

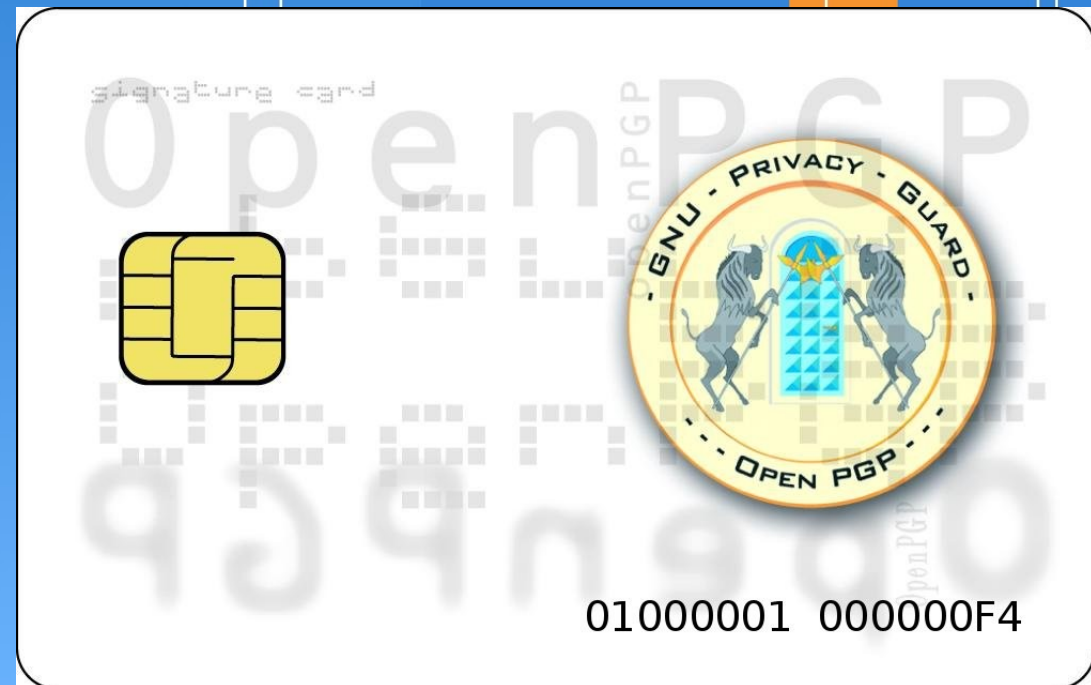
Discovering OpenPGP Card

Dany Nativel

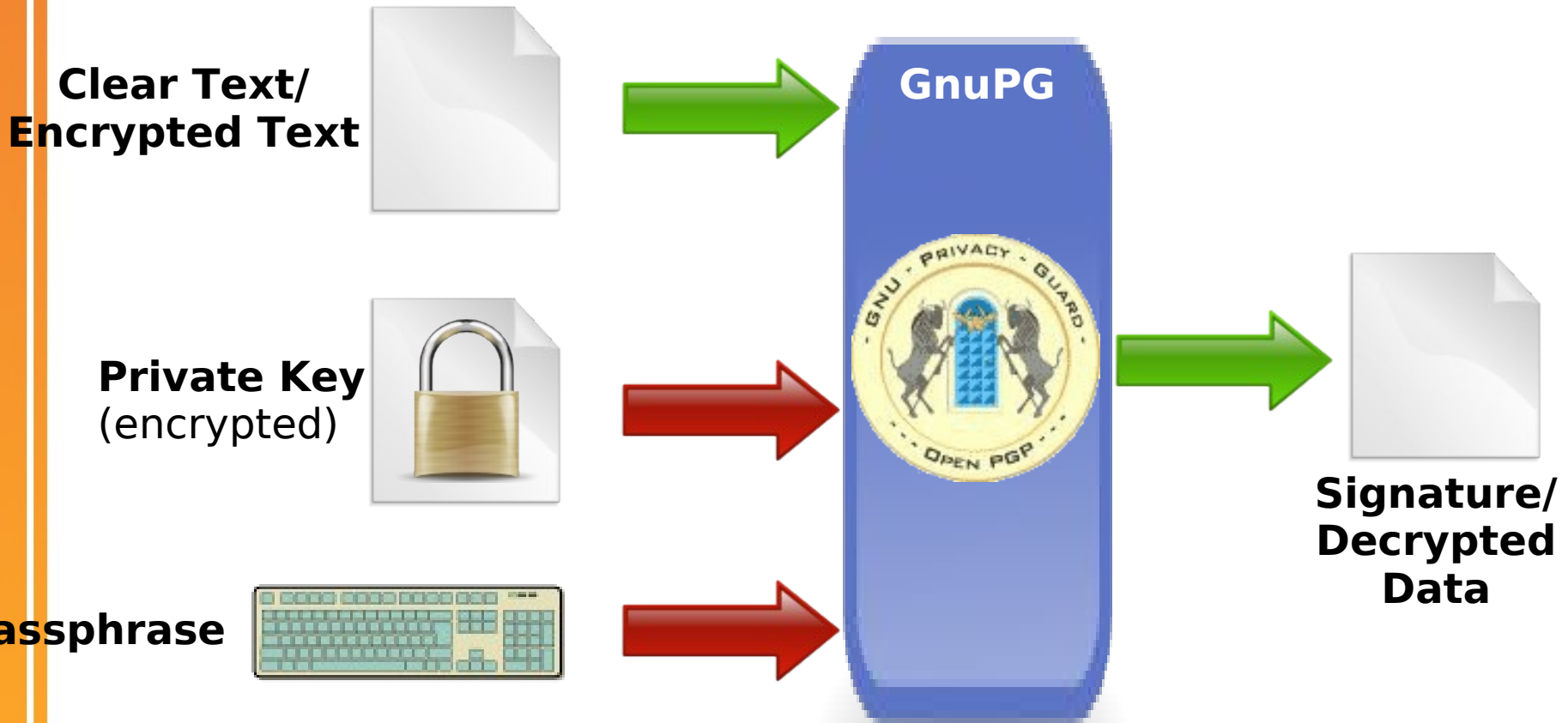
(dany AT nativel DOT net)

