参考資料

Application Note

Keywriter の導入戦略：本番環境でのセキュア デバイス プロビジョニング

Diwakar Dhyani,Frangline Jose, Keerthy J

概要

このアプリケーション ノートは、TDA4x/DRA8x プロセッサを HS-FS (High Security Field-Securable：高度セキュリティ対応、フィールドでのセキュア化可能) の動作モードから HS-SE (High Security Security-Enforced：高度セキュリティ対応、セキュリティ強化) の動作モードに変換するための包括的なガイドです。この資料では、TI のワンタイム プログラマブル (OTP) Keywriter ツールのアーキテクチャと、開発環境および本番環境向けの導入方法について説明しています。また、この資料では、TIFS (テキサス インスツルメンツの基本セキュリティ) API を使用したフィールド更新の手順についても説明します。エンジニアは、適切なセキュアな鍵のプロビジョニング戦略の選択方法、セキュアなデバイス プログラミング ワークフローの実装方法、OTP キーのプログラミング操作中に発生する一般的な問題のトラブルシューティング方法について学習します。

目次

1 はじめに....2
1.1 Keywriter が必要な理由....2
1.2 Keywriter とは....3
2 Keywriter 証明書 BLOB....4
2.1 証明書生成の概要....4
3 本番環境の導入方法....5
3.1 OSPI ブート方式....5
3.2 eMMC ブート方式....6
3.3 DFU バックアップ方式による eMMC ブート....7
3.4 JTAG / 開発ブート方式....7
3.5 RGMI イーサネット ブート モード....7
4 フィールドの更新....9
4.1 SWREV (ソフトウェア リビジョン)....9
4.2 KEYREV (キー リビジョン)....9
5 一般的なデバッグ手順....10
5.1 ドライ ラン テスト....10
5.2 パッケージ要件....12
5.3 eFuse プログラミングの設定....12
5.4 プログラミング結果の確認....12
5.5 ブート障害のトラブルシューティング....12
5.6 WAKEUP UART を使用しないトレース収集....12
6 まとめ....13
7 参考資料....13

商標

1 はじめに

この資料は、TI プロセッサ用のセキュアな鍵のプロビジョニングを実装する製造エンジニア、セキュリティ アーキテクト、製造チーム向けのものです。ここでは、Keywriter の導入戦略、アーキテクチャ コンポーネント、および本番環境のワークフローについて説明します。Keywriter のユーザーは、適切な展開方法を選択し、デバイスをフィールドでのセキュア化可能 (FS または HSFS) 状態からセキュリティ強化 (HS-SE) 状態に移行し、一般的なプロビジョニングの問題を解決するための実用的な知識を得られます。本書を読むことで、実際の本番環境で Keywriter の動作を実装、最適化、およびトラブルシューティングするために必要な理論的な理解と実用的な知識の両方を得ることができます。

1.1 Keywriter が必要な理由

TI のプロセッサの出荷時は、HS-FS (High Security Field-Securable：高度セキュリティ対応、フィールドでのセキュア化可能) モードになっています。この状態では、信頼ルート (RoT) キーがデバイスの OTP-eFuse にまだプログラムされていないため、セキュア ブートは強制的に適用されません。Keywriter ツールを使用すると、これらの信頼ルート キーと関連セキュリティ フィールドを eFuse にプログラムし、デバイスを HS-FS 動作モードから HS-SE 動作モードに遷移させることができます。デバイスが HS-SE 状態の場合、ハードウェアによってセキュア ブートが強制的に適用されます。

次のフィールドは OTP-eFuse アレイでプログラムできますが、HS-SE デバイスに遷移するためにすべてのフィールドが必須であるとは限りません。

必須フィールド

- SMPK HASH (セカンダリ メーカーの公鍵開ハッシュ)。
- SMEK (セカンダリ メーカーの暗号化鍵)。
- KEYCNT (キー カウント)。
- KEYREV (キー リビジョン)。

オプションのフィールド

- BMPK HASH (バックアップ メーカーの公開鍵ハッシュ)。
- BMEK (バックアップ メーカーの暗号化鍵)。
- SWREV、MSV、JTAG (ロールバック防止保護用のソフトウェア リビジョン、モデル固有の値、JTAG ディスエーブル)。
- 拡張 OTP フィールド。

