Rocky Enterprise Linux 9.2 Manual Pages on command 'tpm2_verifysignature.1'

$ man tpm2_verifysignature.1

tpm2_verifysignature(1) General Commands Manual tpm2_verifysignature(1)

NAME

tpm2_verifysignature(1) - Validates a signature using the TPM.

SYNOPSIS

tpm2_verifysignature [OPTIONS]

DESCRIPTION

tpm2_verifysignature(1) - Uses loaded keys to validate a signature on a message with the message digest passed to the TPM. If the signature check succeeds, then the TPM will produce a TPMT_TK_VERIFIED. Other? wise, the TPM shall return TPM_RC_SIGNATURE. If object references an asymmetric key, only the public portion of the key needs to be loaded. If object references a symmetric key, both the public and private por? tions need to be loaded.

OPTIONS

? -c, --key-context=OBJECT:

Context object for the key context used for the operation. Either a file or a handle number. See section ?Context Object Format?.

? -g, --hash-algorithm=ALGORITHM:

The hash algorithm used to digest the message. Algorithms should follow the ?formatting standards?, see section ?Algorithm Speci? fiers?. Also, see section ?Supported Hash Algorithms? for a list of supported hash algorithms.

? -m, --message=FILE:

The message file, containing the content to be digested.

? -d, --digest=FILE:

The input hash file, containing the hash of the message. If this op? tion is selected, then the message (-m) and algorithm (-g) options do not need to be specified.

? -s, --signature=FILE:

The input signature file of the signature to be validated.

? -f, --scheme=SCHEME:

The signing scheme that was used to sign the message. This option should only be specified if the signature comes in from a non tss standard, like openssl. See ?Signature format specifiers? for more details. The tss format contains the signature metadata required to understand it?s signature scheme.

Signing schemes should follow the ?formatting standards?, see section ?Algorithm Specifiers?.

? --format=SCHEME:

Deprecated. Same as --scheme.

? -t, --ticket=FILE:

The ticket file to record the validation structure.

References

Context Object Format

The type of a context object, whether it is a handle or file name, is determined according to the following logic in-order:

? If the argument is a file path, then the file is loaded as a restored TPM transient object.

? If the argument is a prefix match on one of:

? owner: the owner hierarchy

? platform: the platform hierarchy

? endorsement: the endorsement hierarchy

? lockout: the lockout control persistent object

? If the argument argument can be loaded as a number it will be treat as a handle, e.g. 0x81010013 and used directly._OBJECT_.

Algorithm Specifiers

Options that take algorithms support ?nice-names?.

There are two major algorithm specification string classes, simple and complex. Only certain algorithms will be accepted by the TPM, based on usage and conditions.

Simple specifiers

These are strings with no additional specification data. When creating objects, non-specified portions of an object are assumed to defaults.

You can find the list of known ?Simple Specifiers Below?.

Asymmetric

? rsa

? ecc

Symmetric

? aes

? camellia

Hashing Algorithms

? sha1

? sha256

? sha384

? sha512

? sm3_256

? sha3_256

? sha3_384

? sha3_512

Keyed Hash

? hmac

? xor

Signing Schemes

? rsassa

? rsapss

? ecdsa

? ecdaa

? ecschnorr

Asymmetric Encryption Schemes

? oaep

? rsaes

? ecdh

Modes

? ctr

? ofb

? cbc

? cfb

? ecb

Misc

? null

Complex Specifiers

Objects, when specified for creation by the TPM, have numerous algo? rithms to populate in the public data. Things like type, scheme and asymmetric details, key size, etc. Below is the general format for specifying this data: <type>:<scheme>:<symmetric-details>

Type Specifiers

This portion of the complex algorithm specifier is required. The re? maining scheme and symmetric details will default based on the type specified and the type of the object being created.

? aes - Default AES: aes128

? aes128<mode> - 128 bit AES with optional mode (ctr|ofb|cbc|cfb|ecb). If mode is not specified, defaults to null.

? aes192<mode> - Same as aes128<mode>, except for a 192 bit key size.

? aes256<mode> - Same as aes128<mode>, except for a 256 bit key size.

? ecc - Elliptical Curve, defaults to ecc256.

