



maxim
integrated.TM

MAX32590/MAX32550

HSM User Guide

UG25T03
Revision A
10/13/2014

Maxim Integrated, Inc.
160 Rio Robles
San Jose, CA 95134

MAXIM Integrated PROPRIETARY – CONFIDENTIALITY

This document contains confidential information that is the strict proprietary of Maxim Integrated, and may be disclosed only under the writing permission of Maxim Integrated itself. Any copy, reproduction, modification, use or disclose of the whole or only part of this document if not expressly authorized by Maxim Integrated is prohibited. This information is protected under trade secret, unfair competition and copying laws. This information has been provided under a Non Disclosure Agreement. Violations thereof may result in criminalities and fines.

Maxim Integrated reserves the right to change the information contained in this document to improve design, description or otherwise. Maxim Integrated does not assume any liability arising out of the use or application of this information, or of any error of omission in such information. Except if expressly provided by Maxim Integrated in any written license agreement, the furnishing of this document does not give recipient any license to any intellectual property rights, including any patent rights covering the information in this document.

All trademarks referred to this document are the property of their respective owners.

Copyright © 2014 Maxim Integrated. All rights reserved. Do not disclose.

Revision History

Rev A	2014-Oct-13	1 st release
-------	-------------	-------------------------

Disclaimer

To our valued customers

We constantly strive to improve the quality of all our products and documentation. We have spent an exceptional amount of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please contact us. We appreciate your assistance in making this a better document.

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Maxim Integrated Office.

Maxim Integrated Technologies may only be used in life-support devices or systems with the express written approval of Maxim Integrated, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Table of Contents

MAX32590/MAX32550.....	0
HSM User Guide	0
MAXIM Integrated PROPRIETARY – CONFIDENTIALITY	1
Revision History	2
Table of Contents	3
Glossary.....	4
References.....	5
1 INTRODUCTION	6
2 nCipher Tools installation.....	7
2.1.1 Installation requirements.....	7
2.1.2 Thales nShield Edge HSM installation	7
2.1.3 Installation package	7
3 CRK generation	13
3.1 mxim_hsm_manager application	15
3.2 MAX32590.....	15
3.2.1 Generate new RSA key pair	15
3.2.2 Export public RSA key	17
3.2.3 Remove RSA key pair	19
3.3 MAX32550.....	22
3.3.1 Generate new ECDSA key pair	22
3.3.2 Export public ECDSA key	23
3.3.3 Remove ECDSA key pair	26
4 SCP packets generation.....	29
4.1 nCipher_session_build application	29
4.1.1 Configuration file	29
4.1.2 SCP packets building.....	29
5 Customer application signature	33
5.1 nCipher_ca_sign_build application	33
5.1.1 Configuration file	33
5.1.2 Application signature	33
6 ANNEX: THALES nShield Edge USB FTDI drivers	37

Glossary

ACS	Administrator Card Set
CRK	Customer Root RSA Key
HSM	Hardware Security Module
MRK	Maxim Integrated Root RSA Key
OCS	Operator Card Set
SCP	Secure Communication Protocol
SoC	System On Chip

E/UG25T03/77682591

References

- [1] PCI-PTS modular security requirements. V4.0 Jun-2013
- [2] nShield Edge User Guide.
- [3] UG21T24. MAX32590 Secure ROM User Guide. Rev A.
- [4] UG25H04. MAX32550 Secure ROM User Guide. Rev D.

E/UG25T03/77682591

1 INTRODUCTION

This document explains how to use the HSM from an operator point of view. More particularly, it describes the security procedures to enforce in order to be compliant with security standards like PCI-PTS (see [1]). It also describes the way to use it. For more details about the HSM installation, refer to the Thales nShield Edge documentation (see [2]).

This document describes the procedure for the MAX32590 (RSA keys) and MAX32550 (ECDSA keys).

The HSM is used for four purposes:

- For nCipher Tools installation, see section [2](#).
- For CRK generation and certification, see section [3](#).
- For SCP packets signature and generation, see section [4](#).
- For customer application signature, see section [5](#).

2 nCipher Tools installation

Maxim Integrated has developed a suite of tools in order to use the **THALES nShield EDGE** HSM.

This suite is called: ***nCipher Tools***.

2.1.1 Installation requirements

- Supported OS: Windows XP, 7.
- Supported HSM: Thales nShield Edge

2.1.2 Thales nShield Edge HSM installation

The first step consists of HSM installation. Simply use the CD (so called “ncss-win-Edge-11”) from the THALES installation package.

2.1.2.1 Software installation

DO NOT PLUG THE HSM BEFORE INSTALLING SOFTWARE AND DRIVERS!!!

- Supported OS: Windows XP/ Windows 7
- JRE installation is required before going ahead. “The nCipher software supports JRE/JDK version 1.4.2, 1.5, and 1.6. The nCipherKM JCA/JCE CSP supports JRE/JDK 6.0” extract from THALES documentation.
- Launch setup.exe at root path.
- Follow the guide lines and choose default choice each time.

2.1.2.2 Configuration

Before to continue with ***nCipher Tools*** installation, it's required to perform following operations referring to the Thales nShield Edge HSM documentation:

1. Create a Security World (***new-world.exe*** command line tool).
2. Create OCS (***createocs.exe*** command line tool).

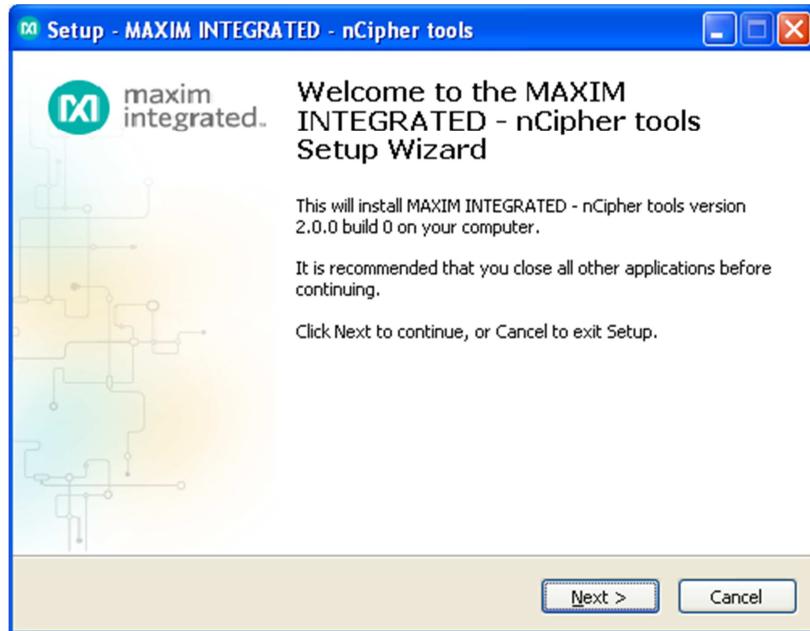
Remark:

These two operations require the user to define a quorum k/n for ACS cards, during Security world creation step, and another one for OCS cards. Achieving a quorum requires a defined chosen number of smart cards (k) from a total set (n) be brought together for an operation to be authorized. For compliance with PCI PTS, we recommend at least a dual control so that means “k” value equal or superior to 2. The “n” value can be defined as follow “n” \geq “k”+1.

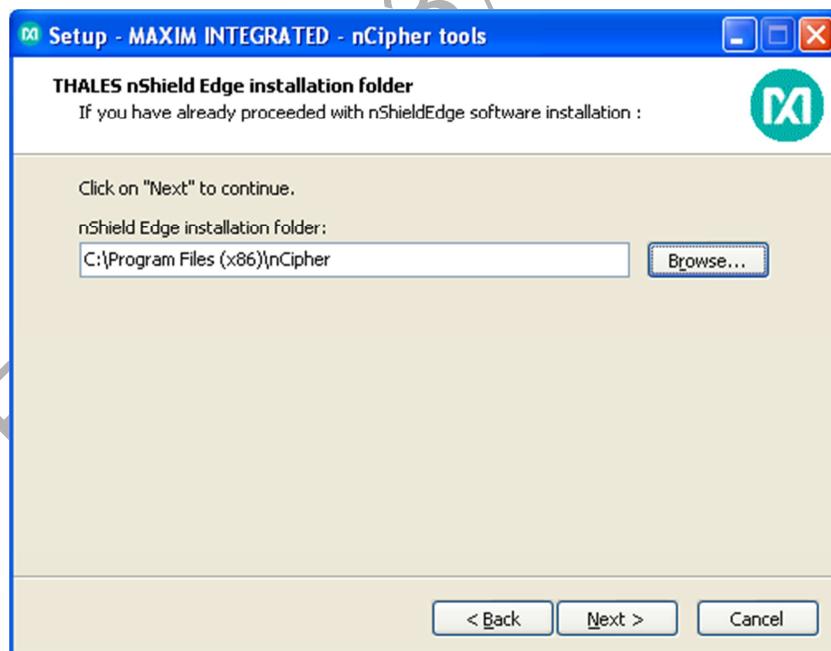
2.1.3 Installation package

Depending on the targeted chip, the following package has to be installed:

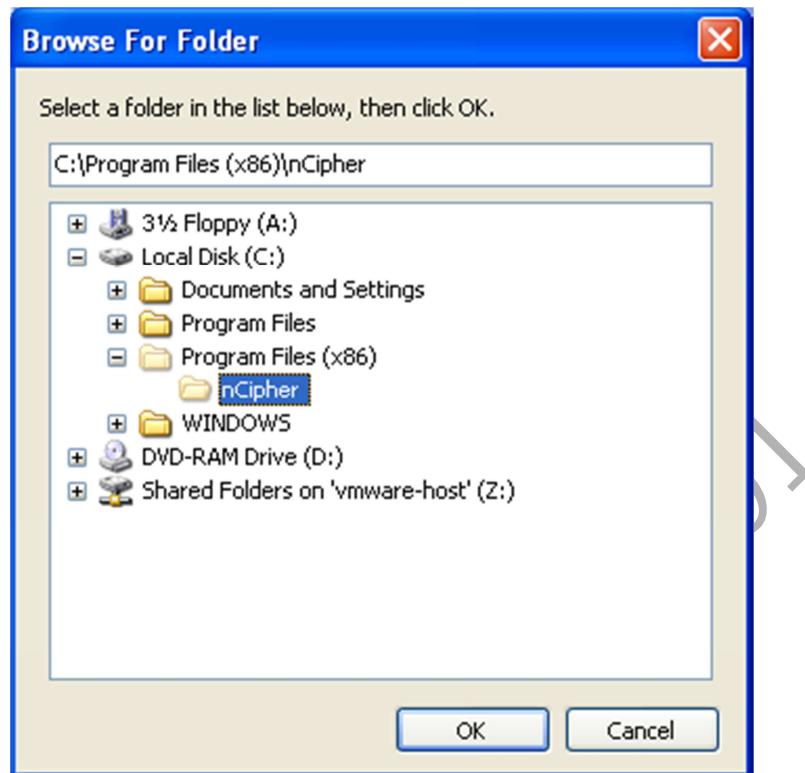
- Installation package name:
 - For MAX32590: ***mxim_ncipher_tools_setup.exe***
 - For MAX32550: ***mxim_ncipher_tools_MAX32550_setup.exe***
- Launch installation:



- Click on “Next”, enter **Thales nShield Edge** Installation folder:
Select the folder where you installed **Thales nShield Edge** software.

