STM32 マイクロコントローラのセキュリティについて

概要

このアプリケーションノートでは、STM32 マイクロコントローラのセキュリティの基本について説明します。

マイクロコントローラにおけるセキュリティは、ファームウェアの知的財産の保護、デバイス内のプライベートデータの保護、サービス実行の保証など、複数の面にわたります。

IoT の普及により、セキュリティはさらに重要性が増しています。非常に多くの数のデバイスがネットワークに接続されるため、攻撃者の主なターゲットになり、いくつかのリモート攻撃により、デバイスの通信チャネルの脆弱性が明らかになっています。IoT では、セキュリティは、秘匿性と認証性の要件を、暗号化を必要とすることが多い通信チャネルまで拡大します。

本書は、さまざまなタイプの攻撃に対して対応策を適用することによってセキュアなシステムの構築を支援することを目的としています。

最初のパートでは、さまざまなタイプの脅威を簡単に要約した後、典型的な攻撃例を挙げて、攻撃者が組込みシステムのさまざまな脆弱性をどのように悪用するかを示します。

後続のセクションでは、これらの攻撃を阻止できるハードウェアとソフトウェアの保護の方法について述べます。

最後のセクションでは、STM32 シリーズで使用可能なすべてのセキュリティ機能の一覧を示し、セキュアなシステムの構築に関するガイドラインを示します。

表 1. 対象とする製品

タイプ	製品シリーズ
マイクロコントローラ	STM32F0 シリーズ、STM32F1 シリーズ、STM32F2 シリーズ、STM32F3 シリーズ、STM32F4 シリーズ、STM32F7 シリーズ、STM32G0 シリーズ、STM32G4 シリーズ、STM32H7 シリーズ、STM32L0 シリーズ、STM32L1 シリーズ、STM32L4 シリーズ、STM32L4+ シリーズ、STM32L5 シリーズ、STM32U5 シリーズ、STM32WB シリーズ、STM32WL シリーズ

1 一般情報

次の表に、本書で使用される略記とその定義の被包括的な一覧を示します。

表 2. 用語

用語	定義
AES	高度暗号化標準
CCM	コア結合メモリ（SRAM）
CPU	中央演算処理装置 — マイクロコントローラのコア
CSS	クロック・セキュリティ・システム
DoS	サービス拒否（攻撃）
DPA	電力差分解析
ECC	エラーコード訂正
FIA	誤動作利用攻撃（フォールトインジェクション）
FIB	集束イオンビーム
GTZC	グローバル TrustZone® コントローラ
HDP	セキュア秘匿保護
HUK	ハードウェア・ユニーク・キー
IAP	アプリケーションによる Flash 書込み
IAT	初期証明トークン
IoT	Internet of things（モノのインターネット）
IV	初期化ベクタ（暗号アルゴリズム）
IWDG	独立型ウォッチドッグ
MAC	メッセージ認証コード
MCU	マイクロコントローラユニット（STM32 Arm® Cortex®-M ベースのデバイス）
MPCBB	ブロックベースのメモリ保護コントローラ
MPCWM	ウォーターマークベースのメモリ保護コントローラ
MPU	メモリ保護ユニット
NSC	非セキュア側から呼出し可能
NVM	不揮発性メモリ
OTFDEC	オンザフライ復号
OTP	ワンタイム・プログラマブル
PCROP	商用コード読出し保護
PKA	公開鍵アルゴリズム（非対称アルゴリズムともいう）
PSA	プラットフォームセキュリティアーキテクチャ
PVD	プログラム可能な電圧検出器
PWR	電源制御
ROM	読出し専用メモリ — STM32 内のシステム Flash メモリ
RoT	信頼の基点
RDP	読出し保護
RSS	ルートセキュアサービス
RTC	リアルタイムクロック

用語	定義
SAU	セキュリティ属性ユニット
SB	セキュア・ブート
SCA	サイドチャネル攻撃
SDRAM	同期ダイナミックランダムアクセスメモリ
SFU	セキュア・ファームウェア・アップデート
SPA	単純電力解析
SPE	セキュア処理環境
SRAM	スタティック・ランダム・アクセス・メモリ（揮発性）
SST	セキュア・ストレージ
SWD	シリアル・ワイヤ・デバッグ
TF-M	トラステッドファームウェア-M
WRP	書込み保護機能

参考資料

各デバイスのリファレンスマニュアルには、搭載するセキュリティ機能とメモリ保護実装の詳細が記載されています。

各 Arm® Cortex® バージョンのプログラミングマニュアルも入手可能であり、メモリ保護ユニット（MPU）について説明されています。

- STM32 Cortex®-M33 MCU プログラミング・マニュアル（PM0264）
- STM32F7 シリーズおよび STM32H7 シリーズ Cortex®-M7 プロセッサプログラミングマニュアル（PM0253）
- STM32 Cortex®-M4 CMU および MPU プログラミング・マニュアル（PM0214）
- STM32F10xxx/20xxx/21xxx/L1xxxx Cortex®-M3 プログラミングマニュアル（PM0056）
- Cortex®- STM32L0、STM32G0、STM32WL、および STM32WB シリーズ用 M0+ プログラミング・マニュアル（PM0223）

セキュリティ機能の詳細な説明については、以下のユーザマニュアルおよびアプリケーションノート（www.st.com で入手可能）を参照してください。

- ユーザマニュアル STM32 crypto library（UM1924）：STM32 暗号ライブラリの API について説明されています。X-CUBE-CRYPTOLIB ソフトウェア拡張パッケージも用意されています。
- ユーザマニュアル Getting started with the X-CUBE-SBSFU STM32Cube Expansion Package（UM2262）：ST の SB（セキュア・ブート）と SFU（セキュア・ファームウェア・アップデート）ソリューションについて説明されています。X-CUBE-SBSFU ソフトウェア拡張パッケージも用意されています。
- アプリケーションノート Proprietary Code Read Out Protection on STM32xx microcontrollers（AN4246、AN4701、AN4758、AN4968）：それぞれ STM32L1、F4、L4、および F7 シリーズについて、PCROP ファームウェアのセットアップと使用方法を説明しています。X-CUBE-PCROP ソフトウェア拡張パッケージも用意されています。
- アプリケーションノート Managing memory protection unit (MPU) in STM32 MCUs（AN4838）：STM32 製品で MPU を管理する方法を説明しています。
- アプリケーションノート STM32WB ST firmware upgrade services（AN5185）

注　Arm は、米国内およびその他の地域にある Arm Limited（またはその子会社）の登録商標です。

2 概要

2.1 セキュリティの目的

保護が必要な理由

マイクロコントローラのセキュリティは、内蔵のファームウェア、データ、およびシステムの機能を保護することを意味します。データ保護の必要性は、暗号化キーや個人データの場合、最も重要です。

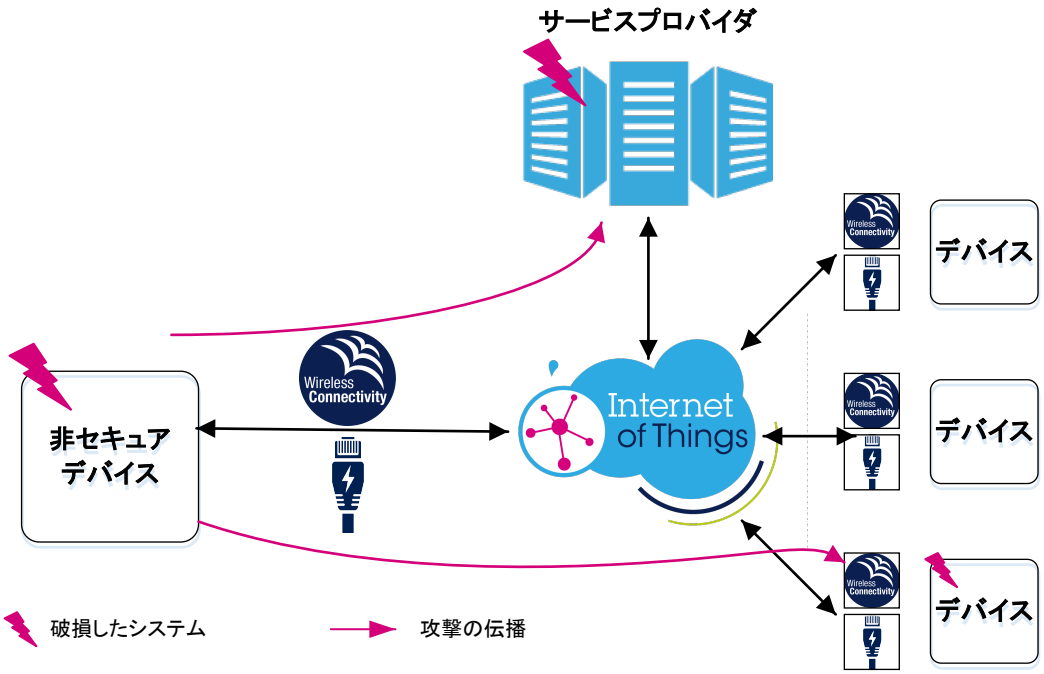
ファームウェアコードも重要な資産です。攻撃者がバイナリにアクセスできれば、プログラムをリバース・エンジニアリングして、さらなる脆弱性を探し、ライセンシングやソフトウェア制限を迂回します。攻撃者はカスタムアルゴリズムをコピーでき、それを使用してハードウェアのクローンを作成することさえできます。オープンソースソフトウェアの場合でも、コードが本物であり、悪意あるファームウェアに置き換えられていないことを確認することは意味があります。

サービス拒否攻撃（DoS 攻撃）は、環境（ガス、火災、侵入）検出アラームや監視カメラなどの保護システムを考慮するときのもう一つの大きな脅威です。システムの機能は堅牢かつ信頼できるものでなければなりません。

システムが複雑になるとしても、セキュリティの要件を軽視すべきではありません。今日、マイクロコントローラを中心として構築されたシステムは、ますます巧妙になり、金銭的利益を得ることを期待している攻撃者の潜在的なターゲットになっています。このような利益は非常に高額な場合があり、攻撃が IoT のように大規模に伝播する場合は特にそうです。完全に安全なシステムなどないとしても、攻撃をよりコストのかかるものにすることは可能です。

実際、IoT またはスマートデバイスに要求されるセキュリティレベルは、ますます高くなっています。接続されたデバイスはリモートアクセスが可能なため、ハッカーにとって非常に魅力的です。接続性は、プロトコルの脆弱性を通して攻撃の手段を提供します。攻撃が成功した場合、1 つのデバイスがハッキングされただけでネットワーク全体の完全性が損なわれることがあります（下の図を参照）。

図 1. 破損した接続デバイスの脅威



保護すべきもの

セキュリティを特定のターゲットや資産に限定することはできません。コードバイナリが暴かれた場合、データを保護することは困難であり、攻撃と保護メカニズムに違いがないことが少なくありません。しかし、それでも、資産とリスクの情報をまとめることは有用です。

下の表に、完全に網羅されているわけではありませんが、攻撃者のターゲットとなる資産のリストを示します。

表 3. 保護すべき資産

ターゲット	資産	リスク
データ	センサデータ(医療データや位置のログなど) ユーザデータ(ID、PIN、パスワード、アカウントなど) トランザクションログ 暗号鍵	個人データの無許可販売 不正使用 スパイ 脅迫メール
デバイスの制御(ブートローダ、悪意あるアプリケーション)	デバイスの正常機能 デバイス/ユーザ ID	サービス拒否 サービスプロバイダへの攻撃 サービス(クラウド)への不正アクセス
ユーザコード	デバイスのハードウェアアーキテクチャ/設計 ソフトウェアの特許/アーキテクチャ テクノロジの特許	デバイスの偽造 ソフトウェアの偽造 ソフトウェアの改ざん セキュア領域へのアクセス

脆弱性、脅威、および攻撃

保護メカニズムは、さまざまな脅威に対処する必要があります。その目的は、攻撃に悪用される恐れがある脆弱性を取り除くことです。基本的なものから高度なものまで、主な攻撃のタイプの概要を セクション 3 攻撃のタイプ に示します。

以下の特定の語句は、セキュリティに関して使用されます。

- 資産:保護する必要があるもの
- 脅威:デバイス/ユーザーがそれに対する保護を必要とするもの
- 脆弱性:保護メカニズムの弱点または隙間

要約すると、攻撃は、資産にアクセスするためにシステムの脆弱性を悪用する脅威の実現です。

3 攻撃のタイプ

この節では、マイクロコントローラが直面する可能性のあるさまざまなタイプの攻撃について、最も基本的なものから非常に高度でコストのかかるものまで示します。最後のパートでは、IoT システムをターゲットにした典型的な攻撃例を示します。

マイクロコントローラへの攻撃は、次のタイプのいずれかにに分類されます。

- ソフトウェア攻撃：ソフトウェアの脆弱性（バグまたはプロトコルの弱点など）を悪用します。
- ハードウェアの非侵襲攻撃：MCU インタフェースと環境情報に焦点を合わせます。
- ハードウェアの侵襲攻撃：シリコンへの直接アクセスを伴う破壊的攻撃。

3.1 攻撃のタイプについて

セキュリティにおける主なルールは、攻撃を成功させることは常に可能であるということです。

まず、予想外の攻撃に対する絶対的な保護はありえません。システムを保護するために、どのようなセキュリティ対策が取られても、デバイスのライフタイム中にセキュリティ侵犯が見つかり、悪用される可能性があります。この最後の点から、ファームウェアの更新方法を考慮し、セキュリティを高める必要があります（セクション 5.2.2 セキュア・ファームウェア・アップデート（SFU）を参照）。

第二に、適切な機器を備えた実験室条件では、マイクロコントローラの内容を取得し、アーキテクチャの詳細を設計することも可能です。これらの技法は、セクション 3.3 ハードウェア攻撃 に簡単に示されています。

攻撃者から見ると、期待できる収益/攻撃コストの比が高いほど、攻撃の採算が合います。収益は、盗む資産価値と攻撃の再現性に依存します。コストは、成功のために費やされる時間と資金（機器）、必要な攻撃者のスキルの獲得状況によって決まります。

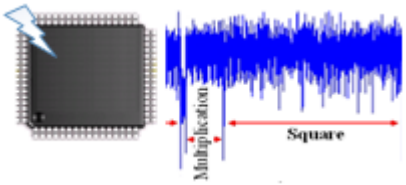
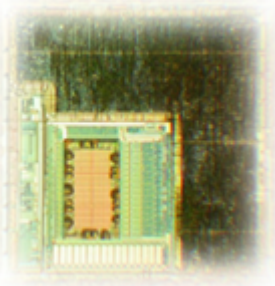
攻撃のタイプ

より詳細な攻撃のグループとカテゴリもありますが、基本的なカテゴリは次のとおりです。

- ソフトウェア攻撃は、特にバグ、プロトコルの弱点、または信頼できないコード片を悪用することによって実行されます。通信チャネルへの攻撃（傍受や不正使用）は、このカテゴリに属します。攻撃の大半はソフトウェア攻撃で占めています。コストは非常に低いです。広く拡散し、繰り返し行われると、多大な損害を与えることがあります。デバイスへの物理アクセスを必要とせず、リモートから攻撃を実行できます。
- ハードウェア攻撃は、デバイスへの物理アクセスを必要とします。最もわかりやすい例は、保護されていないデバッグポートの悪用です。しかし、一般に、ハードウェア攻撃は巧妙で、非常にコストがかかります。特定の機材で実行され、エレクトロニクスエンジニアリングのスキルを必要とします。デバイス破壊を伴わずにボードレベルまたはチップレベルで実行される非侵襲攻撃と、パッケージ破壊を伴い、デバイスシリコンレベルで実行される侵襲攻撃とがあります。ほとんどの場合、そのような攻撃の採算が合うのは、広く適用可能な新しいリモート攻撃につながる情報が明らかになる場合に限られます。

下の表に、各タイプの攻撃のコストと使用される技法の概要を示します。

表 4. 攻撃のタイプとコスト

攻撃のタイプ	ソフトウェア	ハードウェア非侵襲	ハードウェア侵襲
-			
範囲	リモートまたはローカル	ローカルボードおよびデバイスレベル	ローカル・デバイス・レベル
技法	ソフトウェアのバグ プロトコルの弱点 トロイの木馬 盗聴	デバッグポート パワーグリッチ フォールト・インジェクション サイドチャネル解析	プロービング レーザー FIB リバース・エンジニアリング
コスト/知識	ターゲットのセキュリティ障害に応じて、非常に低いものから高いものまで	非常に低コスト。実装にはある程度の精巧な機器と知識のみが必要。	非常に高価。専用の機器と非常に具体的なスキルが必要。
目標	機密資産(コードとデータ)へのアクセス。 不正使用 サービス拒否	秘密データまたはデバイスの内部動作(アルゴリズム)へのアクセス。	デバイスのリバース・エンジニアリング(シリコンの知的財産) 非表示のハードウェアおよびソフトウェアの秘密へのアクセス(Flash アクセス)

3.2 ソフトウェア攻撃

ソフトウェア攻撃は、コード片(マルウェア)を CPU に実行させることによって、システム上で実行されます。マルウェアは、デバイスの制御を奪って、システムのリソース(ID、RAM、Flash メモリの内容、ペリフェラルレジスタなど)にアクセスしたり、機能を変更したりします。

このタイプの攻撃は、次のような理由から、デバイスにとって最大の脅威です。

- パーソナルコンピュータ以外の特殊な機器を必要としないため、攻撃コストが低いです。
- 多くのハッカーが協力して、知識やコツを共有でき、セキュリティ違反が存在した場合、攻撃が成功する可能性が高いです。さらに、成功した場合、攻撃プロトコルが短時間で Web に拡散します。

マルウェアはデバイスに注入されることがあり、または、検証されていないあるいは信頼できないライブラリなどを通じて、メイン・アプリケーション・ファームウェアにすでに存在する可能性があります(インサイダー脅威)。

マルウェアには多くのタイプがあり、非常に小さく、容易に隠すことができます。

マルウェアでできることの一例を示します。

- デバイス設定(オプションバイトやメモリ属性など)を変更します。
- 保護を無効化します。
- メモリを読み出して、その内容をダンプし、ファームウェアおよびデータのクローニングを行います。
- デバイスデータをトレースまたはログします。
- 暗号化アイテムにアクセスします。

- 通信チャネル/インタフェースを開きます。
- デバイスの機能を変更またはブロックします。

ユーザアプリケーションが全面的に信頼でき、バグフリーであり、隔離されていて、外界との通信手段がないという場合を除き、ソフトウェア攻撃を考慮する必要があります。

マルウェアの注入

システム内にコード片を注入するには、さまざまな方法があります。マルウェアのサイズはターゲットによって異なりますが、非常に小さい場合があります（数十バイト）。マルウェアを実行するには、デバイスのメモリ（RAM または Flash メモリ）に注入する必要があります。注入後の難問は、CPU に実行させることであり、そのためには PC（プログラム・カウンタ）が分岐する必要があります。

