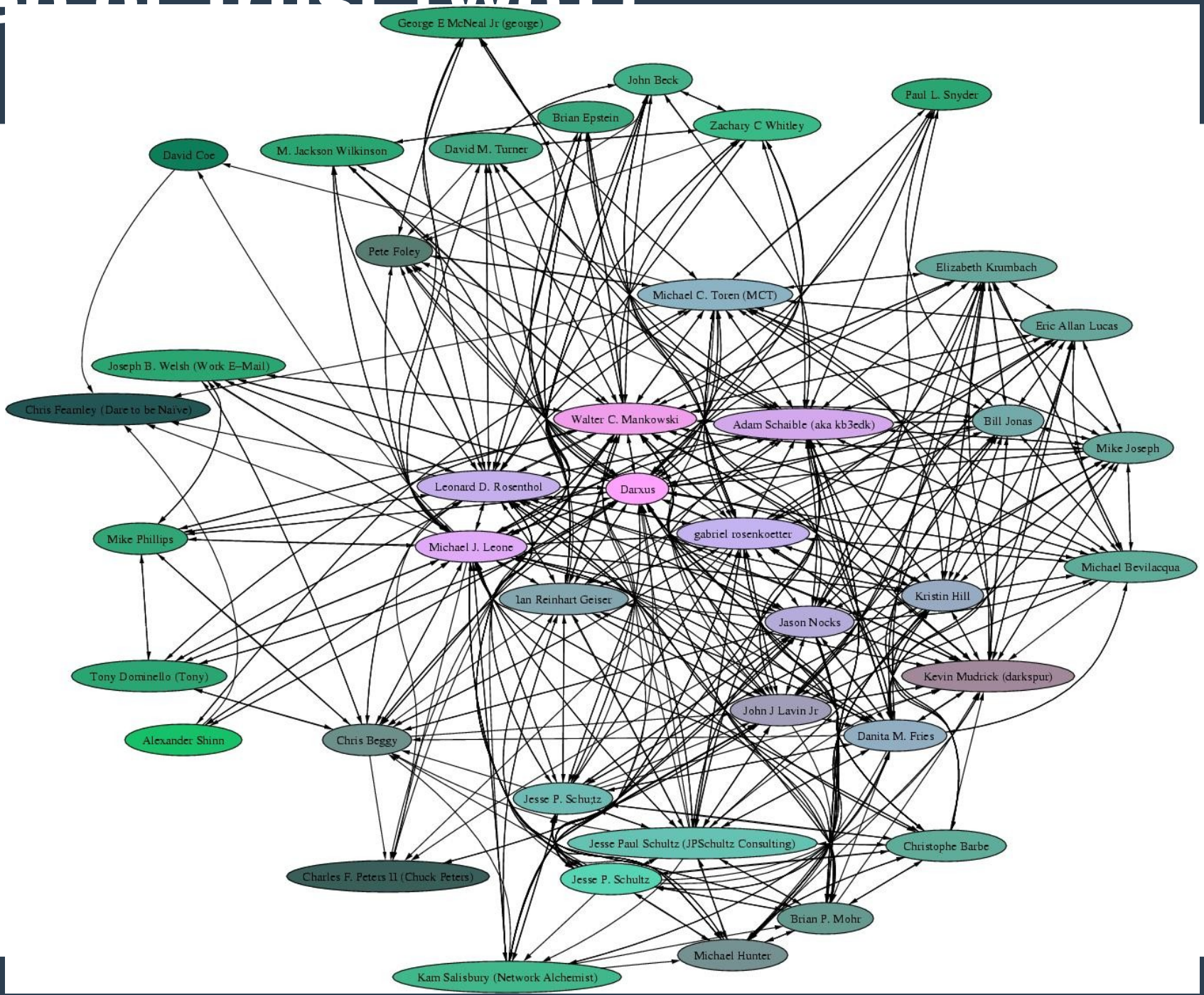
Practical GnuPrivacyGuard (GPG)

Joshua I. James

Seoul Tech
Society

Web of Trust (WoT)



GnuPG – what is it?

- Free, open implementation of OpenPGP standard (PGP)
- Tools for public key encryption
- https://www.gnupg.org/

GnuPG – where to get it?

