PGP everywhere

How it helps us in everyday life

Michal Hrušecký
Michal@Hrusecky.net

Cryptography 101 - Soap opera

Alice meets Bob, they want to chat.

But world and especially Mallory and Eve are against it.

Eve is just a harmless stalker.

Mallory tries to actively break them apart by pretending to be one or another and sending them fake messages…

Little bit of crypto-theory

Symmetric ciphers

- same key for encryption and decryption
- both parties have to share same secret
- the secret key has to remain secret

Asymmetric ciphers

- two keys - public and private
- Alice uses her private and Bobs public to encrypt
- Bob uses his private and Alices public to decrypt
- slower than symmetric
- public keys can be public

History

- 1977 Ron Rivest, Adi Shamir and Leonard Adleman published RSA
- 1985 Koblitz & Miller suggests using elliptic curves in cryptography
- 1991 Phil Zimmermann wrote PGP (custom symmetric cryptography)
  - made it available including sources
- 1992 Web of Trust protocol - PGP 2.0
- 1993 criminal investigation of Zimmermann
  - "munitions export without a license"
- 1995 book with sources
  - protected under first amendment
- 1996 criminal investigation ended, PGP Inc. founded

History

- 1997 PGP 5 supporting RSA released
- 1997 OpenPGP proposed to IETF
- 1998 RFC2440 OpenPGP message format
- 1999 GPG 1.0 released
- 2012 signing and key exchange using ECC in RFC6637
- 2014 support for encryption using ECC proposed
- 2014 GPG 2.1 released with ECC support

How does it work

Encryption/Decryption

- generate random symmetric key
- encrypt message using symmetric key
- encrypt symmetric key using asymmetric one
- send encrypted text and encrypted key
- decrypt key and decrypt the message

Signing

- compute message digest
- sign the digest using asymmetric key

Typical use-case

Mail

- most well known use-case
- signing mails
- encrypting mails
- most e-mail clients have some integration available
    - usually really user friendly

Signing releases

- Linux distributions

Pass

- password manager with unix philosophy in mind
- each password lives inside of a gpg encrypted file
- these encrypted files may be organized into folder hierarchies
- all passwords in `~/.password-store`
- `pass` is just a simple script to manage it
  - `pass -c web/github` - copy github password into clipboard
  - `pass otp -c web/github/totp` - get totp code

Secure Shell

You can use gpg to authenticate

If you have authentification subkey

Enable it in `~/.gnupg/gpg-agent.conf`

```
enable-ssh-support
```

Add a key to `.ssh/authorized_keys`

Get public key by running `gpg --export-ssh-key keyid`

Keeping key secure

- keep it on secure computers
  - you are root
  - no one else has access to
  - encrypted harddrive
  - ideally no internet connection
- preferably use a subkey that you can throw away

Or use hardware token!

- various available on market: Yubikey, Nitrokey, …
- use the open source one - Gnuk

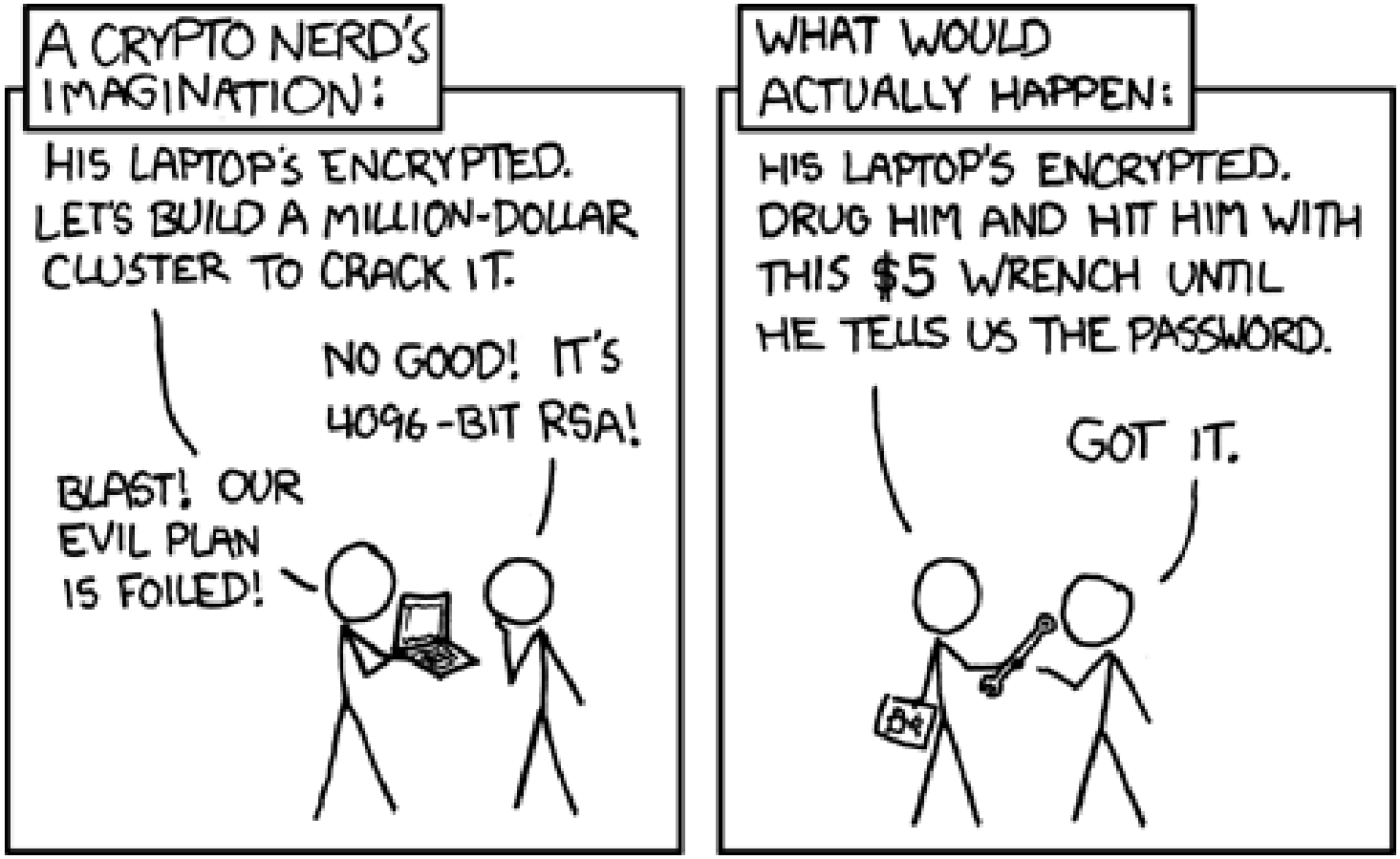
GNUK

- Free Software Initiative of Japan project
- opensource software
- originally based on STM32F103
- now supports more chips and even Linux emulation
- you can buy HW for few bucks
- you can compile SW by yourself and flash it
- fuse to protect flash readout
- key is encrypted on flash anyway
  - using AES and your PIN
  - optional experimental KDF-DO feature

http://www.fsij.org/category/gnuk.html

How does it work for real

Conclusion by XKCD



Actual actual reality: nobody cares about his secrets.

(Also, I would be hard-pressed to find that wrench for $5.)