GPG Key fingerprint

BFD5 345A 5DA1 F3AF DD85
D91F 956E 5AD2 E089 B922



How does GnuPG signature/decryption work?



Where are the weak spots?



- Private Key :

- Malicious software (Trojans, worms) can copy and send your private key over the network
- Swap memory (disk) can leak private key material
- Backup media may contains your private key






**Easy to forge : Just a file copy.
You won't probably notice that your private key
has been copied.**

- Passphrase :

- Keylogger 
- "Over-the-shoulder" key grabbing
- Social engineering 



What happens if your secret key becomes “public” ?

- Private key is encrypted, passphrase is required and could be obtained :
 - Brute Force / Dictionary attack 
 - Keylogger 
 - Social engineering 
 - “Hello Sir, this is GnuPG corp. we need your passphrase for verification purposes ;)”

With private key+passphrase, an attacker can :

- Decrypt all messages (past and future)*
- Sign keys and messages on your behalf*

**Expiration date on keys can limit future operations*

How to protect your private key?

- Protect the key itself
 - Prefer removable media (USB Flash key, floppy)
 - Restrict access permissions on your keyring
 - Keep backup in a safe (place)



- Protect the passphrase

- Use a good quality passphrase
 - = **LONG** with **mixed** types (upper/lower/numbers..)
 - ≠ Webmail, IM or other everyday's passwords
- Don't write it down..ever!
 - No more post-it sticked on the back of your keyboard

4eR" ^p@lap^3&_+/PY7



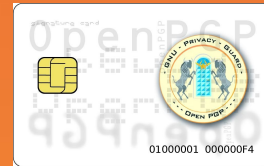
Adapt security to your concerns

- Identity theft, quality of web of trust :
 - Need to increase security for signing operations
- Protect your privacy, keep your secrets...secrets!
 - Have your private key material stored in a Smart Card

Strengthen security for signing

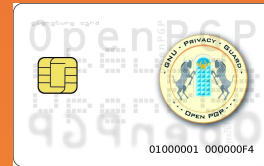
- Have different policies for signing keys/messages and decryption
 - Dedicated computer running Linux (e.g. Live-CD) and no network connection

OpenPGP card increases security



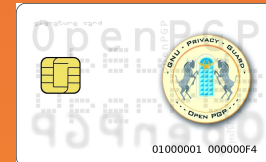
- Keeps your private key...private :
 - Private key material never leaves the card
 - Key can be generated on-card for ultimate security
 - No memory swap leak
 - Immunity to spywares and trojans
- Tamper evidence :
 - Much more difficult to copy or forge a Smart Card
 - You'll notice if someone tries as you no longer have the card in your hands
- Protection against brute force attack :
 - Card locks down after three unsuccessful attempts
 - Can be reset with Admin PIN
- Signature usage monitoring
 - A signature counter provides valuable feedback

