Telink Secure Boot Post Tool

User Guide

Keyword

Post; Signature; Public Key; Secure Boot;

Brief

This document mainly introduces the use of Telink Post Tool.

Published by
Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

Legal Disclaimer

Information

For further information on the technology, product and business term, please contact Telink Semiconductor Company www.telink-semi.com

For sales or technical support, please send email to the address of:

telinksales@telink-semi.com

telinksupport@telink-semi.com

Version

Version	Change Description
V1.0.0	Initial release

Contents

List of Tables

1 Overview

This tool is a supporting tool for the SecurityBoot function, mainly used to generate run descriptors that match the running code. The definition of runtime descriptors for different chips or kernels may vary, and when using them, it is necessary to first modify the relevant configuration files according to the actual situation.

2 Command Options

2.1 --config

Description：

This option is used to set the configuration file for generating run descriptors.

Usage：

```
--config C:/desktop/config.ini
```

2.2 --input

Description：

This option is used to set the runtime code file for generating run descriptors.

Usage：

```
--input C:/desktop/test.bin
```

Note：

Input file must end with ". bin"

2.3 --output

Description：

This option is used to set the output of the run descriptor.

Usage：

```
--output C:/desktop/test_run_desc.bin
```

Note：

Output file must end with ". bin"

2.4 `--private-key`

Description:

This option is used to set the private key and is not a necessary option. By default, the private key value will be read from the configuration file. If the private key value is illegal, a pair of secret keys will be randomly generated and the private key value will be saved in the configuration file.

Usage:

```
--private-key C:/desktop/test_run_desc.bin 10c3f60066a6b51981b4135155a23fe62d7aa7376e8fe31343184788f7eaa10e
```

Note:

The private key must be entered in hexadecimal format, with a length of 32 bytes.

3 Examples

Next, use the Windows command environment to illustrate the usage.

3.1 Generate run descriptors with the private key in configuration file

Command:

```
Post_Tool_Win.exe -- config C:/ desktop/ config.ini -- input C:/ desktop/ test.bin -- output C:/
desktop/test_run_desc.bin
```

3.2 Generate run descriptors with other private key

Command:

```
Post_Tool_Win.exe -- config C:/ desktop/ config.ini -- input C:/ desktop/ test.bin -- output C:/
desktop/test_run_desc.bin --private-key 10c3f60066a6b51981b4135155a23fe62d7aa7376e8fe31343184788f7eaa10e
```

4 Configuration File

This configuration file consists of the "Post_Data" section, "Offset_Record" Section and the data sections which used to specify the data content.

4.1 Section - "Post_Data"

This section is used to specify the data composition of the run descriptor and some custom information.

4.1.1 Key - "data_list"

This key used to specify the data composition of the generated file, with values consisting of the names of the data sections separated by commas in English.

4.1.2 Key - "private_key"

This key is used to set the private key value. The value must be entered in hexadecimal format and has a length of 32 bytes.

4.1.3 Key - "public_key_hash_name"

This key is used to set the output file name, which is used to store the public key hash value. It should be noted that the file name must end with ". bin".

4.2 Section - "Offset_Record"

This section is used to configure the data section that need to be recorded with offset addresses.

4.2.1 Key - "record_list"

This key is used to set the data section that the offset address needs to be recorded. The value consists of the name of the data section and is separated by commas in English. It should be noted that the name of the data section must be in "PostData ->data_list".

4.2.2 Key - "record_list_max_size"

This key is used to set the maximum number of "data section" that can be recorded as offset addresses.

4.2.3 Key - "record_unit_byte_size"

This key is used to set the data length for saving offset addresses, which should not exceed 4 bytes at most.

4.2.4 Key - "record_unit_big_endian"

This key is used to specify whether the data used to store offset addresses is stored in large end or small end mode. If the value is "True", the large end mode is used, and if not, the small end mode will be used.

4.3 Section - data section

This section is used to specify the data content, and its specific composition is as follows.

4.3.1 Key - "value"

This key is used to specify the content of valid data, and its value must be the key of the "EnumerateList" section.

4.3.2 Key - "value_byte_size"

This key is used to specify the length of valid data, in bytes, and the corresponding value must be filled in decimal form.

4.3.3 key - "extend_byte_size"

This key is used to specify the total length of data (the sum of valid and filled data), in bytes, and the corresponding value must be filled in decimal form.

4.3.4 key - "fill_value"

This key is used to set the filling value, and the corresponding value must be filled in hexadecimal format.

4.3.5 Sub Section - "Enumerate_List"

This section is used to configure the values that can be set for the "value" key.