OMG Seekrits

An introductory GnuPG Guide

Ben Kero

2011.10.25

What is GnuPG?

GnuPG == **GNU Privacy Guard**

- OpenPGP Implementation
- GPL Licensed

Start installing it now

- Linux: (apt-get|yum|emerge) gnupg
- Mac OS X: **MacGPG**
- Windows: **Gpg4win**

It can be built into your mail client too!

- Thunderbird: **Enigmail**
- Evolution: Built in, **RTFM**
- Mutt: Built in, **RTFM**
- Mail.app: **GPGTools**

What is it good for?

- Private communications and identity verification

In what form?

- Encrypted files
- Encrypted emails
- Electronically signed files
- Electronically signed emails

How does it work?

- Public key cryptography
  - You have two keys: **Public** and **Private**
  - Keep your private key private
  - Keep your public key public (by putting it on keyservers!)
- Messages are (optionally) signed with your private key
- Messages are encrypted with the recipient's public key
- Messages are decrypted with the recipient's private key

Lay some theory on me

- Nah, this is covered in your MTH231 class
- Encrypting is computationally cheap
- Decrypting is computationally expensive
- Decrypting is therefore hard to crack (brute force)

Getting Started

- Generate your key
  - $ gpg –gen-key
- Set a default keyserver
  - $ echo "keyserver pgp.mit.edu" >> $HOME/.gnupg/gpg.conf
- Upload your key to a keyserver
  - $ gpg –keyserver pgp.mit.edu –send-keys $KEY_ID
- Keyserver HTTP Demo

Interacting with others

- You'd like to sign someone's key and create a cryptographic trust

1. Get a copy of your partner's key and check the fingerprint
   - $ gpg –recv-keys $KEY_ID
   - $ gpg –fingerprint $KEY_ID
2. Verify the identity of your partner (and a printed copy of their fingerprint if they brought it with them)
3. Sign your partner's key
   - $ gpg –sign-key $KEY_ID
4. Upload the signed key
   - $ gpg –send-keys $KEY_ID

Let's say

- I am Bradley Manning and I have Secret_Diplomatic_Cables.docx on my Lady Gaga USB drive
- I am in Iraq, but I need to securely send the file to an associate in Switzerland for distribution
- I...
  1. Encrypt the file with my associate's public key [1]
  2. Attach to email
  3. Send it to to the recipient

[1] yes, this really did happen: http://tinyurl.com/3q8k2um

Interactive time!

- Generate a key

Interactive time!

- Generate a key
- Retrieve the recipient's key

Interactive time!