マルウェアを注入する方法は、次のように分類することができます。

- 基本的なデバイスアクセス/「オープンドア」:
  - デバッグポート：JTAG または SWD インタフェース
  - ブートローダ：アクセス可能な場合、これを使用して、使用可能なインタフェースからメモリの内容を読み書きできます。
  - 外部メモリからの実行

  これらのマルウェア注入は、セクション 4 デバイス保護に述べられている単純なハードウェアメカニズムで容易に対処できます。
- アプリケーションのダウンロード：
  - ファームウェア更新手順：新しい FW の代わりにマルウェアが転送されます。
  - 新しいアプリケーションをダウンロードする機能を備えた OS。

  このカテゴリの対応策は、デバイスとサーバ間の認証、またはコード認証に直接基づきます。認証は暗号化アルゴリズムに依存します。
- 通信ポートの弱点とバグの悪用：
  - データの実行。ときには、マルウェアをデータとして忍び込ませて、正しくない境界チェックを悪用して実行することが可能です。
  - スタックベースのバッファ・オーバーフロー、ヒープベースのバッファ・オーバーフロー、jump-to-libc 攻撃、および data-only 攻撃

  この 3 番目のカテゴリは、定義では回避が困難です。ほとんどの組込みシステムアプリケーションは、C/C++ などの低水準言語を使用してコード化されています。これらの言語は、攻撃者によって利用されるメモリ管理エラーにつながることがあるため（スタック、ヒープ、またはバッファ・オーバーフローなど）、安全でないとみなされています。一般的な考え方として、信頼できない、または検証されていないファームウェア部分を最小化することによって、アタックサーフェスと呼ばれるものを可能な限り削減します。あるソリューションは、異なるプロセスの実行とリソースを隔離することです。たとえば、TF-M には、そのようなメカニズムが含まれています。
- デバイスのバックドアと信頼できないライブラリの使用
  この最後のカテゴリは、デバイスの破損を容易にする意図的なマルウェアの導入です。今日、ファームウェア開発の多くは Web で共有されるソフトウェアに依存し、複雑なものにはトロイの木馬を隠すことができます。前のカテゴリと同様、このような脅威に対応するには、プロセス実行を可能な限り隔離し、重要なコードとデータを保護することによって、アタックサーフェスを削減します。

ブルートフォース

このタイプの攻撃は、共有の秘密に基づく認証をターゲットにします。セキュアデバイスでは、（たとえばクラウドの）サービスにアクセスする前にセッション認証が必要とされますが、ヒューマン・マシン・インタフェース（HMI）を自動プロセスで利用して、膨大なパスワードを連続して試みることができます。

興味深い対応策を以下に示します。

- モノトニック・カウンタ（タイマで、または可能な場合はバックアップドメインで実装）によってログインの試行回数を制限します。
- ログインの連続試行間隔を大きくします。
- チャレンジレスポンス・メカニズムを加えて、自動試行を不能にします。

3.3 ハードウェア攻撃

ハードウェア攻撃は、デバイス、または多くの場合、複数のデバイスへの並行した物理アクセスを必要とします。

コスト、時間、必要な知識に差がある 2 種類の攻撃があります

- 非侵襲攻撃はデバイスへの外部アクセスだけ行い（ボードレベル攻撃）、それほどコストをかけずに実行できます（数千ドルから数万ドルの設備）。
- 侵襲攻撃はデバイスのシリコンに直接アクセスします（パッケージ開梱後）。専門の実験室で見られるような高度な機器で実行されます。非常にコストがかかり（10 万ドル以上、数百万ドルのことも）、非常に貴重なデータ（鍵または ID）や技術特許をターゲットにします。

汎用マイクロコントローラは、最も高度な物理的攻撃に対抗するための最善の候補ではありません。最も高い保護レベルが必要な場合は、セキュアエレメントと汎用マイクロコントローラの併用を検討することをお勧めします。セキュアエレメントは、最新のセキュリティ標準に従って認証された専用マイクロコントローラであり、特定のハードウェアを備えています。

ST のセキュアマイクロコントローラの Web ページを参照してください（www.st.com/en/secure-mcus.html）。

3.3.1 非侵襲攻撃（Non-invasive attack）

非侵襲、またはボードレベル攻撃は、物理的損害なしに（デバイスは機能状態のまま）、保護の迂回を試みます。アクセス可能なインタフェースとデバイス環境のみが使用されます。このような攻撃には、中程度に精巧な機器とエンジニアリングスキル（信号処理など）が必要です。

デバッグポートアクセス

これは、デバイスに対して実行できる最も基本的な攻撃です。デバッグ機能の無効化は、最初に考慮しなければならない保護レベルです。実際、JTAG または SWD プロトコルを通じてデバッグポートやスキャンチェーンにアクセスすれば、デバイスの内部リソース全体にアクセスできます（CPU レジスタ、内蔵 Flash メモリ、RAM、およびペリフェラルレジスタ）。

対応策：

- デバッグポートの無効化またはヒューズ 読出し保護（RDP）

シリアルポートアクセス

通信ポート（I2C、SPI など）へのアクセスは弱点であり、悪用される恐れがあります。通信ポートは、スパイされたり、デバイスエントリポイントとして使用されたりします。関連のプロトコルの実装方法に応じて（メモリアドレスアクセス範囲、ターゲットのペリフェラル、読出し/書込み操作など）、攻撃者はデバイスリソースにアクセスできる可能性があります。

対応策：

- ソフトウェア：
  - 関連するプロトコル操作をファームウェアレベルに制限して、機密リソースを読み書きできないようにする必要があります。
  - 通信スタックを機密データから隔離します。
  - データ転送の長さをチェックして、バッファ・オーバーフローを回避します。
  - デバイスとターゲットの間の共有鍵で通信を暗号化できます。
- ハードウェア：
  - 物理的な通信ポートを多層ボードに埋め込んで、アクセスしにくくできます。
  - 使用しないインタフェースポートは無効化する必要があります。

フォールト・インジェクション：クロックおよび電源障害/グリッチ攻撃

フォールト・インジェクションは、データシートで定義されたパラメータの範囲外でデバイスを使用して、システムの誤動作を起こさせます。攻撃が成功すると、プログラム状態の破損、メモリ内容の破損、プロセス実行の停止（縮退故障）、命令のスキップ、条件分岐の変更、無許可アクセスの提供など、プログラムの動作をさまざまに変更することができます。

典型的な脅威は、クロックの改変（フリーズまたはグリッチ）と電源の改変（電圧不足、過電圧、グリッチ）です。フォールトは意図的でない場合もあるため、対応策は安全性のために使用されるものと同様に、冗長性、エラー検出、およびモニタリングです。

対応策：

- ソフトウェア：
  - 関数の返り値をチェックします。
  - 条件分岐の際には厳格な比較を使用します。
  - 分岐ごとに専用の変数を素数で増分して、期待値をチェックすることによって、コードが重要な部分をスキップしていないことを確認します。
  - 簡単ではない値を真偽として使用します（0 または-1 との比較を避け、相互のハミング距離が大きい複雑な値を使用します）。
- ハードウェア：
  - 使用可能な場合は、クロック・セキュリティ・システム（CSS）を使用します。
  - 内部クロックソースを使用します。
  - 内部電圧レギュレータを使用します。
  - メモリエラー検出（ECC およびパリティ）を使用します。

サイドチャネル攻撃（SCA）

ファームウェアが実行されるとき、攻撃者はデバイスの実行特性（電力消費、電磁放射、温度、活動時間など）を観察することができます。この観察から、データ値やアルゴリズム実装など、秘密の資産を取得するのに十分な情報が得られます。サイドチャネルベースの攻撃は、暗号化デバイスに対する強力な攻撃であり、システムによって使用される鍵を暴くことができます。SPA（単純電力解析）と DPA（電力差分解析）は、電力消費を悪用するサイドチャネル攻撃の典型例です。

対応策：

- ソフトウェア：
  - 鍵の使用を制限します。可能なときには、セッション・ランダム・キーを使用します。
  - 動作のランダム化（遅延、フェイク命令など）機能を持つ、保護された暗号ライブラリを使用します。
- ハードウェア：
  - モニタリングに対する防御はセキュア・エレメント（STSAFE）で見られますが、通常、汎用マイクロコントローラに内蔵される効率的なハードウェア対応策はありません（STM32U5 シリーズの SAES（Secure AES co-processor）搭載例を除く）。

3.3.2 シリコン侵襲攻撃

そのような攻撃のコストは、非常に高くなります。すべての手段は、プロセス中に破壊されるデバイスの情報を抽出するものとみなされます。攻撃者が成功するには、かなりの数のデバイスを入手する必要があります。専門の研究所にあるような高価な機器で実行され、高いレベルのスキルと知識および時間が必要です。

侵襲攻撃はデバイスパッケージの除去から始まります。パッシベーション層を除去しなくても、ある程度の分析を行うことができますが、デバイスの相互作用の調査（プロービング）には除去が必要です。開梱は、化学エッチング、ドリル穿孔、またはレーザーカッターによって行うことができます。デバイスが開いたら、プロービングや改ざん攻撃を行うことが可能です。

セキュリティ専用のいくつかの ST マイクロコントローラは、このような種類の処理に対する堅牢性を備えています。これらは STM32 ファミリの一部ではなく、本書の範囲外です。ST のセキュア・ハードウェア・プラットフォーム（www.st.com/en/secure-mcus.html）を参照してください）。

リバース・エンジニアリング

目的は、デバイスの内部構造を理解し、その機能性を解析することです。現代のデバイスは数百万のゲートがあるため、これは非常に困難なタスクです。

最初のステップは、マイクロコントローラのマップを作成することです。光学顕微鏡を使用して、デバイス表面の高解像度写真を撮影することによって行うことができます。デバイスをエッチングして金属層を剥離した後、第 2 のステップで、より深い層を分析できます。

データの読出し

電子顕微鏡を使用すると、データが電荷によって表現され、目に見えるようになります。デバイスのメモリ全体を読み出すことが可能です。

マイクロプロービングと内部故障の注入

マイクロプロービングは、金属層レベルでデバイスと相互作用することから成ります。微細電極を使用してデバイス表面と直接的な電気接触を確立することにより、攻撃者はデバイスの実行中に観察、操作、干渉を行うことができます。

デバイスの改ざん

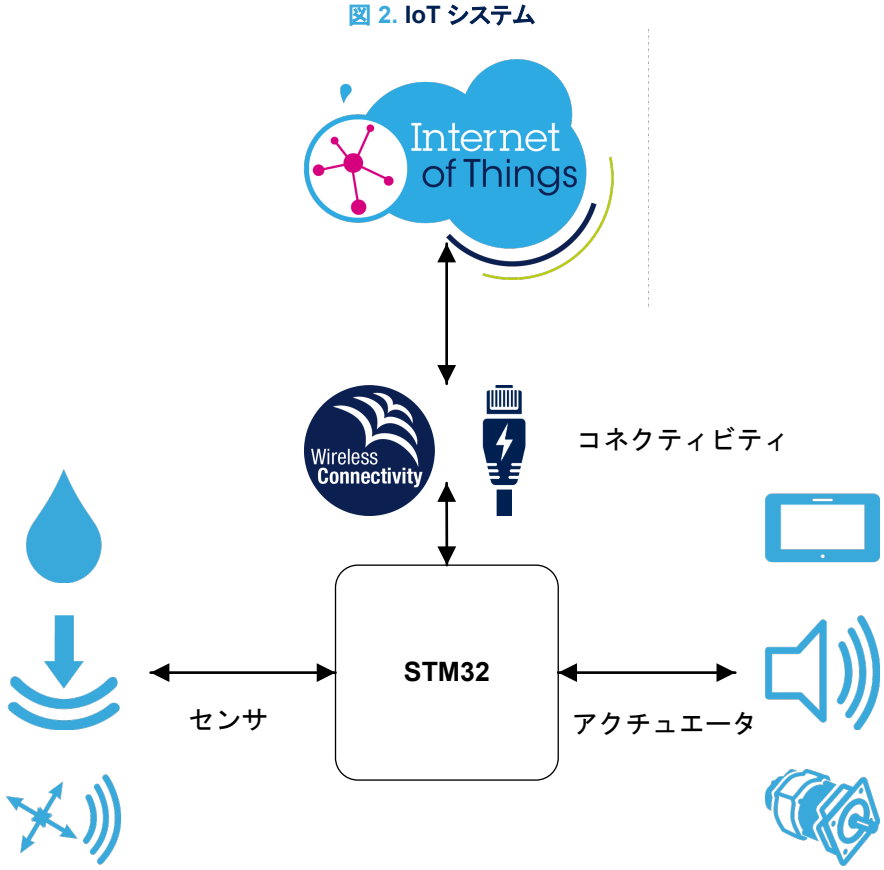
より高度なツールを使用して攻撃を行うことができます。たとえば、FIB:Focused Ion Beam(集束イオンビーム)ワークステーションにより、深い金属ラインおよびポリシリコン・ラインの手動プロービングを単純化します。また、既存の相互接続ラインをカットしたり、新しい相互接続ラインや新しいトランジスタを作成したりすることによって、デバイス構造を改ざんするためにも使用できます。

3.4 IoT システム攻撃の例

この節では、IoT システムに対する典型的な攻撃例を示します。幸いにも、このような攻撃のほとんどはセキュリティ機能(ハードウェア対応策)とセキュアなアプリケーション・アーキテクチャ(ソフトウェア対応策)を有効にすることによって対策できます。これらの対応策について、以下の節で説明します。

IoT システムは、STM32 マイクロコントローラを中心に、接続システム(Ethernet、Wi-Fi®、Bluetooth® Low Energy、LoRa® など)とセンサおよび/またはアクチュエータで構築されます(下の図を参照)。マイクロコントローラはアプリケーション、データ取得、およびクラウドサービスとの通信を処理します。マイクロコントローラは、ファームウェアの更新と整合性チェックを通じてシステム保守も担当します。

図 2. IoT システム



3.5 攻撃のターゲットの一覧

以下の節では、可能な攻撃ターゲットの一覧を示します。

初期プロビジョニング

セキュリティのチェーンの信頼の基点の暗号データは、管理された、信頼できる方法で SoC に注入される必要があります。鍵、証明書、またはハッシュ初期値のいずれであっても、不変で秘密に保たれなければなりません。デバイス内にプログラムされたデータ保護メカニズムを有効にして、許可されたプロセスだけがアクセスできるようにしなければなりません。

- リスク:ファームウェアの破損、不正使用
- 対応策:
  - 信頼できる製造者環境
  - セキュアデータプロビジョニングサービス(SFI)の使用
  - データ保護メカニズム
  - セキュア・アプリケーション隔離
  - OTP メモリの使用

ブート変更

この攻撃の目的は、ブートローダを使用して、デバイスの内容にアクセスすることです。攻撃はブートモードおよび/またはブートアドレスを変更して、ユーザアプリケーションをプリエンプトし、ブートローダを通じて(USB DFU、I2C、または SPI を介して)、デバッグポートを通じて、または RAM にインジェクトされたファームウェアを通じて、CPU の制御を奪うことです。ブートモードとブートアドレスはデバイス設定や入力ピンによって制御され、保護される必要があります。

- リスク:マイクロコントローラの内容へのフルアクセス
- 対応策:
  - 固有の起動エントリ
  - ブートローダとデバッグの無効化(読出し保護(RDP)を参照)

セキュア・ブート(SB)または信頼できるファームウェア(TF-M)

堅牢なシステムは、メインアプリケーション起動前の初期ファームウェア完全性および認証性チェックに依存しています。デバイスの信頼の基点であるため、ユーザファームウェアのこの部分は不変でなければならず、迂回が不能でなければなりません。

攻撃の成功には、検証を迂回し、マルウェアに直接ジャンプして、信頼できないアプリケーションを実行する必要があります。これは、フォールト・インジェクションなどのハードウェア技法によって行うことができます。また、予想されるハッシュ値をマルウェアのハッシュ値で置き換えることによっても実行できます(この章の冒頭の「初期プロビジョニング」セクションを参照してください)。

- リスク:デバイスへなりすまし、アプリケーションの変更
- 対応策:
  - 検証の迂回を避けるための一意なブートエントリポイント
  - SB コードの変更を避けるための不変コード
  - ファームウェアの署名やタグ値のセキュアなストレージ
  - 環境イベントの検出(電源グリッチ、温度、またはクロック速度)

ファームウェアの更新

ファームウェアの更新手順では、デバイスのライフタイムにわたって最善のユーザエクスペリエンスを確保するために、製品の所有者がファームウェアの修正版を提案することができます。ただし、ファームウェアの更新は、攻撃者に独自のファームウェアや既存のファームウェアの破損したバージョンでデバイスに侵入する機会を与えることになります。

このプロセスは、ファームウェアの認証性および完全性検証で保護されなければなりません。攻撃が成功するには、暗号化手順とキーにアクセスする必要があります(この章の冒頭「初期プロビジョニング」セクションを参照してください)。

- リスク:デバイスファームウェアの破損
- 対応策:認証性および完全性チェックを備えた SFU アプリケーション。署名に加えて、ファームウェアを暗号化することによって、秘匿性も加えることができます。

通信インタフェース

シリアルインタフェース(SPI、I2C、USART など)は、ブートローダやアプリケーションによって、デバイスとデータやコマンドを交換するために使用されます。通信の傍受により、攻撃者はインタフェースをデバイスへのエントリポイントとして使用できます。また、ファームウェアプロトコルは、バグが発生しやすいことがあります(オーバーフローなど)。

- リスク:デバイスの内容へのアクセス

- 対応策:
  - 物理的バスをボード上で到達しにくくします。
  - ソフトウェア通信スタックを隔離して、重要なデータや操作へのアクセスを防止します。
  - データ交換に暗号化を使用します。
  - 不要なときには、I/F ポートを無効にします。
  - 入力を入念にチェックします。

デバッグポート

デバッグポートはデバイスの内容全体、すなわち、コアおよびペリフェラルレジスタ、Flash メモリ、および SRAM の内容へのアクセスを提供します。アプリケーション開発に使用され、将来のバグを調査するために動作状態に保ちたくなりますが、攻撃者がデバイスへの物理アクセスを試みる最初の侵犯になります。

- リスク:デバイスへのフルアクセス
- 対応策:デバイスのデバッグ機能を無効にします(読出し保護(RDP)機能を参照)。

外部ペリフェラルアクセス

IoT デバイスは、グローバルアプリケーションに応じてセンサおよびアクチュエータを制御します。攻撃者はセンサからのデータを変更することによって、またはアクチュエータへ向かう出力データをシャントすることによって、システムを方向転換させることができます。

- リスク:正しくないシステム動作。
- 対応策:耐タンパによるボードレベルでのシステム侵入の検出

重要なファームウェアとデータ

ファームウェアのいくつかの部分には、特別な保護が必要です。たとえば、暗号化アルゴリズムやサードパーティ・ライブラリなどです。また、データが貴重な資産とみなされる場合(暗号キーなど)、強化された保護を必要とすることがあります。

