Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN229503

Setting Up Semper Secure NOR Flash for RMA

Author: Zhi Feng

Associated Part Families: S35HL-T/S35HS-T
S36HL-T/S36HS-T
S38HL-T/S38HS-T

This application note describes the steps to set up Cypress Semper™ Secure NOR Flash memories into RMA life cycle stage, and provides guidelines and suggestions to implement the host application software.

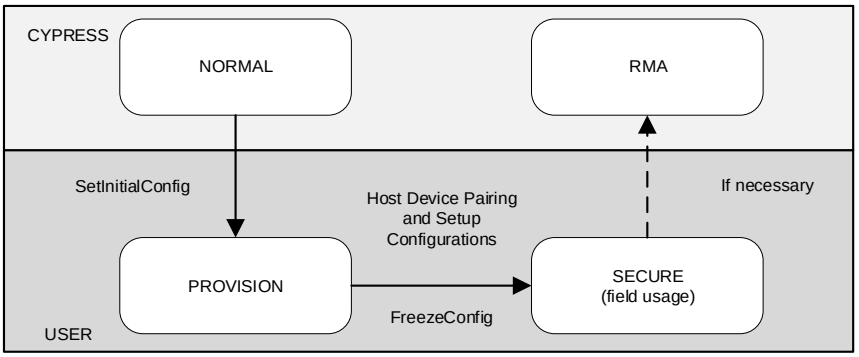
1 Introduction

Semper Secure NOR flash devices follow a unidirectional life cycle that contains several stages, as shown in Figure 1. Devices are shipped in NORMAL life cycle stage without storing any secret. Thereafter, it can move through different life cycle stages. During the PROVISION life cycle stage, users pair up the flash device with the corresponding MCU host, set up secure region configurations and necessary keys. The SECURE life cycle stage is when the device is used in the field. If users expect a possible scenario that requires returning the device to Cypress for failure analysis, they should prepare for the Return Merchandise Authorization (RMA). The RMA life cycle stage allows Cypress to perform tests to analyze the field issue.

This document describes the steps to prepare for the RMA option and to transition the device from SECURE to RMA stage. It is assumed that the reader is already familiar with Semper Secure datasheets and standard operations. Details of the operations are not repeated in this document but can be found in corresponding datasheets. The API (in *Italic*) mentioned in this document can be seen in the datasheet for more details.

This document uses “device” to refer to the Semper Secure flash device and “host” to refer to the paired host MCU if not otherwise specified. Words in *italic* indicate API functions that are available in Semper Solution Development Kit (S-SDK).

Figure 1. Semper Secure NOR Flash Life Cycle



2 Prepare for RMA

2.1 Decide the Option

When you first receive Semper Secure devices, you must decide if you want to allow the RMA life cycle in the future. At the initial pairing with the host, you provision such option into the device through the *SetInitialConfig* transaction. Once the device is provisioned, you cannot change the option. Therefore, if there is any chance that a failure analysis might be required in the future, you should provision the device such that it allows transitioning to the RMA life cycle stage in the future.

The *SetInitialConfig* transaction requires device configuration parameters defined in the datasheet. One of the parameters is “RMA Capable”. If RMA option is allowed, this field should be set to ‘1’; otherwise, ‘0’ should be entered. Inside the same parameter table, the field “RMA Key Index” specifies the index you want to store the RMA key inside the Key Storage area. Cypress suggests using the value ‘65’. Note that any value from 65 to 99 can be chosen as long as the RMA index is different from the Master Key index.

See the *SetInitialConfig* Command section in the datasheet.

2.2 Set Up the RMA Key

If you choose the option to allow RMA, program the RMA key into the device before using the device in the field. This is done with the *ProgramKey* transaction.

For asymmetric devices, the host should program the RMA Public Key into the device. For symmetric devices, the shared RMA secret key should be programmed to the device with encryption. The formats of *ProgramKey* transaction are shown below:

***ProgramKey* Write Packet**

CMD Code	Address	Index	Type	Size	Nonce	Data		TAG or MAC
2 bytes	1 byte	1 bytes	2 bytes	2 bytes	16 bytes	max 512 bytes		16 or 32 bytes
[0:1]	[2]	[3]	[4:5]	[6:7]	[8:23]	[24:43]	[44:size+23]	[size+24:size +55]
0040h	address = 00h	index = 41h	type = 0001h	size of data	nonce	security_parameters	encrypted_key_value or plain_key_value	tag or mac
Asymmetric devices: RMA Public Key (No encryption of key_value) mac = HMAC (master_session_key, CMD_Code||Address||index||type||size||nonce||security_parameters||plain_key_value||(++CMD_Counter)) Symmetric devices: RMA Secret Key (key_value in encrypted by AES-GCM) GCM_Counter = lower 32-bit (++CMD_Counter) GCM_IV = lower 64-bit (master_session_key) || GCM_Counter AAD = CMD_Code || address || index || type || size || nonce || security_parameters encrypted_key_value, TAG = AES-GCM(master_session_key, GCM_IV, AAD, plain_key_value)								

***ProgramKey* Response Packet**

Response	Result	MAC
2 bytes	2 bytes	32 bytes
[0:1]	[2:3]	[4:35]
4000h	result	mac
mac = HMAC(master_session_key, response||result||(++CMD_Counter))		

3 Transitioning to RMA

3.1 Destroy Sensitive Data

After you decide to return a Semper Secure NOR Flash device for failure analysis, it is your responsibility to erase any sensitive data on the device before transitioning the device into RMA life cycle stage. Once the device is in RMA, you can no longer establish session keys to perform any secure transactions.