- Generate a key
- Retrieve the recipient's key
- Encrypt the message

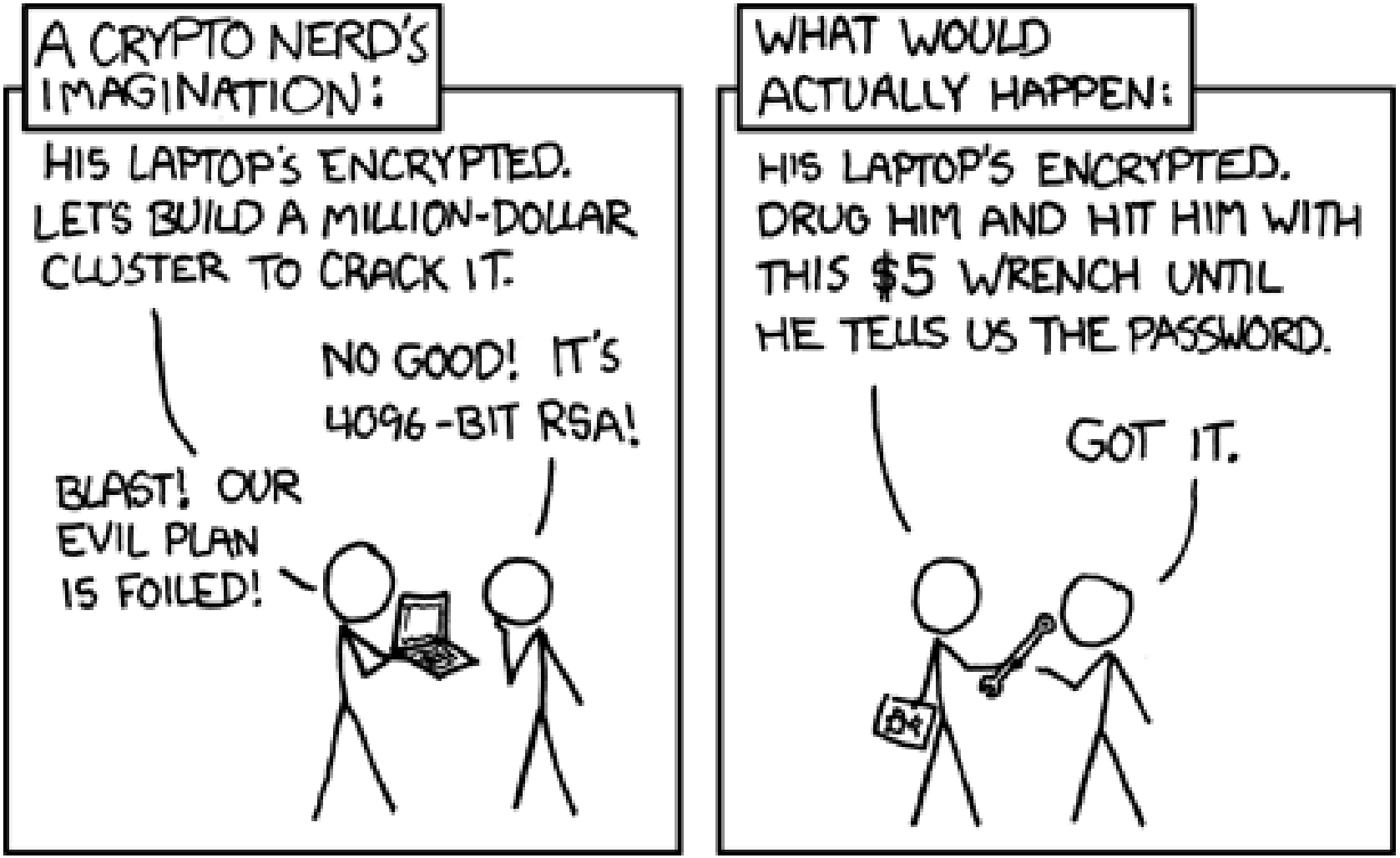
Interactive time!

- Generate a key
- Retrieve the recipient's key
- Encrypt the message
- Attach it to an email and send it

Receiving and decrypting

- Read the email and download the attachment
- Decrypting can be done with:
  - $ gpg decrypt Secret_Diplomatic_Cables.docx.gpg –output omgseekrits.docx
- There is now a decrypted omgseekrits.docx file

Any Questions?



Great

- Now let's have a key signing party!
- I hope you brought ID

Party instructions

1. Generate your key: $ gpg –gen-key
2. Upload your key to a keyserver: $ gpg –send-keys
3. Find your fingerprint: $ gpg –fingerprint|grep fingerprint|head -1

Key fingerprint = B5E9 786D 6527 A4AF 9EC9 9398 691E DEC8 CC42 4ECE

1. Write full name and Fingerprint on the BACK SIDE of your index card (no lines)
2. Go mingle with others, verify their identity, and copy their name and fingerprint down
3. Later:
   - $ gpg –recv-keys $THEIR_KEY # Retrieve their key from the keyserver
   - $ gpg –fingerprint $THEIR_KEY # Verify their fingerprint
   - $ gpg –sign-key $THEIR_KEY # Sign their key
   - $ gpg –send-keys $THEIR_KEY # Send the signature off to the keyserver