AKRAINO

ホワイトペーパー

# Akraino プラットフォームのセキュリティアーキテクチャ

第 1 版 - 2023 年 3 月
www.akraino.org

LF EDGE

# 目次



## 図

# 謝辞

本ホワイトペーパーの作成にあたり、以下の資料を使用しました。

- 本ホワイトペーパーの参考文献セクションで [3]、[4]、[7] と番号付けされている PSA Certified ™ 文書は、すべて © Copyright Arm Limited 2017-2022 です。
- このホワイトペーパーの参考文献セクションで [2] と番号付けされている Arm Aarch64 Exceptions Model 文書は、すべて ©Arm Limited の著作物です。
- インテル ® ハードウェア シールド - OS 以下のセキュリティ」ドキュメント番号本ホワイトペーパーの参考文献の [8] および [9] は、すべて © Copyright Intel® Corporation 。

そのようなコンテンツは許可を得て使用されています。このようなコンテンツの使用には、該当するドキュメントの前面に記載されている条件が適用されます ( 参考文献のセクションにリンクが記載されています )。本規約は、Arm またはインテル ® のテクノロジーまたは製品の知的財産に対する法的権利をお客様に付与するものではありません。

# 1 概要

Akraino ブループリントの開発中、ブループリント所有者は、セキュリティ脅威を分析し、プロジェクトにセキュリティ機能を実装するために多くの労力を費やすことがあります。しかし、多くの場合、ブループリント所有者は、ブループリント実行環境は十分に保護されており、注意を払う必要はないと思い込んでいます。このような思い込みは、ブループリントの機能を妨害し、個人データや重要データの損失を引き起こすプラットフォーム レベルの脆弱性を利用した攻撃につながる可能性があります。このため、プラットフォーム レベルのセキュリティに関する要件は、ブループリント要件の重要な一部と見なされるべきです。

Akraino PSA（プラットフォーム セキュリティ アーキテクチャ）は、Akraino プラットフォームとブループリント実行環境の中核となるセキュリティ要件を定義しています。

Akraino PSA の要件はプラットフォームにとらわれず、プラットフォーム ハードウェアとシステム ソフトウェアのセキュリティ要件を定義しています。文書末尾の付録は、Arm と Intel によるこれらの要件のプラットフォーム固有の実装に関する情報を提供します。

## 1.1 対象となる読者

このホワイトペーパーは以下の読者を対象としています。

1. Akraino ブループリントのオーナーと開発者
2. Akraino プラットフォームのオーナー
3. クラウド環境プロバイダー
4. Akraino ブループリント インテグレーター

# 2 用語と略語

| TERM | 説明 |
|---|---|
| ACM | 認証されたコンピュート モジュール |
| ACRAM | 認証 Code RAM |
| AIK | (TPM) 認証識別キー |
| AP | アプリケーション プロセッサー |
| BL | ブートローダー |
| CPLD | 複雑なプログラマブル ロジック デバイス |
| CoT | 信頼の連鎖 |
| EK | (TPM) エンドースメント キー |
| EL | 例外レベル |
| FPF | フィールド プログラミング ヒューズ |
| hRoT | ハードウェア ルート オブ トラスト |
| IBB | イニシャル ブート ブロック |
| mTLS | 相互トランスポート層セキュリティ |
| OEM/ODM | 相手先商標製品メーカー / オリジナル デザイン メーカー |
| NSPE | 非セキュアな処理環境 |
| PCH | プラットフォーム コントローラー ハブ |
| PFR | プラットフォーム ファームウェア レジリエンス |
| PK | 公開鍵 |
| PKCS | 公開鍵暗号方式標準 |
| PRoT | プラットフォーム RoT |
| RMA | 返品保証 |
| ROM | リードオンリーメモリ |
| RoT | 信頼の根源 |
| ROTPK | 信頼の根源 公開鍵 |
| SCP | システム制御プロセッサ |
| SiP | シリコン プロバイダー（SoC メーカー） |
| SMM | システム管理モード |
| SoC | システム オン ア チップ |
| SPE | セキュアな処理環境 |
| SPS | サーバー プラットフォーム サービス |
| TBB | トラステッド ボード ブート |
| TCB | トラステッド コンピューティング ベース |
| TCG | トラステッド コンピューティング グループ |
| TLS | トランスポート層のセキュリティ |
| TPM | トラステッド プラットフォーム モジュール |
| UEFI | ユニファイド エクステンシブル ファームウェア インタフェース |
| VMM | 仮想マシンマネージャー |