内部メモリの内容は外部アクセス(通信インタフェースなど)と内部アクセス(他のソフトウェアプロセス)に対して保護されなければなりません。メモリの属性とファイアウォールは、プロセスとデータの隔離のための主要な保護です。

- リスク:重要なファームウェアコピーやデータ盗難
- 対応策:
  - 実行専用アクセス権(XO)。
  - ファイアウォール
  - メモリ保護ユニット
  - セキュア領域
  - 外部メモリの暗号化

SRAM

SRAM は、メモリを実行するデバイスです。ランタイムバッファと変数(スタック、ヒープなど)を内蔵し、ファームウェアとキーを内蔵することができます。不揮発性メモリでは、秘密は暗号化されて格納されることがあり、SRAM にロードされるときには、平文になっていなければ使用できません。同時に、SRAM は通常、通信バッファを保持します。この 2 つの理由から、攻撃者は SRAM を集中して狙うことがあります。このメモリに対しては、少なくとも 3 つのタイプの攻撃が行われます。すなわち、コード(マルウェア)インジェクション、バッファ・オーバーフローによるメモリ破損、および一時的に格納された変数による秘密の取得です。

- リスク:バッファ・オーバーフロー、データ盗難、またはデバイス制御
- 対応策:
  - ファイアウォール
  - メモリ保護ユニット
  - セキュア領域

乱数の生成

乱数は、しばしば、セッションキーの暗号化、暗号ノンス、または初期化ベクトル(IV)の生成で使用されます。乱数発生器が弱いと、セキュアプロトコルが脆弱になることがあります。

ソフトウェア攻撃は、ランダムシーケンスの隠れた周期性や構造を悪用して、共有秘密鍵を推測し、通信の秘匿性を破ろうとします。ハードウェア攻撃は、RNG を無効にしたり、出力の統計的ランダム性を弱めたりしようとします。

堅牢な乱数発生器は、エントロピーソース（アナログ）の品質に依存します。

- リスク：暗号化プロトコルのセキュリティの低下
- 対応策：
  - 対応策：真のハードウェアエントロピー発生器の使用
  - RNG 出力をテストして、生成された乱数の統計的プロパティを確認します。

通信スタック

接続プロトコル（Bluetooth、Ethernet、Wi-Fi、LoRa など）は、複雑な通信ファームウェアスタックを持ちます。これらのスタックは、しばしばオープンソースで入手でき、必ずしも信頼できるものとみなすべきではありません。潜在的な弱点が大規模に悪用されることがあります。

- リスク：ネットワーク経由のデバイスアクセス（内容、制御）
- 対応策：
  - 通信プロセスの隔離
  - サーバ認証
  - セキュアファームウェア更新によるバグのパッチ

通信の盗聴

デバイスと IoT サービス間のデータ交換は、互換性のある RF デバイスによって直接、またはネットワーク経由で盗聴されることがあります。ハッカーはデータの取得、デバイス ID の入手、サービスへのアクセスなどを試みることがあります。

暗号化は、すべての通信プロトコルに採用できます。多くの場合、複数の暗号化ステップが、あらゆる階層（デバイス、ゲートウェイ、アプリケーション）の間の通信を保護するとみなされています。

- リスク：ネットワークトラフィックの観察となりすまし。
- 対応策：暗号化バージョンの通信スタックの使用（Ethernet の場合は TLS など）

4 デバイス保護

このセクションで述べるセキュリティ保護はハードウェアメカニズムによって制御されます。これらは、オプションバイトによってデバイスを設定することによって、または、ハードウェアコンポーネントの設定によって動的にセットされます。

- メモリ保護:コードとデータを内部(ソフトウェア)攻撃と外部攻撃から保護するために使用される主要なセキュリティ機能
- ソフトウェア隔離:内部攻撃を避けるためのプロセス間保護
- インタフェース保護:シリアルまたはデバッグポートなど、デバイスのエントリポイントを保護できます。
- システムモニタリング:デバイスの外部改ざんの試みや異常な動作を検出します。

4.1 Armv8-M アーキテクチャの TrustZone®

Armv6 または Armv7 アーキテクチャ(Cortex-M0、M3、M4、および M7)に基づくマイクロコントローラは、ファームウェアとリソースの隔離のほとんどをソフトウェア実装に依存しています。本書で後述しますが、これらのメカニズムは、堅牢ですが、セキュアファームウェアと非セキュアファームウェアを同時実行できるほど柔軟ではありません。

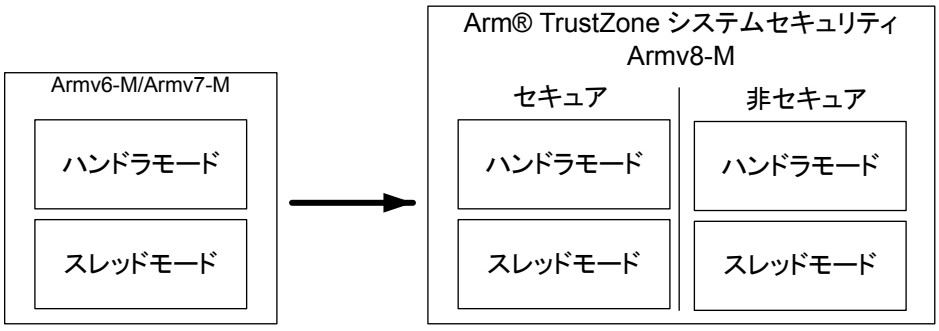
Armv8-M アーキテクチャは、Arm マイクロコントローラに新しいセキュリティの模範をもたらします。マイクロコントローラのシステムレベルで TrustZone テクノロジを実装して、実行時の堅牢な隔離を通じて、信頼できるファームウェアの開発を可能にします。

TrustZone テクノロジは、セキュアドメインと非セキュアドメイン用のデュアルレジスタバンクを持つプロセッサ(Cortex-M23 または Cortex-M33)と、セキュア属性をシステム全体(ペリフェラルとメモリ)に伝播するバス・インフラストラクチャ(AHB5)に依存します。

TrustZone は、実行時の堅牢かつ柔軟なセキュリティ制御を目的としています。セキュアドメインから非セキュアドメインおよびその逆の切り替えは、数サイクルのペナルティで容易に行われます。アプリケーションプロセッサの Cortex-A 用の TrustZone と同様のハイパバイザは不要です。

セキュアモードは既存の Thread および Handler モードとは別です。各セキュアモードに Thread または Handler モードがあります(下の図を参照)。

図 3. Armv8-M ThrustZone 実行モード



Armv8 TrustZone 上で実行する典型的なファームウェアアーキテクチャでは、非セキュアドメインはアプリケーションと OS のタスクを実行し、セキュアドメインはセキュア・アプリケーションとシステムの信頼の基点(ルート・オブ・トラスト)メカニズムを実行します。

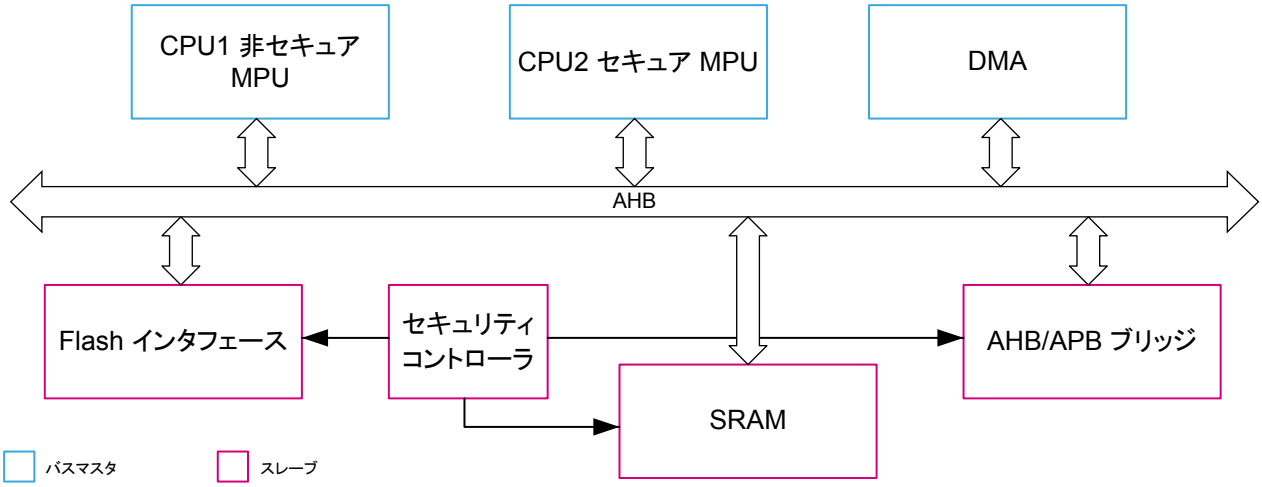
4.2 デュアルコアのセキュリティ

デュアルコア製品では、一方のコアはセキュアとして機能し、もう一方のコアは非セキュアとして機能します。特にデュアルコア STM32WL デバイスなどの一部の製品では、セキュア属性をメモリおよびペリフェラルに伝達するハードウェア・サポートを備えており、堅牢なランタイム分離実装が可能です。

デュアルコア STM32WL デバイスには専用のセキュリティ・コントローラが追加され、アイソレーションが容易に行えます。

属性ユニットを使用する代わりに、セキュリティ保護可能な CPU2 専用のセキュア NV メモリが Flash メモリ・インタフェース設定で定義されます。DMA などの主要ペリフェラルは、セキュアなコンテキストを伝達する必要があります(UM2643 を参照)。このハイブリッド・アーキテクチャの使用方法の詳細については、「UM2643 Getting started with STM32CubeWL for STM32WL Series」を参照してください)。

図 4. デュアルコア・システム・アーキテクチャの簡略図



4.3 メモリ保護

メモリ保護は、システムのセキュリティを考えるとき、最も重要なことです。機密コードやデータを含んでいるメモリは、予期しないインタフェース（デバッグポートなど）や無許可のプロセス（内部の脅威）からアクセス可能であってはなりません。

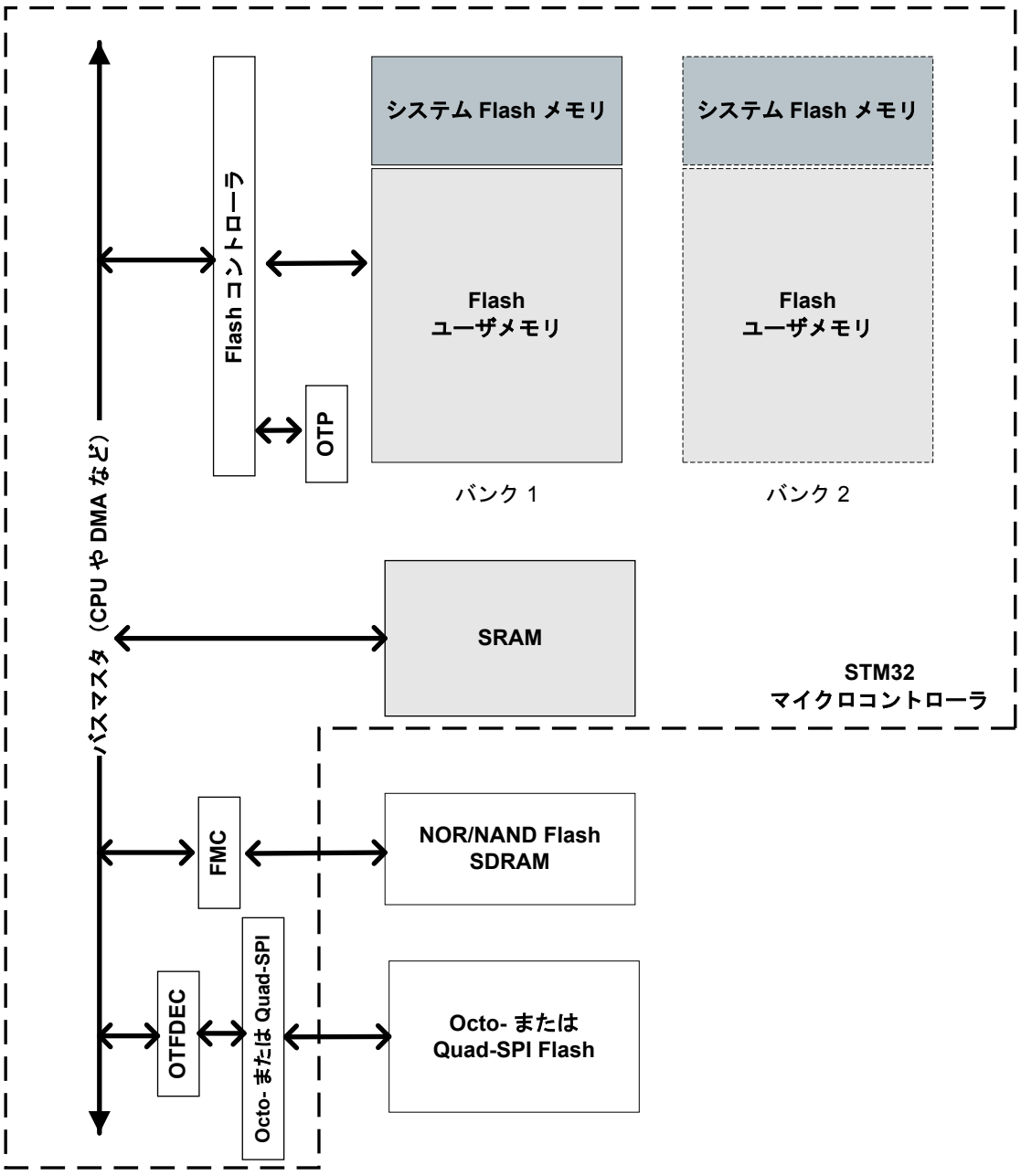
保護する資産（コードまたはデータ）に応じて、さまざまなメカニズムを設定して、無許可アクセス（外部ポート、内部プロセス）の発生源に応じて、または保護するメモリタイプ（Flash、SRAM、または外部メモリ）に応じて、保護を確立することができます。

アクセスフィルタリングは、メモリインタフェース（Flash コントローラ）、バスコントローラ IP（ファイアウォール）、またはコア MPU を通じて（使用可能な場合）実行できます。独自仕様の保護（セキュア非表示保護、PCROP、WRP、RDP）の詳細については、セクション 6 STM32 セキュリティ機能を参照してください。

内蔵 Flash メモリ、内蔵 SRAM、および外部メモリは、さまざまな目的で設計されています。それぞれの保護メカニズムは、このような違いを反映しています。

下の図に、マイクロコントローラにおけるメモリアクセスアーキテクチャの概略を示します。

図 5. メモリタイプ



下の表に、各タイプのメモリの特徴と一般的な保護機能の要約を示します。

表 5. メモリタイプと関連する保護

メモリ	タイプ	説明	保護
システム Flash メモリ	。内部 。NVM 。ROM	Flash メモリの ROM 部分。デバイスブートローダとその他の ST サービスを内蔵します。	更新（消去/書込み）できません。 一部の部分は読出しもできません。
ユーザ Flash メモリ	。内部	ユーザアプリケーションの Flash メモリ。	内部保護: • RDP

メモリ	タイプ	説明	保護
	。NVM		• WRP（SRAM 用ではない） • TrustZone • PCROP（SRAM 用ではない） • OTP（SRAM 内にない） • ファイアウォール • セキュア非表示保護（SRAM 用ではない） • MPU
SRAM	。内部 。揮発性	スタック、ヒープ、またはバッファ用の作業メモリ。内部または外部の不揮発性メモリからダウンロードされた ファームウェアの実行に使用できます。	
NAND、NOR、Octo-SPI、または Quad-SPI メモリ	。外部 。NVM	アプリケーションまたはデータストレージ用の追加メモリ	暗号化 書込み保護（Flash デバイス） TrustZone
SDRAM	。外部 。揮発性	アプリケーション実行用の追加 RAM。	暗号化

4.3.1 システム Flash メモリ

STM32 MCU では、システムメモリは内蔵 Flash メモリの読出し専用部分（ROM）です。ST ブートローダ専用です。この領域に追加のセキュアサービス（RSS）を含んでいるデバイスもあります。この部分は変更できず、認証性と完全性が保証されます。ブートローダは機密アルゴリズムを含まないため、読出し可能です。RSS の一部の部分は隠されていて、ユーザが読み出すことはできません。

システム Flash メモリの保護属性を変更することはできません。

4.3.2 ユーザ Flash メモリ

これは、ファームウェアと不揮発性データの格納に使用されるメインユーザメモリです。内蔵 Flash メモリの一部であり、すべての STM32 マイクロコントローラ で使用可能なメモリ保護機能のセットによって保護できます。

外部攻撃

内蔵 Flash メモリは、外部 Flash メモリと違って、外部攻撃に対する保護が容易です。RDP によるデバッグポートアクセスを無効にし、接続インタフェースの制御されたアクセスにより、外部からの十分な隔離が可能です。

関連する保護：RDP によるデバッグアクセスの無効化

内部攻撃

メモリの内部読出しまたは書込みアクセスは、デバイス SRAM または信頼できないライブラリに注入されたマルウェアから行われることがあるため、重要なコードおよびデータは許可されたプロセスによってのみアクセス可能でなければなりません。

関連する保護：PCROP、MPU、ファイアウォール、セキュア非表示保護、または TrustZone

未使用メモリの保護

コードの改ざんやインジェクションを防止するために、使用されていない領域についても、Flash メモリには常にデフォルトで書込み保護が設定されなければなりません。使用されていないメモリは、ソフトウェア割込み（SWI）オペコード、不正オペコード、NOP など、既知の値で埋めておくのも良い方法です。

関連する保護：MPU または WRP

エラーコード訂正（ECC）

Flash メモリはエラー検出と訂正が可能な ECC を備えている場合があります（最大 2 ビットのエラー検出と 1 ビットのエラー訂正）。安全機能とみなされていますが、フォールト・インジェクションに対する補足的保護としても機能します。

4.3.3 内蔵 SRAM

内蔵 SRAM は、デバイスのワーキングメモリです。実行時にスタック、ヒープ、グローバルバッファ、および変数に使用されます。SRAM には、ウェイトステートなしの最大システムクロック周波数で、バイト、ハーフワード（16 ビット）、またはフルワード（32 ビット）によるアクセスが可能です。

コード実行

より高速な性能を必要とするファームウェアの一部を、ユーザ Flash または外部 Flash メモリからダウンロードして、SRAM から実行できます。SRAM からコードを実行するもう一つの理由は、暗号化された外部 Flash メモリをその場で復号せずにデバイス上で使用するときです。コードは SRAM 内部で復号されてから実行されます。そのため、コードを含んでいる SRAM アドレス範囲では、適切なメモリ保護を有効にする必要があります。SRAM でコードを実行する必要がないときには、MPU で適切な属性（決して実行しない）を設定することによって、マルウェアの実行を防止することをお勧めします。

