ModusToolbox™

"Secure Policy" Configurator guide

About this document

Version

1.20

Scope and purpose

The "Secure Policy" Configurator is used to open, create or change policy configuration files for the PSoC™ 64 "Secure Boot" MCU devices, also modify policy tools and provision devices.

Intended audience

This document helps application developers understand how to use the "Secure Policy" Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- Application – One or more projects related to each other.
- Configurator – A GUI-based tool used to configure a resource.
- `cysecuretools` – A set of Python scripts and policy templates to perform provisioning for the PSoC™ 64 devices.
- DAPLink – Arm Mbed DAPLink is an open-source software project that enables programming and debugging application software running on Arm Cortex CPUs.
- Image, Single/Multi – Single-image and Multi-image types can be created to define how the CM0+ "secure" co-processor and CM4 binaries are generated.
  - Single-image mode – The default mode, in which the CM0+ "secure" co-processor code binary is attached to the beginning of the CM4 binary to form a single binary. Therefore, the CM0+ "secure" co-processor and CM4 must be updated at the same time, as they require a single update binary.
  - Multi-image mode – The CM0+ "secure" co-processor and CM4 binaries are updated individually with two different update binaries.
- JSON file – JavaScript Object Notation is a standard data-interchange file format that uses human-readable text to transmit data objects and store them.

user guide
www.infineon.com

Please read the Important Notice and Warnings at the end of this document
page 1 of 25

002-31917 Rev. *C
20-07-2021

About this document

- JWT – JSON Web Token is an open, industry standard RFC 7519 method to securely represent claims between two parties.

- Monotonic counter ID – The counter to prevent a rollback during the upgrade process. Indicates the monotonic counter value associated with the image, which is booted. During a "secure" boot, this counter value is compared to this image version code. During the upgrade process, this counter is incremented to the value from the image header of the upgrade image.

- Policies – A collection of pre-defined (name, value) pairs that describe what is allowed and not allowed on the device.

- Provisioning – The act of configuring a device with an authorized set of keys, certificates, credentials, and firmware images. Once provisioned, the device can be accessed or modified only with the keys injected adhering to the relevant policies. The act is executed in two steps:

  - 1. Provisioning identity; the unique device secret (UDS) and the device public/private key pair.

Note: This occurs only once in a secure manufacturing environment.

  - 2. Provisioning keys and policies.

- Reprovisioning – The act of reconfiguring a device with a new certificate.

- Public key certificate – The X.509 type certificate is a standard public key certificate used in many Internet protocols, including TLS/SSL.

- Certificate – A message, digitally signed by a "Certificate Authority (CA)", linking a public key to the identity of the certificate holder (can be a URL, name/address, ID/serial number).

- RMA (Return Merchandise Authorization) mode – The device transitions to this mode when the user wants Cypress to perform failure analysis on the device, which means that any areas of the user's flash that may contain proprietary code or sensitive data will be erased automatically.

- Rollback Counter – A special counter accessed by "Secure Boot" code. This counter prevents downgrading the device to an older version of its software that has been deprecated due to security.

- "Secure Boot" – A bootup process where the firmware being run by the chip is trusted by using strong cryptographic schemes and an immutable RoT.

- SMIF – Serial Memory interface, hereinafter refers to the highspeed Quad-SPI interface on PSoC™ 6.

- SWD – Single Wire Debug, a two-wire debug port defined for Arm Cortex CPU's.

Reference documents

Refer to the following documents for more information as needed:

- Eclipse IDE for ModusToolbox™ user guide
- API reference guides
- Device datasheets
- Device technical reference manuals
- "Secure Boot" SDK user guide
- KitProg3 user guide