# 3 はじめに

## 3.1 問題意識

プラットフォーム セキュリティとは、コンピューティング プラットフォーム全体のセキュリティを保証するセキュリティ アーキテクチャ、ツール、およびプロセスを指します。プラットフォーム セキュリティは、バンドル／統合されたセキュリティ ソフトウェア、システム、およびプロセスを使用して、コンピューティング プラットフォームのハードウェア、ソフトウェア、ネットワーク、ストレージ、およびその他のコンポーネントのセキュリティを実現します[3]。**図 1** はプラットフォーム セキュリティに関係する領域 / レイヤーを示しています。

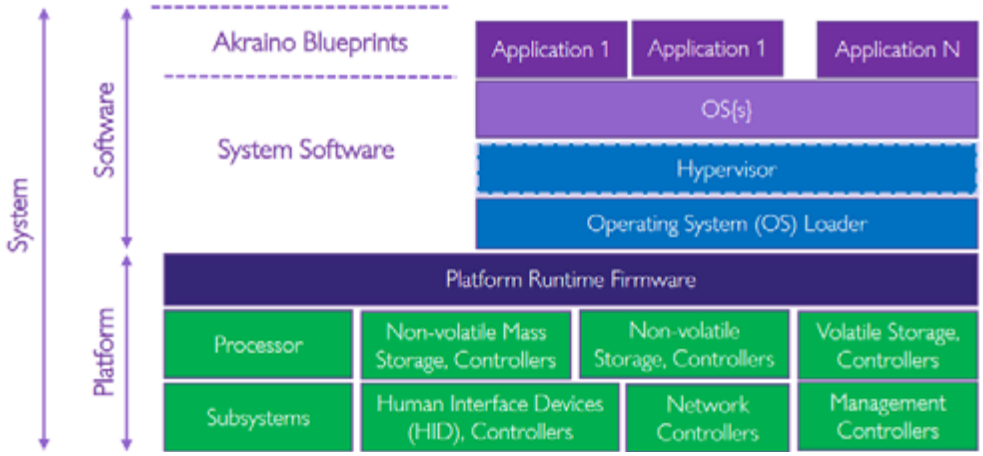


**図 1** システム プラットフォームと ソフトウェアレイヤー

プラットフォーム セキュリティの目標は、プラットフォーム内のすべてのレイヤーとコンポーネントをセキュアにすることである。これにより、統一されたセキュリティ要件を使用することで、プラットフォーム全体の安全性を確保することができます。

Arkaino プロジェクトにおけるプラットフォーム セキュリティの目的は以下のとおりです。

- アーキテクチャにとらわれない
- プラットフォーム層の整合性を維持し、Akraino ブループリントのソフトウェア スタックに安全な実行環境の提供
- プラットフォームの Root-of-Trust に基づいてセキュアブート要件の定義
- プラットフォームが安全で完全な状態であることの証明
- プラットフォームにおける主な資産の保護：
  - プラットフォームの重要データ（プラットフォーム ID、暗号化キー、設定データなど）
  - 変更可能なファームウェア コンポーネント
  - 安全なプラットフォームのファームウェア アップデート
- プラットフォームのランタイム環境とデータの保護
- Akraino Blueprint ソフトウェアスタックのプラットフォーム セキュリティ機能とセキュリティ サービスの活用

## 3.2 Akraino プラットフォームのセキュリティ目標

プラットフォームのセキュリティ目標には以下のものが含まれます。

1. 一意の識別。デバイスは一意に識別可能でなければならない。

2. セキュリティ ライフサイクル。デバイスは、セキュリティ ライフサイクルをサポートしなければならない。デバイスの状態は証明可能でなければならず、デバイスに関連付けられているデータへのアクセスに影響を与える可能性があります。

3. 認証。デバイスは確実に認証可能でなければならない。

4. ソフトウェアの認可。デバイスは、認可されたソフトウェアのみが実行されることを保証しなければならない。セキュア ブートおよびセキュア ローディング プロセスは、許可されていないソフトウェアの実行を防ぐために必要です。

5. 安全なアップデート。デバイスは、ソフトウェア、またはハードウェア構成などのプラットフォームの重要なデータの安全な更新をサポートしなければならない。

6. ロールバック防止。更新の不正なロールバックを防止すること。

7. 分離。デバイスは分離をサポートしなければならない。信頼されたサービスを互いに、また信頼されていないサービスから分離することは、これらのサービスの機密性と完全性を保護するために不可欠です。

8. 相互作用。デバイスは、分離境界を越えたインタラクションをサポートしなければならない。インタフェースを使用して、相互作用するサービスやデバイスの機密性と完全性を損なってはならない。

9. 暗号および信頼されるサービス。すべてのデバイスは、他のセキュリティ目標をサポートするために必要な、最低限の信頼されるサービ スと暗号操作をサポートするものとする。