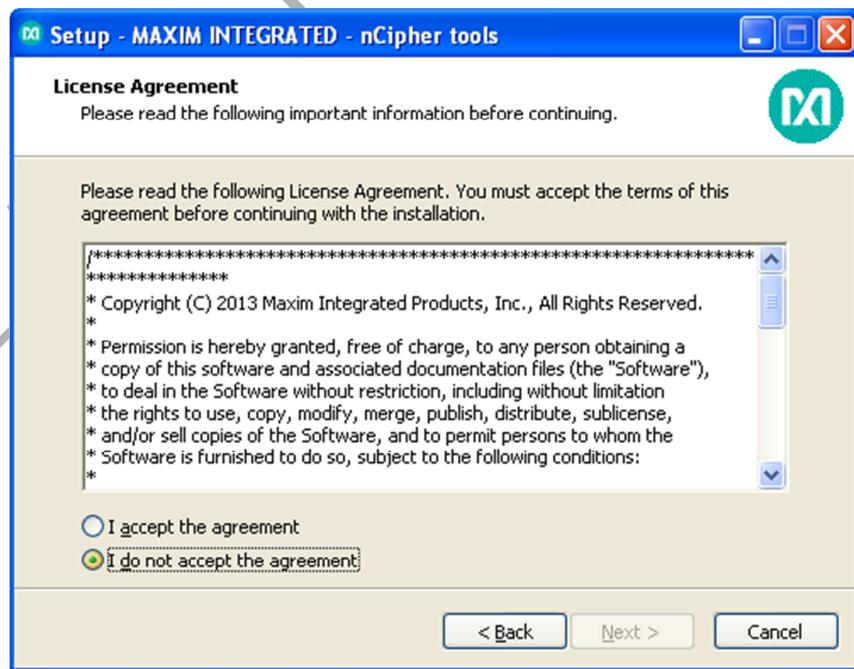


- Browse Thales nShield Edge software installation:

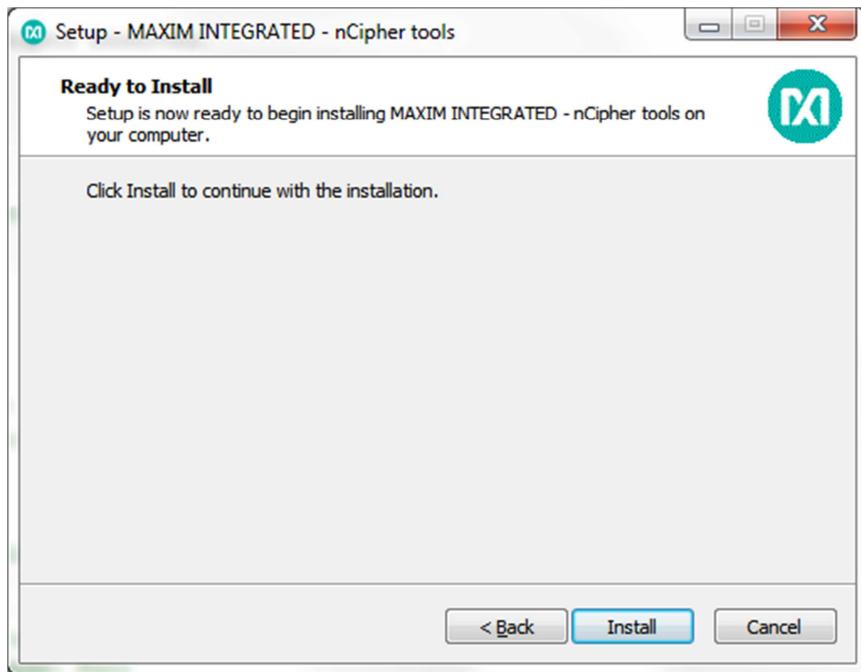


Click on **OK** to validate the choice.

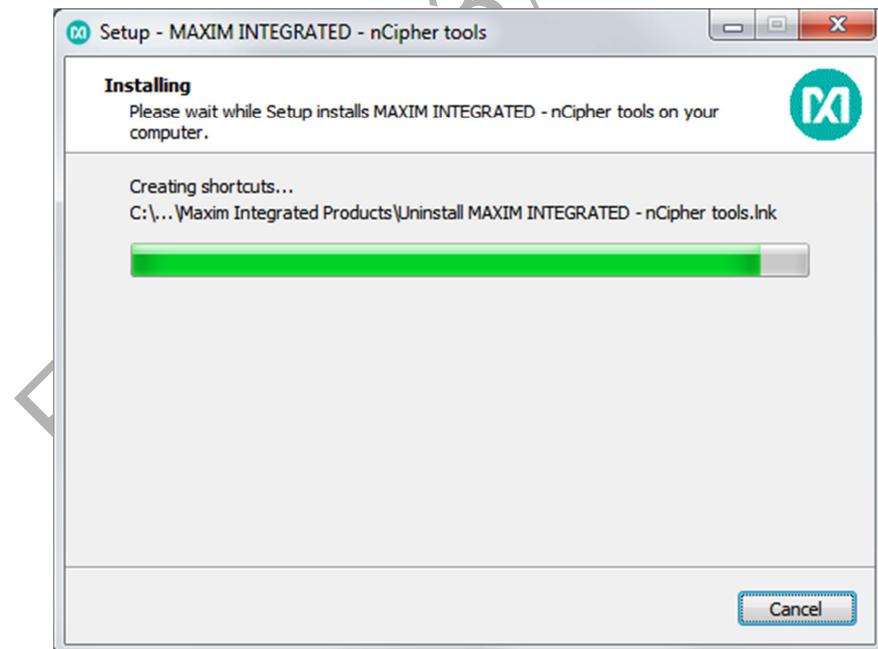
- Read and accept license agreement. Finally click on **Next**:



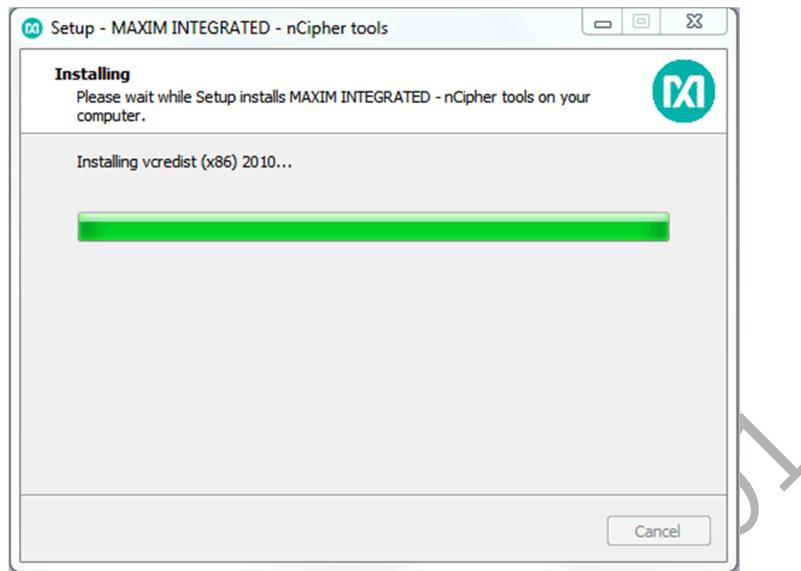
- Click on "Install":



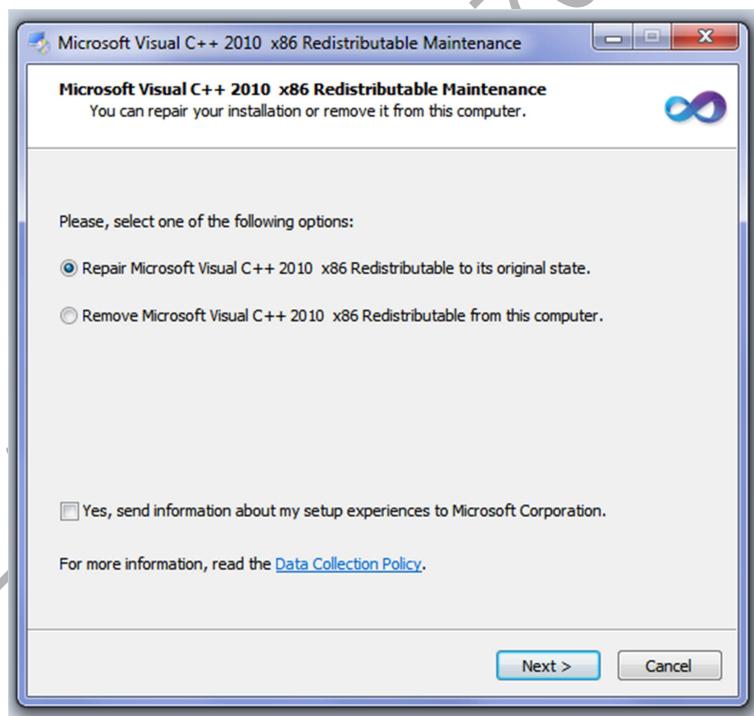
The setup first has to stop Thales nShield Edge services: “nFast Edge” and “nFast Server”.



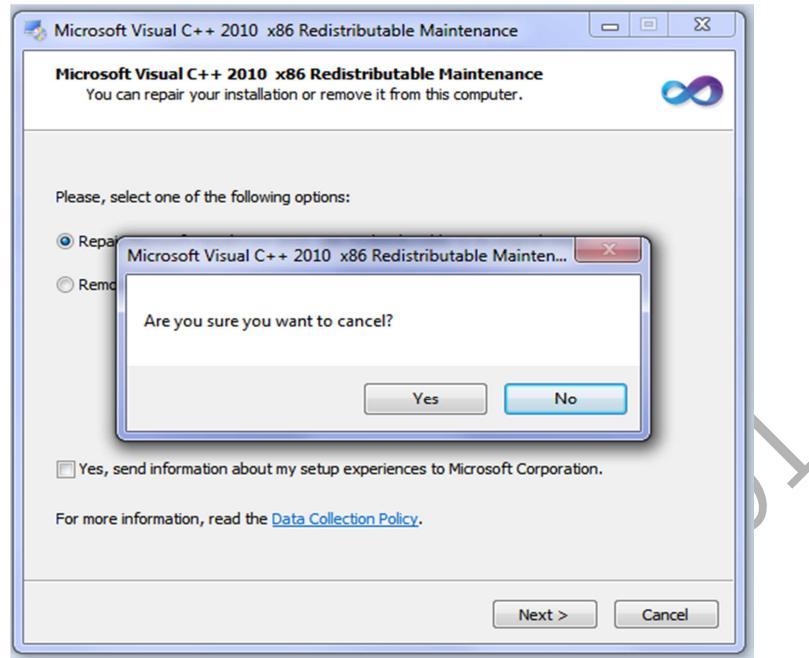
- Redistributable Visual Studio C++ package installation:



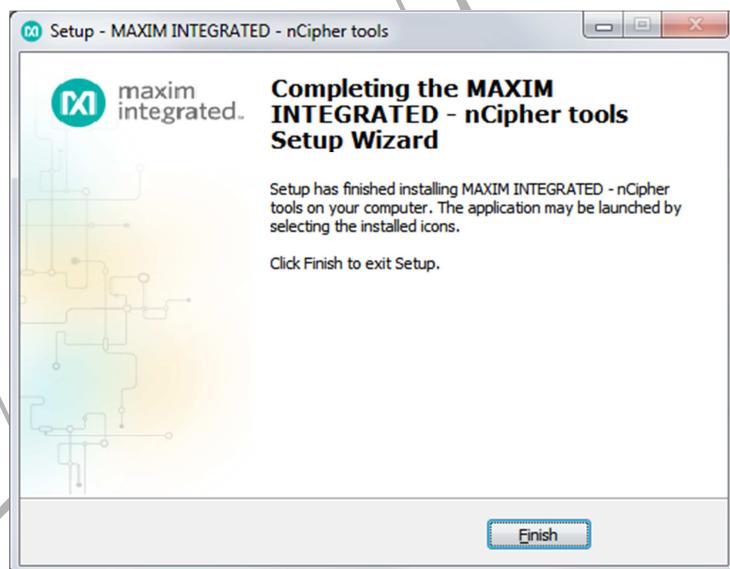
The package will proceed with the components installation but it will stop if it has been already installed and display the following window:



Ignore this window and click on **Next**. The installer requires you to confirm, so click **Yes**:



- Wait for the end and click on **Finish**:



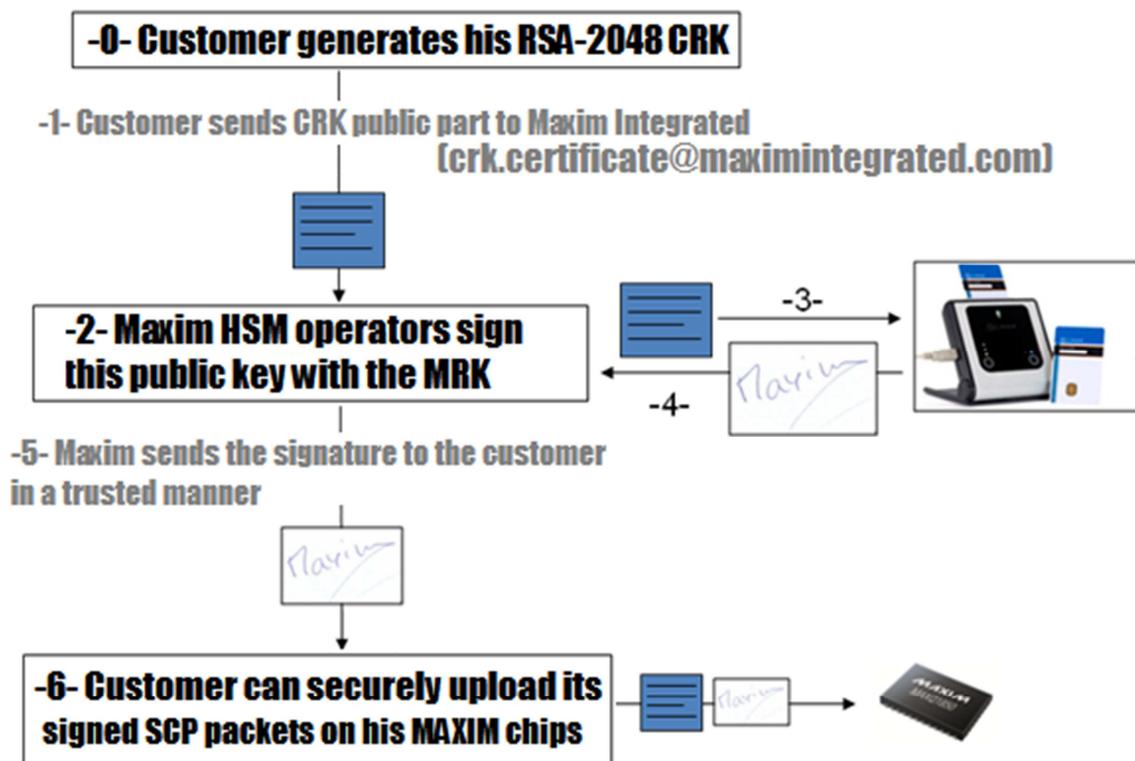
3 CRK generation

Customers have to generate their own key, the CRK and ask Maxim Integrated to certify this key, by issuing certified (i.e. signed) SCP packets, which include the CRK, signed by the MRK.

The diagram below describes this process, to be performed once per CRK.

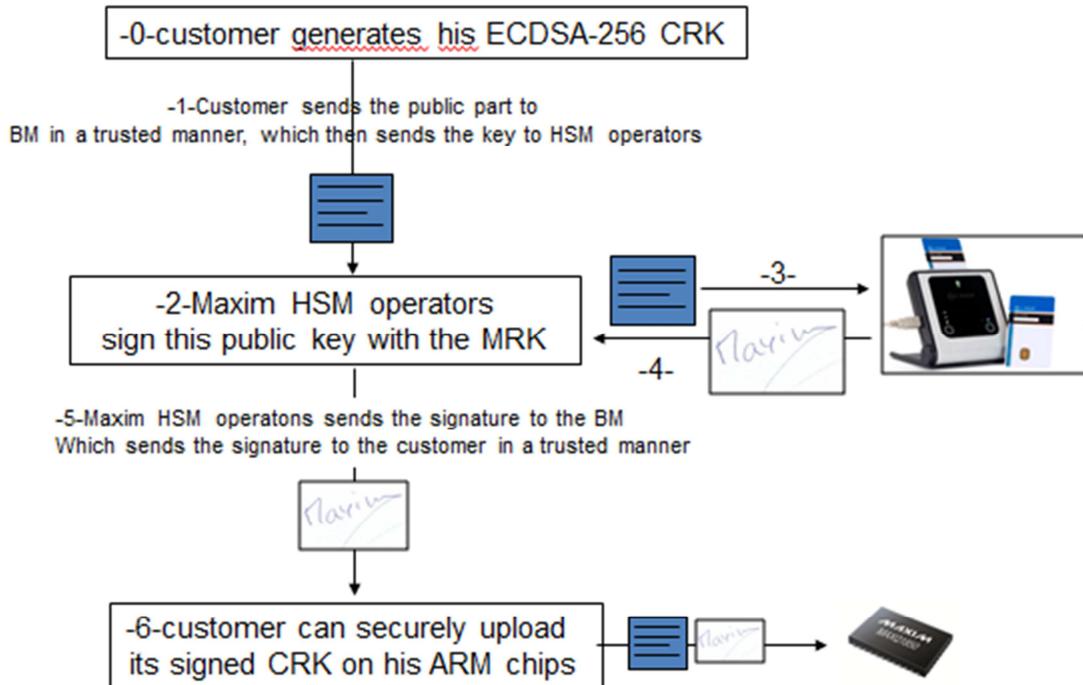
E/UG25T03/77682591

- For MAX32590 (RSA key process):



- For MAX32550 (ECDSA key generation process)

▪ Customer ECDSA key signing



3.1 mxim_hsm_manager application

This application belongs to **nCipher Tools** suite. This command line allows to generate the CRK.

First of all, ***run as administrator*** Windows console and type following commands:

- cd <ncipher_tools_installation_folder>\bin
 - mxim_hsm_manager.exe --help

The last command displays the help information.

3.2 MAX32590

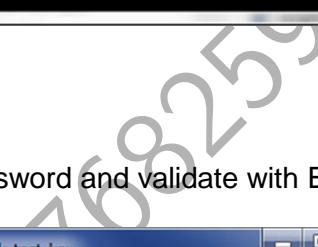
3.2.1 Generate new RSA key pair

Considering a security world (see section 2.1.2.2) is present on the PC and all **OCS** cards required accordingly to the quorum are available, you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- `mxim_hsm_manager.exe --generate-key <name_of_rsa_key_pair>`

Example:

Type following command: `mxim_hsm_manager.exe --generate-key crk_test_key`



```

mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key
0x1000000D: Encrypt operation failed.
0x1000000E: Decrypt operation failed.
0x1000FFFF: unknown error.

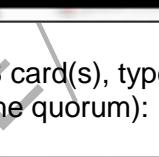
EXAMPLES
mxim_hsm_manager --generate-key crk_test_key
mxim_hsm_manager --export-key crk_test_key.txt --object-label crk_test_key
mxim_hsm_manager --delete-key crk_test_key
mxim_hsm_manager --objects-info --object-label crk_test_key

BUGS
No known bugs.

C:\Program Files (x86)\Maxim Integrated Products\ncipherTools\bin\mxim_hsm_manager.exe --generate-key crk_test_key
Using CKNFAST_DEBUGFILE C:\ProgramData\ncipher\Log Files\nfast.debug failed
NFLLog_AddFileDriver No error failed

```

Insert first OCS card, so called **token**, type the password and validate with ENTER:



```

mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key
bf63cef79d 0x00000465 009D14B8
Object uc7a2be79c0bf2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

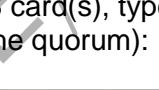
- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done. >:

```

Insert next OCS card(s), type the password and validate with ENTER (repeat this operation accordingly to the quorum):



```

mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

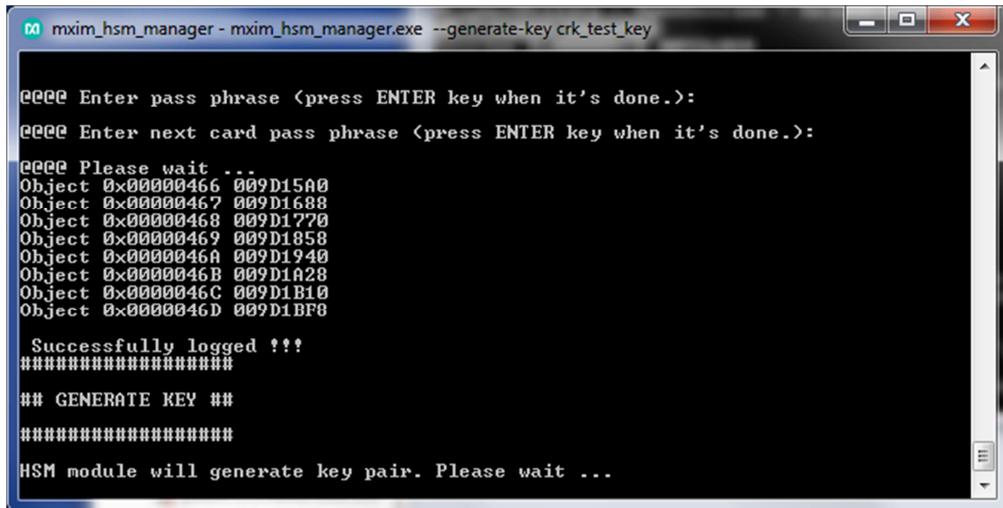
- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done. >:
#### Enter next card pass phrase <press ENTER key when it's done. >:

```

HSM will generate RSA key pair. The operation takes some time.