関連する保護：MPU またはファイアウォール

SRAM の消去

SRAM には機密データや、何らかの秘密を取得するための一時的な値が含まれることがあります。典型的な例は、保護された Flash メモリ領域から SRAM への平文での秘密暗号鍵の転送です。機密データを操作する関数の処理後はすぐに、作業バッファと変数を明示的に消去することを強くお勧めします。

注 リセットの場合、STM32 マイクロコントローラ では SRAM の自動消去が可能です(各デバイスのリファレンスマニュアルを参照)。一部の製品では、RDP がセットされている場合、SRAM の一部は外部アクセスや信頼できないブート(SRAM ブート)から保護されます。

書込み保護機能

書込み保護を使用して、領域の一部を隔離して別のプロセスによって破損されるのを防いだり、オーバーフロー攻撃を防いだりすることができます。オーバーフロー攻撃は、目的のバッファサイズより多くのデータを書き込むことで構成されます(インタフェースポートを通じたデータ転送時など)。境界チェックが実行されない場合、バッファより上のメモリアドレスが破損し、このようにしてマルウェアを注入できます。この保護は、主にコードの実行に使用される SRAM 領域でのみ機能します(この保護はデータに対しては実用的ではありません)。SRAM 書込み保護は、一部の STM32 シリーズの SRAM2 領域に対してのみ使用できます(セクション 6.1 セキュリティ機能の概要 および製品のリファレンス・マニュアルを参照してください)。

関連する保護:MPU、TrustZone、または SRAM 書込み保護(いくつかの STM32 シリーズでのみ使用可能)

パリティチェックと ECC

SRAM のパリティチェックにより、潜在的なエラーをワード(32 ビット)単位で制御できます。バイトあたり余分な 1 ビットがメモリの内容に追加されます(データバス幅は 36 ビット)。これは、Class B や SIL 標準などで必要とされるメモリの信頼性を高めるためです。ECC は、より高度であり、SECDED 機能を備えていますが、特定の マイクロコントローラ製品上の SRAM でのみ使用可能です。無効にすることはできず、デフォルトで故障保護を高めます。

4.3.4 外部 Flash メモリ

外部 Flash メモリは、専用インタフェース(NAND、NOR、Octo-SPI、または Quad SPI)を通じてマイクロコントローラに接続されます。内蔵 Flash メモリと同様にコードとデータを含みますが、外部ストレージには秘匿性(内容の保護)と認証性(デバイス保護)の問題が生じます。ハードウェア保護は、内容の消去または変更を避けるための書込みロックに限られます。それ以上の保護は、暗号化アルゴリズムによって与えられます。認証されていないファームウェアの実行を避けるには、少なくとも内容に署名が必要です。暗号化は、内容が機密の場合にのみ必要です。

内蔵コードは、その場で実行するか、SRAM にロードしてから実行することができます。暗号化されたファームウェアのその場での実行は、デバイスにオンザフライ復号機能がある場合にのみ可能です。そうでない場合、ファームウェアは SRAM にロードしたときに復号する必要があります。復号化されたコードまたはその一部が読出し(RDP2)から保護されない場合、コードの機密性が侵害されます。暗号化と完全性保護を組み合わせることも推奨されます。

関連する保護:OTFDEC

4.3.5 STM32 メモリ保護の概要

さまざまなケースを考慮して、いくつかの STM32 機能を使用できます。これらを、それぞれの適用範囲とともに下の表にリストします(セクション 6 STM32 セキュリティ機能 の説明を参照)。

表 6. STM32 内蔵メモリ保護機能の範囲

機能	外部攻撃保護	内部攻撃保護	Flash メモリ	SRAM
RDP	可能	不可	可能	可能
ファイアウォール	不可	可能	可能	可能
MPU	不可	可能	可能	可能
PCROP	可能	可能(読出し/書込み)	可能	不可
WRP (1)	可能	可能	可能	不可
HDP	可能	可能	可能	可能(実行)(2)
TrustZone	可能	可能	可能	可能

1. 書込み保護は RDP レベル 2 ではない場合にのみ解除できます。
2. SRAM はセキュア・コード実行時にのみ、セキュア領域によって保護されます。セキュア領域を離れる前に消去する必要があります。

4.4 ソフトウェアの隔離

ソフトウェアの隔離とは、さまざまなプロセスを互いから保護するランタイムメカニズムを指します(プロセス間保護)。これらのプロセスは、順に、または同時に(オペレーティングシステムのタスクなど)実行できます。SRAM でのソフトウェアの隔離により、各プロセスのそれぞれのスタックと作業データは、他のプロセスからはアクセスできません。このプロセス間保護は、Flash メモリ内のコードと不揮発性データに拡張することもできます。

ソフトウェアの隔離の目的:

- プロセスが別の機密プロセスの実行を探るのを防ぎます。
- メモリリークやオーバーフロー(正しくないメモリ管理実装)によるスタック破損からプロセス実行を保護します。

このメモリ保護は、下の表と セクション 6 STM32 セキュリティ機能 に詳しく説明されているさまざまなメカニズムによって達成されます。

表 7. ソフトウェア隔離メカニズム

保護	タイプ	隔離
MPU	動的	特権属性により (1)
ファイアウォール	静的	バスアドレスハードウェア制御により
セキュア非表示保護	静的	リセット時のプロセス・プリエンプション(横取り)
デュアルコア	静的	コア ID 順(2)
TrustZone	静的保護と動的保護	コアからすべてのリソースに伝播されるセキュア属性によって

1. 属性保護は CPU アクセス専用であり、他のバスマスタ(DMA など)は考慮されません。
2. CPUID を読み出すと、現在コードを実行している CPU が示されます。例は HAL_GetCurrentCPUID 関数で確認できます。

4.5 デバッグポートとその他のインタフェース保護

デバッグポートは内部リソース(コア、メモリ、およびレジスタ)へのアクセスを提供するため、最終製品では無効にする必要があります。最も基本的な外部攻撃であり、セキュアで変更不可なファームウェア(セクション 5.2.1 セキュア・ブート(SB)を参照)によって JTAG(または SWD)ポートを無効化することによって、またはなるべくなら機能を永続的に無効化することによって(RDP2 の JTAG ヒューズ)、容易に避けることができます。

その他のシリアルインタフェースも使用できます。ブートローダが使用可能な場合、I2C、SPI、USART、または USB-DFU を通じてデバイスの内容にアクセスできます。実行時にインタフェースが開いている場合、アプリケーション転送プロトコルは、そのアクセス機能(操作モードやアドレスアクセス範囲など)を制限する必要があります。

関連する STM32 の機能:

- 読出し保護(RDP)
- 使用していないポートを無効にします。
- ブートローダアクセスを禁止します(STM32 デバイスの RDP によって設定)。

4.6 ブート保護

ブート保護は、システム内の最初のソフトウェア命令を保護します。攻撃者がデバイスブートアドレスの改変に成功した場合、自分のコードを実行して、初期の動的保護設定を迂回したり、セキュリティで保護されていないブート・ローダ・アプリケーションにアクセスして、デバイスメモリにアクセスしたりできます。

マイクロコントローラでは、通常、ブートの設定が可能であり、ユーザアプリケーションから、ブート・ローダ・アプリケーションから、または SRAM にあるファームウェアから起動するかを選ぶことができます。ブート保護は、信頼できるコードへの単一のエントリポイントに依存し、これはユーザアプリケーション、または使用可能な場合はセキュアサービス領域(RSS)です。

関連する STM32 の機能:

- 読出し保護(RDP)
- 固有の起動エントリ
- セキュア非表示保護(HDP)
- TrustZone

4.7 システム監視

デバイスの電源供給と環境の監視を設定して、不正動作を避け、対応する対応策を取ることができます。タンパ検出などのメカニズムは、セキュリティ専用です。その他のメカニズムは主に安全上の理由で使用されますが、セキュリティにも役立ちます。たとえば、パワーダウンまたは外部クロック切断の検出は、意図的でない場合もありますが（安全性）、攻撃を明らかにする場合もあります（セキュリティ）。

タンパ検出は、システム/ボードレベルの侵入を検出するために使用されます。コンスーマ製品の筐体を開くと、MCU ピンで検出され、適切なアクションがトリガされます。内部タンパセンサは、異常な電圧、温度、その他のパラメータを検出できます。

クロック・セキュリティ・システムは、外部オシレータ障害から保護するために使用されます。外部クロックで障害が検出されると、マイクロコントローラは安全に実行するために内部クロックに切り替わります。割込み信号によって、ファームウェアはクロック障害イベントに対応できます。

電源供給と電圧レベルを監視して、異常な低電圧レベルを検出できます。特定の電圧レベル未満では、正常な動作を確保できず、フォールト・インジェクション攻撃の兆候の可能性があります。

デバイス温度は内部センサで測定できます。情報は内部 ADC チャネルを通じてデバイスにフィードバックされます。監視アプリケーションは、温度範囲に応じて適切なアクションを取ることができます。温度上昇は、フォールト・インジェクション攻撃手口の一部である可能性があります。

関連する STM32 の機能:

- タンパ保護（RTC コンポーネント）
- クロック・セキュリティ・システム
- 電源供給の監視
- 温度センサ

5 セキュア・アプリケーション

セキュアシステムを作成するためには、セキュア・ファームウェア・アーキテクチャ実装でハードウェア機能を使用する必要があります。業界標準のソリューションは、IoT エコシステム向けに Arm によって提案された PSA です。ST 独自のソリューションは、セキュア・ブート（SB）とセキュア・ファームウェア・アップデート（SFU）です。

この節では、信頼の基点と信頼のチェーンの概念を定義してから、以下に示されている機能を実装する以下の典型的なセキュア・アプリケーションについて述べます。

- セキュア・ブート
- セキュア・ファームウェア・アップデート（SFU）
- セキュア・ストレージ
- 暗号化サービス

これらのアプリケーションは、暗号化と密接な関係を持ちます。すべての暗号化方式は、共有秘密鍵、公開鍵、およびハッシュ生成という 3 つの概念に基づきます。暗号化の基本を 付録 A で説明します。暗号化 — 主な概念。

注
- ユーザマニュアル Getting started with the X-CUBE-SBSFU STM32Cube Expansion Package（UM2262）には、SB と SFU の実装例が示されています（www.st.com/en/product/x-cube-sbsfu）。
- ユーザマニュアル Getting started with STM32CubeL5 TF-M application（UM2671）では、STM32L5 シリーズの マイクロコントローラ での TF-M 実装例が説明されています。
- ユーザマニュアル Getting started with STM32CubeU5 TF-M application（UM2851）では、STM32U5 シリーズの マイクロコントローラ での TF-M 実装例が説明されています。

5.1 信頼の基点および信頼のチェーン

信頼の基点および信頼のチェーン(チェーン・オブ・トラスト)の原則は、ほぼすべてではないにしても、多くのセキュアシステムで共通です。明らかに自由に拡張可能であり、本質的に効率的かつ柔軟です。

信頼のチェーンは、適応可能なコンポーネントのセットとして構築され、各コンポーネントのセキュリティが別のコンポーネントによって保証されます。信頼の基点は、全体的なセキュリティが依存するチェーンの始点となるアンカーです。

セキュア・ブート実装は、デバイスへの単一のエントリポイントでなければならず、リセット後は変更不可能なコードをセキュアモードで開始しなければなりません。その後、後続の機能を認証して、ファームウェアの次の部分を実行します。これにより、続くチェーンリンクを安全にテストするために必要な追加機能が有効になります。たとえば、揮発性メモリ保護を設定して、セキュア・ストレージ・サービスで使用できるようにします。

5.2 ST 独自の SBSFU ソリューション

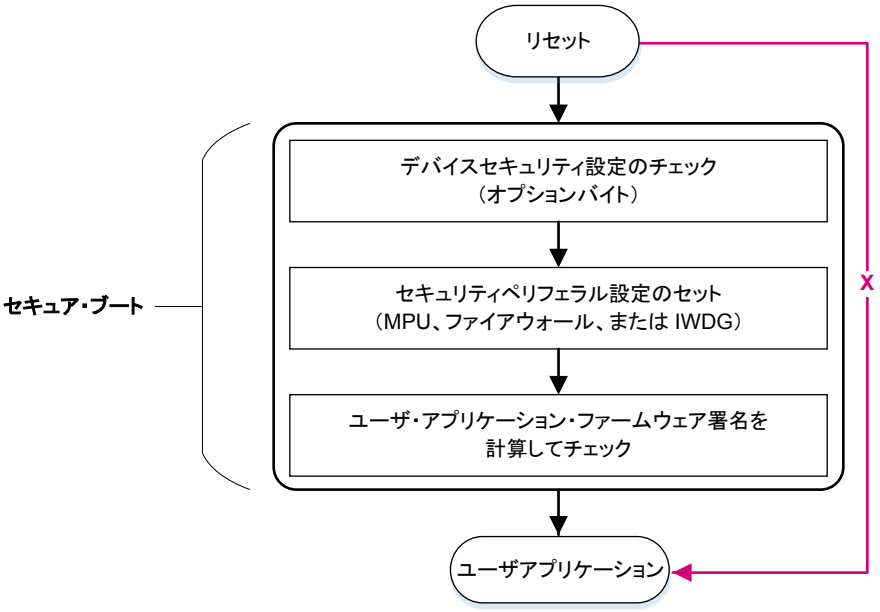
5.2.1 セキュア・ブート（SB）

SB アプリケーションは、リセット時にユーザアプリケーションの前に実行されます。セキュリティの最初の段階を提供し、その後、システムのグローバルな信頼のチェーンの確保を担当します。

SB の主な機能:

- STM32 のセキュリティ構成をチェックし、ランタイム保護をセットアップします。
- 実行されるユーザ・アプリケーション・イメージの完全性と認証性を確認します（下の図を参照）。

図 6. セキュア・ブート FSM



デバイスセキュリティのチェック

SB アプリケーションのこの部分は、静的構成が正しいかどうかをチェックして、動的構成を設定します。静的セキュア構成は、オプションバイトによって定義されます（RDP、PCROP、WRP、およびセキュア非表示保護）。動的保護はプログラムが必要です（ファイアウォール、MPU、タンパ検出、および IWDG）。

完全性および認証性チェック

ファームウェアの完全性チェックは、アプリケーションイメージのハッシング（MD5、SHA1、または SHA256 ハッシュアルゴリズム）と、ダイジェストと期待値の比較によって行われます。このように、アプリケーションファームウェアはエラーフリーとみなされます。

認証性チェックは、期待されるタグがファームウェア所有者とデバイス間の共有鍵で暗号化された場合に追加されます。この鍵はデバイスの保護領域に格納されます。

保護属性

SB ファームウェアがその役割を果たすには、以下の属性を備えている必要があります。

- デバイスの一意エントリポイントでなければなりません（迂回なし）。
- コードが不変でなければなりません。
- 機密データ（証明書やアプリケーション署名など）へのアクセスを持たなければなりません。

SB の最も重要な部分は、ファイアウォール、MPU、またはセキュア非表示保護など、プロセスおよびデータ隔離機能を利用します。実装は、STM32 シリーズで使用可能な機能に依存します。

5.2.2 セキュア・ファームウェア・アップデート（SFU）

SFU は、実用現場でのファームウェア更新を安全に実行して、新しいファームウェアイメージをデバイスにダウンロードできるようにします。

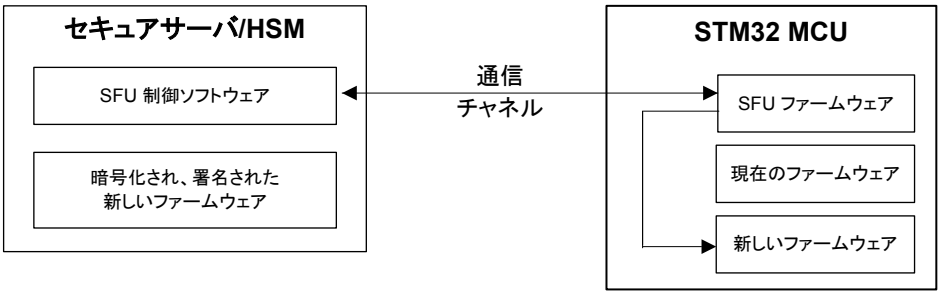
ファームウェア更新は、次の 2 つの当事者を保護しなければならない重要な操作です。

- デバイス所有者：目的は、（意図的にしろ、そうでないにしろ）デバイスを傷つける恐れがある破損したファームウェアのロードを回避することです。
- アプリケーション所有者（OEM）：ファームウェアのクローニングや認証されていないデバイスへのロードを防止する必要があります。

アーキテクチャ

SFU 転送には、ファームウェア所有者（OEM）と更新されるデバイスという 2 つのエンティティが関与します（下の図を参照）。通信チャネルは盗聴の対象であり、一般に安全とはみなされないため、全体的なセキュリティの責任は、送信者（ファームウェア所有者サーバ）と受信者（デバイス）とで共有されます。

図 7. セキュアサーバ/デバイス SFU アーキテクチャ



アプリケーション

OEM 側からは、セキュアサーバは、暗号化され（秘匿性が必要な場合）、署名されたファームウェアを認証済みのデバイスに送信する責任があります。

デバイス上で実行する SFU アプリケーションには、次のような責任があります。

- インストール前に、ロードされたイメージの認証および完全性チェック
- 秘匿性が必要な場合は、新しいファームウェアの復号
- 新しいファームウェアのバージョンのチェック（アンチロールバック・メカニズム）

5.3 ARM TF-M ソリューション

Arm Trusted Firmware は、Armv8-M アーキテクチャとともにセキュア Cortex M-33 コアが導入されたときに、しばらく存在していました。よりコンパクトな、PSA 標準の TF-M オープンソース実装がリファレンス・セキュア・ファームウェア・フレームワークとして提供されました。

STM32L5 デバイスなど、ARMv8 アーキテクチャを利用する ST MCU の場合、SBSFU は TF-M ソリューションに置き換えられます。

TF-M 自体のマニュアルとしては、コードのコメントのほか、ARM のリソースを使用してください。

STM32L5 シリーズ MCU での TF-M 統合については、ファームウェアのユーザマニュアルを参照してください。

SBSFU から TF-M に移行するユーザは、基本的な機能の違いを比較した比較表を参照してください。

表 8. 基本機能の違い

機能	SBSFU v2.3.1	STM32L5 上の TF-M
RoT サービス	信頼の伝播	HUK および証明情報管理
ブートローダ	可能	不可
FW イメージ暗号化	OTFDEC を含むいくつかのオプション	含まれない
暗号鍵管理	NMV 内の鍵、静的コード	揮発性の鍵、更新可能コード
セキュア・ストレージ	キー管理	HUK に基づき可能
初期証明	不可	可能
セキュア・エレメントのサポート	STSAFE-A110	不可

