XCP Seed and Key Shared Library - Reference Manual

1.0.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

XCP Seed and Key shared library (SeedNKey)

1.1 Introduction

The OpenBLT bootloader is an open source project, allowing everyone to access the sources. One downside of this is that if someone knows the OpenBLT bootloader is used in your product, it is relatively easy for them to figure out how to update the firmware in your product. Although this could be a feature of your product, in most cases it is not desirable.

For this reason the bootloader contains a seed/key security module. If this security module is enabled in the bootloader's configuration, updates can only be made by users that have the correct algorithm inside this Seed and Key shared library. If not, then new firmware update requests are rejected by the bootloader.

The SeedNKey project is preconfigured to build this Seed and Key shared library. Its default implemented seed and key algorithm works together with those implemented in the OpenBLT demo bootloaders. You are free and encouraged to update this algorithm to match your security needs.

On the bootloader target side (microcontroller), this security module is enabled by setting configurable BOOT_XCP_SEED_KEY_ENABLE to a value of 1 in the bootloader's configuration header file (blt_conf.h). The implementation of the seed and key algorithm can be made in the hook-functions XcpGetSeedHook() and XcpVerifyKeyHook().

On the host side (PC), you simply specify the Seed and Key shared library file that this preconfigured project built. In MicroBoot this shared library can be specified through the settings user interface. In BootCommander it is specified via a command line option.

Refer to the OpenBLT website for additional information regarding the XCP Seed and Key shared library, including step-by-step instructions on how to build it: https://www.feaser.com/openblt/doku.php?id=manual:security php?id=manual:libopenblt.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Module Documentation

4.1 XCP Seed/Key

XCP Seed and Key shared library.

Files

- file seednkey.c

  XCP Seed and Key shared library source file.
- file seednkey.h

  XCP Seed and Key shared library header file.

4.1.1 Detailed Description

XCP Seed and Key shared library.

This shared library implements an example XCP Seed and Key protection algorithm. If the OpenBLT bootloader on the microcontroller is configured to support this protection, this shared library file must be configured in the firmware update tool on the host (for example MicroBoot or BootCommander). The OpenBLT bootloader will reject new firmware update requests, if an incorrect XCP Seed and Key protection algorithm is specified.

You are free and even encouraged to change the protection algorithm in this shared library to however you see fit to protect your target from unwanted firmware updates.

Chapter 5

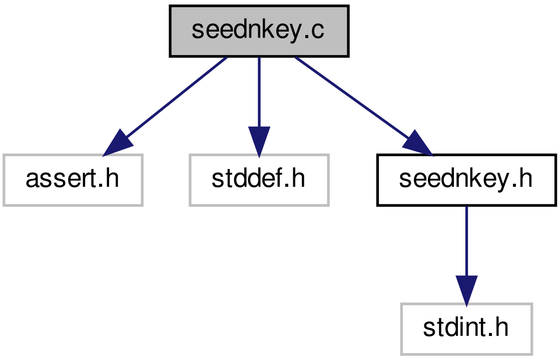
File Documentation

5.1 seednkey.c File Reference

XCP Seed and Key shared library source file.

```
#include <assert.h>
#include <stddef.h>
#include "seednkey.h"
```

Include dependency graph for seednkey.c:



Functions

- LIBOPENBLT_EXPORT uint32_t XCP_ComputeKeyFromSeed (uint8_t resource, uint8_t seedLen, uint8_t const ∗seedPtr, uint8_t ∗keyLenPtr, uint8_t ∗keyPtr)

    Computes the key for the requested resource.
- LIBOPENBLT_EXPORT uint32_t XCP_GetAvailablePrivileges (uint8_t ∗resourcePtr)

    Obtains a bitmask of the resources for which an key algorithm is available.

5.1.1 Detailed Description

XCP Seed and Key shared library source file.

5.1.2 Function Documentation

5.1.2.1 XCP_ComputeKeyFromSeed()

```
LIBOPENBLT_EXPORT uint32_t XCP_ComputeKeyFromSeed (
            uint8_t resource,
            uint8_t seedLen,
            uint8_t const * seedPtr,
            uint8_t * keyLenPtr,
            uint8_t * keyPtr )
```

Computes the key for the requested resource.

Parameters

resource	resource for which the unlock key is requested
seedLen	length of the seed
seedPtr	pointer to the seed data
keyLenPtr	pointer where to store the key length
keyPtr	pointer where to store the key data

Returns

XCP_RESULT_OK on success, otherwise XCP_RESULT_ERROR.

5.1.2.2 XCP_GetAvailablePrivileges()

```
LIBOPENBLT_EXPORT uint32_t XCP_GetAvailablePrivileges (
            uint8_t * resourcePtr )
```

Obtains a bitmask of the resources for which an key algorithm is available.

Parameters

resourcePtr	pointer where to store the supported resources for the key computation.

Returns

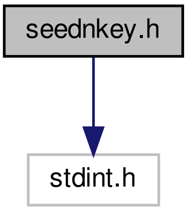
XCP_RESULT_OK on success, otherwise XCP_RESULT_ERROR.

5.2 seednkey.h File Reference

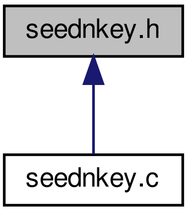
XCP Seed and Key shared library header file.

```
#include <stdint.h>
```

Include dependency graph for seednkey.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define XCP_RESULT_OK (0u)

    Result value in case of success.
- #define XCP_RESULT_ERROR (1u)

    Result value in case of error.
- #define XCP_RESOURCE_PGM (0x10u)

    XCP ProGraMing resource.
- #define XCP_RESOURCE_STIM (0x08u)

    XCP data STIMulation resource.
- #define XCP_RESOURCE_DAQ (0x04u)

    XCP Data AcQuisition resource.
- #define XCP_RESOURCE_CALPAG (0x01u)

    XCP CALibration and PAGing resource.

Functions

- LIBOPENBLT_EXPORT uint32_t XCP_ComputeKeyFromSeed (uint8_t resource, uint8_t seedLen, uint8_t const ∗seedPtr, uint8_t ∗keyLenPtr, uint8_t ∗keyPtr)

  Computes the key for the requested resource.
- LIBOPENBLT_EXPORT uint32_t XCP_GetAvailablePrivileges (uint8_t ∗resourcePtr)

  Obtains a bitmask of the resources for which an key algorithm is available.

5.2.1 Detailed Description

XCP Seed and Key shared library header file.

5.2.2 Function Documentation

5.2.2.1 XCP_ComputeKeyFromSeed()

```
LIBOPENBLT_EXPORT uint32_t XCP_ComputeKeyFromSeed (
            uint8_t resource,
            uint8_t seedLen,
            uint8_t const ∗ seedPtr,
            uint8_t ∗ keyLenPtr,
            uint8_t ∗ keyPtr )
```

Computes the key for the requested resource.

Parameters

resource	resource for which the unlock key is requested
seedLen	length of the seed
seedPtr	pointer to the seed data
keyLenPtr	pointer where to store the key length
keyPtr	pointer where to store the key data

Returns

XCP_RESULT_OK on success, otherwise XCP_RESULT_ERROR.

5.2.2.2 XCP_GetAvailablePrivileges()

```
LIBOPENBLT_EXPORT uint32_t XCP_GetAvailablePrivileges (
            uint8_t ∗ resourcePtr )
```

Obtains a bitmask of the resources for which an key algorithm is available.

Parameters

resourcePtr	pointer where to store the supported resources for the key computation.

Returns

XCP_RESULT_OK on success, otherwise XCP_RESULT_ERROR.

Index