OpenPGP Card Advantages



- Simplified GnuPG keyring management
 - Can now be backup without any security concern
 - Allows use of less trusted machines (a.k.a MS Win)
- On-card key generation possible for ultimate security
- Shorter password to remember
 - Passphrase is now called PIN
- Provides additional services :
 - SSH Login, PAM authentication
 - Signature and PIN attempts counter
 - Admin PIN
- Based on public OpenPGP Card specification
 - Implemented by GnuPG's author himself (Werner Koch)

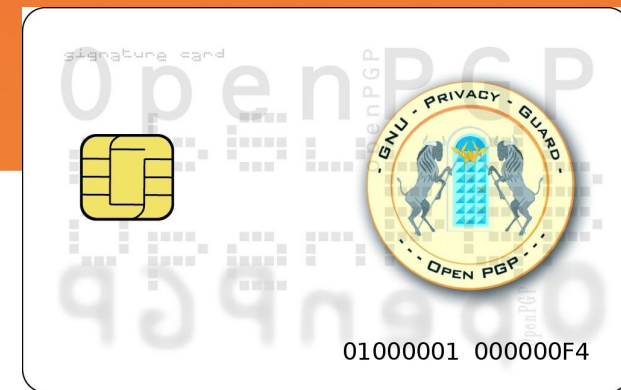
OpenPGP Card Disadvantages



- Need a card and compatible Smart Card reader
 - Export regulations limits diffusion of cryptographic hardware to several countries
 - Not cheap (reader + card), limited portability
- Learning curve
- If you ever lose your card, you lose your private key
 - Backup solutions exist
- Closed-source implementation of the card
 - Bugs ? Backdoor ?
- Based on (closed-source) Basic Card OS
 - Backdoor ?

OpenPGP Card features

- 3 independent 1024 bit RSA keys
 - Signing, encryption, authentication
- Key generation on card or import of existing keys
- FIPS 140-1 Hardware Random Generator
- Signature counter
- Data objects for :
 - Storing an URL to access the full OpenPGP public key
 - Card holder name etc.
 - Login specific data.
- Length of PIN between 6 and 254 characters
 - Not restricted to numbers
- T=1 protocol; compatible with most readers.
- Specification freely available and usable without any constraints.
- Reasonably priced.



Smart Card Reader Support

- CCID readers directly supported by GnuPG (e.g. USB SCM SCR335)
 - Works out of the box.
 - Requires libusb (<http://prdownloads.sourceforge.net/libusb>)
- CCID readers supported by pcsclite package
 - Supports mosts CCID readers on the market
 - <http://pcsclite.alioth.debian.org/ccid.html>
 - Requires libusb (<http://prdownloads.sourceforge.net/libusb>)
 - Requires libpcsclite (<https://alioth.debian.org/projects/pcsclite/>)
- Other Smart Card Readers (CTAPI Driver)
 - Various drivers available (examples):
 - libgempc410 | PC/SC driver for the GemPC 410, 412, 413 and 415 smart
 - libtowitoko2 : Towitoko smartcard reader PCSC and CT-API driver...



Software Integration

- Most frontend applications will ask for the passphrase and pass it to the card as PIN.

- Linux/*BSD : 

- gpg-agent may be required with some frontends
- Enigmail plugins for Mozilla Thunderbird:
 - Features Smart Card Management Interface
 - Provides PIN cache functionality

- Windows :

- WinPT :
 - Features Smart Card Management Interface
- Enigmail plugins for Mozilla Thunderbird



- Lots of Live-CD don't include GnuPG or offer an outdated version of it.

- Can be easily compiled

- Knoppix / kanotix / Kaella



KANOTIX



Kaella

- Works out of the box!

- Another good candidate is PuppyLinux (64MB)



- GnuPG not part of the base system but can be installed as an add-on.

