サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

AN229503

Semper Secure NOR フラッシュの RMA 用セットアップ

著者：Zhi Feng

関連する製品ファミリ：S35HL-T / S35HS-T
S36HL-T / S36HS-T
S38HL-T / S38HS-T

このアプリケーションノートでは、サイプレス Semper™ Secure NOR フラッシュメモリを RMA ライフサイクルステージにセットアップする手順について説明し、ホストアプリケーションソフトウェアを実装するためのガイドラインと提案を提供します。

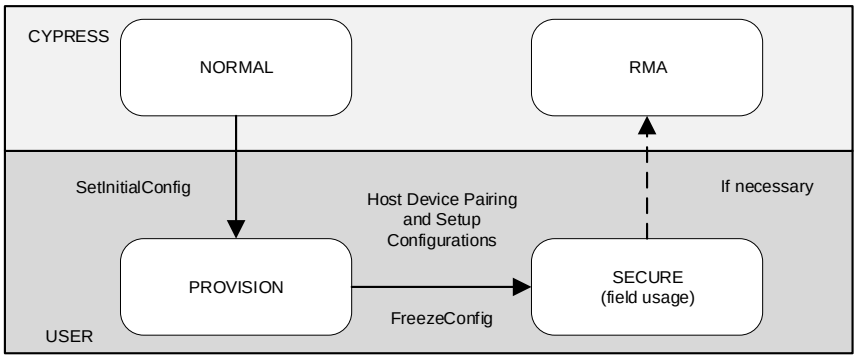
1 はじめに

Semper Secure NOR フラッシュデバイスは、図 1 に示すように、いくつかの段階を含む単方向のライフサイクルに従います。デバイスは機密を保持せずに NOMAL ライフサイクルステージで出荷されます。その後、さまざまなライフサイクルステージを移動できます。PROVISION ライフサイクルステージで、ユーザーはフラッシュデバイスを対応する MCU ホストとペアリングし、セキュアリージョン構成と必要なキーをセットアップします。デバイスがフィールドで使用されるとき、SECURE ライフサイクルステージです。ユーザーが障害分析のためにデバイスをサイプレスへ返却する必要があるシナリオの可能性を期待している場合は、返品承認 (RMA) の準備をする必要があります。RMA ライフサイクルステージにより、サイプレスはフィールドの問題を分析するためのテストを実行できます。

このドキュメントでは RMA オプションを準備し、デバイスを SECURE ステージから RMA ステージに移行する手順について説明します。ユーザーはすでに Semper Secure のデータシートと標準操作に精通していることを前提とします。操作の詳細はこのドキュメントでは繰り返されていませんが、対応するデータシートに記載されています。このドキュメントに記載されている API (*イタリック体*) の詳細についてはデータシートを参照してください。

このドキュメントでは特に指定がない限り、「デバイス」は Semper Secure フラッシュデバイスを指し、「ホスト」はペアのホスト MCU を指します。*イタリック体* の単語は Semper Solution Development Kit (S-SDK) で使用できる API 関数を示します。

図 1. Semper Secure NOR フラッシュライフサイクル



2 RMA の準備

2.1 オプションの決定

Semper Secure デバイスを最初に受け取ったときに、将来的に RMA ライフサイクルを許可するかどうかを決定する必要があります。ホストとの最初のペアリング時に、*SetInitialConfig* トランザクションを介してそのようなオプションをデバイスにプロビジョニングします。デバイスがプロビジョニングされると、オプションを変更できません。したがって、障害分析が将来必要になる可能性がある場合、将来的に RMA ライフサイクルステージに移行できるようにデバイスをプロビジョニングする必要があります。

SetInitialConfig トランザクションはデータシートで定義されているデバイス構成パラメーターが必要です。パラメーターの 1 つは「RMA Capable」です。RMA オプションが許可されている場合、このフィールドは'1'に設定する必要があります。それ以外の場合は'0'を入力する必要があります。同じパラメーター表内のフィールド「RMA Key Index」はキーストレージ領域内に RMA キーを保存するインデックスを指定します。サイプレスは値'65'の使用を提案します。RMA インデックスがマスターキーインデックスと異なる限り、65 から 99 までの任意の値を選択できることに注意してください。