# 4 プラットフォーム セキュリティ

Akraino プラットフォーム セキュリティ アーキテクチャは、プラットフォームにとらわれず、以下を定義します。

- Verified boot（セキュア ブートとも呼ばれる）とは、各 SW/FW コンポーネントがロードされ、前のコンポーネントによって暗号的に検証されることであり、別名「信頼の連鎖（Chain of Trust）」です。最初のコンポーネントは、HW ベースの Root of Trust によって検証されます。信頼の連鎖のどのステップでも検証に失敗した場合、ブートは終了します。
- Measured boot とは、各 SW および FW コンポーネントがロードされ、前のコンポーネントによって測定されることです。すべての測定値は安全なストレージに記録されるが、検証はされない。プロセスの最初のコンポーネントは例外です。まず、HW ベースの Root of Trust によってロードされ、検証されます。次に、そのコンポーネントは自分自身を測定します。この最初の検証に失敗した場合、ブートは終了します。
- 信頼されたプロセスの分離。

プラットフォームのアーキテクチャと構成に応じて、信頼されたハードウェアは、不正アクセスからデータを保護し、暗号化機能を実装することができます、暗号鍵、データ測定値、プラットフォーム設定データ、システムランタイム変数を含むデータの完全性を保護します。

セキュリティが確保され、検証され、信頼されたハードウェア、ファームウェア、ソフトウェアの組み合わせが、プラットフォームのトラステッド コンピューティング ベース（TCB）を提供します。TCB はセキュリティ ポリシーを実施し、ポリシーで定義された制限を破ったり、許可された特権を得たり、ハードウェアを改ざんしたりすることを防ぐ保護メカニズムを提供します。

エッジロケーションはプライベート クラウドやパブリック クラウドよりも物理的なセキュリティが低いことが多いため、このプラットフォーム セキュリティ アーキテクチャでは、すべてのプラットフォームブートタイプに対して、HW ベースの不変の Root of Trust を要求しています。

以下のセクションでは、Verified boot、Measured boot、プラットフォームの Root of Trust、およびプラットフォームのブートフローのハイレベルな説明を提供します。

## 4.1 プラットフォーム ルーツ オブ トラスト

プラットフォーム ルーツ オブ トラスト（PRoT）には、システム レベルのセキュリティ コンフィギュレーションを担うハードウェア コンポーネントとソフトウェア コンポーネントが含まれ、検証された（セキュアな）ブート プロセスのアンカーとなり、プラットフォームの信頼の連鎖を確立し、信頼された測定データとプラットフォームの検証可能な認証データを提供します。

トラステッド コンピューティング グループ（TCG）は、RoT（Root of Trust）を、測定、保存、報告、検証、および／または更新のような、1 つ以上のセキュリティに特化した機能を実行するコンポーネントとして定義しています [10]。RoT は、常に期待された振る舞いをすることが信頼されます。なぜなら、RoT の誤った振る舞いは、測定、認証、観察などによって検出することができないからです。

Root of Trust には Immutable（不変）と Mutable（可変）の 2 つのアーキテクチャ タイプがあります。両者は実装、保守性、信頼性において異なります。Immutable RoT は、RoT メーカーが提供した後、所有者、管理者、ユーザーが変更できません。

Immutable RoT の信頼特性は、RoT ハードウェアとベンダーの製造プロセスで展開される変更不可能なコードによって決まります。Immutable RoT は、定義された脅威モデルに基づく一連のデバイス モデル内のすべてのデバイスで同一性を保つことが期待されます。また、時を経ても変化しないため、デバイスの寿命が尽きるまで同じ振る舞いをすることが期待されます。利害関係者は正しい動作を確認するためにデバイスのサンプルを評価し、その後デバイスの製造工程を変更しないよう製造者を信頼することを選択することができます。

Mutable RoT は、認可されたエンティティによって変更可能です。Mutable RoT をデプロイする目的には、バグや脆弱性の修正、拡張機能の追加が含まれます。

以下の Root of Trust は通常、Platform Root of Trust の様々な定義の一部として含まれます。

1. Root of Trust for Measurement (RTM)
   - 信頼のおける完全性の計測を行うことができる演算エンジン。
   - レポーティングのためのルート オブ トラストによる証明に使用できる、ファームウェア、ソフトウェア、またはコンフィギュレーション データの測定値を提供します。

2. Root of Trust for Reporting (RTR)
   - Root of Trust がストレージのために保持する情報を確実に通知できる演算エンジン
   - アイデンティティと認証アサーションを管理します

3. Root of Trust for Storage (RTS)
   - 完全性ダイジェストの値とダイジェストのシーケンスの正確な要約を維持することができる演算エンジン。
   - アクセスを防止することにより、資産の機密性と完全性を保護します。