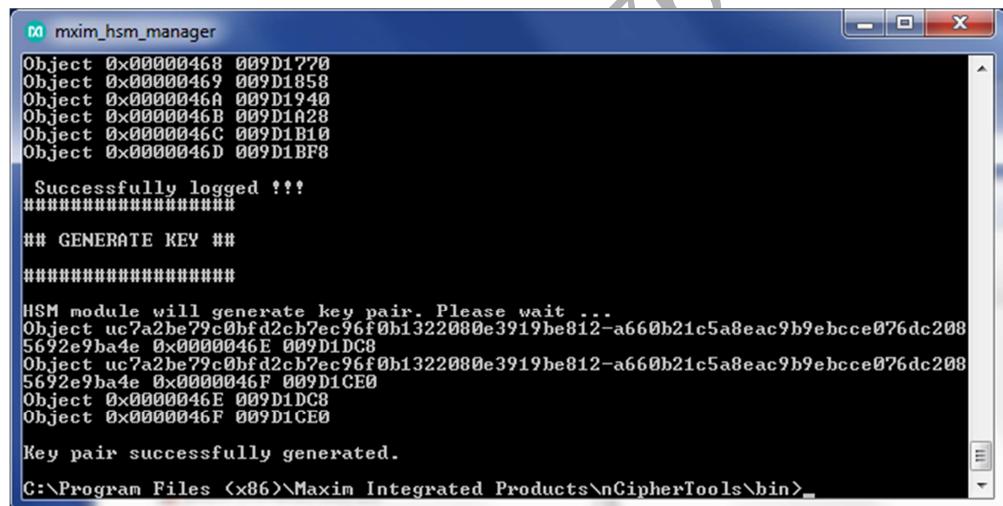
```
mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key

eeee Enter pass phrase <press ENTER key when it's done.>:
eeee Enter next card pass phrase <press ENTER key when it's done.>:
eeee Please wait ...
Object 0x00000466 009D15A0
Object 0x00000467 009D1688
Object 0x00000468 009D1770
Object 0x00000469 009D1858
Object 0x0000046A 009D1940
Object 0x0000046B 009D1A28
Object 0x0000046C 009D1B10
Object 0x0000046D 009D1BF8

Successfully logged !!!
#####
## GENERATE KEY ##
#####

HSM module will generate key pair. Please wait ...
```

And finally ...



```
mxim_hsm_manager

Object 0x00000468 009D1770
Object 0x00000469 009D1858
Object 0x0000046A 009D1940
Object 0x0000046B 009D1A28
Object 0x0000046C 009D1B10
Object 0x0000046D 009D1BF8

Successfully logged !!!
#####
## GENERATE KEY ##
#####

HSM module will generate key pair. Please wait ...
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046E 009D1DC8
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046F 009D1CE0
Object 0x0000046E 009D1DC8
Object 0x0000046F 009D1CE0

Key pair successfully generated.

C:\Program Files (x86)\Maxim Integrated Products\cipherTools\bin\
```

3.2.2 Export public RSA key

Considering a security (see section 2.1.2.2) is present on the PC, all **OCS** cards required accordingly to the quorum are available, and one RSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- `mxim_hsm_manager.exe --export-key <path_and_file_name> --object-label <name_of_rsa_key>`

Example:

To check that the RSA key exists:

```
mxim_hsm_manager.exe --objects-info --object-label crk_test_key
```

```
#####
## OBJECTS INFORMATION ##
#####
WARNING: the class attribute of the object has not been defined.
WARNING: the identifier attribute of the key has not been defined.

Cryptoki object <Handle=0x0000046D>:
- Class name: CKO_PRIVATE_KEY
- Label: crk_test_key
- Key identifier:
- Modulus bits size: 2048
- Modulus bytes size: 256

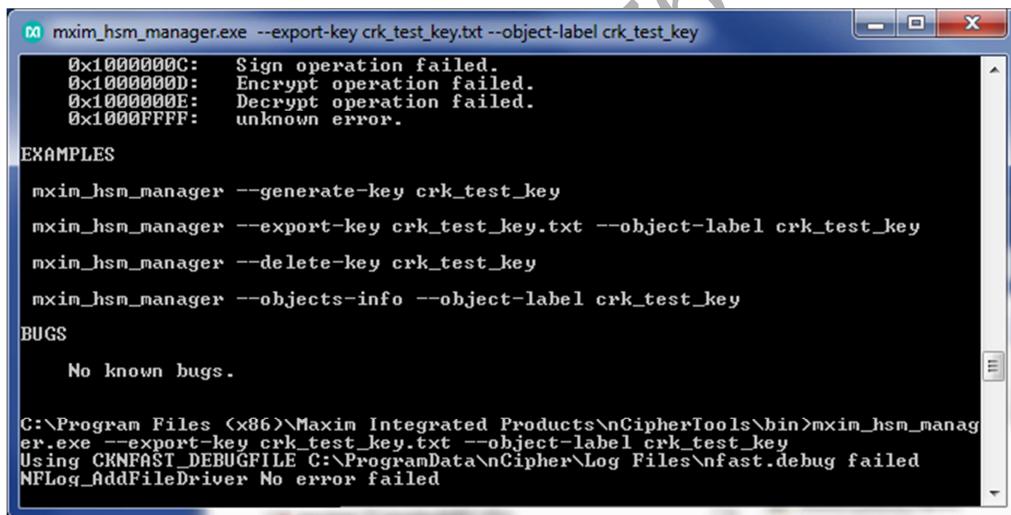
- Modulus data:
a75b4be7926012791f04d8cd77d58b0e1230e0e7206ba0856109198781004fd400f2aeee6ceca1488c4e812fbh60ec1bbfa196d314213c9207
d4907487943916627b40e9ae615b1a00529c51848a6085d5ec5e9486ab29e9a11fc2bd14e5a771296a4a9e73caf957cc78379634cb46e470d2f8
88c94384286eccdf23b7d544c11991bb8de60f91a5d9fd4b236126613bf6f3bc1b03496e238a8f6baf671386885843558a404ce9257b19c7409

MKIM::Cryptoki
Unable to get cryptoki object attribute <CKA_PRIVATE_EXPONENT> value <error CKR_ATTRIBUTE_SENSITIVE: 0x00000011>.

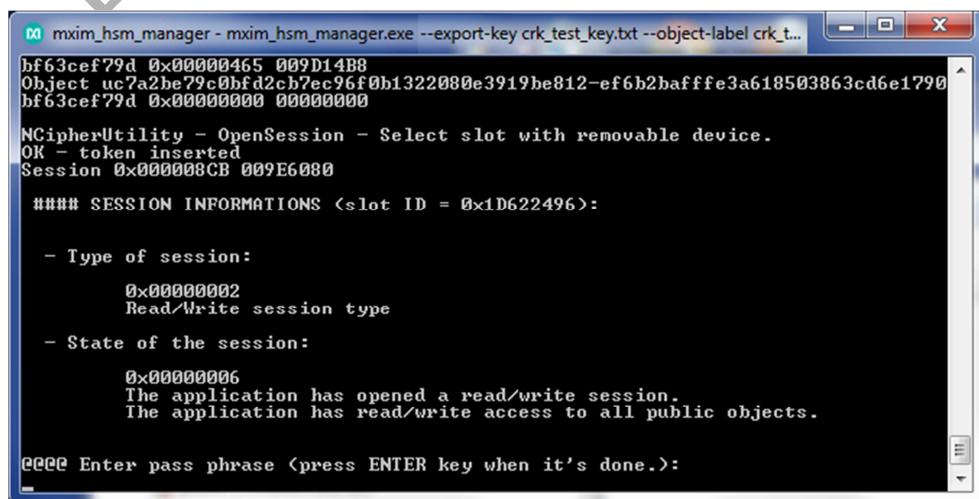
Cryptoki object <Handle=0x00000464>:
- Class name: CKO_PUBLIC_KEY
- Label: crk_test_key
- Key identifier:
- Modulus bits size: 2048
- Modulus bytes size: 256
- Modulus data:
a75b4be7926012791f04d8cd77d58b0e1230e0e7206ba0856109198781004fd400f2aeee6ceca1488c4e812fbh60ec1bbfa196d3142
9e04ede0d4907487943916627b40e9ae615b1a00529c51848a6085d5ec5e9486ab29e9a11fc2bd14e5a771296a4a9e73caf957cc78379634cb46
4db8dd3488e94384286eccdf23b7d544c11991bb8de60f91a5d9fd4b236126613bf6f3bc1b03496e238a8f6baf671386885843558a404ce9257b
- Exponent data:
65537 <0x00010001>
```

Then type following command:

```
mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label
crk_test_key
```



Insert first OCS card, so called **token**, type the password and validate with **ENTER**:



Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):

```
mxim_hsm_manager - mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_t...
bf63cef79d 0x00000000 00000000
NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000000CB 009E6000
##### SESSION INFORMATIONS <slot ID = 0x1D622496>:
- Type of session:
  0x00000002
  Read/Write session type
- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

EEEE Enter pass phrase <press ENTER key when it's done.>: 
EEEE Enter next card pass phrase <press ENTER key when it's done.>: 
```

HSM will export public RSA key.

```
EEEE Enter next card pass phrase <press ENTER key when it's done.>: 
EEEE Please wait...
Object 0x00000467 00511688
Object 0x00000468 00511720
Object 0x00000469 00511858
Object 0x0000046A 00511940
Object 0x0000046B 00511A28
Object 0x0000046C 00511B10
Object 0x0000046D 00511BF8
Object 0x0000046E 00511CE0
Object 0x0000046F 00511DC8

Successfully logged !!!
#####
## EXPORT RSA KEY ##
#####
WARNING: the class attribute of the object has not been defined.
WARNING: the identifier attribute of the key has not been defined.

Public key successfully exported.
C:\Program Files <x86>\Maxim Integrated Products\ncipherTools\bin>
```

The exported file result:

1	a75b4be7926012791f04d8cd77d58b8e1230e0e7206b ...
2	10001
3	

First line is corresponding to the modulus of the public RSA key.

Second line is corresponding to the exponent of the public RSA key.

