高速セキュアブート用の Semper セキュアフラッシュのセットアップ

About this document

Scope and purpose

このアプリケーションノートでは、高速セキュアブートを実行するために Semper™セキュア NOR フラッシュメモリを設定する手順について説明します。この目的で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も提供します。

関連製品ファミリ

- S35HL-T/S35HS-T
- S36HL-T/S36HS-T
- S38HL-T/S38HS-T

Table of contents

1 はじめに

一部のアプリケーション、特に自動車セグメントにおいて、タイミングが重要なメッセージに迅速に応答するためには、非常に高速な起動時間が必要です。例えば、コントローラーエリアネットワーク (CAN) バスをリスニングしている車内のシステムは、CAN バスメッセージを処理するために 100 ミリ秒以内に起動する必要があります。従来のシステムでは、この要件を簡単に満たせます。ただし、セキュアブートを必要とするシステムでは、セキュアブートの性質上、実行する前にさまざまなブートアップステージのファームウェアを検証する必要があり、余分な時間がかかるため、このような時間要件を満たすのは困難です。

Semper セキュア NOR フラッシュデバイスは、このようなアプリケーションが安全な起動時間の要件を満たすのに役立つように設計されています。このアプリケーションノートでは、高速で安全なブートを目的としてフラッシュをセットアップする方法と、実際のブートフローについて説明します。このアプリケーションノートに従って、フローを実装できます。さらに、Semper Solution Development Kit (S-SDK) は、このドキュメントで説明されるように、高速で安全なブートユーザーの例を提供します。S-SDK を使用する場合は、ソースサンプルコードを直接使用できます。

Semper セキュア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。操作の詳細については、対応するデータシートおよびアプリケーションノートを参照してください。

特に指定がない限り、このドキュメントで使用している「デバイス」は Semper セキュア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。

2 高速セキュアブート要件

セキュアブートのプロセスでは、ホスト MCU がブートコードを実行する前に、ブートコード自体の認証が必要です。認証には、コードストレージハードウェアが元のハードウェアであり、ブートコードが改ざんされていないかどうかの確認が含まれます。セキュアブートプロセスを実装する方法は多くありますが、このアプリケーションノートでは、高速セキュアブートを実行する方法の 1 つを示します。

高速セキュアブートの目的は、比較的高速な方法で、可能な限り 100 ミリ秒以内にセキュアブートプロセスを完了することです。この期間は、システムの電源がオンになってからシステムが起動し、ブートコードで実行されるまで測定されます。高速セキュアブートの主な手順は次のとおりです。

1. デバイスの準備。これは、プロビジョニング中に 1 回だけ実行されます。
2. フラッシュデバイスの認証。
3. ブートコードの認証。これは、Semper セキュアフラッシュのオプションの手順です。Semper セキュアはブートコード用の安全なストレージを提供するため、プログラムされた後、許可されていない第三者によってコードが改ざんされることはありません。したがって、フラッシュデバイスが承認されると。安全なストレージ内のコードも無傷であるとみなされます。
4. デバイスからブートコード読取り、またはフラッシュからその場で実行。

3 高速セキュアブートのためのデバイスの準備

3.1 正しいオーダーオプションの選択

Semper セキュアには、対称デバイスと非対称デバイスの 2 つの主要なオーダー製品カテゴリがあります。非対称キーの性質上、ホスト MCU とフラッシュデバイスが相互認証プロセスを完了するためには、非常に長い時間がかかります。したがって、非対称デバイスは高速セキュアブートを実行するようには設計されていません。対称デバイスは共有秘密スキームを使用するため、セキュアブートプロセスははるかに高速です。このドキュメントで説明されている高速セキュアブートプロセスは、対称デバイスにのみ適用されます。

3.2 ホストとの最初のペアリング

ホスト MCU を対称 Semper セキュアデバイスとペアリングするための、一般的な初期プロビジョニング手順は次のとおりです。

1. デバイスのファームウェアを検証します。
2. `SetInitialConfig` トランザクションを使用して、デバイスの初期構成を設定します。
3. マスターキー (共有シークレット) をデバイスにインストールします。
4. リージョン構成をセットアップします。
5. プログラムリージョンシークレットキー。
6. `FreezeConfig` トランザクションを使用して、すべての構成設定をフリーズします。