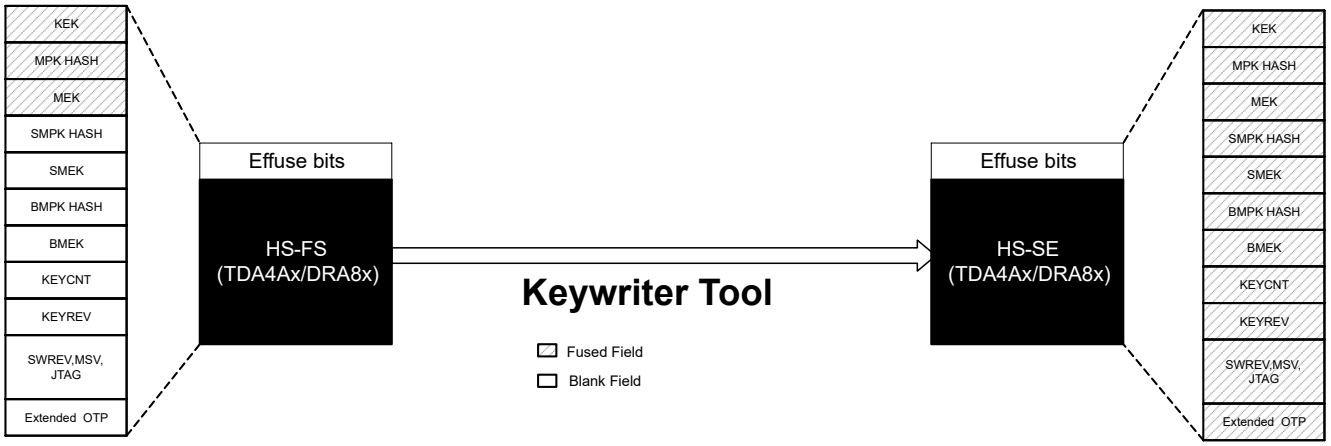

図 1-1. デバイスのライフサイクル：Keywriter eFuse プログラミング ワークフローによる HS-FS から HS-SE への遷移

1.2 Keywriter とは

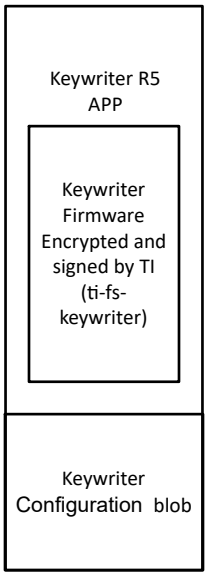

図 1-2. TIFS ファームウェア、R5 アプリケーション、および証明書 BLOB コンポーネントを示す Keywriter システム アーキテクチャ

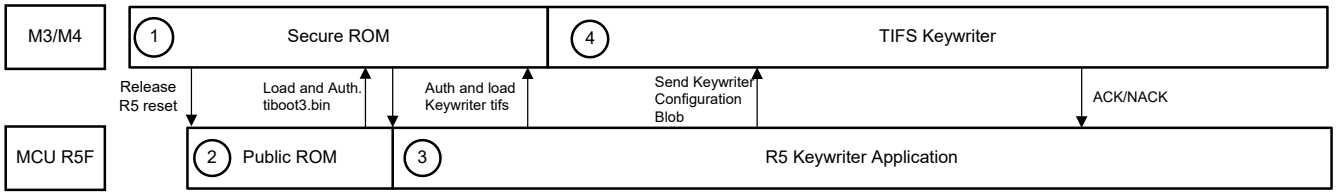

図 1-3. Keywriter ブート フロー

Keywriter ツールは、次の 3 つの主要コンポーネントで構成されています。

- **Keywriter ファームウェア**:これは、デバイスのセキュア サブシステム (M3/M4_0) 上で動作するセキュア OTP Keywriter ファームウェアです。x509 証明書の OTP 構成を検証し、デバイスの OTP-eFuse にデータをプログラミングする役割を果たします。
- **Keywriter 構成 BLOB**:OTP-eFuse 内でプログラミングされた eFuse 構成を含む証明書。
- **Keywriter R5 アプリケーション**:これは、公開 Keywriter アプリケーションです。他のブートローダーなしで、デバイスのプライマリ ブート コアで直接ブートされます。デバイスの OTP-eFuse にキーをプログラムするために、セキュリティ サブシステム (M3/M4) 上で動作するセキュア Keywriter コンポーネントに OTP 構成を送信する役割を果たします。