3.2.3 Remove RSA key pair

Considering a security (see section 2.1.2.2) is present on the local machine, all **OCS** cards required accordingly to the quorum are available, and one RSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- mxim_hsm_manager.exe --delete-key <name_of_rsa_key>

Example:

To check that the RSA key exists:

```
mxim_hsm_manager.exe --objects-info --object-label crk_test_key
```

```
#####
## OBJECTS INFORMATION ##
#####
WARNING: the class attribute of the object has not been defined.
WARNING: the identifier attribute of the key has not been defined.

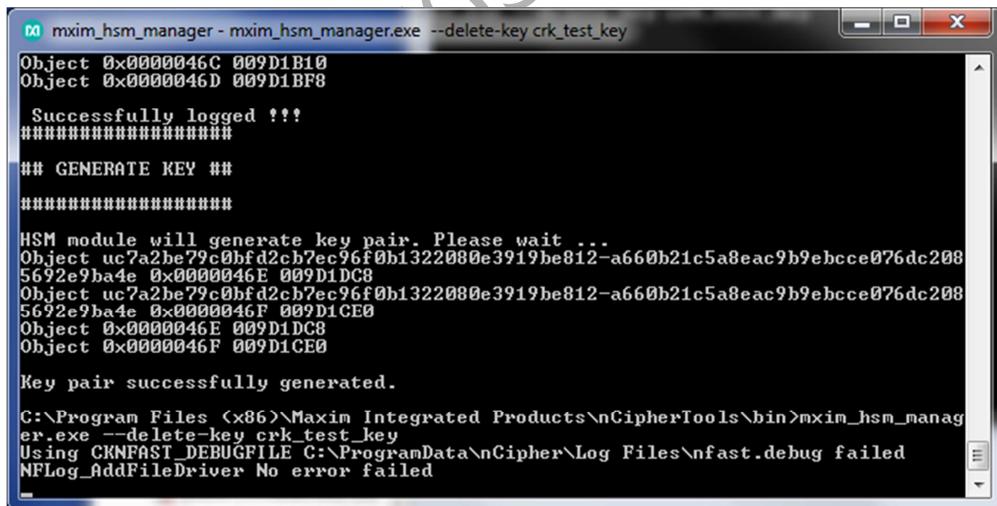
Cryptoki object <handle=0x0000046D>
- Class name: CKO_PRIVATE_KEY
- Label: crk_test_key
- Key identifier:
- Modulus bits size: 2048
- Modulus bytes size: 256

- Modulus data:
a75b4be7926012791f04d8cd77d58b8e1230e0e7206ba0856109198781004fd400f2aeee6ceca1488c4e812fb60ec1bbfa196d314213c9207
d4907487943916627hb40e9ae615b1a00529c51848a6085d5ec5e9486ab29e9a11fc2bd14e5a771296a4a9e73caf957cc78379634cb46e47002f8
88c94384286eccdf23b7d544c11991bb8de60f91a5d9fd4b236126613bf6f3bc1b03496e238a8f6faf671386885843558a404ce9257b19c7409

MKIMs::Cryptoki
Unable to get cryptoki object attribute <CKA_PRIVATE_EXPONENT> value <error CKR_ATTRIBUTE_SENSITIVE: 0x00000011>.

Cryptoki object <handle=0x00000464>
- Class name: CKO_PUBLIC_KEY
- Label: crk_test_key
- Key identifier:
- Modulus bits size: 2048
- Modulus bytes size: 256
- Modulus data:
a75b4be7926012791f04d8cd77d58b8e1230e0e7206ba0856109198781004fd400f2aeee6ceca1488c4e812fb60ec1bbfa196d3142
9e04ded04907487943916627hb40e9ae615b1a00529c51848a6085d5ec5e9486ab29e9a11fc2bd14e5a771296a4a9e73caf957cc78379634cb46
4db8dd3d88c94384286eccdf23b7d544c11991bb8de60f91a5d9fd4b236126613bf6f3bc1b03496e238a8f6faf671386885843558a404ce9257b
- Exponent data:
65537 <0x00010001>
```

Type following command: mxim_hsm_manager.exe -delete-key crk_test_key



Insert first OCS card, so called **token**, type the password and validate with **ENTER**:

```

m mxim_hsm_manager - mxim_hsm_manager.exe --delete-key crk_test_key
bf63cef79d 0x00000465 009E6080
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

@eee Enter pass phrase <press ENTER key when it's done.>:

```

Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):

```

m mxim_hsm_manager - mxim_hsm_manager.exe --delete-key crk_test_key
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

@eee Enter pass phrase <press ENTER key when it's done.>:
@eee Enter next card pass phrase <press ENTER key when it's done.>:

```

HSM will delete RSA key pair.

```

m mxim_hsm_manager

@eee Please wait ...
Object 0x00000467 010B0688
Object 0x00000468 010B0770
Object 0x00000469 010B0858
Object 0x0000046A 010B0940
Object 0x0000046B 010B0A28
Object 0x0000046C 010B0B10
Object 0x0000046D 010B0BF8
Object 0x0000046E 010B0CE0
Object 0x0000046F 010B0DC8

Successfully logged !!!
#####
## DELETE RSA KEY ##
#####
WARNING: the class attribute of the object has not been defined.
WARNING: the label attribute of the object has not been defined.
WARNING: the identifier attribute of the key has not been defined.

Key pair successfully deleted.

C:\Program Files (x86)\Maxim Integrated Products\NCipherTools\bin\_

```

3.3 MAX32550

3.3.1 Generate new ECDSA key pair

Considering a security world (see section 2.1.2.2) is present on the PC and all **OCS** cards required accordingly to the quorum are available, you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

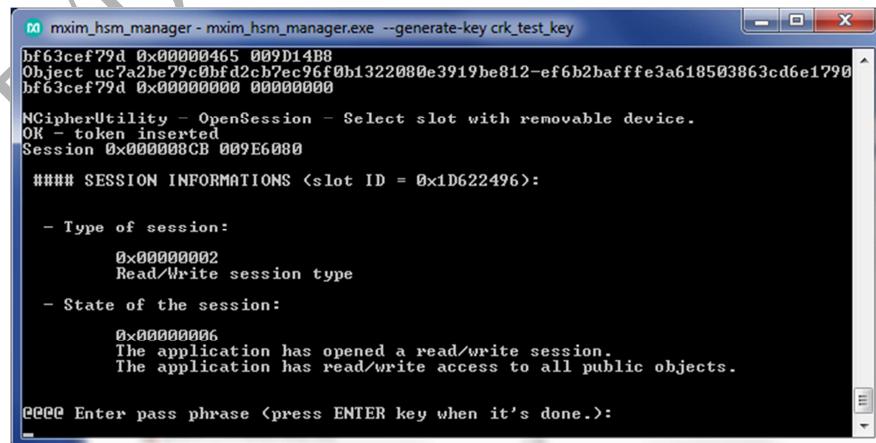
- `mxim_hsm_manager.exe --generate-key <name_of_ecdsa_key_pair> -m [CKM_EC_KEY_PAIR_GEN|CKM_ECDSA_KEY_PAIR_GEN]`

Example:

Type following command: `mxim_hsm_manager.exe --generate-key crk_test_key -m CKM_EC_KEY_PAIR_GEN`



Insert first OCS card, so called **token**, type the password and validate with ENTER:



Insert next OCS card(s), type the password and validate with ENTER (repeat this operation accordingly to the quorum):

```
mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key
bf63cef79d 0x00000000 00000000
NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000000CB 009E6000
#### SESSION INFORMATIONS <slot ID = 0x1D622496>:
- Type of session:
  0x00000002
  Read/Write session type
- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

EEEE Enter pass phrase <press ENTER key when it's done.>:
EEEE Enter next card pass phrase <press ENTER key when it's done.>:
```

HSM will generate ECDSA key pair. The operation takes some time.

```
mxim_hsm_manager - mxim_hsm_manager.exe --generate-key crk_test_key
EEEE Enter pass phrase <press ENTER key when it's done.>:
EEEE Enter next card pass phrase <press ENTER key when it's done.>:
EEEE Please wait ...
Object 0x00000466 009D15A0
Object 0x00000467 009D1688
Object 0x00000468 009D1770
Object 0x00000469 009D1858
Object 0x0000046A 009D1940
Object 0x0000046B 009D1A28
Object 0x0000046C 009D1B10
Object 0x0000046D 009D1BF8

Successfully logged !!!
#####
## GENERATE KEY ##
#####
HSM module will generate key pair. Please wait ...
```

And finally ...

```
mxim_hsm_manager
Object 0x00000468 009D1770
Object 0x00000469 009D1858
Object 0x0000046A 009D1940
Object 0x0000046B 009D1A28
Object 0x0000046C 009D1B10
Object 0x0000046D 009D1BF8

Successfully logged !!!
#####
## GENERATE KEY ##
#####
HSM module will generate key pair. Please wait ...
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046E 009D1DC8
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046F 009D1CE0
Object 0x0000046E 009D1DC8
Object 0x0000046F 009D1CE0

Key pair successfully generated.
C:\Program Files (<x86>)\\Maxim Integrated Products\\nCipherTools\\bin\\
```

3.3.2 Export public ECDSA key

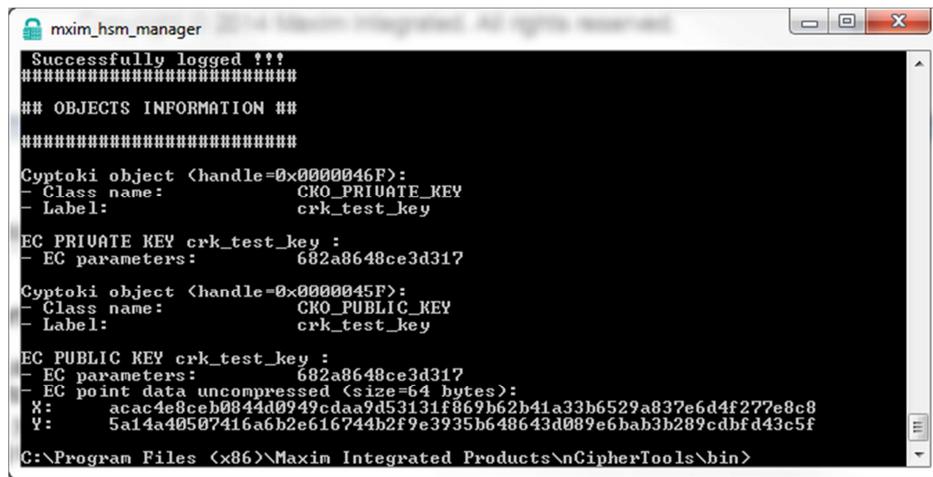
Considering a security (see section 2.1.2.2) is present on the PC, all **OCS** cards required accordingly to the quorum are available, and one ECDSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- `mxim_hsm_manager.exe --export-key <path_and_file_name> --object-label <name_of_ecdsa_key>`

Example:

To check that the ECDSA key exists:

```
mxim_hsm_manager.exe --objects-info --object-label crk_test_key
```



The screenshot shows the command-line interface of the mxim_hsm_manager.exe application. It displays the following output:

```

mxim_hsm_manager
Successfully logged !!!
#####
## OBJECTS INFORMATION ##
#####

Cryptoki object <handle=0x0000046F>:
- Class name: CKO_PRIVATE_KEY
- Label: crk_test_key

EC PRIVATE KEY crk_test_key :
- EC parameters: 682a8648ce3d317

Cryptoki object <handle=0x0000045F>:
- Class name: CKO_PUBLIC_KEY
- Label: crk_test_key

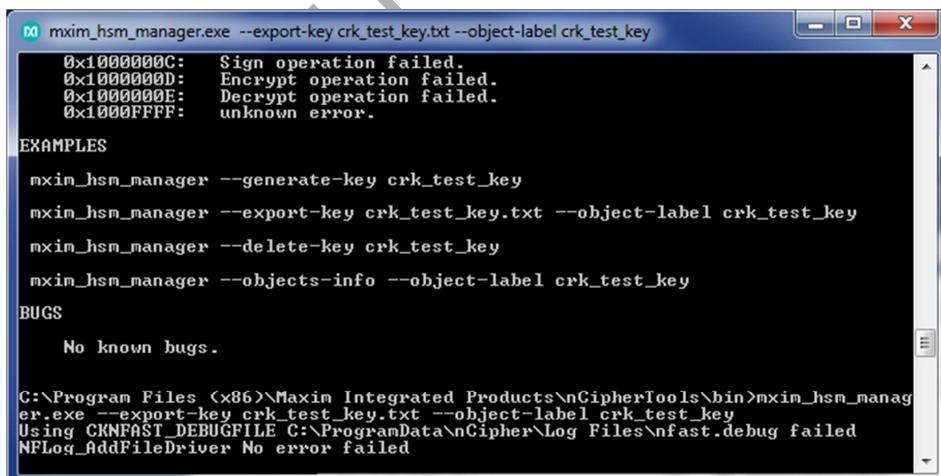
EC PUBLIC KEY crk_test_key :
- EC parameters: 682a8648ce3d317
- EC point data uncompressed (size=64 bytes):
  X: acac4e8ceb0844d0949cdaa9d5313f869b62b41a33b6529a837e6d4f277e8c8
  Y: 5a14a40507416a6b2e616744b2f9e3935b648643d089e6bab3b289cdbfd43c5f

C:\Program Files (<x86>) Maxim Integrated Products\nCipherTools\bin>

