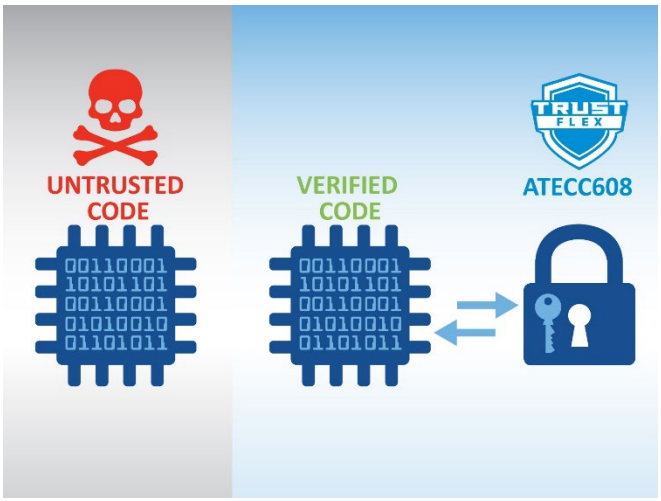
セキュア エレメントを使ったファームウェア検証の代表的な 3 つのユースケース (パート 2)

このシリーズのパート 1 は、アプリケーションのコードにファームウェア検証機能を実装する事の重要性について解説しました。 パート 2 では、パート 1 で述べた課題を克服する方法について解説します。

Microchip 社のマイクロコントローラとセキュア エレメントの併用による効果

セキュアブート時とセキュア ファームウェア更新時にファームウェア検証を実行するアプリケーションでは、マイクロコントローラ内部の改変不可能なブートメモリ領域が非常に重要です。最新の PIC24F 低消費電力マイクロコントローラ(MCU)と高性能な dsPIC33 デジタルシグナル コントローラ(DSC)は ICSP™ Write Inhibit によるフラッシュ OTP 機能と CodeGuard™セキュリティ機能を備えるため、改変不可能なセキュアブートを実装できます。ICSP Write Inhibit はアクセス制限機能であり、読み書き保護機能との併用により、全てのフラッシュメモリに対する外部プログラマ/デバッガからのアクセスを制限できます。ICSP Write Inhibit を有効にすると ICSP フラッシュ書き込み/消去動作は禁止され、フラッシュは OTP (One-Time-Programmable)メモリであるかのように動作します。CodeGuard セキュリティ機能は、フラッシュ プログラムメモリをアクセス権限の異なる 2 つの領域に分割します。第 1 のメモリ領域はブートセグメントであり、ファームウェア検証ルーチンを実装するブートローダ コードを保持します。このブートセグメントは、このセグメント自身およびメモリのアプリケーション領域内からの全ての自己書き込みから保護できます。CodeGuard セキュリティ機能と ICSP Write Inhibit によるフラッシュ OTP 機能の組み合わせにより、ブートローダは改変不可能となります。パート 1 で述べた通り、「改変不可能」とは「アクセス不可能」または「読み出し不可能」を意味するのではなく、単にデジタル攻撃等による「書き換えが不可能」である事を意味します。第 2 のメモリ領域は汎用セグメントであり、アプリケーション イメージを保持します。



CodeGuard セキュリティ機能のブートセグメント内に実装されたファームウェア検証コードは改変不可能であり、CryptoAuthlib ライブラリを使って ECDSA 検証コマンドをセキュア エレメントに対して発行します。 Microchip 社の CryptoAuthentication ファミリ向け Trust Platform からの TrustFLEX ATECC608 セキュア エレメントは、セキュアな鍵ストレージとハードウェア ベースの暗号アクセラレータを提供し、ECDSA 検証動作を高速

に(ms レベルで)実行可能です。 また、コード署名およびコード ダイジェストに対してこの暗号関数を実行する際に使う公開鍵をセキュアに保護します。 公開鍵はセキュア エレメントの改変不可能なセキュアメモリ領域に保存され、耐タンパーおよび耐サイドチャンネル攻撃防御により物理的に保護されます。この方法により、鍵とコードに対するアクセスと改ざんが大幅に困難となります。 Microchip 社は PIC24F MCU および dsPIC33 DSC 向けのソフトウェア ライブラリを提供しています。これらは無償のグラフィカル プログラミング環境 MPLAB® Code Configurator (MCC),内でサポートされ、シームレスで明解な C コードを生成してプロジェクトに挿入する事ができます。 MCC 内の CryptoAuthlib ライブラリを使う事で、TrustFLEX ATECC608 と PIC24/dsPIC33 デバイスを容易に連携させて各種セキュア機能を実装できます。 さらに MCC の 16 ビット ブートローダ ライブラリにより、ファームウェア検証機能とセキュア ファームウェア更新機能を追加できます。 詳細と実装方法は、組込みセキュリティ デザインセンターとセキュアブートと OTA をご覧ください。