2 つのソリューションの要件は異なり、アーキテクチャに違いがあります。交換可能または競合するわけではありません。

STM32 デバイスでの TF-M の構築の詳細については、STM32 TF-M ユーザマニュアル（UM2671）を参照してください。

詳細な比較については、AN5447 アプリケーションノートを参照してください（X-CUBE-SBSFU 対 TF-M の比較のセクション）。

5.4 製品認証

さまざまなセキュア・アプリケーションにおいて、認証により、機能がセキュアな方法で実行されていることを証明する必要がある場合がよくあります。マイクロコントローラ または マイクロコントローラ をベースとしたシステムへの認証ステータスの付与は、評価やテストに基づいて、独立機関または行政機関によって行われます。

STM32 マイクロコントローラに関連する認証と評価は次のとおりですが、これに限られません。

- PSA - Platform Security Architecture（プラットフォーム・セキュリティ・アーキテクチャ） - IoT セキュリティ、マイクロコントローラ認証、3 つのレベルの評価を中心に Arm が管理
  - STM32L4 シリーズはレベル 1、TF-M を搭載した STM32L5 シリーズはレベル 2、TF-M を搭載した STM32U5 シリーズはレベル 3 の認証を取得しています。
  - Arm PSA 認証可能なセキュリティ・レベルを達成する方法については、ユーザ・マニュアル「STM32U585 security guidance for PSA Certified™ Level 3 with SESIP Profil」（UM2852）を参照してください。
- SESIP - Security Evaluation Standard for IoT Platforms（IoT プラットフォームのセキュリティ評価標準） - いくつかの主要なセキュリティ評価ラボで採用されている国際的な方法論、5 レベル
  - SBSFU または TF-M を使用するシステムは、STM32L4、L4+、L5、および U5 シリーズデバイスによりレベル 3 に準拠しています。
- PCI - Payment Card Information（ペイメント・カード情報） - 販売時点管理（POS）アプリケーションを対象とした重要なセキュリティ規格
  - STM32L4 シリーズ・デバイスなどを使用したシステム評価の良好な記録
- CC - Common Criteria（共通基準） - ユニバーサル認証規格には、開発、テスト、および文書の品質について高い基準が必要です。

6 STM32 セキュリティ機能

この節では、以前の節で示したさまざまなセキュリティ概念に対応して、高いレベルのセキュリティを達成できるすべてのSTM32 機能について述べます。

6.1 セキュリティ機能の概要

静的保護と動的保護

保護機能が静的か動的かを区別することができます。

- 静的保護とは、オプションバイトでセットされる機能を指します。設定はパワーオフ時も保持されます。
  静的保護は RDP、PCROP、WRP、BOR、OTP、およびセキュア非表示保護(使用可能な場合)です。
- 動的(またはランタイム)保護は、リセット時にステータスを保持しません。ブートのたびに(たとえば、セキュア・ブート(SB) 時に)設定する必要があります。
  STM32 によって提供される動的保護は、MPU、タンパ検出、およびファイアウォールです。
  その他の動的保護は、セキュリティと安全性の両方に関連します。異常な環境動作は偶発的(安全性)な場合と、攻撃を実行するための意図的な場合があります。これらの保護には、クロックおよび電源監視システム、メモリ整合性ビット、および独立型ウォッチドッグ(IWDG)が含まれます。

STM32 シリーズ別セキュリティ機能

下の表に、STM32 シリーズで使用可能な機能を示します。

表 9. STM32Fx シリーズのセキュリティ機能

機能	STM32F0	STM32F1	STM32F2	STM32F3	STM32F4	STM32F7
FPU 内蔵型 Cortex-M4	M0	M3	M3	M4	M4	M7
RDP 追加保護	+ バックアップレジスタ	2 レベル RDP のみ (1)	+ バックアップ SRAM	+ バックアップレジスタ	+ バックアップ SRAM	+ バックアップ SRAM
Flash WRP	セクタ別（4 KB）	ページ別（4 KB または 8KB）	セクタ別（16 KB、64 KB、または 128 KB）	セクタ別（4 KB）	セクタ別（16 KB、64 KB、または 128 KB）	セクタ別（16 KB、64 KB、128 KB、または 256 KB）
SRAM WRP	不可	不可	不可	不可	不可	不可
PCROP	不可	不可	不可	不可	セクタ別	セクタ別
HDP	不可	不可	不可	不可	不可	不可
ファイアウォール	不可	不可	不可	不可	不可	不可
MPU	不可	可能(2)	可能	可能	可能	可能
OTP	不可	不可	可能	可能	512 バイト	528 ~ 1040 バイト
OTFDEC	不可	不可	不可	不可	不可	不可
固有のブート・エントリ(3)	不可	不可	不可	不可	不可	不可
セキュア・ブート：システム Flash	不可	不可	不可	不可	不可	不可
内部タンパ検出	不可	不可	不可	不可	不可	不可
ハードウェア暗号化：アクセラレータ	不可	不可	AES、HASH	不可	AES、HASH	AES、HASH
ハードウェア暗号化：TRNG	該当なし	該当なし	SP800-90-A	該当なし	SP800-90-A	SP800-90-A
セキュア・ソフトウェア：セキュア・ファームウェア・インストール	不可	不可	不可	不可	不可	不可
セキュア・ソフトウェア：SBSFU	不可	不可	不可	不可	可能	可能
セキュア・ソフトウェア：TF-M	不可	不可	不可	不可	不可	不可
セキュア・ソフトウェア：KMS	不可	不可	不可	不可	不可	不可

1. STM32F1 シリーズの RDP は Flash メモリ保護に限られます。RDP はセットされている（RDP1）かセットされていない（RDP0）かのいずれかです。RDP レベル 2 は実装されていません。
2. この MPU を備えているのは特大容量部品のみです。
3. UBE なしは、ブートが RDP レベル 2 のみに依存することを意味します。UBE が「可能」の場合、専用のブート・ロック・サービスが存在することを意味します。

表 10. STM32Lx および STM32Ux シリーズのセキュリティ機能

機能		STM32L0	STM32L1	STM32L4 STM32L4+	STM32L5	STM32U5
FPU 内蔵型 Cortex-M4		M0	M3	M4	M33 TrustZone	M33 TrustZone
RDP 追加保護		+ EEPROM	+ EEPROM	+ バックアップレジスタ + SRAM2	RDP 4 レベル + バックアップレジスタ + SRAM2	RDP 4 レベル + バックアップレジスタ + SRAM2
Flash WRP		セクタ別（4 KB）	セクタ別（4 KB）	領域別（2 KB の粒度） バンクあたり 1 つの領域	最大 4 つの保護領域（2 KB または 4 KB の粒度）	バンクあたり 2 つの領域、ページ範囲で定義
PCROP		セクタ別	セクタ別	領域別（8 バイトの粒度） バンクあたり 1 つの領域	不可	不可
HDP[1]		不可	不可	不可	TrustZone セキュアドメイン内に最大 2 つのセキュア非表示領域（HDP）	TrustZone セキュア・ドメイン内に、バンクあたり 1 つのセキュア非表示領域（HDP）
ファイアウォール		可能	不可	可能	TrustZone ベース	TrustZone ベース
MPU		可能	可能	可能	可能	可能
固有のブート・エントリ[2]		不可	不可	不可	可能	可能
内部タンパ検出		不可	不可	不可	可能	可能
IWDG		可能	可能	可能	可能	可能
デバイス ID（96 ビット）		可能	可能	可能	可能	可能
ハードウェア暗号化アクセラレータ	対称	AES	AES	AES	AES、OTFDEC	AES、セキュア AES、OTFDEC
	非対称	不可	不可	不可	PKA	PKA
	サポート	不可	不可	HASH、TRNG	HASH、TRNG	HASH、TRNG
保護付き RAM		不可	不可	1 KB の粒度の SRAM2	1 KB の粒度の SRAM2	1 KB の粒度の SRAM2

1. HDP は、製品によってセキュアユーザメモリ、スティッキー領域、またはセキュアメモリと呼ばれます。
2. UBE なしは、ブートが RDP レベル 2 のみに依存することを意味します。UBE が「可能」の場合、専用のブート・ロック・サービスが存在することを意味します。

表 11. STM32H7、STM32G0、STM32G4、STM32WB、および STM32WL シリーズのセキュリティ機能

機能		STM32H7 シリーズ			STM32G0	STM32G4	STM32WB	STM32WL
		STM32H72x/73x	STM32H74x/75x	STM32H7Ax/Bx				
FPU 内蔵型 Cortex-M4		M7			M0+	M4	M4 および M0+	M4 および M0+(1)
RDP 追加保護 (2)		+ バックアップ SRAM + バックアップレジスタ + OTFDEC	+ バックアップ SRAM + バックアップレジスタ	+ バックアップ SRAM + バックアップレジスタ + OTFDEC	+ バックアップレジスタ	+ バックアップレジスタ + CCM-SRAM	+ バックアップレジスタ + SRAM2	+ バックアップレジスタ + SRAM2
WRP		セクタ別（128 KB）		4 個の 8 KB セクタのグループ別	領域別（2 KB の粒度） 2 つの領域が使用可能	ページ別（2 KB または 4KB）	領域別（4 KB の粒度） 2 つの領域が使用可能	領域別（2 KB の粒度） 2 つの領域が使用可能
PCROP		領域別（256 バイトの粒度）	領域別（256 バイトの粒度） バンクあたり 1 つの領域		領域別（512 バイトの粒度） 2 つの領域が使用可能	領域別（64 ビットまたは 128 ビットの粒度） 最大 2 つの領域 (3)	領域別（2 KB の粒度） 最大 2 つの領域	領域別（1 KB の粒度） 2 つの領域が使用可能
HDP(4)		可能（セキュア・ユーザ・メモリ、256 バイトの粒度）			あり（セキュリティ保護可能なメモリ領域）	あり（セキュリティ保護可能なメモリ領域）	あり（CM0+ ファームウェア専用）	可能
ファイアウォール		不可			不可	不可	不可	可能
MPU		可能			可能	可能	可能（CM4）	可能
固有のブート・エントリ(5)		可能（セキュア・アクセスにおける固有のエントリ・ポイント）			可能（ブートロック機能）	可能	不可	可能（ブートロック機能）
内部タンパ検出		可能			可能	可能	不可	可能
IWDG		可能			可能	可能	可能	可能
デバイス ID（96 ビット）		可能			可能	可能	可能	可能
ハードウェア暗号化 (2)	対称	AES、			AES	AES	AES	AES
	非対称	不可			不可	不可	PKA	PKA
	サポート	HASH、TRNG			TRNG	TRNG	TRNG	TRNG
SRAM 書込み保護		不可			不可	A KB の粒度の CCM SRAM	SRAM2、1 KB の粒度	SRAM2、1 KB の粒度

1. M0+ コアは、一部の STM32WL デバイスでは使用できません。
2. デバイスの部品番号によって異なります。
3. 領域の数は、デバイスのカテゴリとデュアル/シングルバンク構成によって異なります。
4. HDP は、製品によってセキュアユーザメモリ、スティッキー領域、またはセキュアメモリと呼ばれます。
5. UBE なしは、ブートが RDP レベル 2 のみに依存することを意味します。UBE が「可能」の場合、専用のブート・ロック・サービスが存在することを意味します。

6.2 読出し保護(RDP)

読出し保護は、包括的な Flash メモリ保護です。内蔵ファームウェア・コードをコピー、リバース・エンジニアリング、ダンピング、デバッグ・ツールの使用、または SRAM でのコードインジェクションから保護できます。ユーザは、バイナリコードが内蔵 Flash メモリにロードされた後に、この保護を設定する必要があります。

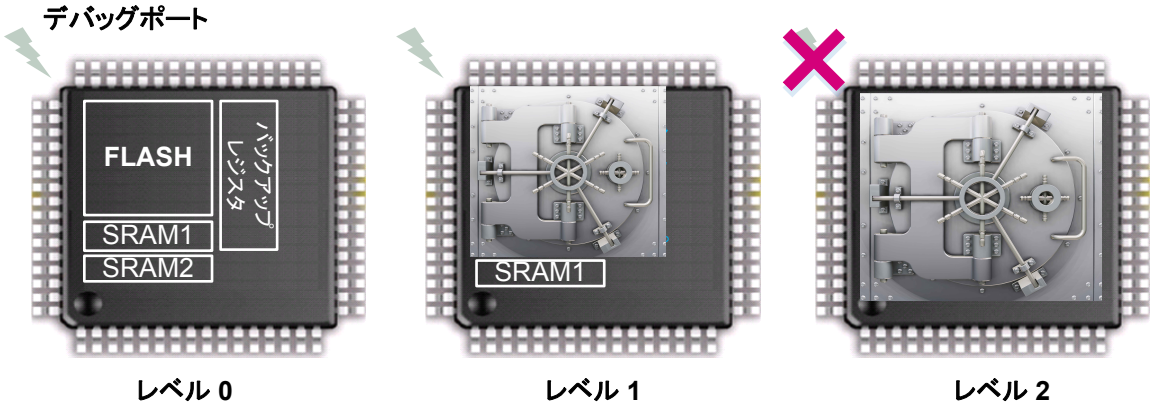
読出し保護は、すべての STM32 シリーズに適用されます。

- メイン Flash メモリ
- オプションバイト(レベル 2 のみ)

STM32 シリーズには、次のような追加の保護を使用できるものがあります。

- リアルタイムクロック(RTC)のバックアップレジスタ
- バックアップ SRAM
- EEPROM

図 8. RDP 保護の例(STM32L4 シリーズ)



RDP レベル(0、1、および 2)は、次のように定義されています。

- レベル 0:これはデフォルトの RDP レベルです。Flash メモリは完全にオープンであり、すべてのブート設定ですべてのメモリ操作が可能です(デバッグ機能、RAM からのブート、システムメモリブートローダからのブート、Flash メモリからのブート)。この設定モードではいかなる保護もないため、開発およびデバッグに適しています。
- レベル 1:SRAM またはシステムメモリブートローダからブートする場合でも、Flash メモリへのアクセス(読出し、消去、プログラム)や、デバッグ機能(シリアル・ワイヤまたは JTAG)による SRAM2 アクセスは禁止されます。このような場合、保護領域の読出し要求はバスエラーになります。
  ただし、Flash メモリからブートする場合、Flash メモリおよび SRAM2 へのアクセス(ユーザコードから)は許可されます。
- レベル 2:RDP レベル 2 が有効なとき、レベル 1 で提供されるすべての保護がアクティブであり、MCU が完全に保護されます。RDP オプションバイトとその他すべてのオプションバイトは凍結され、変更できなくなります。
  JTAG、SWV(シングルワイヤ・ビューア)、ETM、およびバウンダリ・スキャンはすべて無効になります。

ARMv8 アーキテクチャで構築されたデバイスでは、4 つ目の RDP レベルが使用可能です。

- レベル 0.5:非セキュアデバッグのみ。非セキュアな Flash メモリに対するすべての読出し/書込み操作が可能です(書込み保護がセットされていない場合)。セキュア領域へのデバッグアクセスは禁止されます。非セキュア領域へのデバッグアクセスは可能なままです。

RDP レベル回帰

RDP は常にレベルアップできます。レベル回帰は、次の結果を伴うことがあります。

- RDP レベル 1 から RDP レベル 0 に回帰すると、Flash メモリが全体消去され、SRAM2 とバックアップレジスタが消去されます。
- RDP レベル 1 から RDP レベル 0.5 に回帰すると、Flash メモリが部分的(非セキュア部分のみ)に消去されます
- RDP レベル 0.5 から RDP レベル 0 に回帰すると、Flash メモリが全体消去され、SRAM2 とバックアップレジスタが消去されます。

RDP レベル 2 では、回帰はできません。

RDP 保護された STM32 マイクロコントローラでの内部 Flash メモリの内容更新

RDP レベル 1 または 2 では、Flash メモリの内容を外部アクセス（ブートローダまたは SRAM からのブート）によって変更することはできなくなります。ただし、内部アプリケーションによる変更は常に可能です。これは SFU アプリケーションまたは（セキュリティの観点からは推奨できないとしても）シンプルなアプリケーション内プログラミングプロセス（IAP）から実行できます。

次の表に、RDP 保護の要約を示します。

表 12. RDP 保護

領域	RDP レベル	ユーザ Flash からのブート			デバッグまたは SRAM からのブートまたはブートローダー		
		読出し	書込み	消去	読出し	書込み	消去
Flash メインメモリ	0	可能	可能	可能	可能	可能	可能
	1	可能	可能	可能	不可	不可	不可
	2	可能	可能	可能	N/A	N/A	N/A
システムメモリ	0	可能	不可	不可	可能	不可	不可
	1	可能	不可	不可	不可	不可	不可
	2	可能	不可	不可	N/A	N/A	N/A
オプションバイト	0	可能	可能	可能	可能	可能	可能
	1	可能	可能	可能	可能	可能	可能
	2	可能	不可	不可	N/A	N/A	N/A
その他の保護資産 (1)	0	可能	可能	可能	可能	可能	可能
	1	可能	可能	N/A	不可	不可	不可
	2	可能	可能	N/A	N/A	N/A	N/A

1. バックアップレジスタ/SRAM

RDP を使用すべきとき

コンスーマ製品では、RDP は常に少なくともレベル 1 でセットしなければなりません。これにより、デバッグポートまたはブートローダからの基本的な攻撃を防ぎます。ただし、RDP レベル 1 では、Flash メモリの全体消去によって RDP レベルが 0 に戻り、サービス拒否が発生するリスクがあります。

より高いセキュリティレベルを持つアプリケーション（不変コードなど）を実装するには、RDP レベル 2 が必須です。副作用として、RDP レベル 2 では、顧客からの返品後などに、デバイス設定の更新ができないことがあります。

新しい RDP レベル 0.5 は、ARMv8 を備えた製品で使用可能です。非セキュア・アプリケーションのデバッグに使用され、セキュア領域境界内の内容をデバッグアクセスから保護します。この保護の詳細については、アプリケーションノート AN5347（セクション 10 TrustZone® を使用した開発推奨事項）を参照してください。

注 RDP は、すべての STM32 シリーズで使用可能です。

6.3 OTP:One-time programmable（ワンタイム・プログラマブル）

OTP は Flash メモリ内の独立した専用領域で、書込みのみまたはロックアウトができ、変更を防止できます。通常、OTP はユーザ Flash メモリのサイズに比べて小さい領域です。

この機能は、ライフサイクル管理、プロビジョニング、個人用設定、または設定に非常に役立ちます。OTP が書き込まれると、チップに物理的な損傷を与える以外にデータを消去する方法はありません。書き込まれたデータの読出しには、暗黙の制限はありません。