```

Then, type following command:

```
mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_test_key
```



The screenshot shows the command-line interface of the mxim_hsm_manager.exe application. It displays the following output:

```

mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_test_key
0x1000000C: Sign operation failed.
0x1000000D: Encrypt operation failed.
0x1000000E: Decrypt operation failed.
0x1000FFFF: unknown error.

EXAMPLES
mxim_hsm_manager --generate-key crk_test_key
mxim_hsm_manager --export-key crk_test_key.txt --object-label crk_test_key
mxim_hsm_manager --delete-key crk_test_key
mxim_hsm_manager --objects-info --object-label crk_test_key

BUGS
No known bugs.

C:\Program Files (<x86>) Maxim Integrated Products\nCipherTools\bin>mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_test_key
Using CKNFAST_DEBUGFILE C:\ProgramData\nCipher\Log Files\nfast.debug failed
NFLLog_AddFileDriver No error failed

```

Insert first OCS card, so called **token**, type the password and validate with **ENTER**:

```

mxim_hsm_manager - mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_t...
bf63cef79d 0x00000465 009D14B8
Object uc7a2bc79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

EEEE Enter pass phrase <press ENTER key when it's done.>:

```

Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):

```

mxim_hsm_manager - mxim_hsm_manager.exe --export-key crk_test_key.txt --object-label crk_t...
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

EEEE Enter pass phrase <press ENTER key when it's done.>:
EEEE Enter next card pass phrase <press ENTER key when it's done.>:

```

HSM will export public ECDSA key.

```

mxim_hsm_manager

Object 0x0000046B 00AA6D30
Object 0x0000046C 00AC64D8
Object 0x0000046D 00AC6F40
Object 0x0000046E 00AC7028
Object 0x0000046F 00AC7110
Object 0x00000470 00AC71F8
Object 0x00000471 00AC72E0
Object 0x00000472 00AC73C8
Object 0x00000473 00AC74B0
Object 0x00000474 00AC7598
Object 0x00000475 00AC7680
Object 0x00000476 00AC7768
Object 0x00000477 00AC7850
Object 0x00000478 00AC7938

Successfully logged !!!
#####
## EXPORT KEY ##
#####

Public key successfully exported.

C:\Program Files (<x86>\Maxim Integrated Products\nCipherTools\bin>

```

The exported file result:

```

C:\Program Files (x86)\Maxim Integrated Products\nCipherTools\bin\crk_test_key.txt - Notepad++
Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Com
crk_test_key.txt
1 acac4e8ceb0844d0949cd...
2 5a14a40507416a6b2e616744b2f9e3935b648643d089e6bab3b289cdbfd43c5f
3

```

First line is corresponding to the modulus of the public ECDSA key.

Second line is corresponding to the exponent of the public ECDSA key.

3.3.3 Remove ECDSA key pair

Considering a security (see section 2.1.2.2) is present on the local machine, all **OCS** cards required accordingly to the quorum are available, and one ECDSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- mxim_hsm_manager.exe --delete-key <name_of_ecdsa_key>

Example:

To check that the ECDSA key exists:

```
mxim_hsm_manager.exe --objects-info --object-label crk_test_key
```

```

mxim_hsm_manager
Successfully logged !!!
#####
## OBJECTS INFORMATION ##
#####

Crypto object <handle=0x0000046F>:
- Class name: CKO_PRIVATE_KEY
- Label: crk_test_key

EC PRIVATE KEY crk_test_key :
- EC parameters: 682a8648ce3d317

Crypto object <handle=0x0000045F>:
- Class name: CKO_PUBLIC_KEY
- Label: crk_test_key

EC PUBLIC KEY crk_test_key :
- EC parameters: 682a8648ce3d317
- EC point data uncompressed <size=64 bytes>:
X: acac4e8ceb0844d0949cd...
Y: 5a14a40507416a6b2e616744b2f9e3935b648643d089e6bab3b289cdbfd43c5f

C:\Program Files (x86)\Maxim Integrated Products\nCipherTools\bin>

```

Type following command: `mxim_hsm_manager.exe -delete-key crk_test_key`

```
mxim_hsm_manager - mxim_hsm_manager.exe --delete-key crk_test_key
Object 0x0000046C 009D1B10
Object 0x0000046D 009D1BF8
Successfully logged !!!
#####
## GENERATE KEY ##
#####
HSM module will generate key pair. Please wait...
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046E 009D1DC8
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-a660b21c5a8eac9b9ebcce076dc208
5692e9ba4e 0x0000046F 009D1CE0
Object 0x0000046E 009D1DC8
Object 0x0000046F 009D1CE0
Key pair successfully generated.

C:\Program Files (<x86>)\Maxim Integrated Products\ncipherTools\bin\mxim_hsm_manager.exe --delete-key crk_test_key
Using CKNFAST_DEBUGFILE C:\ProgramData\ncipher\Log Files\nfast.debug failed
NLog_AddFileDriver No error failed
```

Insert first OCS card, so called **token**, type the password and validate with **ENTER**:

```
mxim_hsm_manager - mxim_hsm_manager.exe --delete-key crk_test_key
bf63cef79d 0x00000465 009D14B8
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6080

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done.>:
```

Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):

```
mxim_hsm_manager - mxim_hsm_manager.exe --delete-key crk_test_key
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 009E6000

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done.>:
#### Enter next card pass phrase <press ENTER key when it's done.>:
```

HSM will delete ECDSA key pair.

The screenshot shows a Windows application window titled "mxim_hsm_manager". The main text area displays a list of objects and their identifiers, followed by a series of log messages indicating the deletion process. The log messages are:

```
Object 0x0000046B 00D70708
Object 0x0000046C 00D70D18
Object 0x0000046D 00D76FE8
Object 0x0000046E 00D770D0
Object 0x0000046F 00D771B0
Object 0x00000470 00D772A0
Object 0x00000471 00D772B8
Object 0x00000472 00D77470
Object 0x00000473 00D77558
Object 0x00000474 00D77640
Object 0x00000475 00D77728
Object 0x00000476 00D77810
Object 0x00000477 00D778F8
Object 0x00000478 00D779E0

Successfully logged !!!
#####
## DELETE KEY ##
#####

Key pair successfully deleted.

C:\Program Files <x86>\Maxim Integrated Products\nCipherTools\bin>
```

4 SCP packets generation

Maxim Integrated has sent signed CRK public key to the customer who can securely upload its SCP packets on his Maxim integrated chip.

4.1 nCipher_session_build application

It is strongly recommended to read the UG21T24 (see [3]) for MAX32590 and UG25H04 (see [4]) for MAX32550 and notably the part dedicated to session-build tool and details related to its configuration file. ncipher_session_build application use the same configuration file except new parameters related to the use of HSM (see section 4.1.1).

4.1.1 Configuration file

The application program and the configuration file shall be located in the same directory. Accordingly to the session_build tool documentation set the parameters in the “session-build.ini” file. Then configure the parameters related to HSM:

```
#-----  
#-- HSM ECDSA KEY  
#-----  
name_of_ecdsa_key=CRK_PROD  
quorum_k=2  
quorum_n=3  
#-----  
#-- HSM RSA KEY  
#-----  
name_of_rsa_key=CRK_PROD  
quorum_k=2  
quorum_n=3  
#-----
```

- “name_of_rsa_key”: MAX32590 RSA key label used to sign SCP packets. The key has been generated by **Thales nShield Edge** HSM (see section 3.2).
- “name_of_ecdsa_key”: MAX32550 RSA key label used to sign SCP packets. The key has been generated by **Thales nShield Edge** HSM (see section 3.2).
- “quorum_k” and “quorum_n”: OCS cards quorum values (see section 2.1.2.2).

This operation has to be done once.

4.1.2 SCP packets building

Considering a security (see section 2.1.2.2) is present on the PC, all **OCS** cards required accordingly to the quorum are available, and one RSA or ECDSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

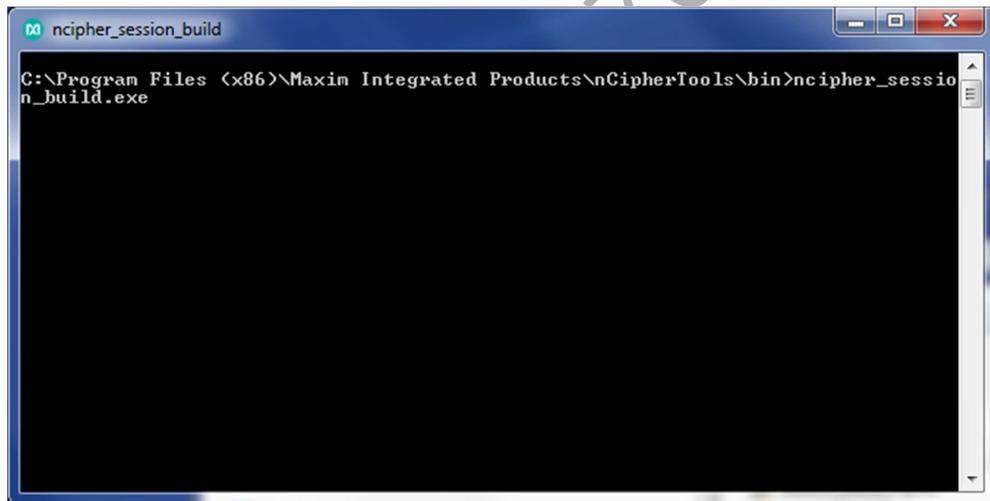
- cd <ncipher_tools_installation_folder>\bin
- ncipher_session_build.exe

Example:

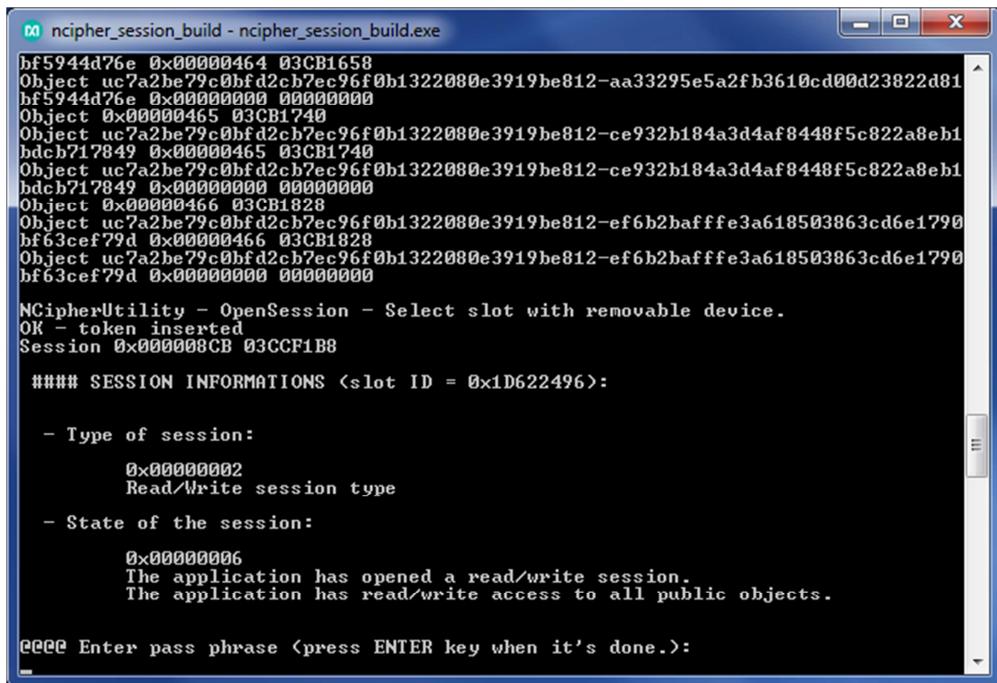
The configuration has default parameters used to perform a test:

- output_file: define where are saved the SCP packets files:
<ncipher_tools_installation_folder>\bin\session_build_outputs.
- script_file: corresponding to the script file used to generate SCP packets:
<ncipher_tools_installation_folder>\bin\session_build_inputs\script.txt

Type following command: ncipher_session_build



Insert first OCS card, so called **token**, type the password and validate with **ENTER**:



```

ncipher_session_build - ncipher_session_build.exe
bf5944d76e 0x00000464 03CB1658
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-aa33295e5a2fb3610cd00d23822d81
bf5944d76e 0x00000000 00000000
Object 0x00000465 03CB1740
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ce932b184a3d4af8448f5c822a8eb1
hdcb717849 0x00000465 03CB1740
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ce932b184a3d4af8448f5c822a8eb1
hdcb717849 0x00000000 00000000
Object 0x00000466 03CB1828
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000466 03CB1828
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 03CCF1B8

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

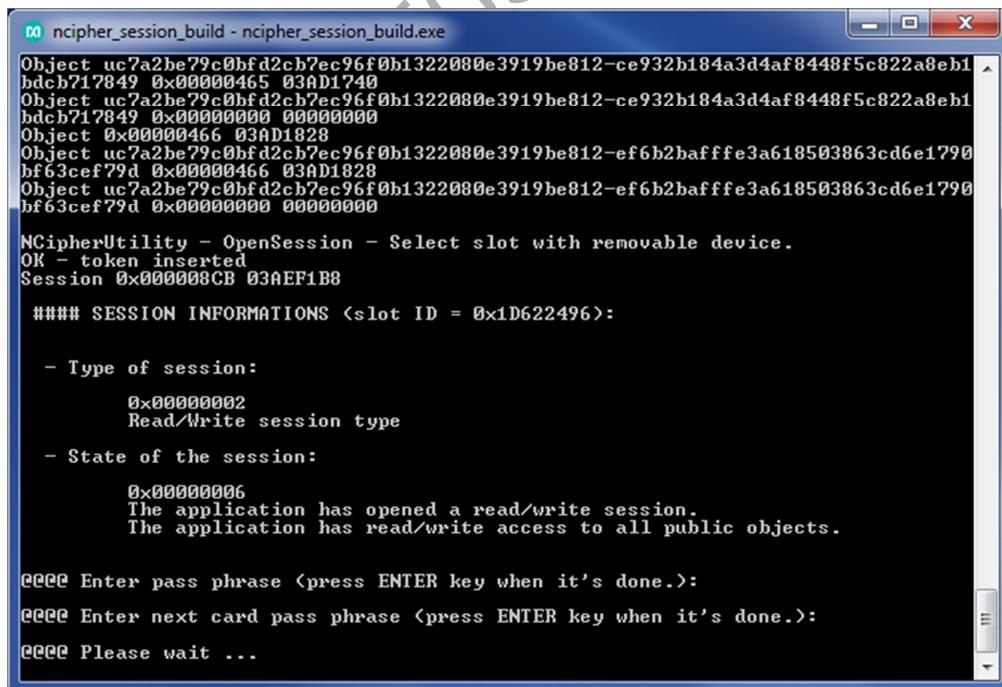
- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done.>:

```

Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):



```

ncipher_session_build - ncipher_session_build.exe
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ce932b184a3d4af8448f5c822a8eb1
hdcb717849 0x00000465 03AD1740
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ce932b184a3d4af8448f5c822a8eb1
hdcb717849 0x00000000 00000000
Object 0x00000466 03AD1828
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000466 03AD1828
Object uc7a2be79c0bfd2cb7ec96f0b1322080e3919be812-ef6b2bafffe3a618503863cd6e1790
bf63cef79d 0x00000000 00000000

NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 03AEF1B8

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done.>:
#### Enter next card pass phrase <press ENTER key when it's done.>:
#### Please wait ...

```

Application successfully ended:

```
ncipher_session_build
61c2c09a0c2cae268f3223f80aca39408b13c6d6851c834885b3a9199eae8c4b12039a3ddac09b4d
b04dbb9a0842a55ecd6fd4daf87fd88a092621dc350e1833706c1a9c89bec460150fc38cf7d02d
0c70ca1668cccd57120f5f82de062f91a973ed563d7e61cc2b35e573fc8de2517a99b069c8e31d1d0
f87ed754598dd9b0a978d8f03f4587a30889c445ea9ceabe1311b9b7283847chc000b9028204a59c
45b04bc4f7a96590e55b00f4d02ba8a238263bc07998c0ac974849dfadce389898d2acdc8112b687
5f1d62633de7ee9b0bdfe0a1635c8dec1cc1c63157a5dc878b3d23de6387295e
<.\session_build_outputs\..0000008.host.write_crk.packet> created
<usip>.0000009.ACK
beefed0600009240
<.\session_build_outputs\..0000009.bl.ack.packet> created
<usip>.0000010.DATA_TRANSFER-write_crk_response
beefed05000893bd59000004000000000e7384c6
<.\session_build_outputs\..0000010.bl.write_crk_response.packet> created
<host>.0000011.ACK
beefed0600009302
<.\session_build_outputs\..0000011.host.ack.packet> created
<host>.0000012.DISC_REQ
beefed0300009496
<.\session_build_outputs\..0000012.host.disconnection_request.packet> created
<usip>.0000013.DISC REP
beefed0400009488
<.\session_build_outputs\..0000013.bl.disconnection_reply.packet> created
<.\session_build_outputs\..log> created
C:\Program Files (x86)\Maxim Integrated Products\nCipherTools\bin>
```

Errors:

If the RSA or ECDSA key name is wrong or it has not been generated (see 3.2) in the configuration file:

```
ncipher_session_build
rsa-data:470a02046b260e89f5494e0e0f5e01eeac7ac2801d7e58745b2157bc75222c41e1c247b
2308e2cb1b85c1d5c800055442bccfaa72753d7d3fffa19803c53448ce55da36ef16f4ce567dc19fc
4e1245a039378b6d1a60e43bee2b2b3ee18993cd4a4b5a22c95dc785099839fe84bdd47d7f448c45
b2c96a765867efdb9dec682eb168e122ffc3ce63c14997b49ba86878a32f8262df07df49d0b81a5a
1b3h39c76d02831228dd7153307d6fb264c55df3f0c595649d0b1a2b809657ee69d8b4abea190a55
0b9c54cbd55f90d16a5d8ec6fbef637302a9f50dcadfeafaa82d529ee59ea22e7b03889b4fc399e0
ddd7e4e5766ed4b6b2e02ab3f81cac4e096170a488cee4c7f000100019f7f97a8f817d2d9b54521b
124353163a61224aea6fc3d5bafe25e99c09593f6260488f2489712dac8631b8b23d0b3c8eca4a0f
761742bd8ea4d7d7526d36fb77hd12f3509d7729cce36dedfhe3498004adaa3921hcdba9179a2671
b3f7083b7f6b4d94cb27abb5886c7b91ba541735d8d23221073536e0648af73c7e1964a5ae68039c
fe0e2a546619179d56e7368c02913a2237e94c4030f3e519638d9c5ad94609011ef5a7ee255c0c8f
d9c0f41d69b21f2aa32450d9b0dbe881234245481df3a734240f10d158c272820e1fd60735hb2ee7
b93d89b5af2b667cf1422325be2af06e91e97ca14e51345d12c9145972bdde23a18a87cb6638f957
f01502723

2013-05-23T14:16:57.212712 - ERROR:
MXIMNCipher library -
MXIMHSM$ign - Cannot find key "crk_key" to sign.

ERROR on rsa_pkcs1 sha256 sign <268435463> 520 0
ERROR: write-crk
ERROR: aborting
C:\Program Files (x86)\Maxim Integrated Products\nCipherTools\bin>
```

5 Customer application signature

Customer application that is launched by the MAX32590 or the MAX32550 Secure ROM code has to be signed by the CRK as the ROM code performs digital signature verification with this key before agreeing to run the application.

5.1 nCipher_ca_sign_build application

It is strongly recommended to read the UG21T24 for MAX32590(see [3]) and UG25H04 (see [4]) for MAX32550 and notably the part dedicated to ca-sign tool and details related to its configuration file. ncipher_ca_sign_build application use the same configuration file except new parameters related to the use of HSM (see section 4.1.1).

5.1.1 Configuration file

The application program and the configuration file shall be located in the same directory. Accordingly to the ca_sign_build tool documentation set the parameters in the “ca_sign_build.ini” file. Then configure the parameters related to HSM:

```
#-----
#-- HSM ECDSA KEY
#-----
name_of_ecdsa_key=CRK_PROD
quorum_k=2
quorum_n=3
#-----
#-- HSM RSA KEY
#-----
name_of_rsa_key=CRK_PROD
quorum_k=2
quorum_n=3
#-----
```

- “name_of_rsa_key”: for MAX32590, RSA key label used to sign binary file. The key has been generated by **Thales nShield Edge** HSM (see section 3.2).
- “name_of_ecdsa_key”: for MAX32550, key label used to sign binary file. The key has been generated by **Thales nShield Edge** HSM (see section 3.2).
- “quorum_k” and “quorum_n”: OCS cards quorum values (see section 2.1.2.2).

This operation has to be done once.

5.1.2 Application signature

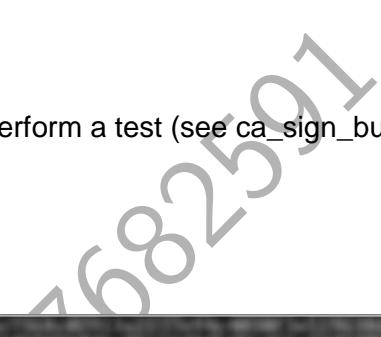
Considering a security (see section 2.1.2.2) is present on the PC, all **OCS** cards required accordingly to the quorum are available, and one RSA or ECDSA key has already been created (see section 3.2), you have to plug the **Thales nShield Edge** in. Then, in Windows console, type the following command and follow the instructions:

- cd <ncipher_tools_installation_folder>\bin
- ncipher_ca_sign_build.exe

Example:

The configuration has default parameters used to perform a test (see ca_sign_build.ini file).