- Download <http://dotpups.de/JR/gnupg.pup>
- To install, click on the file using Rox-Filer
- GnuPG can be launched using :
`/usr/local/gnupg/bin/gpg`

Can fit on a USB Flash drive

Private Key Backup Strategies

Private Key Backup Strategies

- A Smart Card can be easily be damaged or lost
 - Need for a backup policy
- Backup Options:
 - Dual-key policy
 - On-card key generation and off-card backup of the sensitive material only
 - Off-card key generation with full off-card copy backup of the entire key
 - Sign/Decrypt without a card

Dual-Key Policy (using subkeys)

- As recommended on FSFE website
 - http://www.fsfe.org/en/card/howto/subkey_howto
- Basically you always encrypt using two keys so in case you lose your card you can use a backup key stored on a removable media
- **Problem:** Not an easy setup

On-card key generation and off-card backup of the sensitive material only

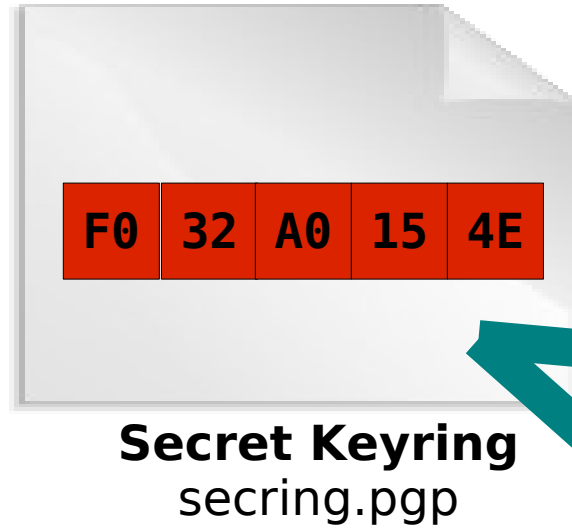
- Procedure:
 - `$ gpg --card-edit`
 - `command> admin`
 - `command> generate`
 - During the key generation a backup will be saved in a file.
 - Store this file in a safe place and “safely” wipe the original from your hard-drive
 - If the card is lost, the backup can be transferred back to a new card using :
 - `$ gpg --edit-key user@domain.com`
 - `command> toggle`
 - `command> bkuptocard`
 - **Problem:** Backup is not a full GnuPG key...
a new Smart Card is required to use it!!

Off-card key generation with full off-card copy

- **Idea:**
 - Generate a private key on a trusted machine (Live-CD, no network or HDD)
 - Backup this private key on a floppy/Flash drive
 - Use the keytocard command to transfer the private key material to the card
 - Backup the new “sanitized” private key to another floppy/Flash drive
 -
- **Advantage:** If you lose your card and are unable to get a new one, you'll still be able to sign/decrypt.

Off-card key generation with full off-card copy

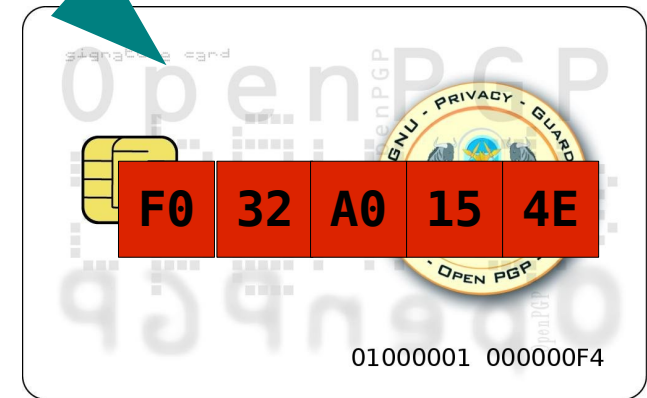
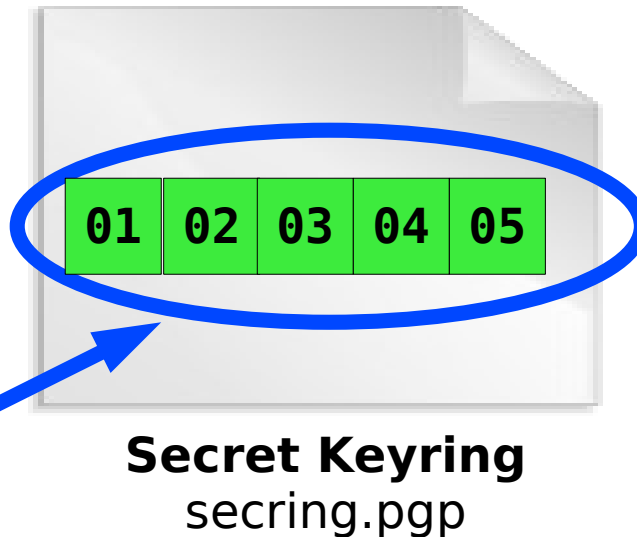
Before



