Jacinto™ 7 プロセッサのセキュリティ・イネーブラ

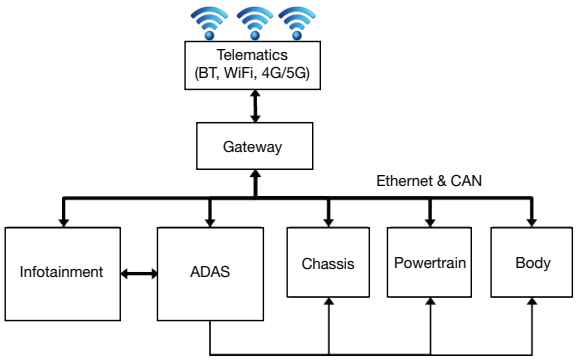
Steve Reis
Systems Applications & Architecture
(システム・アプリケーションとアーキテクチャ担当)
Jacinto processors
(Jacinto プロセッサ部門)
Texas Instruments
(テキサス・インスツルメンツ)

これまで以上に強力な組込みプロセッサとシステム・オン・チップ (SoC) ソリューションを使用すると、設計者の皆様は、より機能が豊富で、より強力なシステムを製作できます。現在、大半の組込みシステムには、有線とワイヤレス両方のコネクティビティが要件として課されています。これらを活用すると、リモート制御や管理の各機能とのインターフェイスを確立できるほか、ファクトリ (工場)、自動車、家庭などの、より複雑で高性能な各種システムへの統合が可能になります。

加えて、機能の追加や誤りの修正を目的とするリモート更新機能も、標準的な要件となっています。この場合、これらの機能を使用するには、より高いセキュリティが必要になります。この種のシステムが他のシステムに取り込まれる、悪用される、または安全ではない状態に置かれる、などの事態を防ぐことが目的です。

図 1 に、シャーシ、パワートレイン、ボディの各システムで構成されている 1 台の自動車システムを示します。ほかに、インフォテインメント・システムと先進運転支援システム (ADAS) も搭載しており、これらはすべてネットワーク・ゲートウェイ経由で互いにネットワーク接続されています。このゲートウェイにより、各電子制御ユニット間でデータを共有することができます。ADAS では、標準的な車載組込みシステムを使用して、いくつかの自動車移動機能を自動的に制御することができます。セルフ・パーキング (自動駐車)、車線逸脱防止支援、他の自動運転機能などです。自動車はテレマティクス・ゲートウェイを通じてクラウドにアクセスし、ソフトウェアの更新や他のデータを利用することができます。

この場合の各種外部インターフェイス、特にワイヤレス・インターフェイスは、リモート・アクセスに関する脆弱性 (弱点) を抱えています。この脆弱性に加え、これらのシステムがますますネットワークへの依存度を高めているという傾向により、何らかのセキュリティ侵害が発生した場合には、広い範囲に影響が及ぶことが想定されます。対策として、高い水準の保護機能を実現することが不可欠です。



Example interconnect vehicle architecture with wireless connectivity

図 1.相互接続された自動車のアーキテクチャ。

このホワイト・ペーパーでは、TDA4x と DRA8x の各プロセッサで構成されている Jacinto 7 プロセッサ・デバイス・ファミリについて説明するほか、TI の Jacinto 7 SoC ファミリのセキュリティ機能の概要について紹介します。システム設計者の皆様は、これらの機能を活用することで、セキュリティの目標を達成しやすくなります。このホワイト・ペーパーでは、これらの機能全般をセキュリティ・イネーブラ (セキュリティ実現機能) と呼びます。セキュリティ・イネーブラの詳細は、TI.com/security でご確認になれます。