各ステップの詳細な説明については、**Semper Secure Early Access Program** で利用可能な AN228332 を参照してください。

ステップ 1 は、デバイスファームウェア (レイヤー0 (L0) およびレイヤー1 (L1)) を検証します。ホストは、L0 と L1 のハッシュ値を格納し、高速セキュアブートシーケンス中に値を検証できます。

ステップ 3 は、共有シークレットをフラッシュデバイスにインストールします。この共有シークレットは、デバイスのマスターキーでもあります。これは、ホストとデバイス間の相互認証の基礎です。共有シークレットは、フラッシュの安全なキーストレージに保持され、Composite Device Identifier (CDI) で暗号化されます。CDI 自体は、Unique Device Secret (UDS) と不変のレイヤー0 ファームウェアのハッシュ値から派生します。つまり、CDI はデバイスごとに一意であり、外の世界に公開されることはありません。共有シークレットが CDI を使用して保存された値を復号化することから正しく回復できる場合、フラッシュ UDS、レイヤー0 ファームウェアがまだ無傷であることを証明します。したがって、フラッシュはホストによって認証されます。

共有シークレットがフラッシュに正常にインストールされた後、ブートコードがすでにセキュアストレージにプログラムされていると仮定して、デバイスは高速セキュアブートプロセスを実行する準備ができています。

4 高速セキュアブート手順の例

4.1 マスターセッションキーの設定

電源を入れると、ホスト MCU はフラッシュデバイスとのマスターセッションキーを確立します。マスターセッションキーが正常に生成されると、デバイスの CDI と共有シークレットがすべて無傷であることが保証されます。