? ecc192 - 192 bit ECC

? ecc224 - 224 bit ECC

? ecc256 - 256 bit ECC

? ecc384 - 384 bit ECC

? ecc521 - 521 bit ECC

? rsa - Default RSA: rsa2048

? rsa1024 - RSA with 1024 bit keysize.

? rsa2048 - RSA with 2048 bit keysize.

? rsa4096 - RSA with 4096 bit keysize.

Scheme Specifiers

Next, is an optional field, it can be skipped.

Schemes are usually Signing Schemes or Asymmetric Encryption Schemes.

Most signing schemes take a hash algorithm directly following the sign?

ing scheme. If the hash algorithm is missing, it defaults to sha256.

Some take no arguments, and some take multiple arguments.

Hash Optional Scheme Specifiers

These scheme specifiers are followed by a dash and a valid hash algo?

rithm, For example: oaep-sha256.

? oaep

? ecdh

? rsassa

? rsapss

? ecdsa

? ecschnorr

Multiple Option Scheme Specifiers

This scheme specifier is followed by a count (max size UINT16) then

followed by a dash(-) and a valid hash algorithm. * ecdaa For example,

ecdaa4-sha256. If no count is specified, it defaults to 4.

No Option Scheme Specifiers

This scheme specifier takes NO arguments. * rsaes

Symmetric Details Specifiers

This field is optional, and defaults based on the type of object being

created and it?s attributes. Generally, any valid Symmetric specifier

from the Type Specifiers list should work. If not specified, an asym?

metric objects symmetric details defaults to aes128cfb.

Examples

Create an rsa2048 key with an rsaes asymmetric encryption scheme

tpm2_create -C parent.ctx -G rsa2048:rsaes -u key.pub -r key.priv

Create an ecc256 key with an ecdaa signing scheme with a count of 4 and sha384 hash

/tpm2_create -C parent.ctx -G ecc256:ecdaa4-sha384 -u key.pub -r key.priv cryptographic algorithms ALGORITHM.

COMMON OPTIONS

This collection of options are common to many programs and provide in? formation that many users may expect.

? -h, --help=[man|no-man]: Display the tools manpage. By default, it attempts to invoke the manpager for the tool, however, on failure will output a short tool summary. This is the same behavior if the ?man? option argument is specified, however if explicit ?man? is re? quested, the tool will provide errors from man on stderr. If the ?no-man? option if specified, or the manpager fails, the short op? tions will be output to stdout.

To successfully use the manpages feature requires the manpages to be installed or on MANPATH, See man(1) for more details.

? -v, --version: Display version information for this tool, supported tctis and exit.

? -V, --verbose: Increase the information that the tool prints to the console during its execution. When using this option the file and line number are printed.

? -Q, --quiet: Silence normal tool output to stdout.

? -Z, --enable-errata: Enable the application of errata fixups. Useful if an errata fixup needs to be applied to commands sent to the TPM.

Defining the environment TPM2TOOLS_ENABLE_ERRATA is equivalent. in? formation many users may expect.

TCTI Configuration

The TCTI or ?Transmission Interface? is the communication mechanism with the TPM. TCTIs can be changed for communication with TPMs across different mediums.

To control the TCTI, the tools respect:

1. The command line option -T or --tcti

2. The environment variable: TPM2TOOLS_TCTI.

Note: The command line option always overrides the environment vari?
able.

The current known TCTIs are:

? tabrmd - The resource manager, called tabrmd (https://github.com/tpm2-software/tpm2-abrmd). Note that tabrmd and abrmd as a tcti name are synonymous.

? mssim - Typically used for communicating to the TPM software simula?
tor.

? device - Used when talking directly to a TPM device file.

? none - Do not initalize a connection with the TPM. Some tools allow for off-tpm options and thus support not using a TCTI. Tools that do not support it will error when attempted to be used without a TCTI connection. Does not support ANY options and MUST BE presented as the exact text of ?none?.

The arguments to either the command line option or the environment variable are in the form:

<tcti-name>:<tcti-option-config>

Specifying an empty string for either the <tcti-name> or <tcti-op?
tion-config> results in the default being used for that portion respec?
tively.