To remove the data inside secure regions, perform erase options according to the region access level. For example, use *AuthenticatedErase* or *EncryptedErase* transactions.

It is not necessary to delete keys inside the Key Storage area because all keys are stored in encrypted format, and the encryption key is derived from the Unique Device Secret (UDS) which is not accessible with hardware protection in the RMA life cycle stage.

3.2 Set Up the RMA Session Key

Before transitioning the device into RMA, the host must establish the RMA session key. This is done by a two-step transaction: *CreateSesionKey* and *StoreSessionKey*. This transaction follows the same way as creating Master Session Key or Region Session Key, which you should be familiar with because all secure transactions require session keys during the SECURE life cycle stage.

For asymmetric devices, generating the RMA session key requires the RMA public key already programed on the device (see Set Up the RMA Key). The host should also have obtained the Alias Public Key from the device. This can be done by reading the Alias Certificate that contains the Alias Public Key information.

For symmetric devices, generating the RMA session key requires the MCU host and the device both to know the shared RMA secret key. This secret key should be already programmed into the device (see Set Up the RMA Key).

If the necessary keys are in place, you can issue the *CreateSessionKey* and *StoreSessionKey* commands to establish the RMA session key. See the *CreateSessionKey*/*StoreSessionKey* sections in the datasheet for more details. Formats of *CreateSessionKey* and *StoreSessionKey* transaction packets are shown below:

***CreateSessionKey* Write Packet**

CMD Code	Address	Type	Nonce	Security Parameters	CRC-16
2 bytes	4 bytes	2 bytes	16 bytes	20 bytes	2 bytes
[0:1]	[2:5]	[6:7]	[8:23]	[24:43]	[44:45]
000Ah	Address = 00000000h	type = 0002h	nonce_u	security_parameters	crc_checksum

***CreateSessionKey* Read Packet**

Response	Result	Nonce	MAC
2 bytes	2 bytes	16 bytes	32 bytes
[0:1]	[2:3]	[4:19]	[20:51]
0A00h	Result	nonce_v	mac
mac = HMAC(new_session_key, response||result||nonce_v||(++CMD_Counter))			

***StoreSessionKey* Write Packet**

CMD Code	Address	Type	Data	MAC
2 bytes	4 bytes	2 bytes	32 bytes	32 bytes
[0:1]	[2:5]	[6:7]	[8:39]	[40:71]
001Eh	address = 00000000h	type = 0002h	MacTagU	mac

For Asymmetric devices:MacTagU = HMAC(new_session_key, "KC_1_U"||pub_key_u||pub_key_v||nonce_u||nonce_v); where pub_key_u and pub_key_v are the public keys used to derive the new_session_key on the host and the device side respectively mac = HMAC(new_session_key, CMD_Code||address||type||MacTagU||(++CMD_Counter)) For Symmetric devices: MacTagU = HMAC(new_session_key, nonce_v||nonce_u) mac = HMAC(new_session_key, CMD_Code||address||type||MacTagU||(++CMD_Counter))

***StoreSessionKey* Read Packet**

Response	Result	MAC
2 bytes	2 bytes	32 bytes
[0:1]	[2:3]	[4:35]
1E00h	Result	mac
mac = HMAC(new_session_key, response||result||(++CMD_Counter))		

3.3 Execute the *TransitionToRMA* Transaction

After the RMA session key is established, you can issue the *TransitionToRMA* transaction to move the device into RMA life cycle stage.

***TransitionToRMA* Write Packet**

CMD Code	Reserved	MAC
2 bytes	2 bytes	32 bytes
[0:1]	[2:3]	[4:35]
0030h	0000h	mac
mac = HMAC(rma_session_key, CMD_Code||Reserved||(++CMD_Counter))		

***TransitionToRMA* Read Packet**

Response	Result	MAC
2 bytes	2 bytes	32 bytes
[0:1]	[2:3]	[4:35]
3000h	result	mac
mac = HMAC(rma_session_key, response||result||(++CMD_Counter))		

After the *TransitionRMA* transaction is completed, you can ship the device back to Cypress for failure analysis.

4 Using Semper Solution SDK

Cypress Semper Solution Development Kit (S-SDK) is a software package that is designed to help customers develop their own driver or directly use the provided code examples. Transitioning to RMA is one of the examples the S-SDK provides. You can follow the S-SDK code example or use the platform-independent C code to perform these steps mentioned in this document.

5 Conclusion

Returning Semper Secure NOR Flash devices to Cypress for failure analysis requires planning ahead, installing RMA keys, protecting sensitive data, and the actual step to transition the device into the RMA life cycle stage. This document summarizes the steps that need to be taken before shipping the device back to Cypress. Software developers can follow these steps to prepare the software in case such needs arise.

6 References

002-26101 S35HS-T, S35HL-T Semper Secure Flash with Quad SPI Datasheets

002-28332 AN228332 – Initial Provisioning in Cypress Semper Secure NOR Flash

Document History

Document Title: AN229503 – Setting Up Semper Secure NOR Flash for RMA

Document Number: 002-29503

Revision	ECN	Date	Description of Change
**	6837587	03/25/2020	Initial release

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Code Examples | Projects | Videos | Blogs | Training| Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709