Full Private Key



After



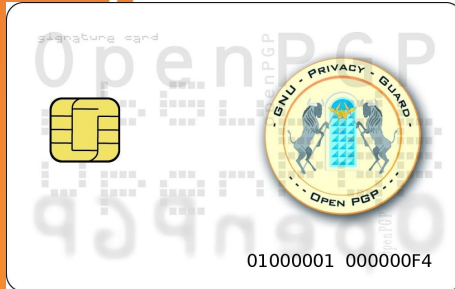
Getting Started with OpenPGP card

Getting started with OpenPGP card

- 1) Get an OpenPGP card
- 2) Get a supported Smart Card reader
- 3) Install required libraries
- 4) *Optional* - pcsclite installation/configuration
- 5) *Optional* - CTAPI installation/configuration
- 6) Get first card response

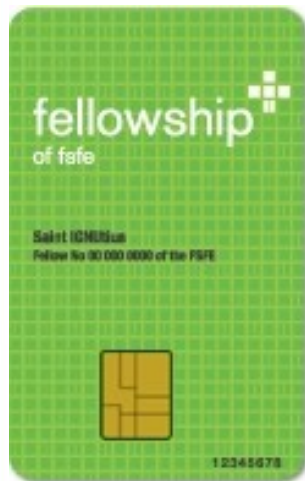
1- Get an OpenPGP card

- Kernel Concepts



- <http://www.kernelconcepts.de/products/security->
- About 16€ piece

- Fellowship of Free Software Foundation Europe



- <http://www.fsfe.org/en>
- Free when becoming a member (donation required)
- Personalized with your name printed on the card



EUROPE ONLY!!!

2- Get a supported Smart Card reader

- Kernel Concepts
 - <http://www.kernelconcepts.de/products/security-en.shtml>
 - USB Chipcard reader SCM SCR-335 for 29€
- Other places
 - Just look for compatibility and driver support



American Express in the US has been offering free Smart Card reader to their Blue Card customers (maybe available on ebay).

They are re-packaged Gemplus GCR415 readers.

3- Required Libraries

- See Smart Card Reader support



On some Debian systems you need to create the following symbolic link :

- `su`
- `ln -s /usr/lib/libpcsclite.so.1 /usr/lib/libpcsclite.so`

4-5- PCSCLite or CTAPI installation

- Smart Card reader dependent
 - Should be a question of apt-get install xxx
- See my post for installing PCSCLite and CTAPI on Linux Live-CD
 - <http://lists.gnupg.org/pipermail/gnupg-users/2005-February/024683.html>

6– First Card response

- Connect the reader
- Insert the card
- Open a terminal window:
 - `gpg --card-status`



You may want to bypass built-in CCID driver by adding `--disable-ccid`



On Kanotix, you need to be root to issue the command (`su`)

gpg --card-status

```
$ gpg --card-status
gpg: detected reader `SCM SCR 331 (80000A10) 00 00'
Application ID ....: D2760001240101000001000000F40000
Version .....: 1.0
Manufacturer .....: PPC Card Systems
Serial number ....: 000000F4
Name of cardholder: [not set]
Language prefs ...: en
Sex .....: unspecified
URL of public key : [not set]
Login data .....: [not set]
Signature PIN ....: not forced
Max. PIN lengths .: 254 254 254
PIN retry counter : 3 3 3
Signature counter : 1
Signature key ....: 50A5 2121 473E 008A CF10 609D 3843 9B24 F8E2 EFDD
Encryption key....: EE8B 3ABB 01B5 9C0A 4F8B CCA9 5E... EB58 80F9 DFFF
Authentication key: CCFE 5FC2 AA06 A0FD 2361 F1FD 3279 55E1 7CE1 443A
General key info..: [none]
```

S/N of the card

Optional Information

PIN monitoring

Fingerprint of the stored key

OpenPGP card personalization

Off-card Key Generation Backup Strategy

Prerequisites

- Linux Live-CD (Knoppix/Kanotix/Puppy Linux)
 - **Only Live-CD, no installed systems**
- Everything described in the “Getting Started Section”
 - Smart Card Reader, Smart Card, Libraries
- Two USB Flash keys/Floppies :
 - One for private key backup
 - One for “sanitized” private key
- **No network**..please unplug the cable
 - Sounds like a bit paranoid but you don't need it here anyway
- **No hard-drives**... Disconnect all hard drives in the host PC
 - Swap space may be used and therefore leaking private key



I don't recommend performing off-card key generation and backup on anything but Linux Live-CD. MS Win is banned here!