データシートの *SetInitialConfig* コマンドセクションを参照してください。

2.2 RMA キーの設定

RMA を許可するオプションを選択した場合、フィールドでデバイスを使用する前に RMA キーをデバイスにプログラムしてください。これは *ProgramKey* トランザクションで実行されます。

非対称デバイスの場合、ホストは RMA 公開キーをデバイスにプログラムする必要があります。対称デバイスの場合、共有 RMA 秘密キーは暗号化を使用してデバイスにプログラムする必要があります。*ProgramKey* トランザクションの形式を以下に示します。

***ProgramKey* 書き込みパケット**

CMDコード	アドレス	索引	分類	サイズ	Nonce	データ:		TAGまたはMAC
2バイト	1バイト	1バイト	2バイト	2バイト	16バイト	最大512バイト		16または32バイト
[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	データのサイズ	nonce	security_parameters	encrypted_key_valueまたはplain_key_value	tagまたはmac
非対称デバイス：RMA公開キー (key_valueの暗号化なし) mac = HMAC (master_session_key, CMD_Code||Address||index||type||size||nonce||security_parameters||plain_key_value||(++CMD_Counter)) 対称デバイス: RMA秘密キー (AES-GCMによって暗号化されたkey_value) 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* 応答パケット**

対応	結果	MAC
2バイト	2バイト	32バイト
[0:1]	[2:3]	[4:35]
4000h	結果	mac
mac = HMAC(master_session_key, response||result||(++CMD_Counter))		

3 RMA への移行

3.1 機密データの破棄

障害分析のために Semper Secure NOR フラッシュデバイスを返却することを決定した後、デバイスを RMA ライフサイクルステージに移行する前に、デバイス上の機密データを消去するのはユーザーの責任です。デバイスが RMA に入ると、安全なトランザクションを実行するためのセッションキーを確立できません。

安全なリージョン内のデータを削除するためには、リージョンのアクセスレベルに応じて消去オプションを実行します。例えば、*AuthenticatedErase* または *EncryptedErase* トランザクションを使用します。

すべてのキーは暗号化された形式で保存され、暗号化キーは RMA ライフサイクル段階でハードウェア保護ではアクセスできない Unique Device Secret (UDS) から取得されるため、キーストレージ領域内のキーを削除する必要はありません。

3.2 RMA セッションキーの設定

デバイスを RMA に移行する前に、ホストは RMA セッションキーを確立する必要があります。これは *CreateSesionKey* と *StoreSessionKey* の 2 段階のトランザクションによって実行されます。このトランザクションは、マスターセッションキーまたはリージョンセッションキーの作成と同じ方法に従います。これはすべてのセキュアトランザクションが SECURE ライフサイクルステージ中にセッションキーを必要とするため、よく知っている必要があります。

非対称デバイスの場合、RMA セッションキーを生成するためにはデバイスにすでにプログラムされている RMA 公開キーが必要です (RMA キーの設定を参照)。ホストはデバイスからエイリアス公開キーも取得している必要があります。これはエイリアス公開キー情報を含むエイリアス証明書を読み出すことで実行できます。

対称デバイスの場合、RMA セッションキーを生成するためには MCU ホストとデバイスの両方が、共有 RMA 秘密キーを知っている必要があります。この秘密キーはすでにデバイスにプログラムされている必要があります (RMA キーの設定を参照)。

必要なキーが配置されている場合は、*CreateSessionKey* コマンドと *StoreSessionKey* コマンドを発行して、RMA セッションキーを確立できます。詳細については、データシートの *CreateSessionKey/StoreSessionKey* セクションを参照してください。*CreateSessionKey* および *StoreSessionKey* トランザクションパケットの形式を以下に示します。

