



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tpm2\_rsadecrypt.1' command**

### **\$ man tpm2\_rsadecrypt.1**

tpm2\_rsadecrypt(1)      General Commands Manual      tpm2\_rsadecrypt(1)

#### NAME

tpm2\_rsadecrypt(1) - Performs an RSA decryption operation using the TPM.

#### SYNOPSIS

tpm2\_rsadecrypt [OPTIONS] [ARGUMENT]

#### DESCRIPTION

tpm2\_rsadecrypt(1) - Performs RSA decryption on the contents of file using the indicated padding scheme according to IETF RFC 3447 (PKCS#1).

Command line argument defaults to stdin if not specified.

The key referenced by key-context is required to be:

1. An RSA key
2. Have the attribute decrypt SET in it's attributes.

#### OPTIONS

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

Context object pointing to the the public portion of RSA key to use for decryption. Either a file or a handle number. See section ?Con? text Object Format?.

? -p, --auth=AUTH:

Optional authorization value to use the key specified by -c.

? -o, --output=FILE:

Optional output file path to record the decrypted data to. The default is to print the binary encrypted data to STDOUT.

? -s, --scheme=FORMAT:

Optional, set the padding scheme (defaults to rsaes).

? null - TPM\_ALG\_NULL uses the key's scheme if set.

? rsaes - TPM\_ALG\_RSAES which is RSAES\_PKCSV1.5.

? oaep - TPM\_ALG\_OAEP which is RSAES\_OAEP.

? -l, --label=FILE OR STRING:

Optional, set the label data. The TPM requires the last byte of the label to be zero, this is handled internally to the tool. No other embedded 0 bytes can exist or the TPM will truncate your label.

? --cphash=FILE

File path to record the hash of the command parameters. This is commonly termed as cpHash. NOTE: When this option is selected, The tool will not actually execute the command, it simply returns a cpHash.

? ARGUMENT the command line argument specifies the file containing data to be decrypted.

## 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 can be loaded as a number it will be treated as a handle, e.g. 0x81010013 and used directly. `._OBJECT_`.

### Authorization Formatting

Authorization for use of an object in TPM2.0 can come in 3 different forms: 1. Password 2. HMAC 3. Sessions

NOTE: Authorizations default to the EMPTY PASSWORD when not specified?.

## Passwords

Passwords are interpreted in the following forms below using prefix identifiers.

Note: By default passwords are assumed to be in the string form when they do not have a prefix.

## String

A string password, specified by prefix `?str:?` or its absence (raw string without prefix) is not interpreted, and is directly used for authentication.

## Examples

```
foobar
```

```
str:foobar
```

## Hex-string

A hex-string password, specified by prefix `?hex:?` is converted from a hexadecimal form into a byte array form, thus allowing passwords with non-printable and/or terminal un-friendly characters.

## Example

```
hex:0x1122334455667788
```

## File

A file based password, specified by prefix `?file:?` should be the path of a file containing the password to be read by the tool or a `?-?` to use stdin. Storing passwords in files prevents information leakage, passwords passed as options can be read from the process list or common shell history features.

## Examples

```
# to use stdin and be prompted
```

```
file:-
```

```
# to use a file from a path
```

```
file:path/to/password/file
```

```
# to echo a password via stdin:
```

```
echo foobar | tpm2_tool -p file:-
```

```
# to use a bash here-string via stdin:
```

```
tpm2_tool -p file:- <<< foobar
```

## Sessions

When using a policy session to authorize the use of an object, prefix the option argument with the session keyword. Then indicate a path to a session file that was created with `tpm2_startauthsession(1)`. Option? ally, if the session requires an auth value to be sent with the session handle (eg policy password), then append a + and a string as described in the Passwords section.

## Examples

To use a session context file called `session.ctx`.

```
session:session.ctx
```

To use a session context file called `session.ctx` AND send the authvalue `mypassword`.

```
session:session.ctx+mypassword
```

To use a session context file called `session.ctx` AND send the HEX auth? value `0x11223344`.

```
session:session.ctx+hex:11223344
```

## PCR Authorizations

You can satisfy a PCR policy using the `?pcr:?` prefix and the PCR mini? language. The PCR minilanguage is as follows:

```
<pcr-spec>=<raw-pcr-file>
```

The PCR spec is documented in in the section `?PCR bank specifiers?`.

The `raw-pcr-file` is an optional argument that contains the output of the raw PCR contents as returned by `tpm2_pcrread(1)`.

PCR bank specifiers (`pcr.md`)

## Examples

To satisfy a PCR policy of sha256 on banks 0, 1, 2 and 3 use a specifi? er of:

```
pcr:sha256:0,1,2,3
```

specifying AUTH.

## 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 requested, the tool will provide errors from man on stderr. If the `?no-man?` option is specified, or the manpager fails, the short options 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. Information 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 variable.

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 simulator.

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

? none - Do not initialize 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-option-config> results in the default being used for that portion respectively.

#### 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 indicate 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 library 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_TC?`

`TI=?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.ex?`

ample.`FooBar`:

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

Specify the default (`abrmd`) tcti and a config string of `bus_type=ses?`

sion:

```
\--tcti:bus_type=session
```

NOTE: `abrmd` and `tabrmd` are synonymous. the various known TCTI mod?

ules.

## EXAMPLES

Create an RSA key and load it

```
tpm2_createprimary -c primary.ctx
```

```
tpm2_create -C primary.ctx -Grsa2048 -u key.pub -r key.priv
```

```
tpm2_load -C primary.ctx -u key.pub -r key.priv -c key.ctx
```

Encrypt using RSA

```
echo "my message" > msg.dat
```

```
tpm2_rsaencrypt -c key.ctx -o msg.enc msg.dat
```

Decrypt using RSA

```
tpm2_rsadecrypt -c key.ctx -o msg.ptext msg.enc
```

```
cat msg.ptext
```

```
my message
```

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>)

tpm2-tools

tpm2\_rsadecrypt(1)