Off-card key generation backup strategy

- 1) Change PIN
- 2) Personalize the card (name, gender...)
- 3) Generate the private key
- 4) Backup the private key to removable media 1
- 5) Move the private key to card
- 6) Copy the new private key to removable media 2

1- Managing PINs

```
$ gpg --change-pin
gpg: detected reader `SCM SCR 331 (80000A10) 00 00'
gpg: OpenPGP card no. D2760001240101000001000000F40000 detected

1 - change PIN
2 - unblock PIN
3 - change Admin PIN
Q - quit

Your selection?
```

- Default PIN codes are :

- **PIN** : 123456
- **Admin PIN** : 12345678

Use 1) and 3) to change both
User PIN and Admin PIN



After entering three wrong (user) PIN, the cards locks

PIN can be reset using Admin PIN



After entering three wrong Admin PIN, the card destroys itself !

No way to recover from this stage

2- OpenPGP Card Personalization

- Discover the --card-edit menu

```
$ gpg --card-edit

gpg: detected reader `SCM SCR 331 (80000A10) 00 00'
Application ID ...: D2760001240101000001000000F40000
Version .....: 1.0
Manufacturer ..: PPC Card Systems
.....
.....
Authentication key: CCFE 5FC2 AA06 A0FD 2361  F1FD 3279 55E1 7CE1 443A
General key info..: [none]

Command> ?

quit          quit this menu
admin         show admin commands
help          show this help
list          list all available data
fetch         fetch the key specified in the card URL
passwd        menu to change or unblock the PIN
verify        verify the PIN and list all data

Command>
```

2- OpenPGP Card Personalization

- Discover the admin section of the --card-edit menu

```
Admin commands are allowed
```

```
Command> ?
```

```
quit      quit this menu  
admin     show admin commands  
help      show this help  
list      list all available data  
name      change card holder's name  
url       change URL to retrieve key  
fetch     fetch the key specified in the card URL  
login     change the login name  
lang      change the language preferences  
sex       change card holder's sex  
cafpr     change a CA fingerprint  
forcesig  toggle the signature force PIN flag  
generate  generate new keys  
passwd    menu to change or unblock the PIN  
verify    verify the PIN and list all data
```

```
Command>
```

3- Generate the private key

- Create a Key Pair as usual but with the following key sizes/types :
 - Signing Key : 1024-bit RSA
 - Encryption sub-Key : 1024-bit RSA

3- Generate the private key – details (1)

```
$ gpg --gen-key

gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Please select what kind of key you want:
  (1) DSA and Elgamal (default)
  (2) DSA (sign only)
  (5) RSA (sign only) ← RSA Only!
Your selection? 5
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 1024 ← 1024-bit only!
Requested keysize is 1024 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

3- Generate the private key – details (2)

```
You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:
```

```
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

```
Real name: Pink Panther  
Email address: pink.panther@mail.zoo  
Comment:
```

```
You selected this USER-ID:  
"Pink Panther <pink.panther@mail.zoo>"
```

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o  
You need a Passphrase to protect your secret key.
```

```
We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.
```

```
..+++++
```

```
.....+++++
```

```
gpg: /home/test/.GnuPG/trustdb.gpg: trustdb created  
gpg: key 8E05B0C6 marked as ultimately trusted  
public and secret key created and signed.
```

```
gpg: checking the trustdb  
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model  
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u  
pub 1024R/8E05B0C6 2006-05-05  
Key fingerprint = C1B8 F2FE 69F7 EAF0 36BD 0FB3 922B E145 8E05 B0C6  
uid Pink Panther <pink.panther@mail.zoo>
```

```
Note that this key cannot be used for encryption. You may want to use the command "--edit-key" to generate a subkey for this purpose.
```

3- Generate the private key – details (3)

```
$ gpg --edit-key 0x8E05B0C6
gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Secret key is available.

pub 1024R/8E05B0C6  created: 2006-05-05  expires: never      usage: SC
                        trust: ultimate    validity: ultimate
[ultimate] (1). Pink Panther <pink.panther@mail.zoo>

Command> addkey
Key is protected.