セキュリティ・フレームワーク

アプリケーション・レベルを出発点として、各種セキュリティ対策を実装することで、さまざまな資産を脅威から保護できるようになります。半導体レベルで考えると、保護を必要とするシステム内の主な資産とは、データ、コード、デバイス ID、鍵です。システム内に存在する露出ポイント (「攻撃対象領域」とも呼びます) は、アプリケーションの各部分、またシステムの寿命全体にわたる各時点や各動作において、資産の脆弱性を高める可能性があります。

保護を必要とする資産や、存在する複数の露出ポイントについて考慮しながら、すべての適切なセキュリティ・イネーブラについて検討し、デバイス・レベルでセキュリティ機能を選定して適切な保護策を策定する必要があります。**図 2** に、セキュリティ・フレームワークの例を示します。

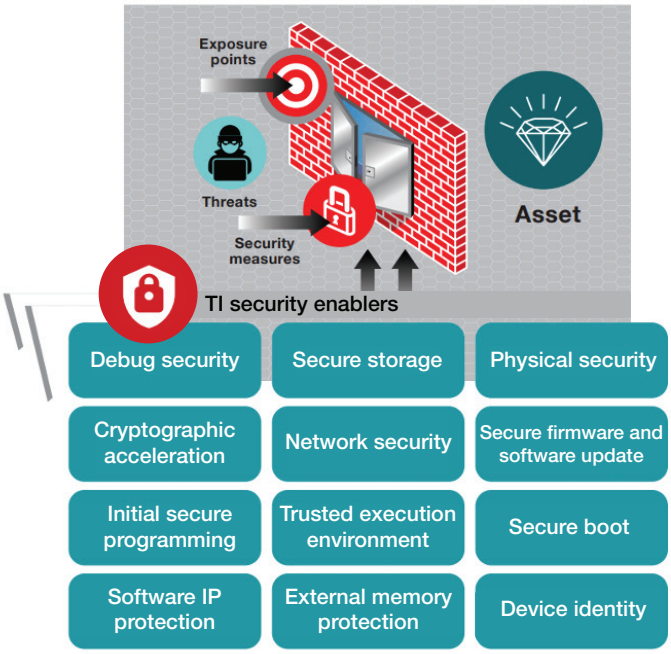


図 2.セキュリティ・フレームワーク。

Jacinto 7 SoC ファミリは多くのセキュリティ・イネーブラをサポートしているので、開発ユーザーの皆様は強力なセキュリティ対策を実装し、開発中システムに合わせてカスタマイズすることで、潜在的な脅威に対抗し、以下のようなシステムの露出ポイントを経由するアクセスを制限または防止できるようになります。

- デバイス ID (一意の ID)。
- セキュア・ブート (信頼済みルートの公開鍵)。
- 初期セキュア・プログラミング。
- 暗号化アクセラレーション機能。
- 外部メモリ保護 (ファイアウォール)。
- デバッグのセキュリティ (パスワードを使用する JTAG [Joint Test Action Group] ロック)。
- ソフトウェアの知的財産 (IP) の保護 (デバッグ・ロックアウト)。

TI の基本的なセキュリティ・プロセッサとファームウェア

Jacinto 7 SoC のセキュリティ・イネーブラの中心になるのは、専用の Arm® Cortex®-M プロセッサとセキュア RAM (ランダム・アクセス・メモリ) であり、これらは基本的なセキュリティ機能を実現するファームウェアをホストします。これらの機能には、セキュア・ブートに加え、eFuse による鍵管理、デバイスのファームウェア管理、JTAG アクセス 承認、ファームウェア・ロールバック (以前のバージョンに巻き戻し) 保護といった各種セキュリティ機能が含まれます。デバイスのバリエーションによっては、さらに付加的な機能もサポートされています。