Table of contents

1   Overview

The "Secure Policy" Configurator is part of a collection of tools included with the ModusToolbox™ software. You can use the "Secure Policy" Configurator to open, create or change policy configuration files for the PSoC™ 64 "Secure Boot" MCU devices, also modify policy tools and provision devices. For details, refer to the "Secure Boot" SDK user guide. The "Secure Policy" Configurator provides easy execution of configuration without using the command line and editing a policy file without editing the XML policy file.
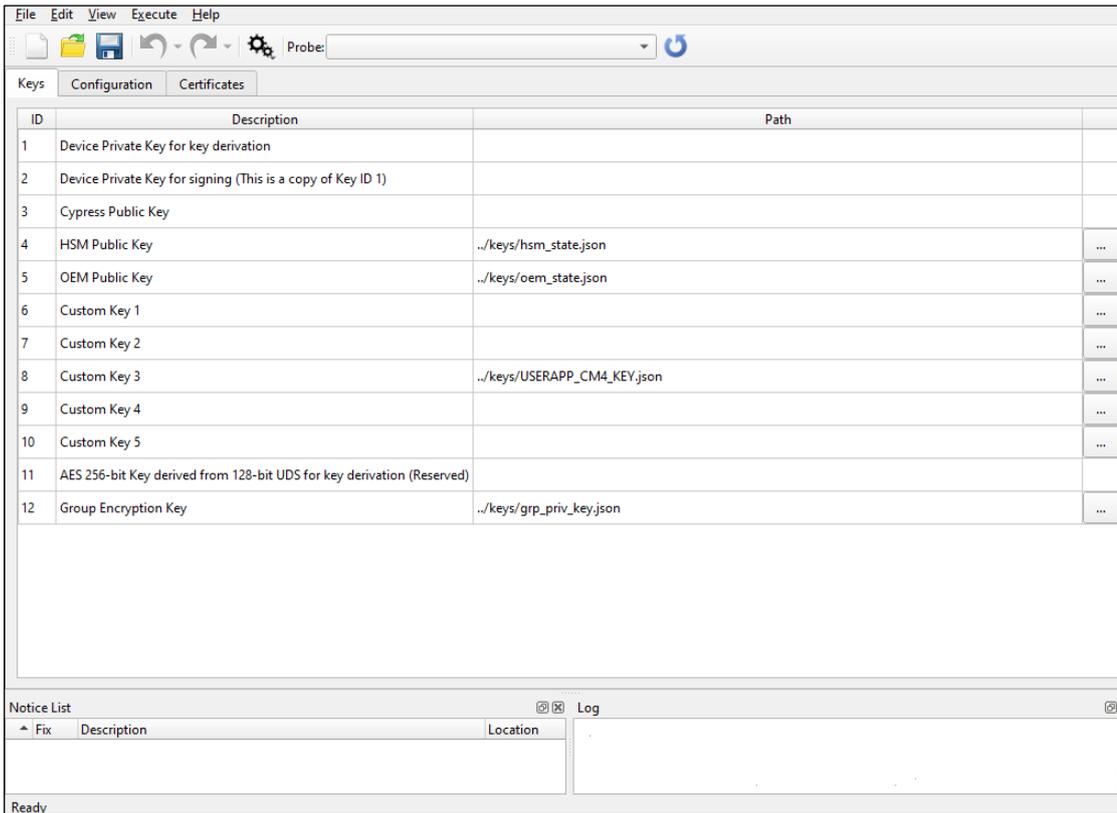
2 Installation

The "Secure Policy" Configurator requires the `cysecuretools` program, which is installed on Windows systems as a part of Python bundled with ModusToolbox™ 2.2 and later. For other operating systems, you can download it from [PyPI](#) or here:

> https://github.com/Infineon/cysecuretools

2.1 Configuring `cysecuretools` location

For **Windows**, `cysecuretools` is installed in the appropriate ModusToolbox™ *tools_<version>* directory as a part of Python bundled with the software. The "Secure Policy" Configurator uses that installation by default.

For **macOS** and **Linux**, use one of the following ways:

- Add the `cysecuretools` installation directory to the `PATH` environment variable.
- Select **File > Settings** and configure a custom path directory to the `cysecuretools` location.

For detailed `cysecuretools` installation instructions, refer to the [_"Secure Boot" SDK user guide_](#).

3   Quick start

The **Quick Start** section provides a simple workflow for how to use the "Secure Policy" Configurator.

3.1   Create an application

1. Follow instructions in the [ModusToolbox™ user guide](#) "Getting Started" chapter to create a new application for the PSoC™ 64 "Secure Boot" kit.

2. After the application has been created, go to the application directory and create a subdirectory named *"secure*."

3.2   Launch the Configurator

Launch the "Secure Policy" Configurator following the instructions described under [Launch the "Secure Policy" Configurator](#).

3.3   Create a policy file

To create a new policy file:

1. Select **File > New** to open the New Configuration dialog.

Note: *Using the **New** command creates multiple policy files (single-image, multi-image, swap), any of which can be selected for use with the project.*

2. Next to the **Project directory**, click the **Browse** [**. . .**] button and navigate to and select the *"secure"* subdirectory.

   The *"secure"* subdirectory contains the following by default:
   - /path/keys – Keys used for code signing and provisioning.
   - /path/packets – Provisioning packets.
   - /path/policy – Copied project policy options
   - /path/prebuilt – A stored prebuilt bootloader.

3. In the **Target** pull-down menu, select the appropriate device family or kit, and click **OK**.

4. Select **File > Save** in the main GUI to generate the user keys.

3.4   Provision a device

These are the minimum steps if you want to use the default policy settings:

1. Create or open an existing policy file.

2. Plug in the kit using the kit's USB cable and connect the host computer to the kit. KitProg3 is powered via the USB cable.

3. The kit must be in DAPLink mode. Press the **Mode** button on the kit until the Status LED blinks fast. For details, refer to the [KitProg3 User Guide](#).