You need a passphrase to unlock the secret key for
user: "Pink Panther <pink.panther@mail.zoo>"
1024-bit RSA key, ID 8E05B0C6, created 2006-05-05

Please select what kind of key you want:
  (2) DSA (sign only)
  (4) Elgamal (encrypt only)
  (5) RSA (sign only)
  (6) RSA (encrypt only) ← RSA Only!
Your selection? 6
```

3- Generate the private key – details (4)

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 1024 ← 1024-bit only!
Requested keysize is 1024 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
Really create? (y/N) y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.+++++
.+++++

pub 1024R/8E05B0C6  created: 2006-05-05  expires: never      usage: SC
                        trust: ultimate    validity: ultimate
sub 1024R/E295B471  created: 2006-05-05  expires: never      usage: E
[ultimate] (1). Pink Panther <pink.panther@mail.zoo>

Command> q
Save changes? (y/N) y ← Don't forget to Quit and Save!
```

4- Backup the private key

- Mount your removable media (Floppy/USB Flash drive) using CLI or icon on the desktop (e.g. /media/sda1) and enable Read/Write
- Copy Public and Private keyring :
 - `cp /home/knoppix/.gnupg/pubring.gpg /mnt/sda1/.`
 - `cp /home/knoppix/.gnupg/secring.gpg /mnt/sda1/.`
- Unmount and disconnect the removable media (CLI or icon)
- Put this invaluable item in a safe or remote secret location



It's also a good idea to generate a revoke certificate at this time.

5- Move the private key to card

• BEFORE

```
$ gpg --edit-key 0x8E05B0C6
pub 1024R/8E05B0C6  created: 2006-05-05  expires: never      usage: SC
                        trust: ultimate    validity: ultimate
sub 1024R/E295B471  created: 2006-05-05  expires: never      usage: E
[ultimate] (1). Pink Panther <pink.panther@mail.zoo>

Command> toggle

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
ssb 1024R/E295B471  created: 2006-05-05  expires: never
(1) Pink Panther <pink.panther@mail.zoo>
```

• AFTER

```
$ gpg --edit-key 0x8E05B0C6
pub 1024R/8E05B0C6  created: 2006-05-05  expires: never      usage: SC
                        trust: ultimate    validity: ultimate
sub 1024R/E295B471  created: 2006-05-05  expires: never      usage: E
[ultimate] (1). Pink Panther <pink.panther@mail.zoo>

Command> toggle

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
ssb 1024R/E295B471  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
(1) Pink Panther <pink.panther@mail.zoo>
```

← S/N of the card



5- Move the private key to card – details (1)

```
$ gpg --edit-key 0x8E05B0C6
gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Secret key is available.

pub 1024R/8E05B0C6  created: 2006-05-05  expires: never      usage: SC
                        trust: ultimate    validity: ultimate
sub 1024R/E295B471  created: 2006-05-05  expires: never      usage: E
[ultimate] (1). Pink Panther <pink.panther@mail.zoo>

Command> toggle

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
ssb 1024R/E295B471  created: 2006-05-05  expires: never
(1) Pink Panther <pink.panther@mail.zoo>

Command> keytocard
Really move the primary key? (y/N) y
gpg: detected reader `SCM SCR 331 (80000A10) 00 00'
Signature key . . . . : 50A5 2121 473E 008A CF10  609D 3843 9B24 F8E2 EFDD
Encryption key . . . . : EE8B 3ABB 01B5 9C0A 4F8B  CCA9 5EAA EB58 80F9 DFFF
Authentication key: CCFE 5FC2 AA06 A0FD 2361  F1FD 3279 55E1 7CE1 443A

Please select where to store the key:
  (1) Signature key ← Select Signature
  (3) Authentication key
Your selection? 1
```

5- Move the private key to card – details (2)

```
gpg: WARNING: such a key has already been stored on the card!

Replace existing key? (y/N) y

You need a passphrase to unlock the secret key for
user: "Pink Panther <pink.panther@mail.zoo>"
1024-bit RSA key, ID 8E05B0C6, created 2006-05-05

gpg: existing key will be replaced
gpg: 3 Admin PIN attempts remaining before card is permanently locked

Admin PIN

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
ssb 1024R/E295B471  created: 2006-05-05  expires: never
(1) Pink Panther <pink.panther@mail.zoo>

Command> key 1 ← Select the encryption sub-key

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
ssb* 1024R/E295B471  created: 2006-05-05  expires: never
(1) Pink Panther <pink.panther@mail.zoo>

Command> keytocard
Signature key . . . . : 50A5 2121 473E 008A CF10  609D 3843 9B24 F8E2 EFDD
Encryption key . . . . : EE8B 3ABB 01B5 9C0A 4F8B  CCA9 5EAA EB58 80F9 DFFF
Authentication key: CCFE 5FC2 AA06 A0FD 2361  F1FD 3279 55E1 7CE1 443A
```