***CreateSessionKey* 書き込みパケット**

CMDコード	アドレス	分類	Nonce	セキュリティパラメーター	CRC-16
2バイト	4バイト	2バイト	16バイト	20バイト	2バイト
[0:1]	[2:5]	[6:7]	[8:23]	[24:43]	[44:45]
000Ah	Address = 00000000h	type = 0002h	nonce_u	security_parameters	crc_checksum

***CreateSessionKey* 読み出しパケット**

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

***StoreSessionKey* 書き込みパケット**

CMDコード	アドレス	分類	データ:	MAC
2バイト	4バイト	2バイト	32バイト	32バイト

CMDコード	アドレス	分類	データ:	MAC
[0:1]	[2:5]	[6:7]	[8:39]	[40:71]
001Eh	address = 00000000h	type = 0002h	MacTagU	mac
非対称デバイスの場合: MacTagU = HMAC (new_session_key, "KC_1_U" || pub_key_u || pub_key_v || nonce_u || nonce_v); ここで、pub_key_u とpub_key_v は、それぞれホスト側とデバイス側でnew_session_keyを導出するために使用される公開キーです。 mac = HMAC(new_session_key, CMD_Code||address||type||MacTagU||(++CMD_Counter)) 対称デバイスの場合: MacTagU = HMAC(new_session_key, nonce_v||nonce_u) mac = HMAC(new_session_key, CMD_Code||address||type||MacTagU||(++CMD_Counter))				

***StoreSessionKey* 読み出しパケット**

対応	結果	MAC
2バイト	2バイト	32バイト
[0:1]	[2:3]	[4:35]
1E00h	結果	mac
mac = HMAC(new_session_key, response||result||(++CMD_Counter))		

3.3 *TransitionToRMA* トランザクションを実行します

RMA セッションキーが確立されたのち、*TransitionToRMA* トランザクションを発行してデバイスを RMA ライフサイクルステージに移動できます。

***TransitionToRMA* 書き込みパケット**

CMDコード	予約済み	MAC
2バイト	2バイト	32バイト
[0:1]	[2:3]	[4:35]
0030h	0000h	mac
mac = HMAC(rma_session_key, CMD_Code||Reserved||(++CMD_Counter))		

***TransitionToRMA* 読み出しパケット**

対応	結果	MAC
2バイト	2バイト	32バイト
[0:1]	[2:3]	[4:35]
3000h	結果	mac
mac = HMAC(rma_session_key, response||result||(++CMD_Counter))		

TransitionRMA トランザクションが完了したら、障害分析のためにデバイスをサイプレスに返送できます。

4 Semper Solution SDK の使用

Cypress Semper ソリューション開発キット (S-SDK) は、お客様が独自のドライバーを開発したり、提供されたコード例を直接使用したりできるように設計されたソフトウェアパッケージです。RMA への移行は S-SDK が提供する例の 1 つです。S-SDK サンプルコードに従うか、プラットフォームに依存しない C コードを使用して、このドキュメントに記載されているこれらの手順を実行できます。

5 結論

障害分析のために Semper Secure NOR フラッシュデバイスをサイプレスに戻すためには、事前の計画、RMA キーのインストール、機密データの保護、およびデバイスを RMA ライフサイクルステージに移行するための実際の手順が必要です。このドキュメントでは、デバイスをサイプレスに返送する前に実行する必要のある手順をまとめています。ソフトウェア開発者は、このようなニーズが発生した場合に備えて、これらの手順に従ってソフトウェアを準備できます。

6 参考文献

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

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

改訂履歴

文書名: AN229503 – Semper Secure フラッシュの RMA 用セットアップ

文書番号: 002-32716

版数	変更内容
**	本版は英語版 002-29503 Rev. **について、CYPRESS DEVELOPER COMMUNITYの参画者によって日本語に翻訳されたドキュメントです。

セールス, ソリューションおよび法律情報

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、サイプレスのロケーションページをご覧ください。

製品

Arm® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチセンシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス	cypress.com/wireless

PSoC®ソリューション

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

サイプレス開発者コミュニティ

コミュニティ | サンプルコード | Projects | ビデオ | ブログ | トレーニング | Components

テクニカルサポート

cypress.com/support

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709