Figure 1 に、ホストの観点からマスターセッションキーを生成する手順を示します。

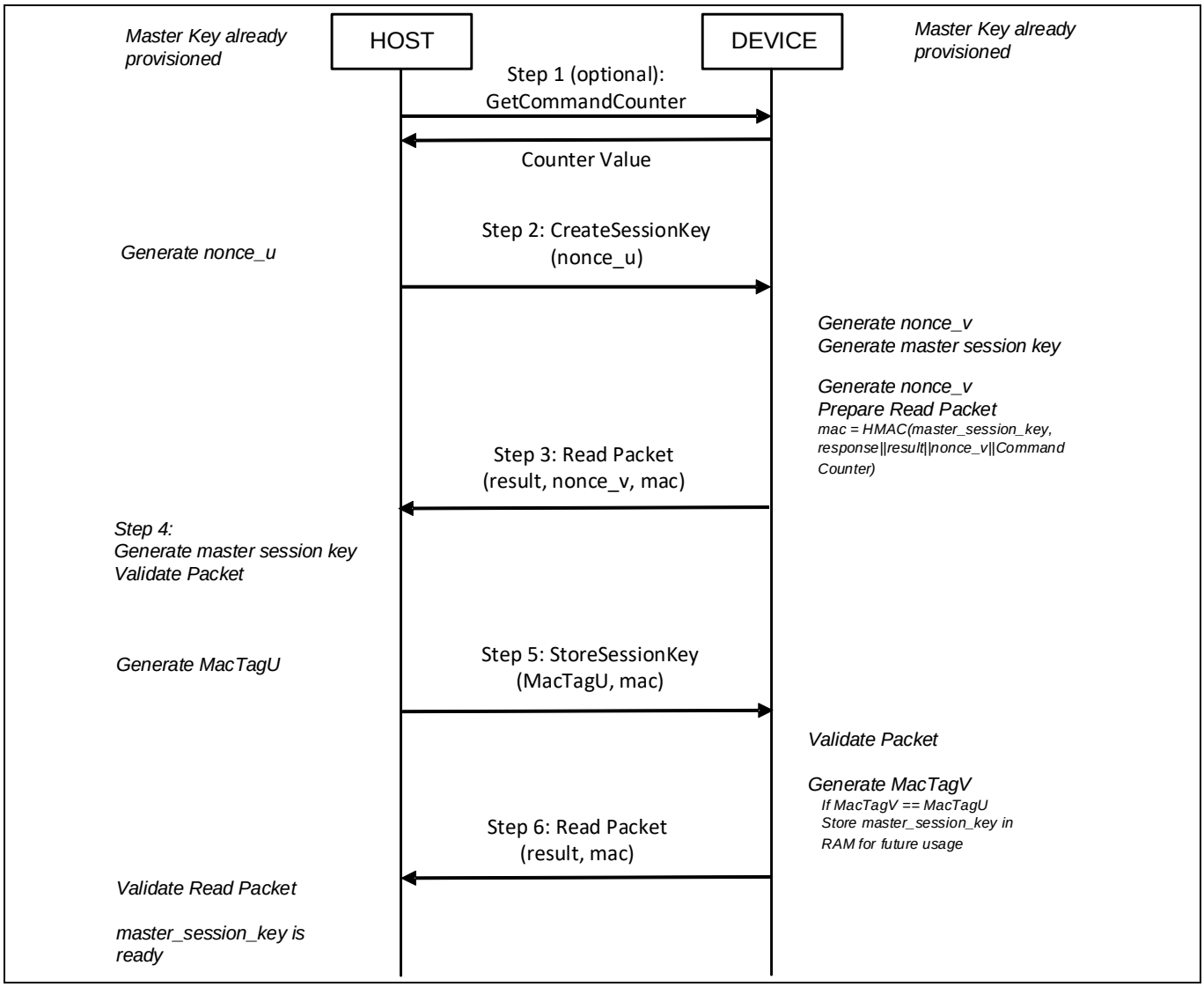

Figure 1 マスターセッションの鍵生成フロー

フローの詳細な説明は次のとおりです。

1. **GetCommandCounter トランザクションを発行します。**これはオプションの手順です。ホストがすでにコマンドカウンター値をフラッシュと同期している場合は、この手順を省略できます。ホスト MCU とフラッシュの両方がパワーオンリセット (POR) を通過した場合、安全なトランザクションプロセスを開始するためには、ホストが初めてコマンドカウンター値を取得する必要があります。これは、セキュリティパラメータを必要としないパブリックトランザクションです。

2. **CreateSessionKey トランザクションを発行します。**ホストは、CreateSessionKey トランザクションでマスターセッションキー生成シーケンスを開始します。タイプ 00h で送信する必要があります。つまり、master_session_key を作成します。このトランザクション書き込みパケットには、16 バイトの nonce_u 値、20 バイトのセキュリティパラメータ、およびパケットの整合性を確保するための CRC-16 チェックサムが含まれます。
書き込みパケットを送信した後、ホストは割込みを監視するか、ステータスレジスタをポーリングして、デバイスが操作を完了したことを確認する必要があります。
3. **デバイスからの応答を読み出します。**デバイスの準備ができたら、ホストはパケット読み出しトランザクションを発行して、デバイスから結果コードを取得します。読み出されたパケットには、フラッシュによって生成された nonce_v 値が含まれます。パケットには、フラッシュで新しく生成されたマスターセッションキーによって計算された HMAC 値も含まれます。ホストがマスターセッションキーを取得する前は、まだパケットを検証できません。
4. **ホストで master_session_key を生成します。**デバイスから nonce_v 値を取得した後、ホストは次の式に基づいて master_session_key を計算する必要があります。

```
salt (256 bits) = 0xDEADBEEF || last 28 bytes of Device Configuration Data;

Z (256 bits) = Master_Key (i.e., Shared_Secret);

Kdk (256 bits) = HMAC(salt, Z);

L = 0x100 (HMAC-256) or 0x140 (AES-GCM);

Label (64 bits) = 0x6DE8BC2177D879B2 (HMAC-256) or 0x921743DE8827864D (AES-GCM);

Context (512 bits) = Life_cycle (16b) || 0000h || ++CmdCounter (64b) ||
security_parameters (160b) || nonce_u (128b) || nonce_v (128b);

master_session_key = KDF (Kdk, L, Label || Context);
```

フラッシュ上のものと同じであるはずのマスターセッションキーを取得した後、ホストは HMAC によって読み出されたパケットを検証して、その信頼性を確認できます。

5. **StoreSessoinKey トランザクションを発行します。**ホストは、この式に従って新しいマスターセッションキーから MacTagU 値を生成し、それを master_session_key のタイプ 0000h の書き込みパケットに含めます。

```
MacTagU = HMAC(master_session_key, nonce_v||nonce_u)
```