Type following command: ca_sign_build



```
Windows PowerShell

PS C:\Program Files (<x86>)\\Maxim Integrated Products\\nCipherTools\\bin> .\ncipher_ca_sign_build.exe
CA signature build v1.0.0 (build 0) (c)Maxim IC 2006-2013
--warning: this tool does handle keys with Thales nCipher Edge HSM --
<display config>
verbose
version: abcdef01
rsa file: casignk.key
customer application <input> file: sla_template.bin
signed customer application <output> file: sla_template.sbin
binary load address: 90000000
binary jump address: 90000160
application arguments: <"string of chars">
Dynamic Memory Slot: SR_PAPD: ac
    SR_PRFSH: abcdef01
    SR_PCFG: 0fefcdab
    SR_PEXT: ac
    MEM_GCFG: 02030405
    DMC_CLK: 04
UCI2 parameters
    KSRC-Config ENC-INT: bc
    uci0-AC1R-start-offset: ef01abcd
    uci0-AC1R-end-offset: 01abcdef
    uci0-DDR-r0: abcdef01

HSM try open connection
Using CKNFAST_DEBUGFILE C:\\ProgramData\\nCipher\\Log Files\\nfast.debug failed
NFIoLog_AddFileDriver No error failed
Object 0x0000045E 0BDD2DD0
Object uc?a2be79c0bfd2cb7ec96f0b1322080e3919be812-15f495cca58c1a7cc417c11fabc367ce244231a4 0x0000045E 0BDD2DD0
Object uc?a2be79c0bfd2cb7ec96f0b1322080e3919be812-15f495cca58c1a7cc417c11fabc367ce244231a4 0x00000000 00000000
```

Insert first OCS card, so called **token**, type the password and validate with **ENTER**:

```
NCipherUtility - OpenSession - Select slot with removable device.
OK - token inserted
Session 0x000008CB 0BDF20F0

##### SESSION INFORMATIONS <slot ID = 0x1D622496>:

- Type of session:
  0x00000002
  Read/Write session type

- State of the session:
  0x00000006
  The application has opened a read/write session.
  The application has read/write access to all public objects.

#### Enter pass phrase <press ENTER key when it's done.>:
```

Insert next OCS card(s), type the password and validate with **ENTER** (repeat this operation accordingly to the quorum):

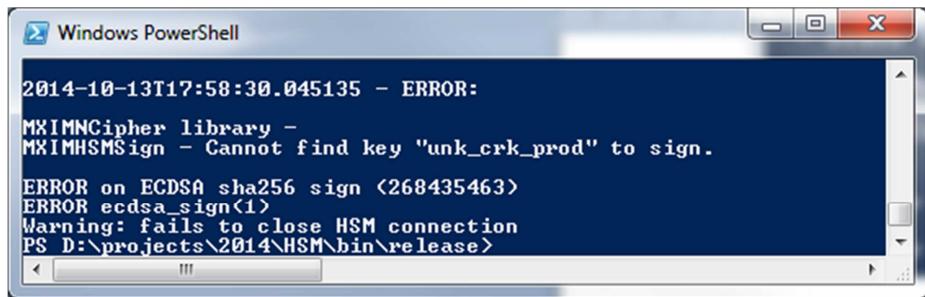
```
##### SESSION INFORMATIONS <slot ID = 0x1D622496>:  
  
- Type of session:  
    0x00000002  
    Read/Write session type  
  
- State of the session:  
    0x00000006  
    The application has opened a read/write session.  
    The application has read/write access to all public objects.  
  
AAAA Enter pass phrase <press ENTER key when it's done.>:  
BBBB Enter next card pass phrase <press ENTER key when it's done.>
```

Application successfully ended:



Errors:

If the RSA or ECDSA key name is wrong or it has not been generated (see [3.2](#)) in the configuration file:



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window displays the following error message:

```
2014-10-13T17:58:30.045135 - ERROR:  
MXIMNCipher library -  
MXIMHSMSSign - Cannot find key "unk_crk_prod" to sign.  
ERROR on ECDSA sha256 sign <268435463>  
ERROR ecdsa_sign<1>  
Warning: fails to close HSM connection  
PS D:\projects\2014\HSM\bin\release>
```

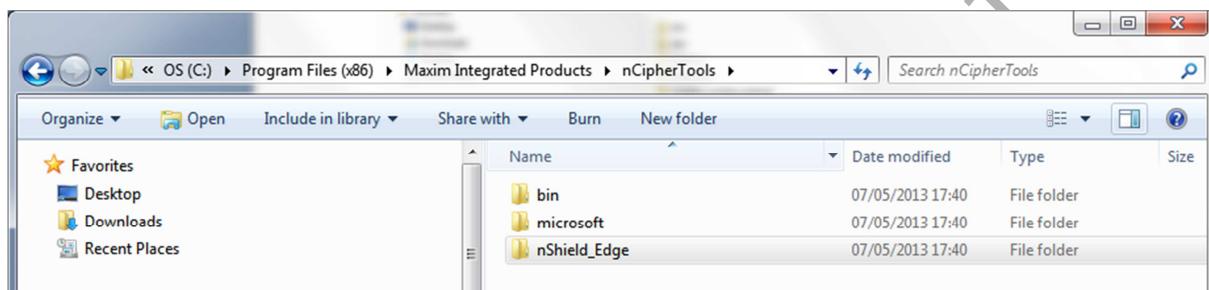
E/UG25T03/77682591

6 ANNEX: THALES nShield Edge USB FTI drivers

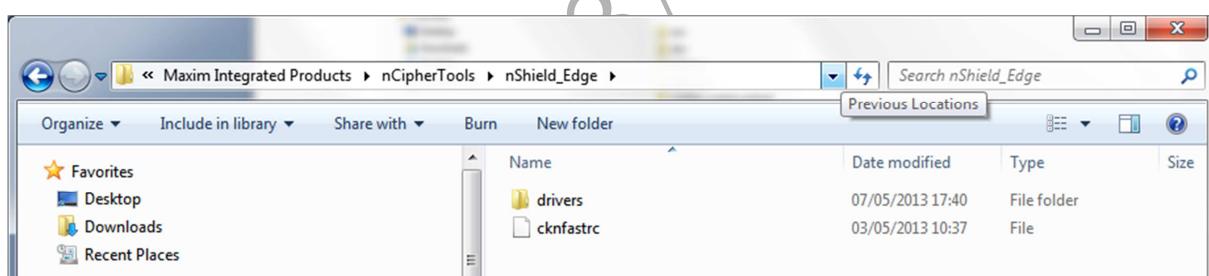
During THALES nShield Edge installation (problem already seen on Windows 7, 64 bits architecture), the HSM may not be recognized by the system due to USB pilots incorrect installation.

Follow this procedure:

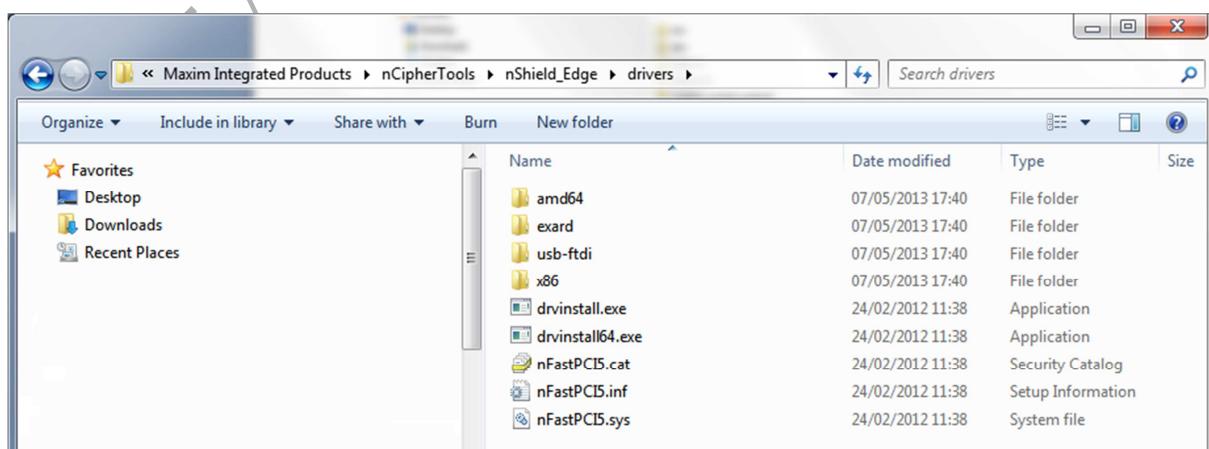
- Go into “MAXIMINTEGRATED nCipherTools” installation directory:



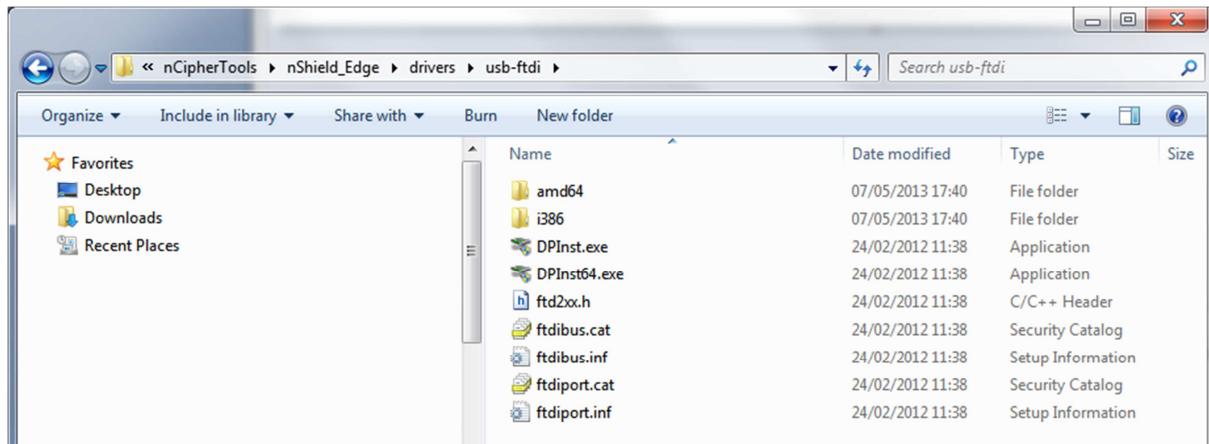
- Then into “nShield_Edge” installation directory:



- In “drivers” directory:



- Finally, in “usb-ftdi”:



- Click on “DPinstall.exe” in case of 32 bits architecture or on “DPinstall64.exe” in case of 64 bits architecture and follow instructions.

Maxim Integrated Reference

MAXIM Integrated



<http://www.maximintegrated.com>

for support, go to www.maximintegrated.com/support

E/UG25T03/77682591