4. Click **Refresh Probe List** on the toolbar.

5. Select the device from the **Probe** list in the toolbar.

6. Select **Execute > Run Entrance Exam**.

7. Select **Execute > Provision Device**.

    If the device is already provisioned, select **Execute > Reprovision Device** instead.

Note: *The device must be initially configured to be reprovisioned (check the Reprovisioning Options in the Advanced tab) in order to be reprovisioned later.*

4 Launch the "Secure Policy" Configurator

The best way to launch the "Secure Policy" Configurator is within the context of an application using either the `make` command or the Eclipse IDE. You can also open the configurator as a stand-alone tool. If you launch the configurator as part of an application, it defaults to the application directory. If you launch the configurator as a stand-alone tool, it defaults to the installation directory.

When launched, the "Secure Policy" Configurator attempts to open the default policy file from the */<path>/policy* subdirectory. If the *policy* subdirectory and/or policy files do not exist, the configurator opens an empty window with a message to create a new or open an existing configuration (policy) file.
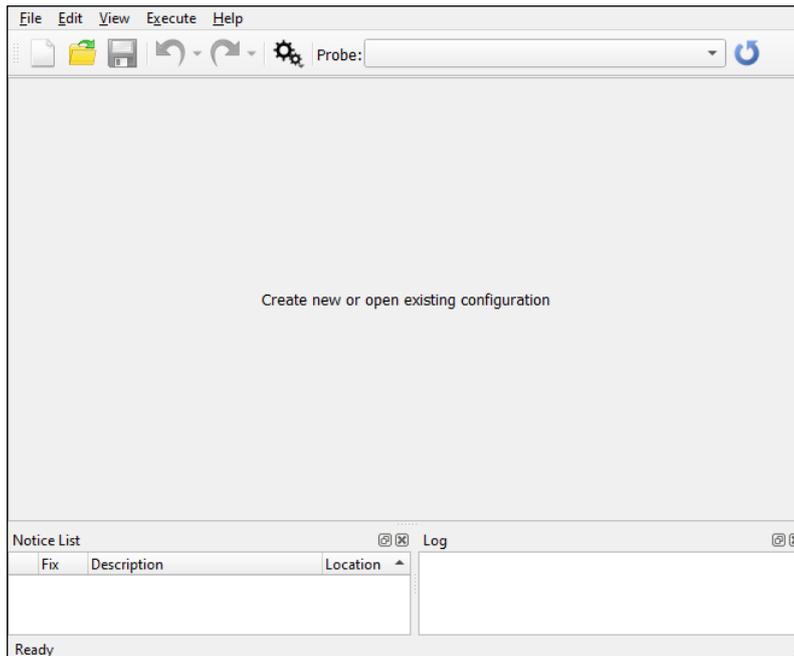
- Create a new policy file as described under [Create a policy file](#).
- Use the **File > Open** menu to locate and open an existing policy file.

