NAME

paperkey − extract secret information out of OpenPGP secret keys

SYNOPSIS

paperkey [**−−secret−key=***FILE*] [**−−output=***FILE*] [**−−output−type=base16|raw**] [**−−output−width=***WIDTH*]

paperkey −−pubring=*FILE* [**−−secrets=***FILE*] [**−−input−type=auto|base16|raw**] [**−−output=***FILE*] [**−−ignore−crc−error**] [**−−comment=***STRING*] [**−−file−format**]

paperkey −−version

MOTIVATION

As with all data, secret keys should be backed up. In fact, secret keys should be backed up even better than other data, because they are impossible to recreate should they ever be lost. All files encrypted to lost keys are forever (or at least for a long time) undecipherable. In addition to keeping backups of secret key information on digital media such as USB-sticks or CDs it is reasonable to keep an if-all-else-fails copy on plain old paper, for use should your digital media ever become unreadable for whatever reason. Stored properly, paper is able to keep information for several decades or longer.

With GnuPG, PGP, or other OpenPGP implementations the secret key usually contains a lot more than just the secret numbers that are important. They also hold all the public values of key pairs, user ids, expiration times and more. In order to minimize the information that has to be entered manually or with the help of OCR, QR code or similar software, **paperkey** extracts just the secret information out of OpenPGP secret keys. For recovering a secret key it is assumed that the public key is still available, for instance from public Internet keyservers.

DESCRIPTION

paperkey has two modes of operation:

The first mode creates "paperkeys" by extracting just the secret information from a secret key, formatting the data in a way suitable for printing or in a raw mode for further processing.

The other mode rebuilds secret keys from such a paperkey and a copy of the public key, also verifying the checksums embedded in the paperkey. This mode is selected when the**−−pubring** option is used, which is required in that case. If a passphrase was set on the original secret key, the same passphrase is set on the rebuilt key.

Input is read from standard−in except when the **−−secret−key** or **−−secrets** option is used; output is printed to standard−out, unless changed with the **−−output** option.

SECURITY CONSIDERATIONS

Please note that **paperkey** does not change the protection and encryption status of and security requirements for storing your secret key. If the secret key was protected by a passphrase so is the paperkey. If the secret key was unprotected the paperkey will not be protected either.

OPTIONS

−−help, **−h** Display a short help message and exit successfully.

−−version, **−V**
: Print version information and copyright information and exit successfully.

−−verbose, **−v**
: Print status and progress information to standard−error while processing the input. Repeat for even more output.

−−output=*FILE*, **−o**
: Redirect output to the file given instead of printing to standard−output.

−−comment=*STRING*
: Include the specified comment in the base16 output.

−−file−format
Paperkey automatically includes the file format it uses as comments at the top of the base16 output. This command simply prints out the file format and exits successfully.

OPTIONS FOR EXTRACTING SECRET INFORMATION

−−output−type=base16, **−−output−type=raw**
Select the output type. The base16 style encodes the information in the style of a classic hexdump, including line numbers and per-line CRC checksums to facilitate localizing errors in the input file during the recovery phase. The raw, or binary, mode is just a raw dump of the secret information, intended for feeding to barcode generators or the like.

−−output−width=*WIDTH*
Choose line width in the base16 output mode. The default is 78 characters.

−−secret−key=*FILE*
File to read the secret key from. If this option is not given **paperkey** reads from standard−input.

OPTIONS FOR RE-CREATING PRIVATE KEYS

−−input−type=auto, **−−input−type=base16**, **−−input−type=raw**
Specify that the given input is either in base16 format, as produced by **paperkey**, or in raw format. The default, auto, tries to automatically detect the format in use.

−−pubring=*FILE*
File to read public key information from. It is assumed that the user can get the public key from sources like public Internet keyservers.

−−secrets=*FILE*
File to read the extracted secrets, the paperkey, from. If this is not given then the information is read from standard−input.

−−ignore−crc−error
Do not reject corrupt input and continue despite any CRC errors.

EXAMPLES

Take the secret key in key.gpg and generate a text file to−be−printed.txt that contains the secret data:

$ paperkey −−secret−key my−secret−key.gpg −−output to−be−printed.txt

Take the secret key data in my−key−text−file.txt and combine it with my−public−key.gpg to reconstruct my−secret−key.gpg:

$ paperkey −−pubring my−public−key.gpg −−secrets my−key−text−file.txt −−output my−secret−key.gpg

If −−output is not specified, the output goes to stdout. If −−secret−key is not specified, the data is read from stdin so you can do things like:

$ gpg −−export−secret−key my−key | paperkey | lpr

SEE ALSO

gpg(1), **http://www.jabberwocky.com/software/paperkey/**

AUTHORS

paperkey is written by David Shaw <dshaw@jabberwocky.com>.