2 Keywriter 証明書 BLOB

2.1 証明書生成の概要

Keywriter 証明書 BLOB は、デジタル署名と暗号化を組み合わせた階層型セキュリティ アプローチを採用しています。OTP 構成データは、ランダムに生成された AES-256 セッション キーで暗号化されます。その後、このセッション キーは TI-FEK (TI フィールド暗号化鍵) を使用して暗号化され、対応する秘密鍵を持つ TI の正規のファームウェアでのみ復号できることを検証します。証明書全体は、認証に SMPK (セカンダリ メーカー秘密鍵) を使用して署名されます。オプションで、証明書は BMPK (バックアップ メーカー秘密鍵) を使用してキー ローテーション シナリオを有効にすることによって、二重署名することもできます。Keywriter BLOB 生成フローの詳細については、次の図を参照してください。

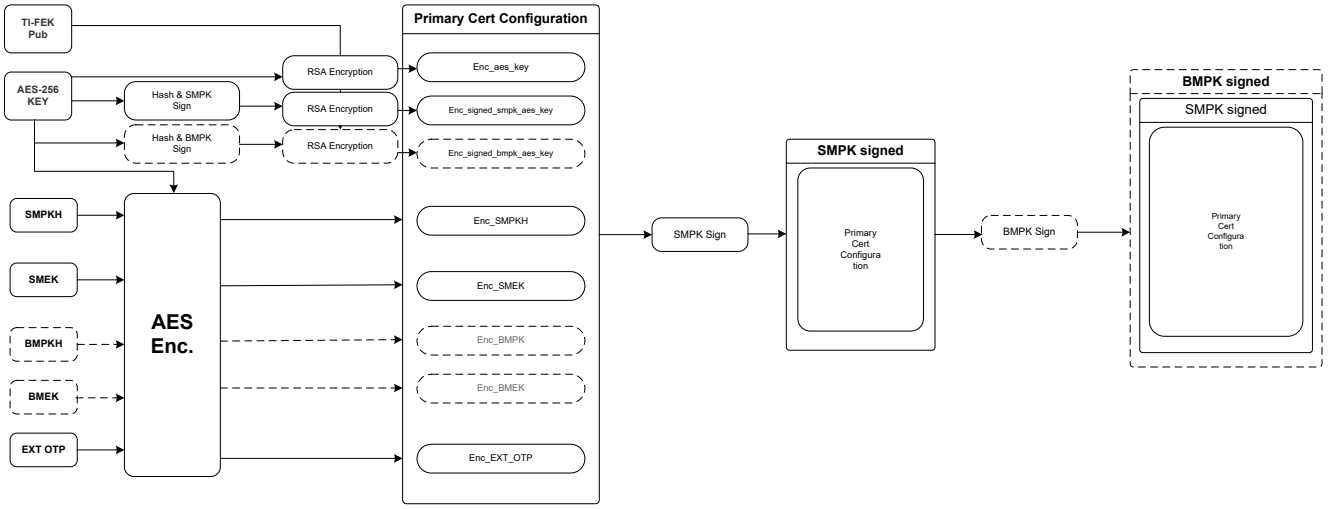

図 2-1. 証明書の BLOB 生成フロー：AES 暗号化、TI-FEK キー ラッピング、デュアル シグネチャ認証

3 本番環境の導入方法

複数の導入方法がありますが、その中で本番環境や要件に合ったものを使用できます。それぞれの方法には特定の実装手順があり、本番環境の設定とニーズに最適なアプローチを選択できます。この柔軟性により、事前定義された使用事例の制約を受けることなく、適切な方法を選択できます。

3.1 OSPI ブート方式

初期設定と鍵のプロビジョニング プロセス:

1. Keywriter イメージを、OSPI フラッシュ メモリのプライマリ ブート位置にフラッシュします。
2. HS-SE 署名済みアプリケーション イメージをバックアップ ブート位置にフラッシュします。
3. プロセッサ (MPU) の電源を投入します。ブート ROM は、OSPI フラッシュ内のプライマリ位置から Keywriter イメージを読み取ります。
4. デバイスは HS-FS (High Security-Field Securable) モードであるため、Keywriter アプリケーションが実行され、セキュリティ キーがデバイスにプログラムされます。

   セキュア ブートへの遷移:

   OTP-eFuse がプログラムされると、デバイスは次のパワー サイクルで HS-SE モードに移行します。顧客キーがプログラムされ、KEYREV、KEYCNT もプログラムされると、ハードウェアは HS-SE 状態に切り替わります。このプロセスは、HS-FS 状態には戻せません。
5. 次のパワーオン リセット時に、ROM ブートローダーでは強制的にセキュア ブートが実行されます。プライマリ ブート位置の Keywriter イメージは最初にロードされますが、OTP-eFuse に新たにプログラムされたお客様の信頼ルート キーで署名されていないため、認証に失敗します。
6. ROM ブートローダーのフォールバック メカニズムがアクティブになり、OSPI フラッシュのバックアップ ブート位置からアプリケーション イメージを読み取ります。eFuse に格納された公開鍵ハッシュを使用して、HS-SE 署名済みアプリケーション イメージを正常に認証します。
7. 顧客キー (SMPKH) で署名されたイメージを使用して、デバイスが正常に起動されます。
8. この段階では、OSPI フラッシュのプライマリ イメージを消去し、OSPI フラッシュのセカンダリ パーティションからロードされたものと同じ、顧客キーで署名されたアプリケーション イメージのコピーを使用して再プログラムすることをお勧めします。

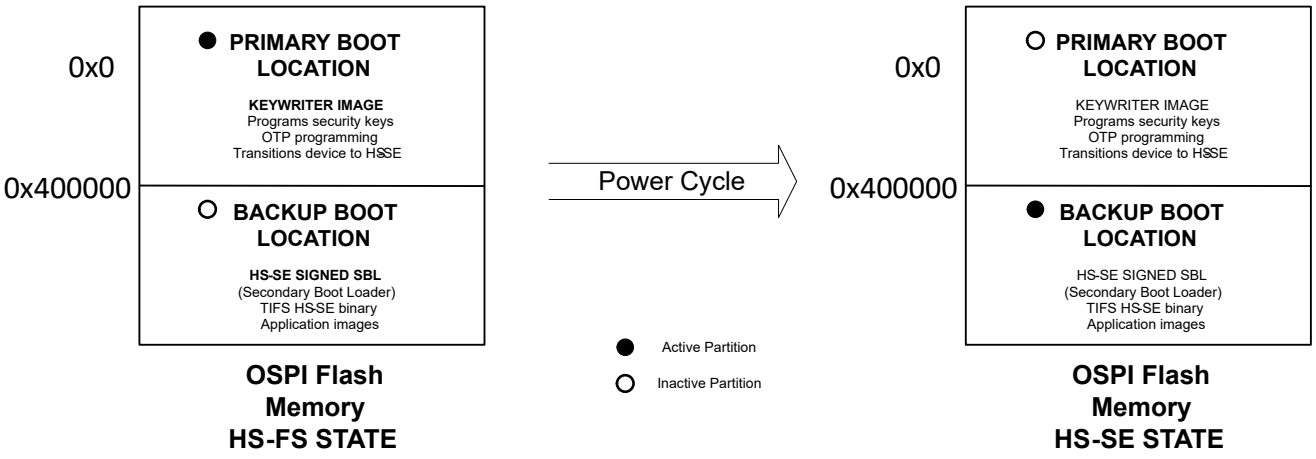

図 3-1. OSPI ブート方式

3.2 eMMC ブート方式

初期設定と鍵のプロビジョニング プロセス:

1. Keywriter イメージを eMMC デバイスの Boot-0 パーティションにフラッシュします。
2. HS-SE 署名済みアプリケーション イメージを eMMC デバイスの Boot-1 パーティションにフラッシュします。
3. eMMC ブートモード用のブート モード ピンを構成し、デバイスの電源をオンにします。
4. ROM ブートローダーが PARTITION_CONFIG レジスタを読み出し (デフォルト:Boot-0 アクティブ)、デバイスが HS-FS 状態であることから、ROM は Keywriter を認証して、実行します。
5. Keywriter が鍵のプロビジョニングを完了すると、keywriter アプリケーションは eMMC の EXT_CSD レジスタの PARTITION_CONFIG フィールドを更新して、アクティブなブート パーティションを Boot-0 から Boot-1 に変更します。リファレンス実装については、FAQ (よくある質問) を参照してください。

セキュア ブートへの遷移:

6. ボードの電源を入れ直すと、ROM ブートローダーは (eMMC ブート モード用に構成されている) ブート モード ピンをチェックし、PARTITION_CONFIG フィールドを読み取ってどのパーティションがアクティブであるかを確認します。
7. 前の手順でアクティブ パーティションが Boot-1 に変更されたため、ROM は Boot-1 からイメージをフェッチします。
8. このデバイスは、HS-SE 署名済みの本番環境イメージを使用して、Boot-1 から正常に起動されます
9. この段階では、eMMC フラッシュのプライマリ イメージを消去し、eMMC フラッシュのセカンダリ パーティションからロードされたものと同じ顧客キーで署名されたアプリケーション イメージのコピーを使用して、再プログラムすることをお勧めします。

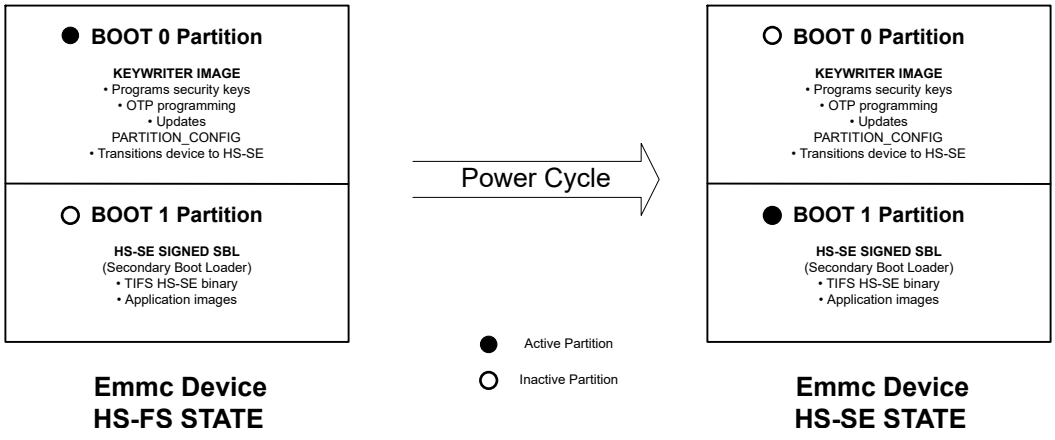

図 3-2. eMMC ブート モード

3.3 DFU バックアップ方式による eMMC ブート

初期設定と鍵のプロビジョニング プロセス:

1. プライマリ ブート メディアを EMMC に設定し、バックアップを DFU に設定します。
2. デバイスの電源を入れます。イメージを eMMC にフラッシュしていないため、ブート ROM はプライマリ ブート メディアからイメージを読み取ることができず、バックアップ ブート モードに戻ります。
3. ブート ROM が DFU ブートを開始します。DFU インターフェイスを使用して Keywriter イメージをロードします。
4. Keywriter バイナリは、OTP にキーをプログラムします。

セキュア ブートへの遷移:

5. ボードの電源を入れ直し、DFU インターフェイス経由で HS-SE 署名済みイメージを使用して U-Boot までブートします。『TDA4 フラッシュ技法』のセクション「5 eMMC フラッシュ」の説明に従って、DFU インターフェイスを使用して HS-SE 署名済みイメージを eMMC にフラッシュします。
6. eMMC がプライマリ ブート メディアであり、有効な署名済みブート イメージがあるため、ボードの電源を入れ直します。ROM は、eMMC からアプリケーション イメージを読み取り、認証してブートします。