4.1 make command

As described in the [ModusToolbox™ user guide "](#)Build System" chapter, you can run numerous `make` commands in the application directory, such as launching the "Secure Policy" Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make open CY_OPEN_TYPE=secure-policy-configurator
```

This command opens the "Secure Policy" Configurator GUI for the specific application in which you are working.

4.2 Eclipse IDE

To launch the "Secure Policy" Configurator from the Eclipse IDE, right-click on the project and select
ModusToolbox™ > Secure Policy Configurator. You can also open the "Secure Policy" Configurator by clicking
the link in the Eclipse IDE for ModusToolbox™ Quick Panel.
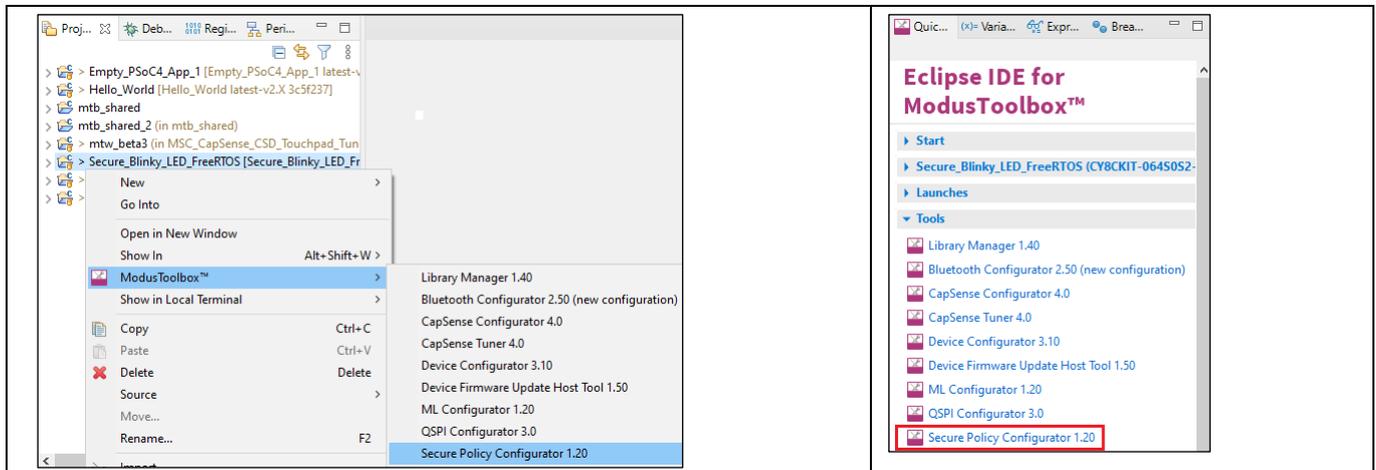
4.3 Executable (GUI)

You can launch the "Secure Policy" Configurator as a stand-alone tool by running its executable as appropriate
for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, it
is installed here:

<install_dir>/ModusToolbox/tools_<version>/secure-policy-configurator<version>

When launched this way, the "Secure Policy" Configurator opens without a policy configuration file, and the
default directory is where the configurator is installed. You can create a policy configuration file and save it in
the application directory in a subdirectory named "secure".

5 GUI description

The "Secure Policy" Configurator GUI contains menus, tabs, and dialogs to configure "secure policy" configuration files.

5.1 Menus

5.1.1 File

- **New** – Opens the New Configuration dialog, where you can create a new "secure policy" directory.
  - **Project directory** – The path to the folder where a project will be created.
  - **Target** – A drop-down menu to choose a device or a device family to create a project.
- **Open …** – Opens an existing "secure policy" configuration file (.json).
- **Save** – Saves changes to the "secure policy" configuration file. This command generates a user key if it does not exist already.
- **Open in System Explorer** – Opens your computer's file explorer tool to the folder that contains the *.modus* file.
- **Settings …** – Opens the Settings dialog, where you can choose either the default path to the `cysecuretools` software or customize the path.
- **Device and Policy** – Opens the Device and Policy dialog, where you can change the target device.
- **Recent Files** – Shows recent files that you can open directly.
- **Exit** – Closes the configurator.

5.1.2 Edit

- **Undo** – Undoes the previous change.
- **Redo** – Redoes the last undone change.

5.1.3 View

- **Advanced** – Opens a dialog to edit additional "secure policy" configuration settings.
- **Notice List**, **Log**, **Toolbar** – Hide or show these options respectively.
- **Reset View** – Restores the GUI to the default view.

5.1.4 Execute

- **Get Device Info** – Provides a list of available device information in the Log window. This information includes but is not limited to the following:
  - Target
  - Flashboot revision
  - Silicon ID, Family, and revision
  - Probe ID
- **Run Entrance Exam** – Must confirm that the device is genuine and blank.
- **Run Entrance Exam and Erase User Flash** – Confirms that the device is genuine and blank and erases the user's flash. The user is responsible for erasing any secret information.

GUI description

- **Provision Device** – Configures the device with an authorized set of keys, certificates, credentials, and firmware images.
- **Reprovision Device** – Provides the device with a new certificate.
- **Read Public Keys** – Reads the selected key into the Log window. This drop-down menu displays the public keys, which can be read from the device.