MCU と ATECC608 TrustFLEX の間のトランザクション

Trust Platform Design Suite (TPDS)ソフトウェア ツールを使うと、Microchip 社製セキュア エレメントを容易に使い始める事ができます。 TPDS を開いてユースケースと開発プラットフォームを選択すると、そのユースケースに応じたトランザクション図が表示されます。

例として、従来型マイクロコントローラと TrustFLEX ATECC608 の間のファームウェア検証向けトランザクションを図 1 に示します。この図は以下のトランザクションを示しています。
- ホスト MCU はファームウェアのダイジェストを計算し、そのダイジェストをセキュア エレメントへチャレンジとして送信する。
- セキュア エレメントはプロビジョニング済みの公開鍵を使ってダイジェストと署名を暗号処理により検証し、レスポンスをホスト MCU へ送信する。

この図には、ホスト MCU がセキュア エレメントと通信するために使う CryptoAuthLib API 呼び出し(atcab_secureboot_MAC 等)と、MCU とセキュア エレメントの間で発生する動作の簡単な説明も記載しています。

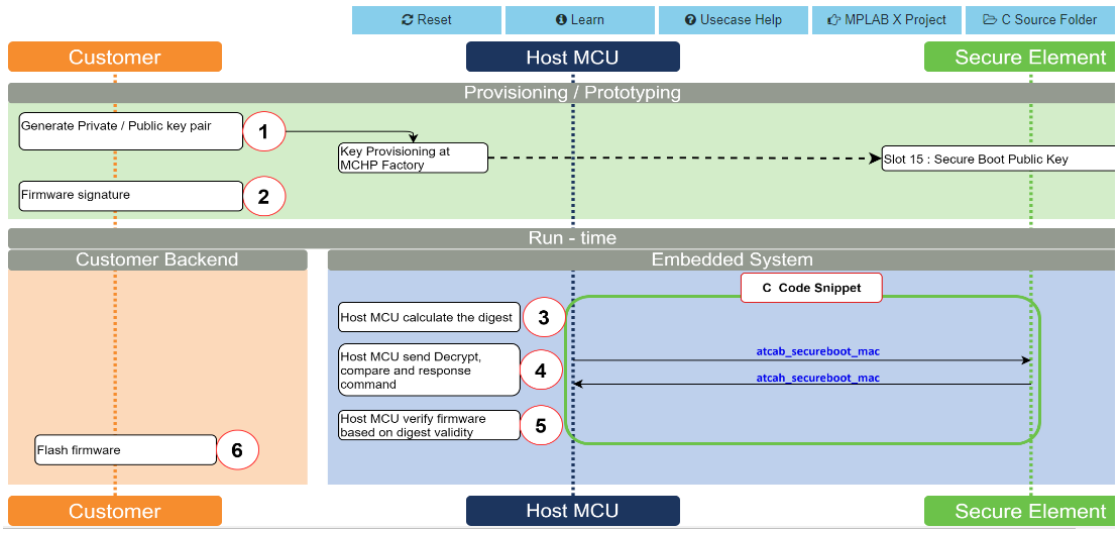


図 1: Trust Platform Design Suite V2 のトランザクション図
(TrustFLEX ATECC608 を使ったファームウェア検証の例)

この図は、以下の 2 つの状況でのトランザクションを示しています。

- プロトタイピング フェイズ:
  - TPDS ツールは、非対称鍵ペアをプロトタイピング用に生成します。
  - TPDS ツールは、ユーザのコンピュータにより、公開鍵をプロトタイピング用にプロビジョニングします。量産向けには、Microchip 社のセキュア鍵プロビジョニング サービスが利用できます。
  - 非公開鍵を使ってコードイメージに署名する事で、コードイメージの署名を生成します。
- ブート時:
  - ホスト MCU はダイジェストを計算します。
  - ホスト MCU はダイジェストと署名をセキュア エレメントへ送信します。
  - セキュア エレメントはそれらを検証し、レスポンス(有効/無効)を返します。
  - ホスト MCU は、受信したレスポンスが有効であれば、そのファームウェアを認証して実行します。

BootROM 機能を備えていないマイクロコントローラでセキュア エレメントを使う場合の効果

以下では、BootROM を備えない従来型の 32 ビット マイクロコントローラ(例: SAMD21 ARM® Cortex® M0+)を使った設計を想定します。これは今日使われている設計の大半の状況でもありますが、新たなマイクロコントローラの採用は再認定のコストと時間の問題から簡単ではない場合もあります。