デバイスの ID、鍵、セキュア・ブート

Jacinto 7 プラットフォームのセキュリティ・イネーブラのうち重要な要素の 1 つは、セキュア・ブートとセキュア信頼ルートの鍵に対するサポートです。これらの機能を組み合わせると、ブート・プロセスをセキュア化し、信頼できないソフトウェアをロードして実行してしまう事態を防止できます。
このセキュリティ・アンカーは、セキュア信頼ルート、言い換えると Jacinto 7 SoC に組み込まれている一連の鍵を土台として構築されています。これらの鍵は、公開鍵および秘密鍵という非対称型のペア (2 個 1 組)、1 個の共有秘密鍵、さらに 1 個のデバイス固有秘密鍵という組み合わせで構成されています。このうち公開鍵は、ハードウェア製造フローの一環として、ヒューズ溶断の形で、ワンタイム・プログラマブル (1 回限り書き込み可能、OTP) eFuse メモリに書き込まれます。システムの起動時には、初期ソフトウェア・イメージをデバイス・セキュリティ用初期構成コンポーネントとともに使用しますが、その際にこの公開鍵を使用して、デジタル証明書と、ソフトウェアに組み込まれているシグネチャ (署名) を検証することで、初期ソフトウェア・イメージを認証します。このプロセスを拡張し、付加的な鍵を通じて信頼を確立すること、および付加的なソフトウェア・コンポーネントを認証する方法で信頼チェーン (信

頼の連鎖) を拡張することもできます。これらのソフトウェア・コンポーネントに該当するのは、付加的なブートローダーや、Jacinto 7 SoC が搭載している複数のコアに対応するオペレーティング・システムのカーネルなどです。

システム・メーカーの皆様は、ルートとなる一連の鍵をセキュア・コンピューティング環境内で維持し、システムの整合性を確保すると同時に、認証済みユーザーのみにアクセスを限定します。これらのユーザーは、それらの鍵に間接的にのみアクセスし、自らのシステムに存在する Jacinto 7 プロセッサのコアに対応するソフトウェアの署名と暗号化を実施することができます。これらのソフトウェアは、標準的な X.509 証明書形式を使用して認証されるため、カスタム証明書を生成する必要も、署名ツールを用意する必要もなく、一般的なツールを使用するだけで該当の証明書を作成できます。その結果、開発ユーザーの皆様が使用しているセキュア・コンピューティング環境内に直接的な方法で実装を行い、一連の秘密鍵のセキュリティを確保することができます。

Jacinto 7 SoC のセキュア・ブート機能を活用すると、このデバイス上で動作するソフトウェアが常に認証済みの状態であることを保証できます。したがって、非常に重要な初期ブート・フェーズで、未認証のソフトウェアをロードしてしまう事態を防止できます。初期セキュア・ブート・プロセスは、セキュア・ブート ROM (読み取り専用メモリ) によって実装され、デバイスの信頼済みルートを基盤として、一連のソフトウェア・コンポーネントに認証を適用することができます。ブート認証オプションは、(最大 4,096 ビットの鍵を使用する) RSA (Rivest-Shamir-Adleman)、または最大 521 ビットの鍵を使用する楕円曲線デジタル証明書認証 (Elliptic Curve Digital Signature Authentication、ECDSA) の楕円曲線暗号化 (elliptic curve cryptography、ECC) のどちらかをサポートできます。これらのどちらかを、ソフトウェアや証明書のシグネチャとして使用できる、強力な SHA2-512 ハッシュと組み合わせることが可能です。また、オプションでブートローダーの AES-256 暗号化もサポートしています。

初期セキュア・プログラミング

秘密鍵をプログラミングする (書き込む) 際には、デバイス鍵のプロビジョニングをセキュアな方法で行うことが必須のプロセスとなります。デバイス鍵のプロビジョニング・プロセスは、鍵のプログラミング (書き込み) のセキュリティ、簡潔さ、最大のフレキシビリティ確保を目的として、TI から提供されたセキュア・プロビジョニング・ツールを使用し、システム・メーカーの皆様が独自のファクトリ (工場) で包括的に管理するものです。暗号により保護しているので、プロビジョニング・プロセスの過程で対称型の秘密鍵が明らかになってしまう事態を防止できます。また、信頼できないファクトリ環境であっても、鍵のプロビジョニングと製造を実施することが可能です。