- Windows (GPG4Win) -
  - **http://gpg4win.org/download.html**
- OSX (GPGTools) -
  - **https://gpgtools.org/gpgsuite.html**
- **Debian/Ubuntu - apt-get install gnupg2**
  - Similar in other package managers (rpm)
  - Source: **www.gnupg.org/download**

These are recommended – other options exist

GnuPG – Getting started

- Commands may be slightly different depending on your system
- We will do everything from cli – most systems also have a gui interface
- Help: **gpg2 --help**

GnuPG – Overview

- **We are going to:**
  - Generate your own public / private keys
  - Edit your keys (add identities, change hash preferences)
  - Create revocation certificates
  - Backup your public/private keypair
  - Send your public key to a keyserver
  - Sign / verify documents / software
  - Encrypt / decrypt documents
  - Sign other people's keys

Creating a keypair

- List keys: **gpg2 --list-keys**
- Create a new key: **gpg2 --gen-key**
- Choose the algorithm you want (**RSA recommended**)
- Choose the keysize you want (**4096 recommended**)
- Choose how long the key will be valid (**3 years recommended**)
- Enter your **real name** [!important!]
- Enter your email
- Enter a **strong** password
- Wait for entropy
- List keys: **gpg2 --list-keys**

Editing a keypair

- Edit key: **gpg2 --edit-key [key ID]**
- See key preferences: **showpref**
- Edit key preferences:

  setpref SHA512 SHA384 SHA256 SHA224 AES256 AES192 AES CAST5 ZLIB BZIP2 ZIP
- Add an identity: **adduid**
  - Fill in the identity info like before
- Save your changes!!!: **save**

Create revocation certificate

- List keys: **gpg2 --list-keys**
- Create revocation cert:

  gpg2 --output [keyID]-gpg-revocation-cert --gen-revoke [key ID]

- Select reason **1 = key has been compromised**
- Save the file [keyID]-gpg-revocation-cert **offline** – used only for emergencies

Send public key to keyserver

- List keys: **gpg2 --list-keys**
- Send keys to public key server:

  gpg2 --send-keys [keyID]

Sign / verify documents

- List keys: **gpg2 --list-keys**
- Find a text file, and a jpg to practice on
- Sign a file (encrypt with private key):

  gpg2 --sign [filename]
- Verify signature:

  gpg2 --verify [filename].gpg
- Decrypt signed file:

  gpg2 --decrypt [filename].gpg

Sign / verify documents

- List keys: **gpg2 --list-keys**
- Detach-sign a file:

  gpg2 --detach-sign [filename]
- Verify detached signature:

  gpg2 --verify [filename].sig
- Verifying detached signature requires the original file and the signature

Encrypt / decrypt documents

- List keys: **gpg2 --list-keys**
- Encrypt **for yourself**:

  gpg2 -r [your key ID] --encrypt [filename]
- Decrypt file (sent to you):

  gpg2 --decrypt [filename].gpg

Getting and trusting keys

- List keys: **gpg2 –list-keys**
- We want to establish how much we can *trust* the key
- What information should you get from your partner?
  - Full fingerprint:
    - **gpg2 --list-keys --fingerprint [key ID]**
  - Photo ID
  - Verify their email (have them send their full fingerprint via email)
  - Import their key
  - Sign their key with level of trust
  - Email their key back to them / upload to keyserver
- **Do not share your private key**

Sharing public keys

- **Share your public key information here: https://goo.gl/vZcebM**
- **Name, email, full fingerprint**
- **We will print, and everyone can verify everyone elses keys while we party to make STS Web of Trust**

Get person's key

- List keys: **gpg2 –list-keys**
- Get key from keyserver:

  gpg2 --recv-keys [keyID]
- Get key from file share:

  gpg2 –-import [public key filename]
- **Do not share your private key**

Sign person's key

- List keys: **gpg2 –list-keys**
- Sign key:

  gpg2 --edit [peron's keyID]

  trust

  sign

  save
- Check signatures:

  gpg2 --check-sigs [person's keyID]
- Send signed key to keyserver:

  gpg2 --send-keys [keyID]

Send encrypted & signed files to partner

- Create two text files with different messages
- **Encrypt** the first text file **with your partner's public key**
- **Sign** the second text file
- Send the encrypted and signed files to your partner
- Decrypt both messages
  - Decryption requires partner's **private key**
  - Signed-file decryption requires the signer's **public key**

Use for encrypting email / on mobile

- Most email clients have plugins to support GPG encryption
- Thunderbird – Enigmail:
  - www.enigmail.net
- Mail (OSX) – GPG Tools:
  - https://gpgtools.org/
- Android:
  - K9Mail with APG