その場合、マイクロコントローラ内のコードが不正に変更される可能性があると考える必要があります。以下では、従来型マイクロコントローラからセキュア エレメントへ送信される ECDSA 検証コマンドが変更された場合に何が起こるか考えます。コードは署名済みで、マイクロコントローラ内部には検証の鍵が存在しない場合、その被害は攻撃を受けた 1 つの製品だけで済みます(公開鍵基盤を使用していると想定)。これは、セキュア エレ

メントが公開鍵を隔離して保管するためです。これにより、被害が他の製品に波及する事を防ぐ事ができます。

従来型マイクロコントローラとセキュア エレメントを併用する場合、以下の点を考慮する必要があります。

- 1 つの製品に物理的にアクセスし、その 1 つの製品を攻撃するためだけにコードをリバース エンジニアリングする事に価値はあるでしょうか? ほとんどの場合、答えは「否」でしょう。市場に出回っている多数の製品に攻撃を加えようとしても、各製品に 1 つずつ物理的にアクセスする必要があるのでは、現実的ではありません。純正の署名済みコードを検証するために 1 回の OTA 更新が必要であり、そのコードに鍵は含まれていないと想定される場合、リモート攻撃を試みる価値はないでしょう。
- 加えて、コード検証用の暗号鍵を扱う委託業者が信頼できない場合、Microchip 社のセキュア鍵プロビジョニング サービスを利用する事で、セキュア エレメントの価値は 2 倍に高まります。このサービスは、Microchip 社が自社のセキュアな工場内で鍵をデバイスに書き込むため、委託業者への鍵暴露のリスクを排除できます。委託業者を変更する場合も、鍵がセキュア エレメント内に保管されていれば、Microchip 社に配送先住所の変更を依頼するだけで済みます。
- 最後に、財務リスクレベルに対する損益(P&L)の影響を考慮する事が非常に重要です。弊社のセキュア エレメントは、鍵をセキュアに保管できる非常に費用対効果の高いソリューションであるため、多くのお客様に使われています。マイクロコントローラまたはマイクロプロセッサに鍵を書き込んだ時点でそれらのデバイスはキャンセルも返品もできなくなり、約 3～5 ドルするデバイスの在庫を抱える事になります。セキュア エレメントはこのような財務リスクを大幅に軽減します。

サプライチェーン管理と、検証用鍵を MCU 内部でセキュア プロビジョニングするコスト

一般的にマイクロプロセッサはマイクロコントローラよりも大幅にセキュリティ機能が豊富ですが、どちらを使っても鍵と暗号アルゴリズムの両方を格納するセキュアな物理境界が必要です。しかし、これを低コストに達成する事は困難です。なぜなら、セキュア境界はプロセッサまたはコントローラのダイコストに大きく響くからです。シングルチップ統合のコストだけがコスト増の要因ではありません。そのようなアーキテクチャに対する完全な技術サポート、複数利用者権限を扱うための各種ツール、鍵をセキュアにプロビジョニングするためのサプライチェーン管理等により、複雑さとリスクはどんどん増加します。

100,000 個のマイクロプロセッサの全てに鍵を書き込んだ場合の損益について考えてみましょう。鍵を書き込んだ時点で、それらのデバイスは返品不可の在庫となります。販売業者であろうと OEM であろうと、在庫費用によって損益は悪化します。また、プロジェクトに遅延が発生した場合、時間が経つにつれてプロジェクト リスクは増加します。

このようなリスクを軽減するため、より柔軟なアーキテクチャとして Microchip 社の CryptoAuthentication™コンパニオン デバイス(ATECC608 TrustFLEX 等)が利用できます。鍵は対費用効果の高いセキュア鍵ストレージ内で隔離されます。また、本デバイスは世界中のほとんどどこにでも出荷可能です(EAR99 輸出規則に従う)。Microchip 社のセキュア

鍵プロビジョニング サービスを利用する事で、鍵がサプライチェーンの中間業者へ暴露されるリスクを排除でき、攻撃表面を削減できます。

まとめ

セキュリティの基礎として、1) コード検証(セキュアブート時、実行時、OTA 更新後)、2) サプライチェーンでの各種リスク、3) 暗号鍵を扱うためのコストを考慮する必要があります。改変不可能なブート機能を内蔵した dsPIC33 DSC または PIC24F MCU とセキュア エレメント(ATECC608 等)の組み合わせは、重要な根源でのソリューションを提供します。また、bootROM 機能を備えていないマイクロコントローラであっても、セキュア エレメントを追加して公開鍵を物理的に隔離する事で、脅威モデルとリスク評価の観点からほとんどのアプリケーションにおいて有効な効果が得られると考えられます。

参考資料

- ATECC608 TrustFLEX
- dsPIC33 DSC および PIC24 MCU による組込みセキュリティ ソリューション
- PIC24 MCU および dsPIC33 DSC 向けの 16 ビット ブートローラ

ツール入門:

- Trust Platform Design Suite

Xavier Bignalet, Dec 20, 2021