5.1.5 Help

- **View Help** – Opens this User Guide.
- **About Secure Policy Configurator** – Shows version information.

5.2 Toolbar

The toolbar contains several commands also available from the menus.

- **New** – Creates a new "secure policy" directory.
- **Open** – Opens an existing policy file (.json).
- **Save** – Saves changes to the policy file. Generates a user key if it does not exist already.
- **Undo** – Undoes the previous change.
- **Redo** – Redoes the last undone change.
- **Execute** – Equivalent to the Execute menu.
    - **Probe** – This is a drop-down menu of all the available probes currently connected to the PC. The user may select any of the listed probes for the actions in the Execute menu. A note displays to remind the user to switch the kit or programmer to DAPLink mode.
    - **Refresh Probe List** – Searches for a kit or programmer currently attached to the PC and repopulates the Probe list.

5.3 Tabs

The "Secure Policy" Configurator contains several tabs in which to update various settings.

- Keys Tab
- Configuration Tabs

    The "Secure Policy" Configurator operates in Single-image and Multi (two)-image modes, so different tabs display depending on the selected policy file mode.

- Certificates Tab

5.4 Notice List

The **Notice List** combines notices (errors, warnings, tasks, and any other information) from many places in your design into a centralized list. If a notice shows a location, you can double-click the entry to navigate to the error or warning. For details, refer to the description in the Device Configurator guide.

5.5 Log

The log pane shows any errors or status. A log file collects the output of any executed script to the log pane. The placement of this log file is consistent with other tools in the build system.

5.6 Keys tab

The **Key** tab provides a method to associate a Key ID with the appropriate key file.

5.6.1 ID

- **1, 2, 3, 11** – These keys cannot be modified by the configurator. They are reserved for other purposes.
- **4, 5, 6, 7, 8, 9, 10, 12** – The user keys whose entries can be modified. These keys can be loaded with keys provided by the OEM. Key ID **8** is the default user application key.

5.6.2 Description

This field provides the Keys' assignments.

5.6.3 Path

The **Browse** [. . .] button on the right – to select which key is associated with the **Key ID** and configure the path to each key file.

5.7   Configuration tabs

Depending on the selected device and policy file, you will see varying tabs for configuration.

5.7.1   Single-Image

If you open or create a single-image policy file, the **Configuration** tab displays as follows:
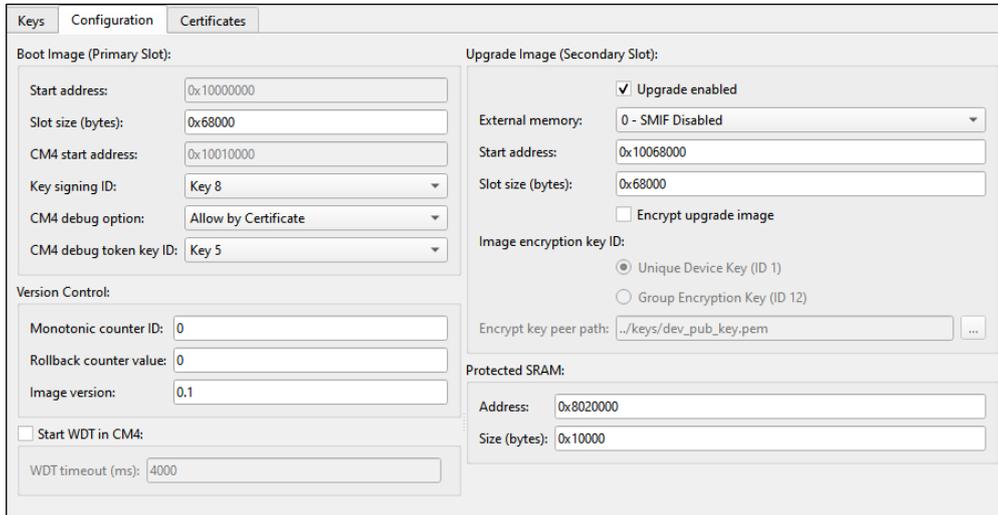
5.7.2   Single-Image Swap mode

If you open or create a single-image policy file with the `swap` suffix, the tool adds a parameter called "Set image OK."
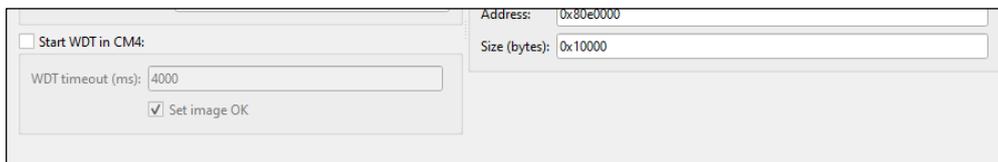
5.7.3   Multi-Image CM4 Configuration

For a multi-image policy file, there are two tabs: **CM4 Configuration** and **CM0+ Configuration**. The following shows the **CM4 Configuration** tab settings:
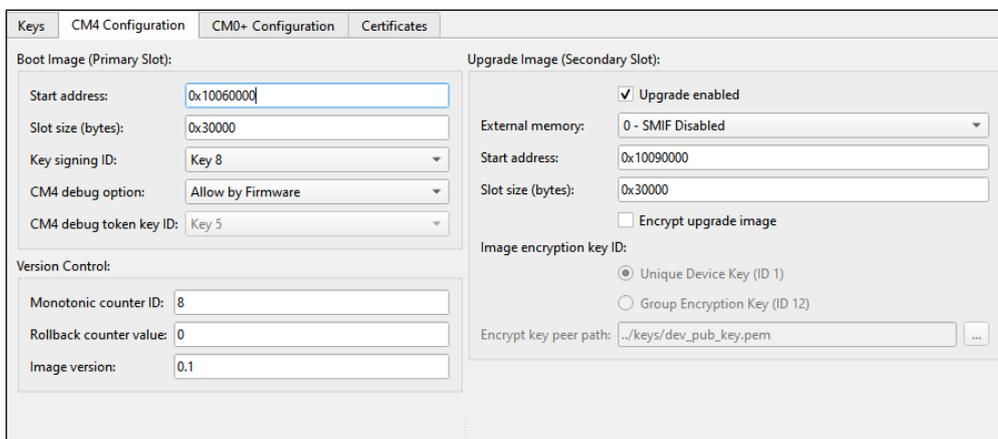
5.7.4 Multi-Image CM0+ Configuration

For a multi-image policy file, there are two tabs: **CM4 Configuration** and **CM0+ Configuration** The following shows the **CM0+ Configuration** tab settings:
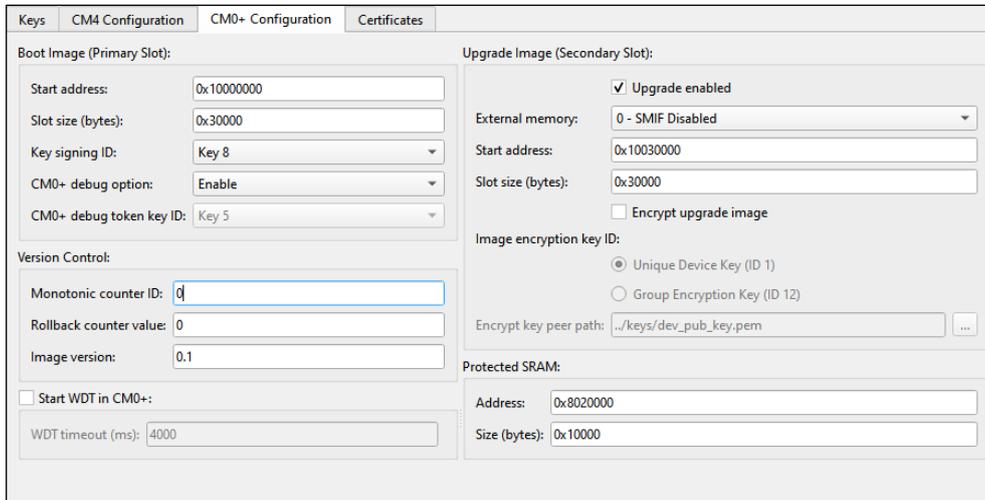
5.7.5 Multi-Image Swap mode

If you open or create a multi-image policy file with the `swap` suffix, the tool adds a parameter called "Set image OK" to the **CM0+ Configuration** tab.
