Build SCP Session - User Guide

UG98S09

Maxim Integrated

2018-09-10

1 Description

The build_scp_session tool computes offline the SCP frames corresponding to the provided parameters.

2 Usage

```
build_scp_session [OPTION] [PARAMETERS]... [ FOLDER [EXTRA_PARAM]... ]
```

OPTION	See General Options
PARAMETERS	See Parameters
FOLDER	Output folder to store SCP Packet
EXTRA_PARAM	See Extra Parameters

3 General Options

3.1 c - chip part number

-c CHIP_NAME - Use default configuration of CHIP_NAME.

3.2 Help

-h - Print this help and quit.

3.3 Version

-v - output software and libraries versions and quit.

3.4 Debug

-d - Activate debug output.

4 Parameters

Parameters are used by priority in the following order :

1. Command line
2. Configuration file " INIFILE " in the current folder
3. Chip default parameters selected by the -c option or the MAXIM_SBT_DEVICE env variable.
4. Software default values.

4.1 SCP Script file

```
script_file=script.txt
```

`script.txt` - text file containing SCP operation to perform. For more information see SCP Script Commands

4.2 Output filename prefix

```
output_file=nameprefix
```

`nameprefix` - filename prefix of SCP generated file (packets and logfile). Default value : `session.txt`

4.3 Output folder

```
output_dir=dir
```

`dir` - folder where SCP files will be saved. If this folder does not exist it will be created.

4.4 Key file

```
key_file=file.key
```

UCL format private key file for SCP packet signing. For more information see *UCL Key Format* documentation.

4.5 Session Mode

```
session_mode=mode
```

`mode` - SCP session mode to be used for the communication with SBL. Please refers to CHIP documentation to select the corect one. Available mode are :

- `SCP_FLORA_RSA`
- `MSP_MAXQ1852_ECDSA`
- `SCP_ECDSA`
- `SCP_LITE_ECDSA`
- `SCP_PAOLA`

4.6 Protection Profile

```
pp=PP
```

`PP` - SCP Protection profile to be used for the communication with SBL. Please refers to CHIP documentation to select the corect one. Available protection profile are :

- `RSA_2048`
- `RSA_4096`
- `ECDSA`

4.7 Verbose

```
verbose=level
```

verbose level (0-5)

4.8 Chunk Size

```
chunk_size=size
```

`size` - maximum data size for one SCP packet (in bytes), this value have to be set according the CHIP used.

4.9 Maximum Flash Size

```
flash_size_mb=size
```

`size` - maximum flash size in Mo, this define the memory allocated when reading a data file (S19, HEX or SBIN)

4.10 USN - Unique serial Number

```
usn=USN
```

`USN` - Unique Serial Number of the device you want to personnalized the SCP session for (i.e. kill-chip command).

4.11 Transaction ID (MAXQ1852 only)

```
transaction_id=trid
```

`trid` - User Selected transaction ID when using MSP_MAXQ1852_ECDSA.

4.12 Transaction ID (MAXQ1852 only)

```
addr_offset=address
```

`address` - address offset added when reading S19 files and base address when reading SBIN files.

5 HSM Parameter

This application can use a Thales(R) HSM for key storage and cryptographics operation. By default the application use it's builtin cryptographics functions.

5.1 HSM

```
hsm=yes
hsm=no
```

Use or not an HSM to manage key and perform cryptographics operations

5.2 HSM Key Name

```
hsm_key_name=name
```

`name` - name of the key to use stored inside the HSM.

5.3 HSM Thales DLL Location

```
hsm_thales_dll=dll_path
```

`dll_path` - path to the Thales cknfast DLL.

5.4 HSM SLot Number

```
hsm_slot_nb=nb
```

`nb` - number of the HSM slot to use (usually : 1).

6 Extra Parameters

In order to make *SCP scripts* more modular extra parmaters can be used. There are **TAGS** used in the script that will be replace at the excution with parameters provided in the commandline.

The format of the at is the following **%PARAM_N%**. With N from 0 to 9.

6.1 Example

Let the following script :

```
wrtie-otp 08bc %PARM_0%
write-otp 09bc %PARM_1%
write-file %PARM_2%
```

With the following command line call

```
build_scp_session scp_folder 02654212 ED45830A firmware.sbin
```

will become :

```
wrtie-otp 08bc 02654212
write-otp 09bc ED45830A
write-file firmware.sbin
```

> Note: When Using Extra parameters the folder **MUST** be specified.