暗号化アクセラレーション機能

暗号化関数の計算は、フレキシビリティやスループットの要件に基づき、汎用コンピューティング・コア、または特化型ハードウェア・アクセラレータで実施できます。Jacinto 7 SoC は、一般的な暗号化関数のアクセラレーション (迅速な実行) に適した一連のコアを搭載しているほか、以下の各機能もサポートしています。

- 非対称型暗号化：一連の RSA 関数と ECC 関数。
- ハッシュ計算：メッセージ・ダイジェスト・アルゴリズム (MDS)、SHA1、SHA2-224/256/384/512。
- 対称型暗号化関数：AES-128/192/256。
- ハードウェア TRNG モジュールと、決定論的乱数ビット生成機能 (deterministic random bit generator、DRBG) による後処理。

これらに加えて、Arm Cortex-A CPU は ARMv8 の暗号化拡張機能をサポートしています。この拡張機能は、AES、SHA1、SHA2 の各アルゴリズムの実行をアクセラレート (高速実行) するための新しい命令を追加します。

ソフトウェア IP (知的財産) の保護 (ファイアウォール)

Jacinto 7 SoC は、一連の異種プロセッサ・コアで構成されています。これらのコアはさまざまなタスク向けに最適化済みであり、64 ビット Arm コアや 32 ビット Arm マイコン・コアのほか、一部のデバイスでは TI のデジタル信号プロセッサ (DSP) や特化型 DSP アクセラレータも搭載しています。このうちいくつかのコンポーネントに対して、セキュア資産を対象にするタスクを割り当てることもできます。その場合、保護策や、他の汎用機能と分離する対策が必要になります。Jacinto 7 は、一連の包括的なシステム・ファイアウォール (分離機能) を搭載しており、ランタイムのセキュリティ保護や、安全性を目的とした分離に使用できます。開発ユーザーの皆様はファイアウォールを活用して、ハードウェア要素やメモリの範囲を定義し (この領域はセキュア用途限定、他の領域は汎用、など)、各プロセッサ・コアやシステム初期化機能がどの範囲にアクセスできるか

を規定することができます。このファイアウォール・インフラは重要なイネーブラとして機能し、秘密が明らかになる事態の防止、干渉されない自由の維持、仮に侵害が発生した場合でも影響の限定という役割を果たします。

デバッグのセキュリティ

JTAG デバッグ・ポートの普及が進んでおり、大半のプログラマブル・デバイスに実装済みなので、アクセスしやすさに関係する多くの機能は JTAG 経由で実施できます。デバイスのレジスタやメモリへの容易なアクセス、初期のフラッシュ書き込みを容易に実施できる方法、プログラムのトレース機能などが該当します。このようなアクセスしやすさは、システム内で露出に関して非常に脆弱なポイントの 1 つは JTAG である、ということも意味します。その結果、Jacinto 7 SoC の JTAG デバッグ・ポートはセキュア・デバイスではデフォルトでディスエーブルになっているので、セキュア・デバイスでこのポートを使用して SoC の動作にアクセスする (どのような暗号化処理を実施しているのか外部から観察する) ことは不可能です。同時に、必要な場合は、デバッグと分析を目的に、Jacinto 7 デバイスの JTAG をセキュアな方法でイネーブル (有効) にすることも可能です。JTAG アクセスをイネーブルにするには、信頼済みルートに結び付いている証明書メカニズムを使用して、承認または認証を実施することが必須です。さらに、各デバッグ証明書は 1 個のデバイスに結び付いています。したがって、1 通の証明書を使用してデバッグをイネーブルにできるのは、その証明書が格納しているデバイス ID を持つ、選択されたデバイスのみです。1 通の証明書を手に入れただけで、多数のデバイスをデバッグ (内部を観察) することはできません。最後に、システムのセキュリティ・プロトコルの要件によっては、ワンタイム・プログラマブル eFuse のプログラミング (書き込み) を通じて、JTAG アクセスを永続的にディスエーブルにすることも可能です。これら一連の機能は、保護とアクセスの階層構造を実現し、開発ユーザーの皆様は開発段階で適切なアクセスを実施できるフレキシビリティを確保しながら、製造段階ではセキュリティを維持することができます。