注 OTP は、ほとんどの STM32 シリーズで使用できます。詳細については、セクション 2 概要を参照してください。

6.4 TrustZone

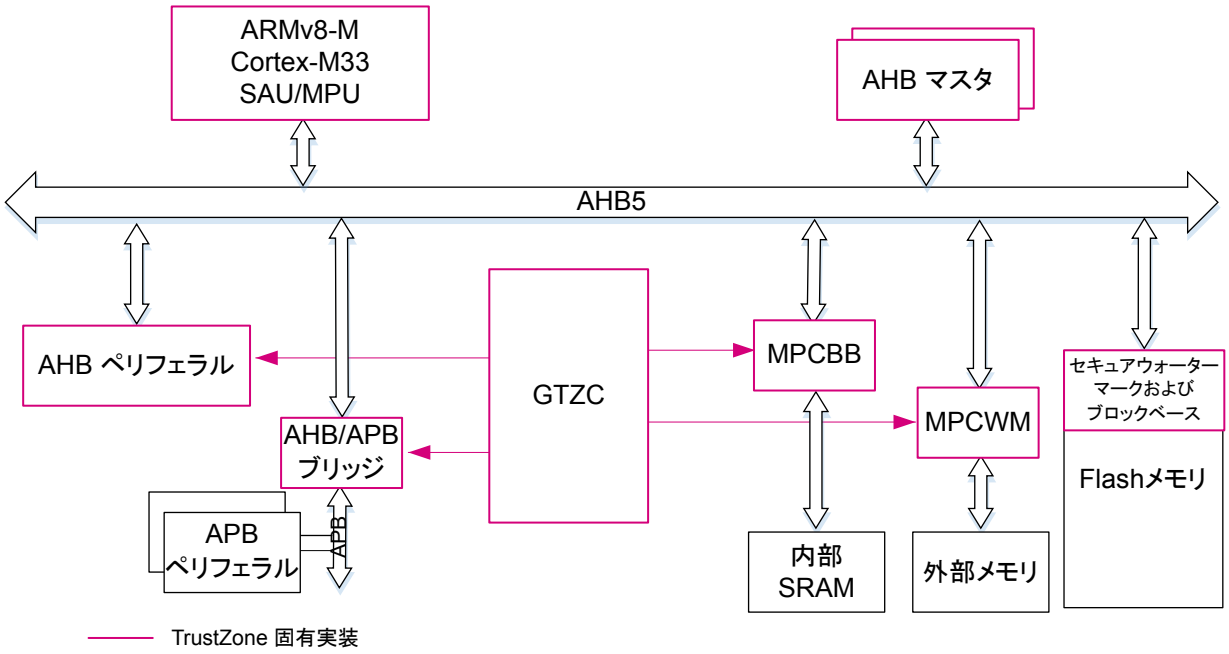
次の節では、TrustZone アーキテクチャの主な機能について説明します。詳細については、アプリケーションノート STM32L5 シリーズ TrustZone® 機能（AN5347）およびリファレンスマニュアル STM32L552xx および STM32L562xx 高度な Arm®ベースの 32 ビット MCU（RM0438）を参照してください。

ARMV8-M TrustZone アーキテクチャは、システムレベルでセキュアと非セキュアの 2 つのドメインを定義します。メモリマップ空間全体がセキュア領域と非セキュア領域に分けられます。これには、すべてのメモリタイプ（Flash、SRAM、および外部メモリ）のほか、一方のドメインまたは他方のドメインに共有（各ドメインの特定のコンテキストで）または専用のすべてのペリフェラルも含まれます。

システムレベルでは、セキュアドメインと非セキュアドメインの隔離は、以下のハードウェアメカニズムに依存します（下の図を参照）。

- 特定のコアアーキテクチャ（ARMV8-M Cortex-M33）。セキュアドメインと非セキュアドメイン用のデュアルバンクレジスタと、アドレス範囲のセキュリティステータスをアサートするセキュア属性ユニット（SAU）を持ちます。
- 実装定義属性ユニット（IDAU）。SAU を補足します。
- バス・インフラストラクチャ。トランザクションのセキュア属性と特権属性を伝播します。
- 2 つのドメイン間の分割を管理する専用ハードウェアブロック（内部 SRAM と外部 FSMC/OCTOSPI メモリおよびペリフェラルのセキュリティ属性を定義する GTZC）。

図 9. システムレベルでの TrustZone 実装



6.4.1 コア状態

コア状態は、現在実行中のコードの領域に依存します。コードがセキュア領域から実行しているときには、コアはセキュア状態です。そうでない場合、コアは非セキュア状態です。

6.4.2 セキュア属性ユニット（SAU）

SAU は（MPU として）コアに連結されたハードウェアユニットです。SAU は、AHB5 トランザクションのセキュリティ属性を設定します。トランザクションのセキュリティ属性は、メモリマップ・リソースのターゲットアドレス（メモリ領域またはペリフェラル）によって固定されます。SAU の設定に応じて、アドレスはセキュア、非セキュア呼出し可能（NSC）、または非セキュアとしてタグが付けられます。NSC はセキュアドメインのサブドメインであり、非セキュアコードが特定のエントリポイントでセキュアドメインにアクセスするためのゲートウェイを定義することができます。

SAU はセキュアファームウェアによって設定できます。ブート時に固定された構成で設定するか、セキュアファームウェアによって動的に変更することができます。

注 セキュリティ属性を IDAU（実装によって定義されたセキュア属性）によって設定されたデフォルト設定より低く変更することはできません（セキュリティ順に、セキュア > NSC > 非セキュア）。各デバイスの実装の詳細については、リファレンスマニュアルを参照してください。

アドレス・エイリアシング

セキュリティ属性は、固定されたリソースアドレスに応じて設定されます。ただし、メモリマップ・リソースは、アプリケーションに応じてセキュアまたは非セキュアとして設定できます。この明らかな矛盾を克服するために、各メモリマップ・リソースに 2 つのアドレスが割り当てられます。1 つはリソースにセキュアモードでアクセスする必要があるときに使用され、もう 1 つは非セキュアモードで使用されます。このメカニズムをアドレス・エイリアシングといいます。

アドレス・エイリアシングによって、すべてのペリフェラルのアクセスを複数の散乱した領域ではなく 2 つの領域にまとめることもできます。最後に、IDAU は、メモリマップ・リソースを次のような領域に分割します。

- ペリフェラルセキュア/非セキュア領域
- Flash メモリセキュア/非セキュア領域
- SRAM セキュア/非セキュア領域

詳細な設定については、デバイスのリファレンスマニュアルを参照してください。

6.4.3 メモリとペリフェラルの保護

SAU はトランザクションセキュリティ属性を定義し、バス・インフラストラクチャは、この属性をターゲットに伝播します。ターゲット(メモリとペリフェラル)は、セキュア属性と特権属性に応じてアクセスをフィルタするハードウェアメカニズムによって保護されます。

TrustZone システムアーキテクチャでは、2 つのタイプのペリフェラルがあります。

- TrustZone 対応ペリフェラル: AHB または APB バスに直接接続され、特定の TrustZone 動作(セキュアなレジスタのサブセットなど)を持ちます。アクセスフィルタリング制御は、これらのペリフェラルに含まれます。
- セキュリティ保護可能なペリフェラル:セキュリティ特性を定義するために、GTZC から制御される AHB/APB ファイアウォールゲートによって保護されています。

TrustZone 対応ペリフェラルは、バスマスタの役割を持つもの(DMA)、GTZC、Flash メモリコントローラ、およびシステム内で基本的な役割を持つその他のペリフェラルです。つまり、PWR、RTC、およびシステム構成ブロックです。残りのシステムペリフェラルは、セキュリティ保護可能なペリフェラルです。

GTZC

GTZC は、セキュリティ保護可能ペリフェラル、内蔵 SRAM、および外部メモリのアクセス状態を定義します。

- ペリフェラルは、セキュアまたは非セキュア(排他的に)、特権または非特権として TZSC を使用して設定できます。
- 内蔵 SRAM は、MPCBB ブロックを通じて、256 バイトのブロックによって保護されます。
- 外部メモリは、領域によって保護されます(ウォーターマーク:開始および長さ)。保護される領域の数は、メモリタイプによって異なります(NAND、NOR、または OCTOSPI)。
- 不正アクセスイベントがあると、TZIC によってセキュア割込みが生成されます。

注 Flash メモリのセキュリティ属性は、セキュア・ウォーターマーク・オプションバイトおよび/または Flash インタフェースブロック・ベースレジスタによって定義されます。

6.5 Flash メモリ書込み保護(WRP)

書込み保護機能は、指定されたメモリ領域の内容を消去や更新から保護するために使用されます。

Flash メモリ技術の場合、更新はゼロで埋めることとみなされなければなりません。

たとえば、Flash メモリのページまたはセクタに書込み保護を設定して、ファームウェアまたはデータ更新時の改変を防ぐことができます。使用されていないメモリ領域にデフォルトで設定して、マルウェアの注入を防止することもできます。その粒度は、ページまたはセクタサイズにリンクされます。

WRP を使用すべきとき

この保護は使用されなければならず、特に、アプリケーション内での書込み操作が予想されるときには使用する必要があります。データストレージまたはコード更新操作が予想される場合がそうです。WRP は、安全でない関数による正しくないアクセスから予想外のオーバーフローが発生するのを防ぎます。

注 書込み保護は、すべての STM32 シリーズで使用可能です。

6.6 実行専用ファームウェア(PCROP)

STM32 Flash メモリ部分は実行専用(execute-only)属性で設定できます。そのように設定された領域に格納されたファームウェアは、CPU 命令バスによってのみフェッチできます。この領域の読出しまたは書込みの試みは禁止されます。保護は内部(ファームウェア)アクセスと外部(デバッグポート)アクセスの両方に対して適用されます。STM32 では、この機能は独自仕様コード読出し保護(PCROP)と呼ばれます。

PCROP は、オプションバイトによってセットされる静的保護です。保護される領域の数と粒度は STM32 シリーズによって異なります(表 9、表 10、および表 11 を参照)。PCROP が使用されているときには、ファームウェアを実行専用属性でコンパイルするように注意する必要があります(ユーザコンパイラオプションを参照)。

PCROP を使用すべきとき

PCROP は、サードパーティ・ファームウェア(知的財産)とユーザファームウェアの最も重要な部分を保護するために使用されます。

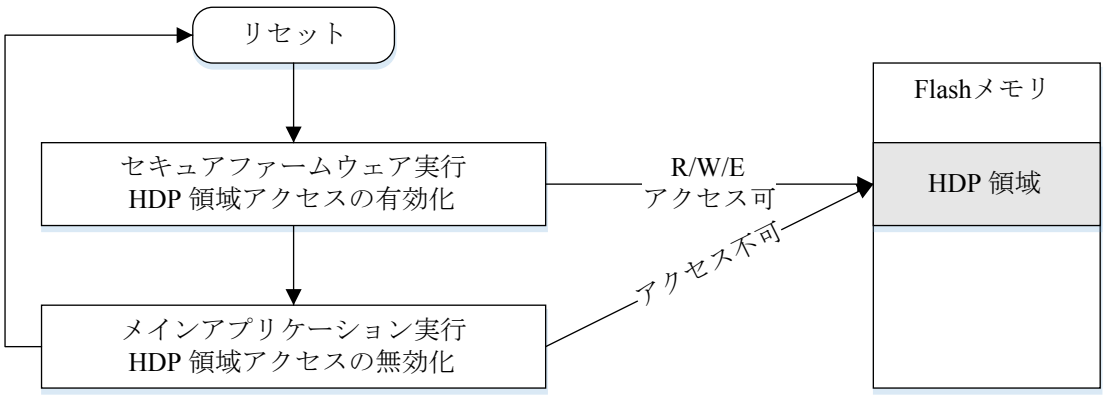
注 PCROP は、表 1 に示されているすべての STM32 シリーズで使用可能ですが、TrustZone 対応デバイスでは、PCROP の機能は別の保護メカニズムにとって代わられます。

6.7 セキュア非表示保護(HDP)

一部の STM32 デバイスは、HDP メモリの概念をサポートしています。HDP は、STM32L5 シリーズではセキュア非表示保護と呼ばれ、STM32H7 シリーズではセキュアユーザメモリ、STM32G0 シリーズではセキュリティ保護可能メモリとも呼ばれています。

HDP 領域は、デバイスのリセット後に一度しかアクセスできない Flash メモリの部分です。HDP は、機密データを内蔵または操作し、ブート時にセキュアに実行されなければならない機密アプリケーションを対象としています。アプリケーションが実行されると、HDP 領域は閉じられ、いかなる手段でもアクセスできなくなります(下の図を参照)。

図 10. HDP で保護されたファームウェアアクセス



HDP は、オプションバイトによって設定される静的保護です。設定されると、ブートピンまたはブートアドレスによって設定されたブート構成に関係なく、CPU は、この領域に内蔵されたファームウェアでブートします。

HDP を使用すべきとき

HDP は、信頼の基点としてのセキュア・ブートなど、リセット後にのみ実行されなければならないコードを格納するのに適しています。

この保護は、STM32H7、STM32G0、STM32G4、および STM32L5 シリーズで使用可能ですが、実装と名前にわずかな違いがあります(詳細については、専用のリファレンスマニュアルを参照)。

6.8 ファイアウォール

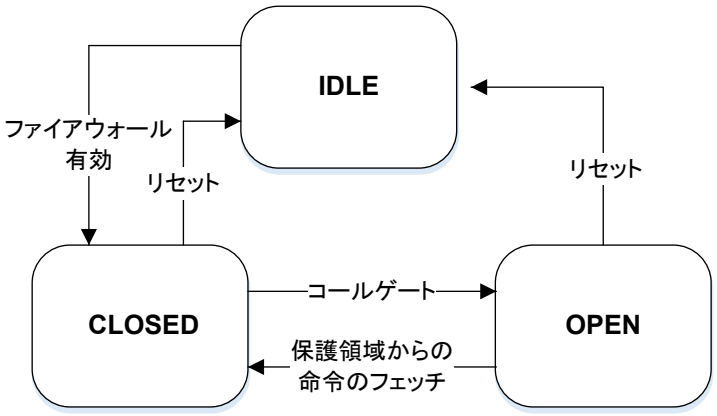
ファイアウォールは、バストランザクションを制御し、3 つの特定領域、すなわち、コード領域(Flash メモリ)、揮発性データ領域(SRAM)、および不揮発性データ領域(Flash メモリ)へのアクセスをフィルタするハードウェア保護ペリフェラルです。保護されたコードは、単一のエントリポイントを通じてアクセス可能です(下記のコールゲート・メカニズムの説明を参照)。エントリポイントを通過することなく、コードセクションにある関数へのジャンプまたは実行を試みると、システムリセットが生成されます。

ファイアウォールは動的保護の一部です。起動時に(たとえば SB アプリケーションによって)セットする必要があります。

コールゲート・メカニズム

ファームウェアはコールゲート・メカニズムを呼び出すことによって開きます。これは、ゲートを開いて、ファイアウォールによって保護されているコードを実行するために使用しなければならない単一のエントリポイントです。保護されたコードにコールゲート・メカニズムを介さずにアクセスした場合、システムリセットが生成されます。保護された領域の外側で命令がフェッチされた場合、ファイアウォールは閉じます（下の図を参照）。

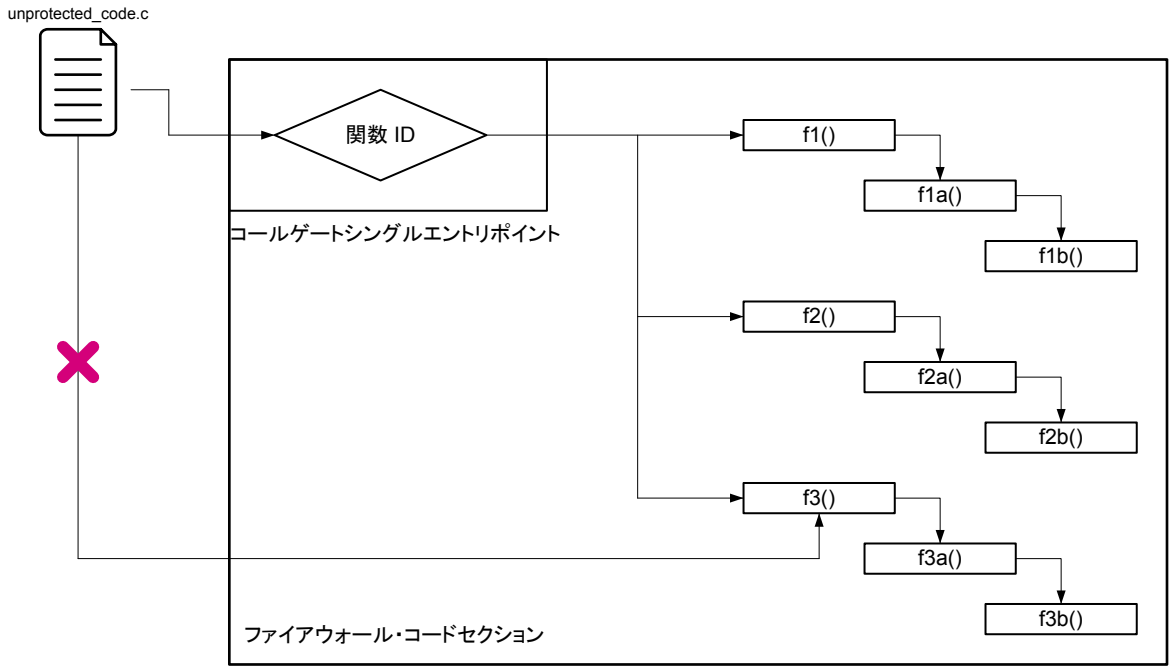
図 11. ファイアウォール FSM



コールゲート・シーケンスを守る唯一の方法は単一のコールゲート・エントリポイントを通ることなので、保護されていないコード領域からファイアウォールによって保護されている複数の関数（暗号化関数と復号関数など）を呼び出すアプリケーションをサポートするためには、メカニズムが提供されなければなりません。`CallGate (F1_ID)`、`CallGate (F2_ID)` というように、パラメータを使用して、実行する関数を指定することができます。パラメータに従って、適切な機能を内部的に呼び出します。

このメカニズムを下の図に示します。

図 12. ファイアウォールのアプリケーション例



ファイアウォールを使用すべきとき

ファイアウォールは、コードとデータの両方を保護します。保護されたコードは、コールゲート・メカニズムが守られている限り、常に呼び出すことができます。

注　ファイアウォールは STM32L0 および STM32L4 シリーズでのみ使用可能です。

注　ファイアウォールの詳細については、アプリケーション・ノート STM32L0/L4 FIREWALL overview (AN4729)を参照してください。

6.9 メモリ保護ユニット(MPU)

MPU は、デバイスのメモリにマップされたリソースに対して、特定のアクセス権を定義できるメモリ保護メカニズムです(Flash メモリ、SRAM、およびペリフェラルレジスタ)。この保護はランタイムに動的に管理されます。

注　MPU 属性は、CPU アクセスについてのみセットされます。他のバスマスタ(DMA などの)要求は MPU によってフィルタされないため、必要がないときには無効にしなければなりません。