注

バックアップ ブート位置は、バックアップ ブート モードとは異なります。ブート ROM は、プライマリ オフセットで有効なイメージが見つからない場合、バックアップ OSPI オフセットをサポートし、同じ OSPI フラッシュ内のバックアップ オフセットに自動的にジャンプします。バックアップ ブート モードは、完全に異なるブート メディアへのハードウェア構成されたフォールバックですが、物理的な BOOTMODE ピンによって決定されます。

3.4 JTAG / 開発ブート方式

初期設定:

1. 適切なブート ピン (BOOTMODE[4] = 0) を設定して、デバイスを開発ブート (DEV BOOT) モードに構成します。
2. DEV BOOT 中、デバイスは MCU R5 からのファームウェア ロード完了メッセージを待機します。
3. JTAG エミュレータ (XDS110、XDS200、または XDS560) をマイクロプロセッサ / MPU に接続します。

プロビジョニング プロセス:

4. MCU Cortex R5F コア 0 に接続します。R5F コアをクリーン状態にリセットします。
5. Keywriter アプリケーションを MCU R5 コアにロードします。
6. Keywriter 証明書 BLOB を指定されたメモリ位置にロードします (たとえば、TDA4AL の場合は 0x41C7F004。KeyWriter 証明書は keywr_end + 1 から開始されます。これは「.keywr_bin_end」セクションに配置されています。マップ ファイルまたはリンカ コマンド ファイルを参照してください)。
7. MCU R5 で Keywriter プログラムの実行を開始します。
8. Keywriter ファームウェアにより、証明書の検証、鍵の復号、OTP-eFuse のプログラムが開始され、デバイスが HS-FS から HS-SE に遷移します。

このリンクの FAQ (よくある質問) には、この方式の詳細が記載されています。この FAQ には、CCS スクリプト シェルで直接使用できる Java スクリプトも記載されています。

3.5 RGMI イーサネット ブート モード

初期設定と鍵のプロビジョニング プロセス:

1. RGMI ブート モードでデバイスを構成します。
2. ホスト PC でマイクロプロセッサの静的 IP を設定し、リンクを 100M 全二重に構成します。
3. ホスト PC で tftpd32 アプリケーションをダウンロードし、tftpd32 アプリケーションを使用して DHCP サーバーを設定します。
4. イーサネット ケーブルを使用して、PC のイーサネット ポートをデバイスの CPSW ポートに接続します。

詳細については、FAQ (よくある質問) を参照してください。

注

イーサネット ブート モードは、TDA4VH クラスのデバイスでのみサポートされています

4 フィールドの更新

4.1 SWREV (ソフトウェア リビジョン)

OTP-eFuse の SWREV フィールドは、ソフトウェアのロールバック保護を実装するためのフィールドで、攻撃者がプライマリ アプリケーション ファームウェアを脆弱性が確認されている古いバージョンにダウングレードすることを防止するために使用します。デバイスがブートすると、TIFS/BOOT ROM コードにより、ブート イメージ証明書に組み込まれているソフトウェア リビジョン値が、OTP-eFuse にプログラムされた SWREV 値と比較されます。証明書のソフトウェア リビジョンが OTP SWREV 値よりも小さい場合、ブート プロセスは拒否されるため、SWREV リビジョン番号以上の認証とロードが保証されます。

SWREV を更新できるコンポーネントは次のとおりです。

1. ソフトウェア リビジョン SBL。
2. ソフトウェア リビジョン SYSFW/TIFS。
3. ソフトウェア リビジョン セキュア ボード構成。

更新手順:

- HS-SE TIFS には、実行時に TISCI_MSG_WRITE_SWREV コマンドを使用して、SWREV の値を更新する機能があります。
- eFuse の書き込み動作には、VPP_MCU が必要です。

注

必要な場合以外は、TIFS/SYSFW の SWREV を変更しないことをお勧めします。存在しない不正な TIFS のソフトウェア リビジョンを使用すると、デバイスにブリックが発生する可能性があります。