信頼できる実行環境

Jacinto 7 SoC の Arm Cortex-A72 TrustZone® 機能は、セキュア・ソフトウェア・コンポーネントの実行を他のコンポーネントから分離し、鍵、データ、特化型アルゴリズムのような重要な資産を保護する目的で使用できます。このセキュア環境の使用方法を簡素化するために、信頼できる実行環境 (trusted execution environment、TEE) により、セキュリティ・アプリケーションを分離するためのセキュア・ソフトウェア環境を実現します。Jacinto 7 デバイス向けの Linux® SDK (ソフトウェア開発キット) を使用すると、Linaro OP-TEE セキュア・スタックを統合できます。このスタックにより、Arm プラットフォーム上でセキュア・アプリケーションを開発するために、標準的な GlobalPlatform API (アプリケーション・プログラミング・インターフェイス) を使用するセキュリティ・アプリケーションを有効にすることができます。TEE のもう 1 つの利点は、セキュア・アプリケーション同士が互いに分離され、Linux スタックの残りの部分からも分離されることです。したがって、複数のクライアントが、クライアント相互間で自らの資産を明らかにせずに、TEE を安全な方法で使用することができます。

ファームウェアとソフトウェアのセキュア更新

ファームウェアのセキュア更新機能が必要になるのは、特に OTA (over-the air、ワイヤレス) 更新であり、現在は組込みシステム全般でこのようなセキュア更新機能が必須になっています。その結果、新機能や拡張機能の更新、バグ修正、セキュリティ・パッチの適用をフィールド (現場) で迅速に実施でき、サービス技術者やファクトリのサービス担当者の出張に伴う時間や費用が不要になります。ただし、この更新プロセスも脆弱性ポイントになる可能性があります。特に他者が偽装 (自らの身分を偽る)、以前のバージョンへのロールバック (巻き戻し)、または更新メカニズムを悪用して、セキュリティを侵害したソフトウェア・イメージの書き込みを行う場合などです。

更新イメージには、ハッシュと署名を常に施しておく必要があります。それにより、そのイメージの整合性と真正さ (改ざんのない正しいイメージである) の両方を検証できます。真正さのチェックを実施すると、既知かつ信頼できる出所からその更新が提供されたものである、ということを検証できます。また、整合性チェックを実施すると、転送やロードの間に、そのイメージの変更や改ざんが実施されなかった、ということを検証できます。セキュア・ブート認証に使用する Jacinto 7 SoC のこれらの機能は、ソフトウェアやデータの更新を認証する目的でも使用できます。

まとめ

Jacinto 7 デバイス・ファミリに用意されたセキュリティ・イネーブラは、一連の包括的な組込みセキュリティ機能で構成されています。設計者やアーキテクトの皆様はこれらの機能を活用し、開発中システムで必要とされているセキュリティの目標を達成することができます。これらは通常、セキュリティ実装サイクルの一環として評価されます。つまり、プロジェクトごとにセ

キュリティに関する具体的な目標とリスクと対策を識別し、それらセキュリティ上の目標を達成するのに役立つセキュリティ・イネーブラを特定する業務です。詳細は ti.com/security をご覧ください。

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が適用される各種規格や、その他のあらゆる安全性、セキュリティ、またはその他の要件を満たしていることを確実にする責任を、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 の販売約款 (https://www.tij.co.jp/ja-jp/legal/terms-of-sale.html)、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。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。

日本語版 日本テキサス・インスツルメンツ合同会社