領域アクセス属性

MPU はメモリマップを複数の領域に分割し、それぞれが独自のアクセス属性を持ちます。アクセス権は、実行可能、実行不可(XN)、読取り/書込み(RW)、読取り専用(RO)、またはアクセスなしに設定できます。

注　他にも、各領域について MPU によって設定される属性として、共有可能、キャッシュ可能、およびバッファ可能があります。このアプリケーション・ノートでは、MPU の複雑さ全体については説明していません。この章の役割は、単に導入的な説明と概要を示すことです。該当するデバイス・プログラミング・マニュアルまたはアプリケーションノート Managing memory protection unit (MPU) in STM32 MCUs (AN4838)を参照してください。

特権および非特権モード

アクセス属性に加えて、Arm Cortex-M アーキテクチャは 2 つの実行モードを定義しているため、プロセスは特権モードまたは非特権モードのいずれかで実行できます。各領域について、各モードのアクセス属性を個別に設定できます。

下の表に、モードとアクセス属性の組み合わせによってサポートされるさまざまなケースを示します。

表 13. MPU によって管理される属性とアクセス許可

特権モード属性	非特権モード属性	説明
Execute Never(XN)(1)		コード実行属性
アクセスなし	アクセスなし	アクセスはすべて許可フォールトを生成します。
RW	アクセスなし	特権ソフトウェアからのアクセスのみ
RW	RO	非特権ソフトウェアからの書込みがあると、許可フォールトが生成されます。
RW	RW	フル・アクセス
RO	アクセスなし	特権ソフトウェアによる読出しのみ
RO	RO	特権ソフトウェアまたは非特権ソフトウェアによる読出し専用

1. Execute Never(XN)属性は領域ごとに設定され、両方のモードで有効です。たとえば、SRAM のコードインジェクションを避けるために使用できます。

特権モードで実行されるコードは、追加の特定命令(MRS)にアクセスでき、Arm コアペリフェラルレジスタ(NVIC、DWT、SBC など)にもアクセスできます。これは、非特権ファームウェアからはアクセスできない機密リソースへのアクセスを必要とする OS カーネルまたはセキュアコードに便利です。

セキュアプロセスの隔離

リセット時には、いかなるプロセスについても特権モードがデフォルトのモードです。したがって、SB アプリケーションは特権モードで実行されます。そのため、セキュアプロセス(SB、OS カーネル、キーマネージャ、SFU など)を安全でない、または信頼できないプロセス(ユーザアプリケーション)から隔離するのが望ましいことです。

表 14. プロセスの隔離

ファームウェアのタイプ	モード	リソースアクセス
セキュアファームウェア(SB や OS カーネルなど)	特権	フル・アクセス
残りすべてのファームウェア	非特権	MPU によって制御されるアクセス:アクセスなし、RO、RW

OS カーネルは MPU 属性を動的に操作して、現在実行中のタスクに応じて特定のリソースへのアクセスを付与できます。アクセス権は、OS がタスクを切り替えるたびに更新されます。

MPU を使用すべきとき

MPU は、実行時に機密コードを隔離するためや、デバイスによって現在実行されているプロセスに応じて、リソースへのアクセスを管理するために使用されます。この機能は、設計にセキュリティが組み込まれている高度な組込みオペレーティングシステムにとって特に有用です。

注 MPU は、STM32F0 シリーズ以外のすべての STM32 シリーズで使用可能です(詳細については、さまざまなプログラミングマニュアルを参照)。

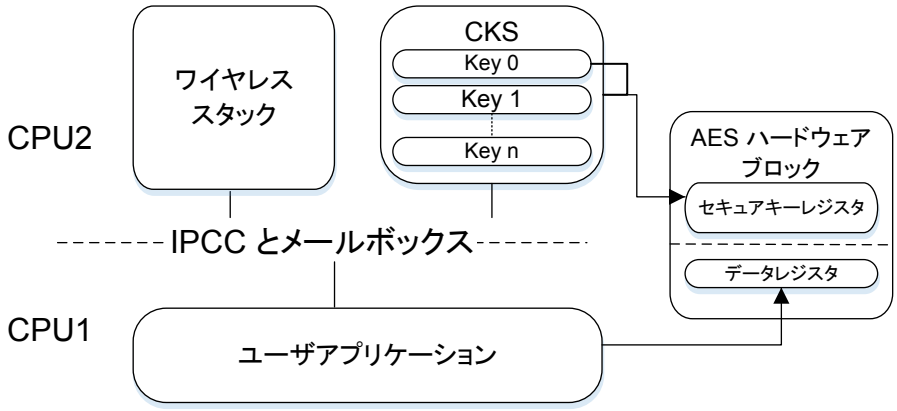
6.10 カスタマキーストレージ(CKS)

STM32WB シリーズはデュアルコアデバイスであり、1 つのコア(CPU1)はユーザアプリケーション用であり、もう 1 つのコア(CPU2)はワイヤレス側のリアルタイム制御(Bluetooth Low Energy または Thread プロトコル)専用です。CPU2 が使用する Flash メモリは、CPU1 や外部アクセスから保護されています。2 つのコアの間の通信は、メールボックスとプロセッサ間通信制御ハードウェアブロック(IPCC)によって確保されます。

ワイヤレススタック実行に加えて、CPU2 は、専用 AES ハードウェアブロックとともに使用される暗号化キー用のセキュア・ストレージ・サービスを提供します(下の図を参照)。このブロックのキーレジスタは CPU2 のみアクセス可能であり、CPU1 で実行している信頼できないプロセスやデバッグポートによるキーへのアクセスを防止します。

キーがセキュア領域内にプロビジョンされた後、ユーザアプリケーションはキーそのものだけを参照するインデックスでセキュア・ロード・サービスを呼び出すことによって、それらを使用できます。

図 13. デュアルコアアーキテクチャと CKS サービス



CKS を使用すべきとき

CKS は、ユーザアプリケーションが AES 暗号化または復号に依存しているときに使用しなければなりません。プロビジョンされたキーは、他の内部プロセスや外部アクセスが値を読み取れないように、セキュア領域に格納できます。

注 CKS は STM32WB シリーズでのみ使用可能です。

6.11 耐タンパ(TAMP)/バックアップレジスタ(BKP)

耐タンパはシステムレベルの保護であり、システム上の物理的なタンパの試みを検出するために使用されます。外部タンパイベントは、専用のデバイスピンのレベル遷移によって検出されます。内部タンパセンサは、電圧、温度、またはクロックをチェックできます。このイベントを使用してコアをウェイクアップし、適切な処置(メモリ消去、アラームなど)を取ることができます。

このペリフェラルには、内容が VBAT によって保持されるバックアップレジスタと、リアルタイムクロック(RTC)が含まれます。これらのレジスタは、タンパが検出された場合、リセットされます。

少し前のデバイスでは、このペリフェラルはバックアップレジスタ(BKP)と呼ばれています。最近のデバイスでは、単調カウンタや TrustZone セキュア領域のためのセキュアセクションなど、追加の機能で進化しています。

耐タンパを使用すべき場面

システム侵入検出(コンスーマ製品の密閉筐体など)に使用すべきです。単調カウンタは、RTC のタンパに対する対応策です。

注　外部タンパ検出は、すべての STM32 シリーズで使用可能です。

6.12 クロック・セキュリティ・システム（CSS）

CSS は、外部クロックソース（クリスタルなど）の障害を検出するように設計されています。クロックソースの喪失は、意図的な場合とそうでない場合があります。いずれにしても、デバイスは回復のための適切な処置を取る必要があります。CSS は、そのような場合、コアに割込みをかけます。

外部クロックソースがメインシステムクロックを駆動している場合、CSS はシステムを内部クロックソースに切り替えます。

CSS を使用すべきとき:

外部クロックが使用されるときには、CSS を使用する必要があります。

注　CSS は、すべての STM32 シリーズで使用可能です。

6.13 電力監視

攻撃の中には、マイクロコントローラの電源供給をターゲットとし、セキュリティ対応策の失敗につながるエラーを起こさせるものがあります。電源供給の喪失は、デバイス状態をフリーズさせて、内部メモリの内容にアクセスしようとする兆候である場合があります。

STM32 デバイスはプログラム可能な電圧検出器（PVD）を内蔵し、電力の低下を検出できます。PVD では、最小電圧閾値を設定でき、これ未満になると、割込みが生成され、コアは適切な処置を取ることができます。

PVD を使用すべきとき

機密アプリケーションが実行中で、何らかの機密データが作業メモリ（SRAM）に残っている可能性があるときには、PVD を使用しなければなりません。パワーダウン検出時にはメモリクリーニングを起動できます。

注　PVD は、すべての STM32 シリーズで使用可能です。

6.14 メモリ完全性ハードウェアチェック

エラーコード訂正（ECC）とパリティチェックは、メモリの内容に関連付けられる安全性ビットです。ECC はメモリワードに関連付けられ、各 Flash または SRAM メモリワード（メモリタイプに応じて、32 ビットから 256 ビットワード）のシングルビットエラーからの回復や最大 2 つのエラービットの検出を可能にします。シンプルなパリティチェックにより、ECC が実装されない SRAM ワードの単一のエラービットを検出できます。ECC の詳細については、AN5342 アプリケーションノートの例を参照してください。

ECC とパリティビットを使用すべきとき

ECC とパリティチェックは、安全上の理由から最もよく使用されます。ECC は何らかの侵襲的ハードウェア攻撃を防止するためにも使用されます。

6.15 独立型ウォッチドッグ（IWDG）

IWDG は、フリーランニング・ダウンカウンタであり、カウンタが所定のタイムアウト値に達したときにシステムリセットをトリガするために使用できます。IWDG は、実行中のコードが誤動作やデッドロックを起こした場合の解決策として使用できます。IWDG は、独自の独立型低速クロック（LSI）によってクロック供給されるので、メインクロックに障害が発生した場合でもアクティブなままです。

IWDG を使用すべきとき

IWDG は、デッドロックを解除するために使用できます。重要なコード（復号または Flash プログラミング）の実行時間を制御するためにも使用できます。

注　IWDG は、すべての STM32 シリーズで使用可能です。

6.16 デバイス ID

各 STM32 デバイスは一意な 96 ビットの識別子を持ち、任意のコンテキストで任意のデバイスの個別のリファレンスとなります。ユーザは、これらのビットを変更できません。

一意なデバイス識別子は、直接デバイス認証に使用でき、またはマスタ OEM キーから一意なキーを生成するためにも使用できます。

6.17 暗号化

セクション 5 で述べたように、暗号化アルゴリズムは組込みシステムの安全確保に不可欠です。暗号化は、データまたはコードの秘匿性、完全性、および認証性を確保します。これらの機能を効率的にサポートするために、ほとんどの STM32 シリーズは組込みハードウェア暗号化ペリフェラル搭載のマイクロコントローラオプションを提供します。これらのハードウェアブロックにより、暗号化計算（ハッシュや対称アルゴリズムなど）を加速できます。そのような特定のハードウェア・アクセラレーションを持たないデバイスの場合、STM32 暗号化ファームウェア・ライブラリ（CryptoLib）により、ソフトウェアで実装された多数の暗号化アルゴリズムを提供します。

6.17.1 ハードウェアアクセラレータ

以下の暗号化ハードウェアペリフェラルは STM32 デバイスで使用可能です。

- 真の乱数発生器
  - 物理的ノイズ源を提供するハードウェアベースのペリフェラル。強力なセッションキーの生成に使用されます。

注 TRNG 検証の詳細については、アプリケーション・ノート STM32 microcontroller random number generation validation using the NIST statistical test suite （AN4230）を参照してください。

- AES アクセラレータ
  - 暗号化／復号化
  - 128 または 256 ビットキー
  - いくつかの連鎖モード（ECB、CBC、CTR、GCM など）
  - DMA サポート
- PKA アクセラレータ
  - 高速剰余乗算に関するモンゴメリ法に基づく RSA、DH、および GF(p) での ECC 演算の高速化
  - モンゴメリ定義域内外への変換を搭載
- HASH アクセラレータ
  - MD5、SHA1、SHA224、SHA256
  - FIPS 準拠（FIPS Pub 180-2）
  - DMA サポート

注 AES ブロックが暗号化または復号化に使用される場合、キーを保持するレジスタへのアクセスは保護しなければならず、使用後にキーレジスタを消去しなければなりません（MPU）。

6.17.2 CryptoLib ソフトウェア・ライブラリ

STM32 X-CUBE-CRYPTOLIB は、すべて STM32 デバイス上で動作するソフトウェアライブラリです。www.st.com/en/product/x-cube-cryptolib から無料でダウンロードできます。STM32 X-CUBE-CRYPTOLIB V3.1.5 は、Cortex-M0、M0+、M3、M33、M4、および M7 コア向きにコンパイルされた完全なファームウェア形式で入手可能です。

X-CUBE-CRYPTOLIB は、次のアルゴリズムをサポートしています。

- DES、3DES（ECB および CBC）
- AES（ECB、CBC、OFB、CCM、GCM、CMAC、KEY ラップ、XTS）
- ハッシュ関数：MD5、SHA-1、SHA-224、SHA-256、SHA-384、SHA-512
- その他：ARC4、ChaCha20、Poly1305、Chacha20-Poly1305
- PKCS#1v1.5 付き RSA 署名
- キー生成、スカラ乗算（ECDH の基本）& ECDSA + ED25519 および Curve 25519 付き ECC

注 X-CUBE-CRYPTOLIB V3.1.5 はすべての STM32 シリーズに対応しています。

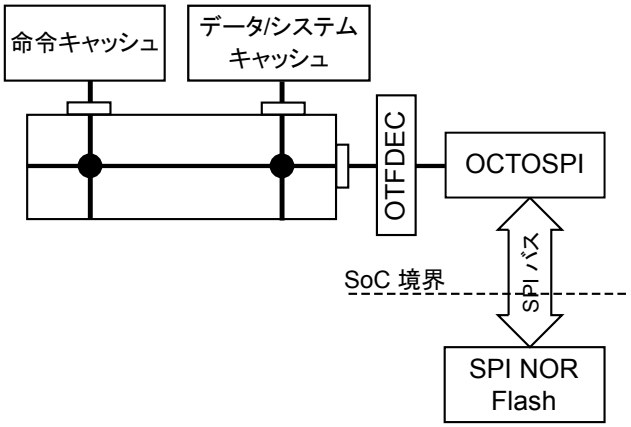
6.18 オンザフライ復号化エンジン（OTFDEC）

外部メモリの内容（コードとデータ）は、従来の読出し / 書込み保護では保護できません。内容を保護する手段は、暗号化して、使用前にデバイス内で復号することです。

一つの方法としては、外部メモリの内容を SRAM にダウンロードし、復号し、コードを実行するか、データを使用します。この方法には 2 つの欠点があります。受け入れられない遅延が発生する恐れがあることと、内容によっては大量の SRAM を使用することです。

OTFDEC は、低遅延で、SRAM 割当てを必要とせずに、内容を直接復号できます。OTFDEC は、読出しリクエストのアドレス情報に基づいてオンザフライバス(AHB)トラフィックを復号するハードウェアブロックです。Octo-SPI インタフェースで使用されます(下の図を参照)。

図 14. SoC での OTFDEC の標準的な使用方法



OTFDEC は AES-128 をカウンタモードで使用し、128 ビットのキーで 12 AHB サイクル未満の遅延を達成します。最大 4 つの独立して重複のない暗号化領域を定義でき(4 キロバイトの粒度)、それぞれ専用のキーを持ちます。

OTFDEC を使用すべきとき

OTFDEC は、システムによって外部メモリが使用されるときに使用されます。TrustZone 対応 MCU の場合、復号キーはセキュアモードでのみアクセス可能です。詳細については、AN5281 を参照してください。

注　OTFDEC は、STM32L5 シリーズおよび STM32H7 シリーズでのみ使用可能です。

7 ガイドライン

セキュアシステムは、多くのセキュリティ支援ハードウェア機能を利用することができます。なかには、どのようなシステムにも有用で、アプリケーションコードをほとんど変更しなくても有効化でき、完全に機能するものもあります。RDP 機能もその一つであり、デバッグポートを無効化することによって Flash メモリへの基本的なアクセスを防止します。その他の機能は、ユーザアプリケーションと必要なセキュリティレベルに応じて選択する必要があります。

この節の目的は、システムの使用事例に応じて採用すべきセキュリティ機能の選択を容易にすることです。

使用事例は、外部(1)と内部(2)の脅威に対する保護、セキュリティ保守(3)、および暗号化(4)に関するその他の使用事例という 4 つの主なグループに分けることができます。

表 15. セキュリティの使用事例

1 外部の脅威に対するデバイス保護:RDP 保護、タンパ検出、デバイス・モニタリング	
-	1.1 デバイス設定(オプションバイト、変更されていないと想定)
	• RDP レベル 2 を使用します。これはデバイスを外部アクセスから閉ざします。
	1.2 デバイスのデバッグ機能を削除する。
	• デバッグを永続的に無効化するために RDP レベル 2 を使用します。
	1.3 外部クロックソース(クリスタル)の喪失に対してデバイスを保護する。
	• クロック・セキュリティ・システム(CSS)を有効にします。
	1.4 システムレベルの侵入を検知する。
	• RTC ペリフェラルのタンパ検出機能を使用します。
	1.5 コードインジェクションからデバイスを保護する。
	• RDP を使用します。 • 通信ポートプロトコルを MPU、ファイアウォール、または HDP で隔離します。 • 通信ポートプロトコルのアクセス範囲を制限します。 • 空のメモリ領域(Flash メモリおよび SRAM)で書込み保護を使用します。
2.内部の脅威に対するコード保護:TrustZone、PCROP、MPU、ファイアウォール、および HDP	
-	2.1 クローニングからコードを保護する。
	• 外部アクセスに対して RDP レベル 1 または 2 を使用します。 • 内部アクセスに対してコードの最も重要な部分に PCROP を使用します。 • OTFDEC を使用して、外部メモリに格納されているコードを保護します。
	2.2 他のプロセスから秘密データを保護する方法
	• ファイアウォールを使用して、コードとデータの両方を保護します。 • MPU を使用して、秘密データ領域が読み取られないようにします。 • データがリセット時にのみ使用されるようにするには、HDP を使用します。 • 使用可能な場合、TrustZone のセキュアドメインを使用します。
	2.3 完全に検証されていないあるいは完全には信頼されていないライブラリが使用されたときに、コードとデータを保護する。
	• PCROP を使用して、ユーザの最も重要なコードを保護します。 • ファイアウォールを使用して、ユーザの重要なアプリケーション(コード、データ、および実行)を保護します。 • MPU を使用し、信頼できないライブラリから権限をはく奪します。 • IWDG を使用して、デッドロックを避けます。 • 使用可能な場合、TrustZone のセキュアドメインを使用します。
3.デバイス・セキュリティ・チェックおよび保守:完全性チェック、SB、SFU	
-	3.1 コードの完全性をチェックする。
	• リセット時にファームウェアコードをハッシュして、期待値と比較します。 • Flash メモリの ECC と SRAM のパリティチェックを有効にします。
	3.2 セキュリティチェックまたは内蔵ファームウェアの認証性
	• 暗号化を備えた SB アプリケーションを実装します。 • SB アプリケーションの秘密データを保護します(以前の節を参照)。

