NAME

ksi sign - Sign data with KSI command-line tool.

SYNOPSIS

ksi sign -i *input* [**-o** *out.ksig*] **-S** *URL* [**--aggr-user** *user* **--aggr-key** *key*] [**-H** *alg*] [**--data-out** *file*] [**--dump**] [*more options*]

DESCRIPTION

Signs the given input such as content of a given file or a pre-computed hash value with KSI. User must have access to KSI signing service (KSI Aggregator) for signing. Note that until not extended, the signatures can be verified using the copy of calendar blockchain at KSI Extender (calendar-based verification) or using the PKI signature in the calendar authentication record (key-based verification, temporary only). See **ksi-verify**(1) for details.

OPTIONS

-i *input* The input is either the path to the file to be hashed and signed or a hash imprint in case the data to be signed has been hashed already. Use '-' as file name to read data to be hashed from *stdin*. Hash imprint format: <*alg*>:<*hash in hex*>. Use **-h** to get the list of supported hash algorithms. Note that only the main data stream of the file is hashed, any extensions of specific file systems are ignored (e.g. the alternate data streams of NTFS are not signed).

-o *out.ksig*
Output file path for the signature. Use '-' as file name to redirect signature binary stream to *stdout*. If not specified, the signature is saved to <input file>.ksig (or <input file>_<nr>.ksig, where <nr> is auto-incremented counter if the output file already exists). If specified, will always overwrite the existing file.

-H *alg* Use the given hash algorithm to hash the file to be signed. Use **ksi -h** to get the list of supported hash algorithms.

-S *URL* URL Signing service (KSI Aggregator) URL.

--aggr-user *user*
Username for signing service.

--aggr-key *key*
HMAC Key for signing service.

--data-out *file*
Save signed data to file. Use when signing a stream. Use '-' as file name to redirect data being hashed to *stdout*.

-d Print detailed information about processes and errors to *stderr*.

--dump
Dump signature created in human-readable format to *stdout*.

--conf *file*
Read configuration options from given file. It must be noted that configuration options given explicitly on command line will override the ones in the configuration file. See **ksi-conf**(5) for more information.

--log *file*
Write libksi log to given file. Use '-' as file name to redirect log to *stdout*.

EXIT STATUS

See **ksi**(1) for more information.

EXAMPLES

In the following examples it is assumed that KSI service configuration options (URLs, access credentials) are defined. See **ksi-conf**(5) for more information.

1 To sign a file *file* and save signature to *sig.ksig* call:

```
ksi sign -i file -o sig.ksig
```

2

To sign a data hash (hashed with SHA256) and save the resulting signature to file *sig.ksig* call:

```
ksi sign -i SHA-256:c8ef6d57ac28d1b4e95a513959\f5fcdd0688380a43d601a5ace1d2e96884690a -o
    sig.ksig
```

3

To sign a data file *file* with non-default algorithm *SHA1* call:

```
ksi sign -i file -H SHA1 -o sig.ksig
```

4

To sign a stream (stdin), save data from stream to *file* and save signature to *sig.ksig* call:

```
ksi sign -i file -H SHA1 -o sig.ksig
```

ENVIRONMENT

Use the environment variable **KSI_CONF** to define the default configuration file. See **ksi-conf**(5) for more information.

AUTHOR

Guardtime AS, http://www.guardtime.com/

SEE ALSO

ksi(1), **ksi-verify**(1), **ksi-extend**(1), **ksi-pubfile**(1), **ksi-conf**(5)