5- Move the private key to card – details (3)

```
Please select where to store the key:
  (2) Encryption key
Your selection? 2

gpg: WARNING: such a key has already been stored on the card!

Replace existing key? (y/N) y

You need a passphrase to unlock the secret key for
user: "Pink Panther <pink.panther@mail.zoo>"
1024-bit RSA key, ID E295B471, created 2006-05-05

gpg: existing key will be replaced

sec 1024R/8E05B0C6  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
ssb* 1024R/E295B471  created: 2006-05-05  expires: never
                        card-no: 0001 000000F4
(1) Pink Panther <pink.panther@mail.zoo>
```

```
Command> q
Save changes? (y/N) y
```

← Don't forget to Quit and Save!

6- Backup the “sanitized” private key

- Mount your removable media (Floppy/USB Flash drive) using CLI or icon on the desktop (e.g. /media/sda1) and enable Read/Write
- Copy Public and Private keyring :
 - `cp /home/knoppix/.gnupg/pubring.gpg /mnt/sda1/.`
 - `cp /home/knoppix/.gnupg/secring.gpg /mnt/sda1/.`
- Unmount and disconnect the removable media (CLI or icon)
- You can now safely shutdown your machine, reconnect hard drives and network and restart your favorite OS
- Just copy back the sanitized version of your pub/sec ring to the usual location

OpenPGP Card Usage

Use GnuPG with your OpenPGP Card

- Use GnuPG as before except it will ask you for your PIN instead of passphrase.
- Some front-end program don't support GnuPG Smart Card out of the box (Kggpg).
 - In some cases you will have to install gpg-agent
 - Enigmail plug-in for Thunderbird and WinPT provide a nice interface for entering and caching the PIN



When asking for the PIN, GnuPG also displays the Serial Number of the concerned Smart Card so you know which one to pick

Use GnuPG with your OpenPGP Card - details

```
$ touch test.txt

$ echo CLEAR TEXT >> test.txt

$ more test.txt
CLEAR TEXT

$ gpg -r pink.panther@mail.zoo -ea test.txt

$ more test.txt.asc
-----BEGIN PGP MESSAGE-----

hIwD5GBYquKVtHEBBAC9XkLYZI5cDioRQKm8sg1c+ILbMFwLCj7oYtaBsNXLqFj0
JQzcSk5LIz6k61Dd5+9jP9nLYP38ri1zKc2T7uPQ8gXaUZeK63TFUgtv/EYKFCBi
Ro9c1kdn1KekbD6H8iZgd9+VMyiJ6RQ9DFUAL4Ml4laKimNfDCgCSyMYswJJj9JS
AU+2E11PLx1koV888ILZLQEADEmu0ImXcwm9NQSgyZW62lZdHa1brh+Qd3AWIICk
5FbVa5B0QFwiXi5NdAx8BIOA9rbm6U8Nmr8mZ3zSkzoxVg==
=PiAn
-----END PGP MESSAGE-----

$ rm test.txt

$ gpg -d test.txt.asc > testout.txt
gpg: detected reader `SCM SCR 331 (80000A10) 00 00'

PIN
gpg: encrypted with 1024-bit RSA key, ID E295B471, created 2006-05-05
      "Pink Panther <pink.panther@mail.zoo>"

$ more test.txt
CLEAR TEXT
```



JM2L 2006



Now it's time for a
Keysigning party!

References

- How to use OpenPGP card with Live-CD including pcsclite and CTAPI driver installation
 - <http://lists.gnupg.org/pipermail/gnupg-users/2005-February/024683.html>
- How to use the Fellowship Smartcard
 - [http://www.gnupg.org/\(fr\)/howtos/card-howto/en/smartcard-howto.html](http://www.gnupg.org/(fr)/howtos/card-howto/en/smartcard-howto.html)
- The GnuPG Smartcard HOWTO (by Werner Koch)
 - <http://www.kernelconcepts.de/products/Smartcard-HOWTO.txt>
- g10 OpenPGP page
 - <http://www.g10code.de/p-card.html>
- OpenPGP Card specification :
 - <http://www.g10code.de/docs/openpgp-card-1.1.pdf>