4.2 KEYREV (キー リビジョン)

OTP-eFuse の KEYREV フィールドは、セキュア ブート動作に使用する信頼ルート キーを制御します。

- KEYREV = 1 : セカンダリ メーカーの鍵セット (SMPK/SMEK) を有効にします
- KEYREV = 2 : バックアップ メーカーの鍵セット (BMPK/BMEK) を有効にします

更新手順:

- KEYREV は、HS-SE デバイス上の実行時に TISCI_MSG_WRITE_KEYREV コマンドを使用して更新できます
- KEYREV を更新するには、次の 2 つの構成要素、つまり SMPK 秘密鍵で署名されたプライマリ証明書と BMPK 秘密鍵で署名されたセカンダリ証明書を使用した二重署名証明書が必要です
- TISCI_MSG_WRITE_KEYREV メッセージは、write_host のセキュア ボード構成に記述されているホストによってのみ送信できます

参考として、TIFS API を使用した KEYREV の更新に関する FAQ (よくある質問) を参照してください。

5 一般的なデバッグ手順

このセクションでは、鍵のプロビジョニング中に発生する問題を診断して解決するための体系的なアプローチについて説明します。効率的なトラブルシューティングの手順に従ってください。

5.1 ドライ ラン テスト

- Keywriter アプリケーション内の VPP 対応ロジックを削除 / コメントアウトし、Keywriter ツールをビルドして実行します。
- VPP が無効の場合、eFuse のプログラミングは失敗します。
- 検証の解析と、実際の eFuse の書き込みに失敗した場合の Keywriter TIFS からの障害メッセージについては、TIFS (WAKEUP UART) トレースを参照してください。

以下のコードは、J721S2 評価基板上の WKUP UART と MCU UART からのリファレンス ログを示しています。証明書 BLOB の生成に使用されるコマンドは、次のとおりです。

```
./gen_keywr_cert.sh -t ti_fek_public.pem
-s keys/smpk.pem --smek keys/smek.key -s-rp --smek-rp
-b keys/bmpk.pem --bmek keys/bmek.key -b-rp --bmek-rp
--mek-opt 0x1 --mpk-opt 0x21
--keycnt 2 --keycnt-wp
--keyrev 1
```

表 5-1.

MCU UART (R5 アプリケーション)	WKUP UART (Keywriter TIFS)
OTP Keywriter Version: 02.00.00.00 (May 20 2026 - 16:02:05) *OTP Keywriter ver: 10.1.4-v10.01.04a-1-gf7c27 (Fie* *Legacy Boot Mode* *Key programming sequence initiated* *Taking OTP certificate from 0x41c70004* *Sciclient_otpProcessKeyCfg returns: -1* *Debug response: 0x800* *Key programming sequence completed*	# Decrypting extensions.. # MPK Options: *0x21* *MEK Options: 0x1* *MPK Opt P1: 0x1* *MPK Opt P2: 0x1* *MEK Opt: 0x1* ** SMPKH Part 1 BCH code: 61b45b59* ** SMPKH Part 2 BCH code: 417d4cb4* ** SMPK Hash (part-1,2):1f6002b07cd9b0b7c47d9ca8d1aae57b8e8784a12f636b2b760d7d98a18f189701 60dfd0f23e2b0cb10ec7edc7c6edac3d9bdfefe0eddc3fff7fe9ad875195527d01* SMEK BCH code: 21224fcc* ** SMEK Hash: 92785809a3dfefea57f6bbed642d730ba5d05e601222a72e815bf01ceb3a50f96ab85d282425f684436fabd4c7da624b791da411615035314103cc64e611f532* ** BMPKH Part 1 BCH code: 81a4977b* ** BMPKH Part 2 BCH code: 613bf931* ** BMPK Hash (part-1,2): b41865771ef3e13933568b1d1d50a72f5037a2a103c4535c713e10a61df1609c01 f6033117fbb56d5f190a17598e151141c6e413e2d853e56aae7997820639909b01* ** BMEK BCH code: 21ec5cec* ** BMEK Hash: 21a5ae74b64b1c9fbd1f9201e60a319fe417edf06c2f9ba71fa3e1a967d171b684a4f85a83b8a4821abffe8e322b3697e299ffa94dec921bd4ac99f8ca59dd6c* *EXT OTP extension programming disabled* *MSV extension programming disabled* ** KEY CNT: 03030000* ** KEY REV: 01010000* *SWREV extension programming disabled* *FW CFG REV extension programming disabled* ** KEYWR VERSION: 0x20000* # *# Programming Keys..* # * MSV: [u32] bch + msv:*0x8BAC0FFE* *MSV extension programming disabled* *[u32] bch + msv: 0x8BAC0FFE* ** SWREV:* [u32] SWREV-SBL:*0x1* *[u32] SWREV-SYSFW: 0x1* *SWREV extension programming disabled* *[u32] SWREV-SBL: 0x1* *[u32] SWREV-SYSFW: 0x1* ** FW CFG REV:* [u32] SWREV-FW-CFG-REV:*0x0* *SWREV SEC BCFG extension programming disabled* *[u32] SWREV-FW-CFG-REV: 0x0* ** EXT OTP:* EXT OTP extension programming disabled * BMPKH, BMEK: Error in programming BMPKH part 1 debug_response: *0x800*