7 SCP Script Commands

7.1 Write File

```
write-file filename [address]
```

This command send the binary data contains in the provided file to the SBL for writing using the `WRITE DATA` SCP Command. It also erase the target memory are using the `ERASE DATA` SCP Command.

`filename`	S19 or sbin file containing the data to be send for writing to the SBL.
`address`	Start address to where writing data from sbin file or offset adddress to add to S19 addresses. Optionnal whith S19 files but mandatory with sbin files.

7.2 Write Only

```
write-only filename [address]
```

This command send the binary data contains in the provided file to the SBL for writing using the `WRITE DATA` SCP Command.

`filename`	S19 or sbin file containing the data to be send for writing to the SBL.
`address`	Start address to where writing data from sbin file or offset adddress to add to S19 addresses. Optionnal whith S19 files but mandatory with sbin files.

7.3 Verify file

```
verify-file filename [address [dump]]
```

This command send the binary data contains in the provided file to the SBL for verification against the content of the memory writing using the `COMPARE DATA` SCP Command.

`filename`	S19 or sbin file containing the data to be send for writing to the SBL.
`address`	Start address to where start data for verification from sbin file or offset adddress to add to S19 addresse Optionnal whith S19 files but mandatory with sbin files.
`dump`	yes/no Add a dummy dump packet for the SCP sender

7.4 Write CRK

```
write-crk filename
```

This command send `WRITE-CRK` SCP Command. It send the CRK with it's signature by the MRK.

`filename`	File containing the CRK sign by the MRK

7.5 Rewrite CRK

```
rewrite-crk old_crk_filename new_crk_filename
```

This command send `REWRITE-CRK` SCP Command. It send the *old* CRK and the *new* CRK with it's signature by the MRK.

`old_crk_filename`	File containing the old CRK sign by the MRK
`new_crk_filename`	File containing the new CRK sign by the MRK

7.6 Echo

```
echo
```

This command check the communication with the SBL by sending an `ECHO` SCP command.

7.7 Write OTP

```
write-otp offset data
```

This command write data inside the CHIP OTP using the `WRITE-OTP` SCP Command.

`offset`	Address offset inside the OTP memory.
`data`	Data to write at the offset specified.

7.8 Write Time-out

```
write-timeout target value
```

This command write the timeout configuration for the different SCP bus using the `WRITE-TIMEOUT` SCP Command.

`target`	Bus for which the timeout will be written. Possible value are :

	`0` - for UART
	`V` - for VBUS
	`U` - for USB
	`E` - for Ethernet
	`S` - for SPI
`Value`	Value of the Timeout in ms.

7.9 Write Parameter

```
write-param target value
```

This command write the parameter configuration for the different SCP bus using the `WRITE-PARAM` SCP Command.

`target`	Bus for which the parameter will be written. Possible value are :
	`0` - for UART
	`V` - for VBUS
	`U` - for USB
	`E` - for Ethernet
	`S` - for SPI
`Value`	Value of the parameter.

7.10 Write Stimulus

```
write-stim target value
```

This command write the stimulus configuration for the different SCP bus using the `WRITE-STIM` SCP Command.

`target`	Bus for which the stimulus will be written. Possible value are :
	`0` - for UART

	`V` - for VBUS
	`U` - for USB
	`E` - for Ethernet
	`S` - for SPI
`Value`	Value of the stimulus.

7.11 Write Deactivation

```
write-deact target
```

This command deactivate the different SCP bus using the `WRITE-DEACT` SCP Command.

`target`	Bus for which the stimulus will be written. Possible value are :
	`0` - for UART
	`V` - for VBUS
	`U` - for USB
	`E` - for Ethernet
	`S` - for SPI

7.12 Kill Chip

```
kill-chip
```

This command send the `KILL-CHIP` SCP command to SBL.

7.13 Kill Chip USN

```
kill-chip2
```

This command send the `KILL-CHIP2` SCP command to SBL with the Chip Unique Serial Number (USN) provided with the corresponding option.

7.14 Execute Code / Register Applet

```
execute-code address
```

This command send the `EXECUTE-CODE` SCP command to SBL. This will register an applet if the adress point to an applet header or will launch an application if the address point to an application header.

`address`	Address of the previously loaded SCP applet or SLA Application.

8 Glossary

SBL	Secure Boot Loader
SCP	Secure Communication Protocol
USN	Unique Serial Number
OTP	One Time Programmable Memory
CRK	Customer Root Key
MRK	Maxim Root Key