TCTI Defaults

When a TCTI is not specified, the default TCTI is searched for using dlopen(3) semantics. The tools will search for tabrmd, device and mssim TCTIs IN THAT ORDER and USE THE FIRST ONE FOUND. You can query what TCTI will be chosen as the default by using the -v option to print the version information. The ?default-tcti? key-value pair will indi?
cate which of the aforementioned TCTIs is the default.

Custom TCTIs

Any TCTI that implements the dynamic TCTI interface can be loaded. The tools internally use dlopen(3), and the raw tcti-name value is used for

the lookup. Thus, this could be a path to the shared library, or a li?

brary name as understood by dlopen(3) semantics.

TCTI OPTIONS

This collection of options are used to configure the various known TCTI modules available:

? device: For the device TCTI, the TPM character device file for use by the device TCTI can be specified. The default is /dev/tpm0.

Example: -T device:/dev/tpm0 or export TPM2TOOLS_TCTI=?device:/dev/tpm0?

? mssim: For the mssim TCTI, the domain name or IP address and port number used by the simulator can be specified. The default are 127.0.0.1 and 2321.

Example: -T mssim:host=localhost,port=2321 or export TPM2TOOLS_TCTI=?mssim:host=localhost,port=2321?

? abrmd: For the abrmd TCTI, the configuration string format is a se? ries of simple key value pairs separated by a `,' character. Each key and value string are separated by a `=' character.

? TCTI abrmd supports two keys:

1. `bus_name' : The name of the tabrmd service on the bus (a string).
2. `bus_type' : The type of the dbus instance (a string) limited to `session' and `system'.

Specify the tabrmd tcti name and a config string of bus_name=com.example.FooBar:

\--tcti=tabrmd:bus_name=com.example.FooBar

Specify the default (abrmd) tcti and a config string of bus_type=session:

\--tcti:bus_type=session

NOTE: abrmd and tabrmd are synonymous. the various known TCTI modules.

Signature Format Specifiers

Format selection for the signature output file. tss (the default) will output a binary blob according to the TPM 2.0 specification and any po?

tential compiler padding. The option plain will output the plain sig?

nature data as defined by the used cryptographic algorithm.

EXAMPLES

Sign and verify with the TPM using the endorsement hierarchy

```
tpm2_createprimary -C e -c primary.ctx
tpm2_create -G rsa -u rsa.pub -r rsa.priv -C primary.ctx
tpm2_load -C primary.ctx -u rsa.pub -r rsa.priv -c rsa.ctx
echo "my message > message.dat
tpm2_sign -c rsa.ctx -g sha256 -m message.dat -s sig.rssa
tpm2_verifysignature -c rsa.ctx -g sha256 -m message.dat -s sig.rssa
```

Sign with openssl and verify with the TPM

```
# Generate an ECC key
openssl ecparam -name prime256v1 -genkey -noout -out private.ecc.pem
openssl ec -in private.ecc.pem -out public.ecc.pem -pubout
# Generate a hash to sign (OSSL needs the hash of the message)
echo "data to sign" > data.in.raw
sha256sum data.in.raw | awk '{ print "000000 " $1 }' | \
xxd -r -c 32 > data.in.digest
# Load the private key for signing
tpm2_loadexternal -Q -G ecc -r private.ecc.pem -c key.ctx
# Sign in the TPM and verify with OSSL
tpm2_sign -Q -c key.ctx -g sha256 -d data.in.digest -f plain -s data.out.signed
openssl dgst -verify public.ecc.pem -keyform pem -sha256 \
-signature data.out.signed data.in.raw
# Sign with openssl and verify with TPM
openssl dgst -sha256 -sign private.ecc.pem -out data.out.signed data.in.raw
tpm2_verifysignature -Q -c key.ctx -g sha256 -m data.in.raw -f ecdsa \
-s data.out.signed
```

Returns

Tools can return any of the following codes:

? 0 - Success.

? 1 - General non-specific error.

? 2 - Options handling error.

? 3 - Authentication error.

? 4 - TCTI related error.

? 5 - Non supported scheme. Applicable to tpm2_testparams.

BUGS

Github Issues (https://github.com/tpm2-software/tpm2-tools/issues)

HELP

See the Mailing List (https://lists.01.org/mailman/listinfo/tpm2)