GPG

LaurenceWarne

June 23, 2019

Contents

1 Introduction

GnuPG is an implementation of the **OpenPGP standard**. It enables the encryption of data and features a versatile key management system. At its core, gpg is a **command line tool**, it can be used with the **gpg** command.

A **public key** consists of the public portion of the master signing key, the public portions of the subordinate signing and encryption subkeys, and a set of user IDs used to associate the public key with a real person.

2 CLI

We can generate a new key pair with:

```
gpg --gen-key
gpg --full-generate-key  # Allows choice of key size, expiration, etc
```

Now if we call:

```
gpg -k                  # Equivalent to the following commands
gpg --list-keys
gpg --list-public-keys
```

We will be met with something along these lines, which show information about keys on your **public keyring**:

```
/home/laurencewarne/.gnupg/pubring.gpg
--------------------------------------
pub   rsa3072 2019-02-13 [SC] [expires: 2021-02-12]
      A3500FD11FFBF3E0D2B1E2FB43ECFD79C3A76493
uid           [ultimate] Laurence Warne <laurencewarne@gmail.com>
sub   rsa3072 2019-02-13 [E] [expires: 2021-02-12]

pub   rsa2048 2018-11-09 [SC]
      9129AB98125EAC18C65DBF2964D3012D80EE3190
uid           [ unknown] John Smith <ragnarok89@yahoo.com>
sub   rsa2048 2018-11-09 [E]
```

Note only public keys are listed here. The first column here denotes the type of the key.

2.1 Fingerprints

The long strings are the **key fingerprints**. A fingerprint is calculated from a constant, the packet length and finally a part of the public key packet (see this stack overflow post for more details). To see the fingerprint of a key:

```
gpg --fingerprint laurencewarne@gmail.com
```

Closely related (but not shown), is the ID of a key (pair), which just denotes the lowest 64 bits of the key's fingerprint.

2.2 Subkeys

In the first column of the cmd output, 'sub' indicates the described key is a **subkey**, of the **master signing key pair** displayed in the 'pub' column. Note to display the fingerprints of subkeys, call:

```
gpg --list-keys --with-subkey-fingerprints
```

When you decrypt a document with your private key, you are most likely using a private encryption subkey to do so, in place of your master signing key. The characters in square brackets shown in the output correspond to the following:

2.2.1 Key roles:

Constant	Character	Explanation
$\mathrm{PUBKEY}_{\mathrm{USAGESIG}}$	S	Key is good for signing
$\mathrm{PUBKEY}_{\mathrm{USAGECERT}}$	C	Key is good for certifying other signatures
$\mathrm{PUBKEY}_{\mathrm{USAGEENC}}$	E	Key is good for encryption
$\mathrm{PUBKEY}_{\mathrm{USAGEAUTH}}$	A	Key is good for authentication

3 Editing Keys

We can interact with the keys on our keyring using:

```
gpg --edit-key laurencewarne@gmail.com
```

This will open a little interpreter from which we can add a key to our public key (We will then be prompted to describe what the subkey will be used for):

```
gpg> addkey
State how much we ’trust’ a key:
gpg> trust
```

4 Digital Signatures

A **digital signature** serves the same purpose as a hand-written signature.