書き込みパケットを送信した後、ホストは割込みを監視するか、ステータスレジスタをポーリングして、デバイスが操作を完了したことを確認する必要があります。

6. **デバイスからの応答を読み出します。**デバイスの準備ができたら、ホストはパケット読み出しトランザクションを発行して、デバイスから結果コードを取得します。読み出されたパケットには、フラッシュによって生成された MacTagV 値が含まれます。パケットを HMAC 値で検証した後、ホストは MacTagV を MagTagU 値と比較する必要があります。比較に合格した場合、それはマスターセッションキーがホストとデバイスの両方によって正常に検証されたことを意味します。

4.2 デバイスファームウェアの検証

マスターセッションキーを設定した後、ホストは ValidateFW トランザクションをデバイスに発行できます。このトランザクションは FMAC 値を返し、その計算には L0 および L1 ファームウェアのハッシュ値が含まれます。次に、ホストは、保存されている L0 および L1 ハッシュ値を検証して、改ざんされていないことを確認できます。

これらの手順の後、ホストとデバイスは相互認証されます。

4.3 リージョンセッションキーの設定

マスターセッションキーを設定するための同じ手順を使用して、ホストは、ブートコードを含むリージョンのフラッシュデバイスでリージョンセッションキーを設定できます。このリージョンのホストとデバイス間のすべての安全なトランザクションは、リージョンセッションキーを使用します。

4.4 読み出し用にリージョンのロックを解除

この例は、ブートコード領域からのインプレース実行 (XiP) を示します。セキュアリージョンは、アクセスレベルが Authenticated Lock リージョンとして設定されています。読み出し操作を実行するには、ホストは最初にリージョンセッションキーを使用してリージョンのロックを解除する必要があります。これは、`AuthenticatedUnlock` 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の後、ホストは XiP のリージョンからレガシーSPI またはクアッド SPI 読み出しの実行を開始できます。

4.5 読み出し前にブートコードを認証

さらに、ホストはブートコードを実行する前に、`AuthenticateMemory` トランザクションを発行してブートコード全体を検証できます。このトランザクションは、リージョンセッションキーを使用して、指定されたアドレス範囲のハッシュ値を計算し、そのハッシュ値をホストに返します。次に、ホストはブートコードを読み出す前にハッシュ値を検証します。

5 Semper Solution SDK の使用

Semper Solution Development Kit (S-SDK) は、お客様が独自のドライバーを開発したり、提供されたサンプルコード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。Fast Secure Boot は、S-SDK で提供されている例の 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手順を実行できます。

6 結論

Semper セキュアフラッシュは、安全な領域でブートコード用の安全なストレージを提供することにより、ホスト MCU がシステム要件を満たすために、高速で安全なブートを実行できるようにします。このプロセスでは、ホスト側とデバイス側の両方で事前にプロビジョニングされた共有シークレットを検証します。ブートコードが使用される前に、デバイスとソフトウェアの両方の整合性を検証できます。

7 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

Note: *これらのドキュメントは、**Semper Secure Early Access Program** で利用できます。*

改訂履歴

Document version	Date of release	Description of changes
**	2021-03-23	本版は英語版 002-30415 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。

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

Edition 2021-03-23
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to www.cypress.com/support

Document reference
002-32847 Rev. **

重要事項
本文書に記載された情報は、いかなる場合も、条件または特性の保証とみなされるものではありません（「品質の保証」）。本文に記された一切の事例、手引き、もしくは一般的価値、および／または本製品の用途に関する一切の情報に関し、インフィニオンテクノロジーズ（以下、「インフィニオン」）はここに、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

さらに、本文書に記載された一切の情報は、お客様の用途におけるお客様の製品およびインフィニオン製品の一切の使用に関し、本文書に記載された義務ならびに一切の関連する法的要件、規範、および基準をお客様が遵守することを条件としています。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

本製品、技術、納品条件、および価格についての詳しい情報は、インフィニオンの最寄りの営業所までお問い合わせください（**www.infineon.com**）。

警告事項
技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。