5.2 パッケージ要件

最新の keywriter アドオン パッケージが使用されていること、セキュア リソースから利用できることを確認します。

5.3 eFuse プログラミングの設定

VPP (プログラミング電圧) は、鍵のプロビジョニング プロセス全体を通じて安定している必要があります。

5.4 プログラミング結果の確認

- eFuse のプログラミングが成功すると、デバッグ応答 0x0 が返されます。MCU UART で R5 Keywriter アプリケーション ログを取得します。次のリファレンス ログに、鍵のプロビジョニングの成功例を示します。

```
OTP Keywriter Version: 02.00.00.00 (Jul 19 2023 - 11:15:35)
OTP Keywriter ver: 9.0.7-v09.00.07 (Kool Koala)
OTP_VppEn
WKUP_GPIO0_54 output high
Key programming sequence initiated
Taking OTP certificate from 0x41c7f004
Debug response: 0x0
Key programming sequence completed
```

- 応答がゼロ以外の場合は、エラーを示しています。特定のエラー コードの詳細については、TISCI の資料を参照してください。

5.5 ブート障害のトラブルシューティング

eFuse プログラミング後にデバイスがブートに失敗した場合:

- ブート イメージの署名に使用する鍵ハッシュが、OTP-eFuse にプログラムされたハッシュと一致していることを確認します。
- デバイスから公開鍵ハッシュを抽出し、Sec Cust MPK Hash (有効な信頼ルート鍵ハッシュ) と比較します。

  デバイスから公開鍵ハッシュを抽出する方法については、FAQ (よくある質問) を参照してください。

5.6 WAKEUP UART を使用しないトレース収集

- デバイス上で WAKEUP UART が使用できない場合は、トレース バッファを使用して TIFS トレースをキャプチャします。

6 まとめ

このアプリケーション ノートには、本番環境でセキュアなデバイス プロビジョニングを行うための Keywriter 展開戦略の包括的な概要を記載しています。OSPI ブートを使用する車載および産業用アプリケーションから、eMMC ブートを使用する大量のコンシューマ エレクトロニクスまで、さまざまな生産シナリオに対応できるよう、複数の導入方法が用意されています。

エンジニアは、個別の生産要件、規模、インフラストラクチャに応じて、適切な導入方法を選択する必要があります。実際の環境で Keywriter の実装と動作時のトラブルシューティングを問題なく行うには、証明書の生成、検証プロセス、および一般的なデバッグ手順を正しく理解しておく必要があります。

7 参考資料

1. テキサス インスツルメンツ、『TISCI Key Writer ユーザー ガイド』、ユーザーズ ガイド。
2. テキサス インスツルメンツ、『Jacinto7 高セキュリティ デバイスの開発』、アプリケーション ノート。
3. テキサス インスツルメンツ、『『PDK Keywriter ユーザー ガイド』ユーザーズ ガイド。

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

最終更新日：2025 年 10 月