-	• SB アプリケーションの一意なブートエントリを保証します。 – 使用可能な場合は、HDP を使用します。 – RDP レベル 2 を使用し、ブートピン選択を無効にします。
	3.3 フィールドでファームウェアを安全に更新する。
	• 暗号化を備えた SFU アプリケーションを実装します。 • SFU 秘密データの周囲にセキュアメモリ保護を適用します（以前の節を参照）。
4.通信と認証性：暗号化	
-	4.1 安全に通信する。
	• 秘匿性と認証性のために、暗号化に依存するセキュア通信スタックを使用または実装します（Ethernet の場合は TLS など）。
	4.2 STM32 デバイスで ST AES/DES/SHA 暗号化機能を使用する。
	• STM32 X-CUBE-CRYPTOLIB などの ST による正式なソフトウェア実装のみを使用します。
	4.3 AES/DES/SHA 暗号化機能を活用する。
	• 暗号化ハードウェアペリフェラルを備えたデバイスを正式な STM32 X-CUBE-CRYPTOLIB とともに使用します。 • OTFDEC を使用して、外部メモリの AES 暗号化コードに遅延なくアクセスします。
	4.4 真の乱数データを生成する。
	• STM32 デバイスに内蔵されているハードウェアの真の乱数発生器を使用します。
	4.5 ST マイクロコントローラを一意に識別する。
	• STM32 マイクロコントローラの 96 ビット UID を使用します。
	4.6 製品デバイスを認証する。
	• 共有暗号化キーをデバイスに内蔵し、暗号化メッセージを交換します。
	4.7 製品デバイスを一意に認証する。
	• デバイス秘密鍵とその証明書をデバイスに内蔵し、暗号化メッセージを交換します。
	4.8 通信サーバを認証する。
	• 共有暗号化キーをデバイスに内蔵し、暗号化メッセージを交換します。 • サーバ公開鍵をデバイスに内蔵し、暗号化メッセージを交換します。

8 結論

どのようなシステムも、ハードウェアのセキュリティ機能を有効にするだけではセキュアにできません。セキュリティは、ソリューション全体のアーキテクチャに根差す必要があります。

脅威を識別して、対策を正しく設計し、他のセキュリティ機能と連携して実装する必要があります。

セキュリティにはかなりのリソースが必要になるため、リスクを正しく評価して、攻撃のコストと保護される資産の価値を念頭に置いて、リソースを効率的に使うことが重要です。

信頼の基点(ルート・オブ・トラスト)という概念は、すべてを他のすべてに依存する全体論的なアプローチと違って、より分析的なアプローチを用いるため、非常に有用です。

STM32 シリーズのマイクロコントローラでは、組込みの IoT セキュリティは非常に費用対効果が高く、効率的です。

付録 A　暗号化 — 主な概念

完全性、認証性、および秘匿性

暗号化の目的は 3 つです。

- 秘匿性：無許可の読出しアクセスからの機密データの保護
- 認証性：メッセージ送信者の身元の保証
- 完全性：転送中のメッセージ破損の検出

これらの目的を満たすために、セキュアデータの流れはすべて、多かれ少なかれ、下記のアルゴリズムの複雑な組み合わせに依存しています。

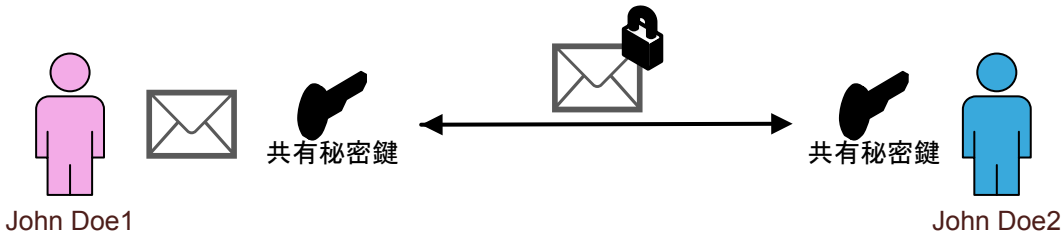
- 共有秘密鍵/対称暗号化
- 公開鍵/非対称暗号化
- ハッシュ

これらのアルゴリズムについて、以下のセクションで説明します。

A.1　共有秘密鍵アルゴリズム

このアルゴリズムファミリは、送信者と受信者で共有される共有秘密鍵で平文を暗号化することによって秘匿性を確保します。暗号化と復号に同じ鍵が使用されるため、この技法は対称暗号化と呼ばれます。

図 15. 対称暗号化



これらのアルゴリズムの本質的な弱点は、両方の当事者による鍵の共有です。セキュアな環境（製造工場など）では問題にならないかもしれませんが、両方の当事者が離れている場合、鍵の転送が課題になります。

すべての共有秘密鍵アルゴリズムの中でも、ブロックベースのアルゴリズムはハードウェアまたはソフトウェアの並列実装によって効率的に加速できるため、非常に一般的です。典型的な AES（高度暗号化標準）アルゴリズムは、128 ビットのクリアブロックを操作します。128、192、または 256 ビットの鍵を使用して、同じ長さの暗号化ブロックを生成します。連続するブロックをつなぐためのさまざまな方法を「操作モード」といいます。これらには、暗号ブロック連鎖（CBC）、カウンタモード（CTR）、およびガロア・カウンタモード（GCM）があります。

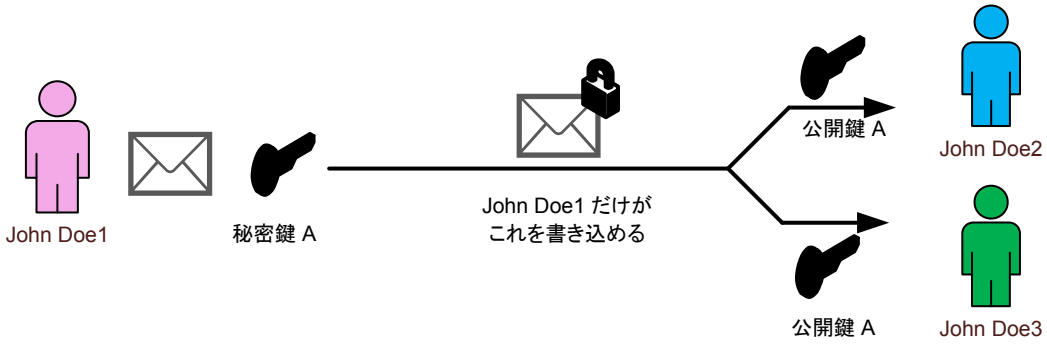
これらのアルゴリズムは決定論的であるため、常に入力データをノンスというランダム値と混合し、1 回のセッションに限り、初期化ベクトルとして使用されます。

A.2 公開鍵アルゴリズム（PKA）

このクラスのアルゴリズムは、一対の鍵に基づきます。1 つは秘密鍵であり、リモートシステムと交換されず、もう 1 つは公開鍵であり、任意の当事者と交換できます。2 つの鍵の関係は非対称です（非対称暗号化）。

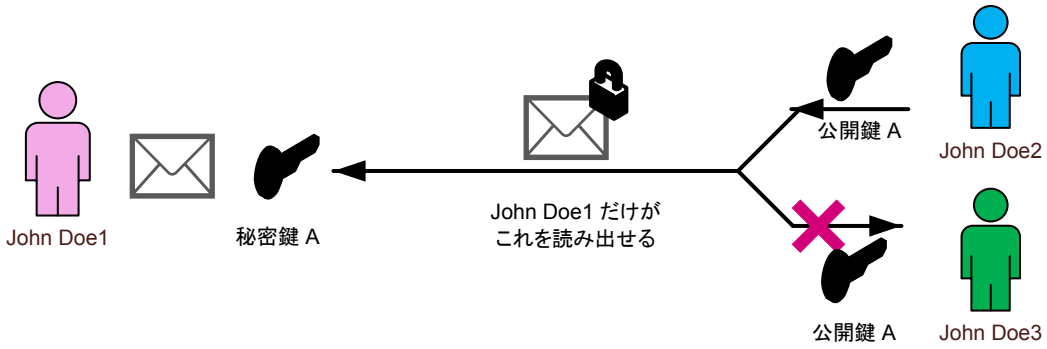
- 秘密鍵によって暗号化されたメッセージは、公開鍵を持つ当事者が読むことができます。秘密鍵は共有されないため、このメカニズムにより、送信者の強力な認証が確保されます。デジタル署名は、このメカニズムに基づきます。

図 16. 署名



- 公開鍵によって暗号化されたメッセージは、秘密鍵の所有者だけが読むことができます。

図 17. PKA 暗号化



公開鍵アルゴリズムの主な用途は認証です。

対称暗号化の「鍵共有」問題の解決にも使用されます。ただし、より複雑な操作、計算時間の増加、およびメモリフットプリントの増加というコストが伴います。

RSA と楕円曲線暗号（ECC）は、最も一般的な非対称アルゴリズムです。

ハイブリッド暗号化

一般的なセキュア転送プロトコル（Bluetooth や TLS など）は、両方のアルゴリズムタイプに依存しています。このスキームは、ハイブリッド暗号化と呼ばれます。

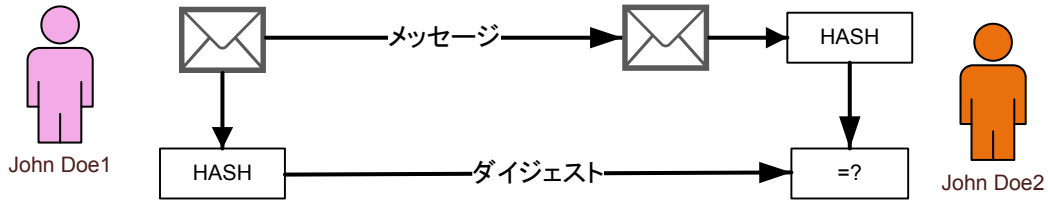
- 非対称暗号化がまず使用されて、対称鍵共有問題を解決します。セッション鍵が公開鍵所有者から秘密鍵所有者へ交換されます。
- 次に、セッション鍵を使用した対称アルゴリズムによって、転送の秘匿性が提供されます。

A.3 ハッシュアルゴリズム

ハッシュアルゴリズムはメッセージの完全性を保証します。メッセージからダイジェストと呼ばれる一意な固定長のビットストリームを生成します。入力メッセージに差があると、まったく異なるダイジェストになります。ダイジェストを逆転して入力メッセージを得ることはできません。

ハッシュはメッセージの暗号化とは別に使用できます。

図 18. メッセージのハッシング



古典的な CRC との違いは、より複雑な演算と、はるかに長いダイジェスト長（16 または 32 ビットではなく、最大 512 ビット）による堅牢性です。例として、CRC はデータ転送時の高速な完全性チェックのために使用されます。ダイジェスト長により、事実上、ダイジェストは一意となり、衝突が起きることはありません。

典型的なアルゴリズムは、MD5（128 ビットのダイジェスト）、SHA-1（160 ビットのダイジェスト）、SHA-2（224、256、384、または 512 ビットのダイジェスト）、および SHA-3（224、256、384、または 512 ビットのダイジェスト）です。

A.4 MAC または署名と証明書

MAC と署名

メッセージ認証コード（MAC）と署名は、メッセージハッシュを暗号化することによって完全性に認証を加えます。MAC と署名の違いは、MAC 生成が対称鍵アルゴリズムを使用するのに対して（図 19）、署名がメッセージ送信者の秘密鍵（図 20）を使用することです。

署名は否認不可の次元を認証に加えます:

- 秘密鍵は抹消が想定されていませんが（ライフタイムは送信操作を超えて続きます）、共有秘密鍵はライフタイムが限られている（この送信に限定されている）ことがあります。
- 署名に使用される秘密鍵は共有されず、共有秘密鍵より高いセキュリティを持ちます。

図 19. 共有秘密鍵アルゴリズムによる MAC 生成

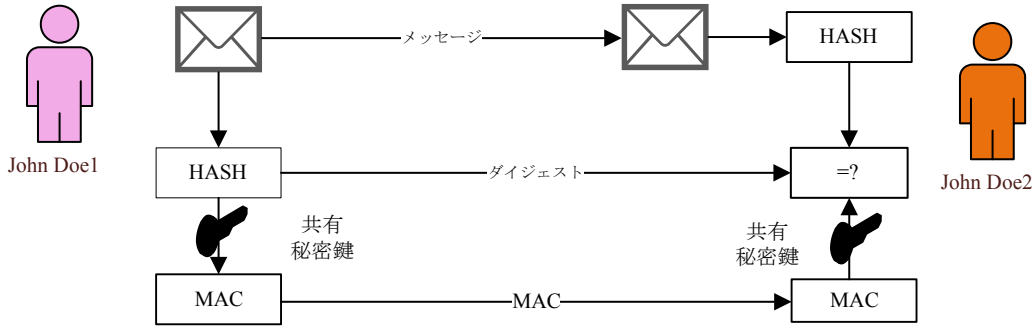
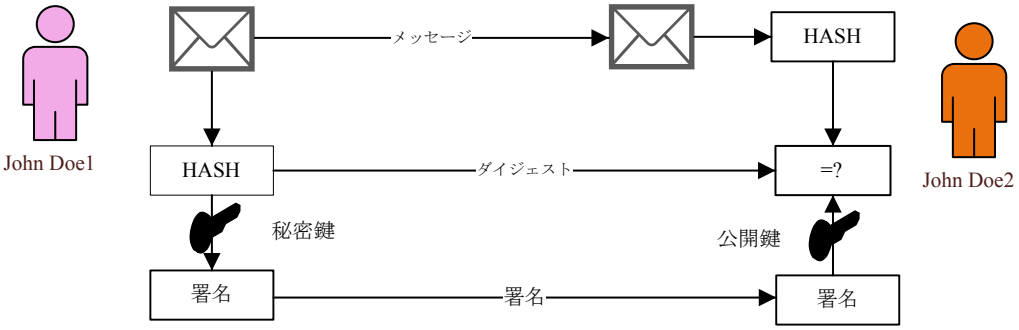


図 20. 公開鍵アルゴリズムによる署名生成



証明書

証明書は、公開鍵アルゴリズムに関連します。非対称転送において公開鍵を認証します。攻撃者が正しい公開鍵を自分自身の鍵に置き換える不正使用に対抗するために使用されます。証明書は、認証局（CA）の秘密鍵によって署名された公開鍵で構成されます。この CA は完全に信頼できるとみなされます。

公開鍵に加えて、証明書はバージョン番号、有効期間、および ID も含みます。

改版履歴

表 16. 文書改版履歴

日付	版	変更内容
2018 年 10 月 17 日	1	初版発行
2019 年 2 月 25 日	2	更新: • 表 2.対象とする製品 • セクション 1 一般情報 • 表 11.STM32H7、STM32G0、STM32G4、および STM32WB シリーズのセキュリティ機能 • 図 9.RDP 保護の例(STM32L4 シリーズ) • セクション 6.6 ファイアウォール 追加: • セクション 6.8 暗号化キーストレージ(CKS)
2019 年 10 月 7 日	3	更新: • 表 2.対象とする製品 • 「紹介」セクションの名前を「概要」に変更。 • 表 2.用語 • 「ハードウェア保護」セクションの名前を「デバイス保護」に変更。 • 図 4.メモリタイプ • 表 5.メモリタイプと関連する保護 • セクション 4.2.4 外部 Flash メモリ • 表 6.STM32 内蔵メモリ保護機能の範囲 • 表 7.ソフトウェア隔離メカニズム • セクション 4.5 ブート保護 • セクション 5 セキュア・プリケーション:表 9、表 10 、および表 11 • セクション 6.2 読出し保護(RDP) • セクション 6.7 セキュア非表示保護(HDP) • セクション 6.17 暗号化 • セクション 7 ガイドライン • すべての図からいくつかの色を除去 追加: • セクション 4.1 Armv8-M アーキテクチャ用 TrustZone® • セクション 6.4 TrustZone • セクション 6.18 オンザフライ復号化エンジン(OTFDEC)
2019 年 2 月 21 日	4	• 「概要」セクションを更新。 • 表 2 に略称を追加。用語 • セクション 2「概要」およびセクション 3「攻撃のタイプ」を更新。 • セクション 3.4「IoT システム攻撃の例」を再構成(セクション 3.5「攻撃のターゲットの一覧」を追加)。 • セクション 4「デバイス保護」を更新。 • セクション 5「セキュアアプリケーション」を更新および再構成。 • セクション 5.3「Arm TF-M ソリューション」を追加。 • セクション 6「STM32 セキュリティ機能」、セクション 7「ガイドライン」、およびセクション 8「結論」を更新。
2020 年 11 月 06 日	5	更新: • ドキュメントの範囲に STM32WL シリーズを追加 • 表 1.対象とする製品 • セクション 1 一般情報 • セクション 3.1 攻撃のタイプについて • セクション 3.2 ソフトウェア攻撃 • セクション 3.3.1 非侵襲攻撃 • セクション 3.3.2 シリコン侵襲攻撃 • セクション 4.1 Armv8-M アーキテクチャ用 TrustZone® • 表 5.メモリタイプと関連する保護 • セクション 5.3 Arm TF-M ソリューション • 表 8.基本機能の違い

日付	版	変更内容
		• セクション 6.1 セキュリティ機能の概要（すべての表の更新を含む） • セクション 6.2 読出し保護（RDP） • セクション 6.4 TrustZone 追加： • セクション 4.2 デュアルコアのセキュリティ • セクション 6.3 ワンタイム・プログラマブル（OTP）
2021 年 7 月 7 日	6	更新： • ドキュメントの範囲に STM32U5 シリーズを追加 • 表 1. 対象とする製品 • セクション 3.3.1 非侵襲攻撃（Non-invasive attack） • セクション 4.3.3 内蔵 SRAM • セクション 4.3.4 外部 Flash メモリ • セクション 5 セキュア・アプリケーション • 表 9. STM32Fx シリーズのセキュリティ機能 • 表 10. STM32Lx および STM32Ux シリーズのセキュリティ機能 • 表 11. STM32H7、STM32G0、STM32G4、STM32WB、および STM32WL シリーズのセキュリティ機能 • セクション 6.3 OTP:One-time programmable（ワンタイム・プログラマブル） • セクション 6.6 実行専用ファームウェア（PCROP） • セクション 6.8 ファイアウォール • セクション 6.9 メモリ保護ユニット（MPU） • セクション 6.17 暗号化 • セクション 6.17.1 ハードウェアアクセラレータ • セクション 6.17.2 CryptoLib ソフトウェア・ライブラリ 追加： • セクション 5.4 製品認証

目次

表一覧

図一覧