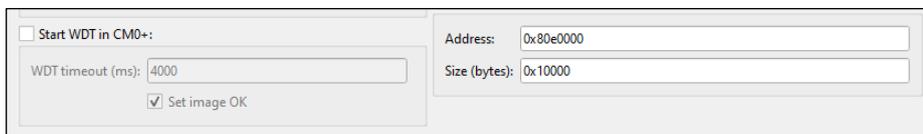
5.8 Configuration parameters

The following describe the configuration parameters available in the **Configuration** tabs.

Note: Some of the parameters display only in the specified tab.

5.8.1 Boot image (Primary Slot)

The memory location to execute code.

5.8.1.1 Start address

The location of the boot image.

5.8.1.2 Slot size (bytes)

The size of the boot image.

5.8.1.3 CM4 start address (Configuration tab)

This is the location in which the CM4 portion of the image starts. In Single-image mode, the CM0+ binary is combined with the user's CM4 code binary.

5.8.1.4   Key signing ID

The key ID to check the boot image signature. The user must select one of the Custom keys (6-10) in the **Keys** tab prior to provisioning.

5.8.1.5   CM4 debug option (Configuration and CM4 tabs)

CM0+ debug option (CM0+ Configuration tab)

Determine how to debug the CM4/CM0. The options:

- **Enable** – Always enabled.
- **Disable** – Always disabled.
- **Allow by Firmware** – The debug access port may be enabled by the firmware.
- **Allow by Certificate** – The debug access port may be enabled by the certificate.

5.8.1.6   CM4 debug token key ID (Configuration and CM4 Configuration tabs)

CM0+ debug token key ID (CM0+ Configuration tab)

The ID to the key to authenticate the debugger.

5.8.2   Version control

This box includes three options for firmware upgrade version control and rollback.

5.8.2.1   Monotonic counter ID

Prevents a rollback during the upgrade process

5.8.2.2   Rollback counter value

The initial counter value.

5.8.2.3   Image version

The version of the image for the MCU Boot header.

5.8.3   Start WDT

This check box is available on the Single-image **Configuration** tab or the Multi-image **CM0+ Configuration** tab. The watchdog timer is used to protect the device from freezing after updating with an incorrect CM4 or CM0+ image. If this check box is enabled, the timer is used to reset the device and revert to the previous image. This setting applies to the entire application, and it is separate from the **Start WDT** setting on the **Advanced** dialog.

Note: *This occurs only if Swap mode is enabled on a CYxx64xA device.*

5.8.3.1   WDT timeout (ms)

The parameter to set the time (in milliseconds) after which the device resets automatically (if the watchdog timer has not been reset previously by the firmware).

5.8.3.2   Set image OK (Swap mode only)

After a swap upgrade is completed, the new image updates the flash contents to mark itself "**OK**", so the bootloader can choose to run it during the next boot. On a startup, the bootloader inspects the flash contents to decide if swapping of the application images was completed.

Depending on the use case, the swap can also be made permanent directly (by setting the "image OK" flag during the image signing). In this case, the bootloader will never attempt to revert the images on the next reset.

Note: *In a multi-image case, the bootloader considers "image OK" flags of each image separately, so both user applications must set the "image ok" flag in their own areas because they consider the "image OK" flags of images separately.*

5.8.4   Upgrade image (Secondary Slot)

The memory location to stage code for the CM0+ upgrade.

5.8.5   Upgrade enabled

If checked, firmware upgrades are allowed.

5.8.5.1   External memory

Specifies the SMIF ID (if any) to upgrade the image. The options:

- 0 – SMIF Disabled
- Slave Select – 1
- Slave Select – 2
- Slave Select – 3
- Slave Select – 4

5.8.5.2   Start address

