|  |
| --- |
| AN230415 |
| 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

|  |
| --- |
| 著者：Zhi Feng |
| 関連するパーツファミリ：S35HL-T / S35HS-TS36HL-T/S36HS-TS38HL-T/S38HS-T |

  |
| 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。また、そのような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。 |

# 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では、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、非常に高速な起動時間が必要です。たとえば、コントローラーエリアネットワーク（CAN）バスをリッスン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すことができます。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 Secure 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ているように、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、ソースコード例を直接使用できます。

SemperSecure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ています。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。

このドキュメントでは、特に指定がない限り、「デバイス」を使用してSemper Secureフラッシュデバイスを指し、「ホスト」を使用してペアのホストMCUを指します。

# 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たくさんありますが、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つを示しています。

高速セキュアブートの目的は、比較的高速な方法で、できれば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、SemperSecureフラッシュのオプションの手順です。Semper Secure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見なされます。
4. デバイスからブートコードを読み取るか、フラッシュからその場で実行します。

# 高速セキュアブートのためのデバイスの準備

## 正しい注文オプションの選択

Semper Secureには、対称デバイスと非対称デバイスの2つの主要な注文パーツ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のにはるか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。

## ホストとの最初のペアリング

ホストMCUを対称SemperSecure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 AccessProgram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。この共有秘密は、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。

# 高速セキュアブート手順の例

## 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。

図1は、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ています。

図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ています。

書き込みパケットを送信した後、ホストは割り込みを監視するか、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。

1. デバイスからの応答を読み取ります。デバイスの準備ができたら、ホストはパケット読み取り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。読み取られたパケットには、フラッシュによって生成されたnonce\_v値が含まれています。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値も含まれています。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。
2. 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によって読み取られたパケットを検証して、その信頼性を確認できます。

1. 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、ホストは割り込みを監視するか、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。

1. デバイスからの応答を読み取ります。デバイスの準備ができたら、ホストはパケット読み取り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。読み取られたパケットには、フラッシュによって生成されたMacTagV値が含まれています。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。

## 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。

## 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。

## 読み取り用にリージョンのロックを解除する

この例は、ブートコード領域からのインプレース実行（XiP）を示しています。セキュアリージョンは、アクセスレベルがAuthenticatedLockリージョンとして設定されています。読み取り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読み取りの実行を開始できます。

## 読み取る前にブートコードを認証する

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取る前にハッシュ値を検証します。

# Semper SolutionSDK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。Fast Secure Bootは、S-SDKで提供されている例の1つです。S-SDKコードの例に従うか、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。

# 結論

Semper Secure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。

# 参考文献

002-26101 S35HS-T、S35HL-TSemperセキュアフラッシュとクアッドSPIデータシート

002-28332 AN228332 – Cypress Semper SecureNORフラッシュでの初期プロビジョニング

注：これらのドキュメントは、Semper Secure Early AccessProgramで利用できます。

ドキュメント履歴

＜＜★Document Title: AN230415 – Setting up Semper Secure Flash for Fast Secure Boot

＜＜★Document Number: 002-30415

| **版** | **Engineering Change Notification (技術変更届)** | **発行日** | **変更内容** |
| --- | --- | --- | --- |
|  |  |  |  |