Supplemental Keysigning Help

markus schnalke
meillo@marmaro.de

Abstract

Methods to organize keysigning events are available in large numbers. They usually describe only what needs be done in which order, and this is exactly what they should do.

This document is a supplemental help to one of the methods by describing concrete ways how to actually do some of the tasks. It suggest tools and shows how to use them.

1 Introduction

This document tries to help people in organizing a Keysigning event. It should be seen as concrete suggestions for how to do things that are already described by the keysigning method in general. The method defines how to organize the keysigning, this document makes concrete suggestions *how* to do things. This document also shows how to generate WOT graphs.

2 Keysigning method

The keysigning method that is focused here is Zimmermann and Sassaman's method [7]. It is easy to use and scales well for any amount of people.

One should become familiar with this method and follow it when organizing a keysigning event. This document provides technical help with some selected tasks.

3 Key management

When you invite people to the keysigning event you will receive their public key(s). To manage the keys it is recommended to add them to a new keyring:

```
$ gpg --no-default-keyring \
      --keyring /path/to/keyring.gpg \
      --import some-public-key.asc
```

It is also possible to directly fetch the keys from a keyserver, but this is not preferred. It is better to receive the keys directly from the owners.

```
$ gpg --no-default-keyring \
      --keyring /path/to/keyring.gpg \
      --keyserver subkeys.pgp.net \
      --recv-key 0xDEADBEEF
```

4 Participant list

You have to generate a list that contains the public keys of all participants. A script to do this automatically with nice formatting is available [4]. The script is not perfect, but sufficient.

```
$ keylist.sh /path/to/keyring.gpg header.txt \
             howto.txt checksums.txt
```

The script generates a public key list from all keys in the keyring (first argument). This list can get prepended by the contents of text files (all further arguments).

A general header is demanded by good style. Descriptions of what the participants need to do are highly recommended in order to support inexperienced participants. Fields to insert the checksums should be provided anyway. Examples for the here included files can be found at [4].

5 WOT graphs

The change of the Web of Trust (short: WOT) does directly show the gain of a keysigning event. The more interweaved and the shorter connections between individuals are, the better is the trust among that group of people.

Created for some people of the LUG Ulm [6], 2009-02-18
This document is available on my website `http://marmaro.de/docs` .

```
Tue, 18 Feb 2009 18:56:14 +0100
markus schnalke <meillo@marmaro.de>

                        A SAMPLE KEYSIGNING EVENT
                        =========================

              18. February 2009 at 19:30 at Location in City


Tutorial
--------
1) Print this list before the keysigning event but _after_ the application
   deadline. You need the final version of the list.
2) Create checksums of the file and insert them into the prepared fields.
   The checksums are easiest created with 'gpg --print-mds <filename>'.
3) Take this list, a pencil, and an identity card to the keysigning event.

Additionally: Check your own fingerprint. If it is wrong, bring pieces of
              paper with the correct one on it for every participant.


SHA224 checksum:  ____  ____  ____  ____  ____  ____  ____

                  ____  ____  ____  ____  ____  ____  ____         [ ] OK


SHA256 checksum:  ________   ________   ________   ________

                  ________   ________   ________   ________        [ ] OK


---------------------------------------------------------------------------
001   [ ] fingerprint OK          [ ] identity OK
pub   1024D/E5FDA812 2007-08-03
      Key fingerprint = 0149 51FC A6E6 023B 48F3  3707 E9A1 6967 E5FD A812
uid                   markus schnalke <meillo@marmaro.de>
sub   2048g/FD68D476 2007-08-03
---------------------------------------------------------------------------
002   [ ] fingerprint OK          [ ] identity OK
.
.
.
```

Figure 1: A sample participant list

Providing WOT graphs is a nice act of a keysigning organizer. However, it is in any case optional and can be done afterwards, too.

Two programs are required to generate the graphs: `sig2dot` [2] and `neato` from `graphviz` [5].

To generate a graph that depicts the WOT, use the following command:

```
$ gpg --no-default-keyring \
      --keyring /path/to/keyring.gpg \
      --list-sigs \
  | sig2dot -d YYYY-MM-DD \
  | neato -Tpng > wot.png
```

The date ('`YYYY-MM-DD`') must be substituted, of course. The generated image shows the WOT at the given date.

To generate graphs of the WOT after the event, one must update the keyring first:

```
$ gpg --no-default-keyring \
      --keyring /path/to/keyring.gpg \
      --keyserver subkeys.pgp.net \
      --refresh-keys
```

New images can be created the same way as described above, only the date needs to be changed.

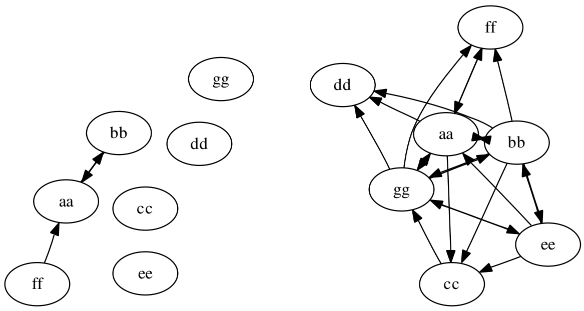

Figure 2: Sample graphs which show the WOT before and after a keysigning event

(Notice that only signatures that were uploaded to a keyserver will be included.)

6 A hint for participants

Receiving keys, signing them, and sending the signatures back to the key owners can be a wasteful job, especially if may people took part in a keysigning event.

The nice tool `caff` [1] is a great helper. It automates the whole process, from key retrieval, to signing, to sending the signatures. (An MTA is required to send signatures.)

7 Acknowledgments

This document bases heavily on how Fabian Fingerle [3] organizes keysigning events. I thank him for being a great inspiration.

References

[1] Various contributors. *PGP Tools*. `http://pgp-tools.alioth.debian.org`.

[2] Darxus. *sig2dot GPG/PGP Keyring Graph Generator*. `http://chaosreihns.com/code/sig2dot/`.

[3] Fabian Fingerle. *Homepage*. `http://fabian-fingerle.de`. His keysigning data is located at `http://datensalat.eu/~fabian/gpg-event`.

[4] Markus Schnalke. *Keysigning utilities*. `http://prog.marmaro.de/keysigning/`.

[5] The Graphviz Team. *Graphviz – Graph Visualization Software*. `http://graphviz.org`.

[6] LUG Ulm. *Homepage*. `http://lugulm.de`.

[7] Zimmermann and Sassaman. *Efficient Group Key Signing Method*. `http://sion.quickie.net/keysigning.txt`.