The location (Secondary/Slot 1) where the upgrade firmware is stored or staged prior to the bootloader transferring it to the Primary Slot.

5.8.5.3   Slot size (bytes)

The size of the upgraded image.

5.8.6   Encrypt upgrade image

This check box determines if the upgrade image must be encrypted.

5.8.6.1   Image encryption key ID

Specifies the key to use the upgrade image encryption.

- **Unique Device Key (ID 1)**
- **Group Encryption Key (ID 12)**

5.8.6.2   Encrypt peer key path

The path to the public key file for image encryption. The key is of the Elliptic Curve Digital Signature Algorithm (ECDSA) type; the file is of the Privacy Enhanced Mail (PEM) format.

5.8.7   Protected SRAM (Configuration and CM0+tabs)

Indicates the memory region for the "secure" CM0+ CPU. The user must not change this from the default unless the CM0+ code has been changed for a special case.

5.8.7.1   Address

The start address of the Protected SRAM location.

5.8.7.2   Size (bytes)

The size of the Protected SRAM region.

5.9   Certificates tab

The **Certificate** tab is used to create and add certificates.

5.9.1   Add certificate

Clicking the **Add Certificate** button provides the user with the options to obtain a certificate.
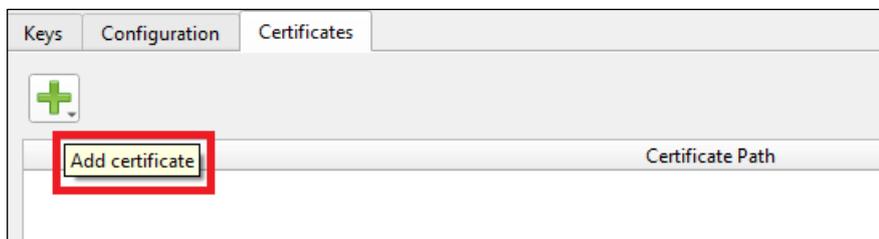
5.9.1.1   Create new certificate
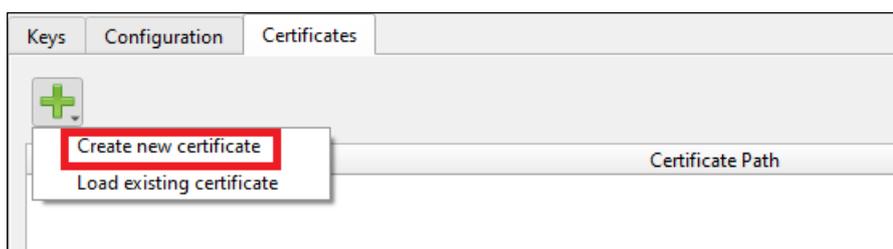
Choosing the "Create new certificate" option displays the Certificates window.
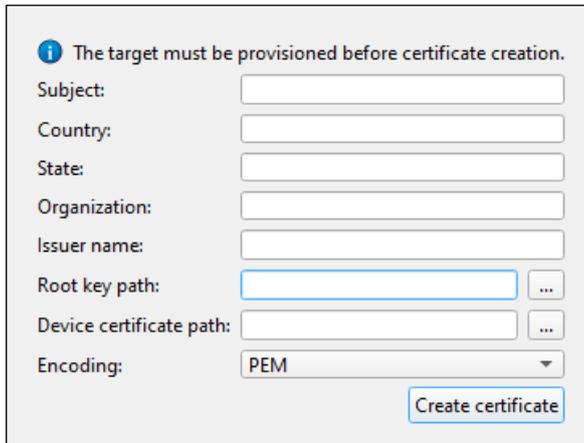
Providing a new public key certificate ensures the secure communication over the Internet. Certificate creation will be skipped if a certificate file already exists.

Note: Certificate creation cannot be executed if the target device is not provisioned.

The user completes the following fields:

- Subject – The title for the certificate.

- Country – The name of the country where the certificate is issued.

- State – The name of the state where the certificate is issued.

- Organization – The name of the organization where the certificate is issued.

- Issuer name – The name of the person by whom the certificate is issued.

- Root key path – The path to the private key file.

- Device certificate path –The location to store the created certificate.

- Encoding – Displays the format of the certificate to be created – PEM or DER. The default is PEM.

Note: The Subject, Country, State, Organization, and Issuer name parameters must include only alphanumeric symbols.

After completing the fields, click the **Create certificate** button to add a new entry to the certificates table.

5.9.1.2   Load existing certificate

Choosing the "Load existing certificate" option displays a file dialog box to allow the user to select an-existing certificate file.

6   Advanced dialog

The **Advanced** dialog contains the parameters used for editing policy files. Two versions of the **Advanced** dialog display depending on the selected policy file: for Overwrite and Swap upgrade modes. For Swap mode, select a policy file with the `swap` suffix.

Overwrite Mode   **Swap Mode**

The following describe the configuration parameters available in the **Advanced** dialog.

Note: *Some of them display only under operation in Swap mode (specified in the description).*

6.1 SysAP Options

These options configure the system access port.

6.1.1 Permissions

- **Enabled** –Always enabled. The default option – debugging of the CM4 application is always allowed.
- **Disabled** – Always disabled. Debugging of the CM4 application is never allowed.
- **Allowed** – Controlled by the firmware.

Note: *Leaving the SysAP in production will create a security risk.*

6.1.2 Enable flash reads

If checked, the SysAP can read the flash memory.

6.1.3 Enable flash writes

If checked, the SysAP can write the flash memory.

6.2 RMA Options

These options control if RMA is allowed, which key to use, and what memory to be erased.

6.2.1 RMA allowed

Enables or disables the RMA process.

- **Enabled** – Always enabled.
- **Disabled** – Always disabled.
- **Allowed** – Controlled by the firmware.

6.2.1.1 RMA token key ID

The IDs to the RMA certificate keys.

6.2.2 Destroy flash region

If there are secrets in the user's flash, the Destroy Flash region option allows erasing the user's secrets in flash to prevent their disclosure.

Note: *This option does not destroy the device; it only erases flash.*

6.2.3 Flash start address

Configures the range of flash to be erased during the RMA process.

6.2.4 Flash size (bytes)

Configures the range of flash to be erased during the RMA process.

6.3   Startup Options

6.3.1   Startup clock

Configures the initial clock speed.

6.3.2   Debug listen window

Configures the amount of time the device waits for an SWD command during the startup.

6.4   Bootloader Options

6.4.1   Bootloader mode

Configures the debugging output.

- **Debug** – The bootloader produces debugging messages.
- **Release** – The bootloader does not produce debugging messages.
- **Custom** – This enables the user to build their own bootloader and provision it. This feature is not currently supported.

6.4.2   Signing key

The private key to sign the bootloader certificate.

6.4.3   HEX file

The bootloader application/program file.

6.4.4   JWT file

The bootloader certificate file.

6.4.5   Status partition (Swap mode)

An extra flash area to store the Swap status for the upgrade and revert images.

6.4.6   Scratch area (Swap mode)

An extra flash area to store parts of images.

6.4.6.1   Start address

The start address of the Swap status area.

6.4.6.2   Size (bytes)

The size of the Swap status area.

6.5   Reprovisioning Options

- **Enable reprovisioning of bootloader** – Check this box to enable the bootloader reprovisioning. This allows the OEM to update the bootloader at a later date (not only during development).

- **Enable reprovisioning of policies** – Check this box to enable the policy reprovisioning. This allows the OEM to reprovision policies at a later date.

6.6 Start WDT in CM0+ (bootloader)

This watchdog timer (WDT) is used to protect the device from freezing after updating with an incorrect CM0+ image. If this check box is enabled for the bootloader, the timer is used to reset the device and revert to the previous image. This setting applies only to the bootloader, and it is separate from the **Start WDT** setting on the **Configuration** tab.

6.6.1 WDT timeout (ms)

The parameter to define the time after which the device resets automatically (if the watchdog timer has not been reset previously by the firmware).

7 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change descriptions
1.0	New tool.
1.10	Fixed minor defects.
	Updated the "Installation" section with ways to install CySecureTools for macOS and Linux.
1.20	Added Scratch area.
	Changed the address format to uppercase.

Revision history

Revision	Date	Description
**	12/17/2020	New spec.
*A	01/22/2021	Updated the MPN name in section "Configuration Parameters" "Start WDT".
*B	03/12/2021	Updated the "Installation" section with ways to install cysecuretools for macOS and Linux.
*C	09/21/2021	Updated to version 1.20.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.