4. Root of Trust for Update (RTU)
   - 変更可能な RoT を更新するための最終権限。
   - 安全なファームウェアとソフトウェアの更新手順を管理します。

5. Root of Trust for Verification (RTV)
   - 暗号メカニズムに基づいて、またはプラットフォーム保護ストレージにプロビジョニングされたプラットフォーム固有の値に対してデータを検証することができる演算エンジン。
   - ファームウェア / ソフトウェアの完全性とデータの検証を管理します。

6. Root of Trust for Recovery (RTR)

- プラットフォーム構成を既知の状態に回復できる演算エンジン。
- ファームウェアが破損または使用不能になった場合の復旧を管理する。

プラットフォーム アーキテクチャに応じて、プラットフォームのニーズに基づいて追加の Root of Trust を定義することができます。以下のセクションでは、プラットフォームの Root of Trust サービスの例を示します。

## 4.2 Platform Verified Boot

Verified Boot（別名、セキュア ブート）は、実行前に、変更可能なファームウェアとソフトウェア コンポーネントの完全性と真正性を検証し、妥当性を確認するプロセスのことです。プラットフォームの実装に応じて、ファームウェアの検証は、ブートプロセスのさまざまな 段階で開始することができます。Verified Boot は、検証の Root of Trust に基づくものとし、この Root of Trust は、Hardware Platform Root of Trust または Immutable Platform Firmware に固定されるものとします。

検証プロセスでは、暗号化アルゴリズムを用いてファームウェアの完全性をチェックします。これには、プラットフォームのファームウェア イメージとメタデータに対して、ハッシュ値を計算する必要があります。計算されたハッシュは、暗号証明書と照合するか、ハードウェアによる改ざんから保護され、Hardware Root of Trust に固定されたハッシュ値を使用して検証されます。

この検証プロセスは、検証されたファームウェア イメージを実行する前に、正常に完了する必要があります。

プラットフォームのセキュリティ ポリシーによっては、検証の失敗がシステムの停止、システム ファームウェアのリカバリ、または制限された機能でのシステムの起動につながる可能性があります。

**図 2** は、不変なプラットフォーム ファームウェアから始まり、ファームウェア コンポーネントの検証の連鎖を実装する、ファームウェア検証プロセスの例を示しています。

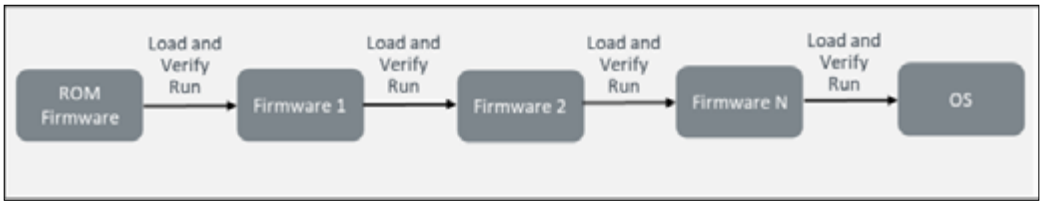


**図 2** 検証済みブート。検証の連鎖。

**図 3** は、最初に検証された可変なファームウェアによって、あるいは、ファームウェア ロードと検証プロセスを制御する信頼されたハードウェアによって、開始されるファームウェア検証プロセスの例を示しています。

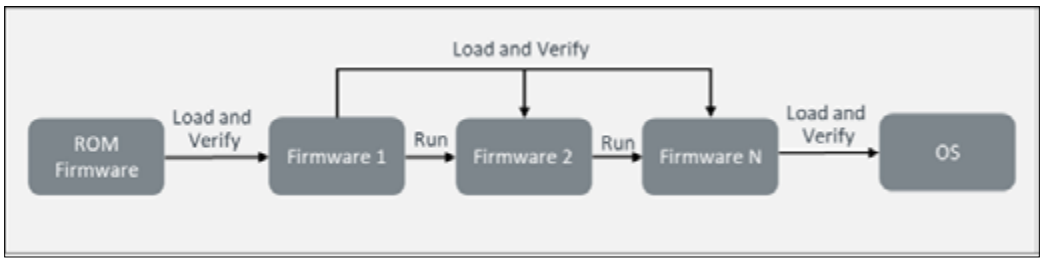


**図 3** 検証済みブート。信頼できる環境による検証。

プラットフォーム固有の Verified Boot の実装および Verified Boot を使用したプラットフォームのブートフローについては、付録 B および 付録 C を参照してください。

## 4.3 Platform Measured Boot

Measured boot とは、コードと重要なデータを暗号的に測定し、セキュリティの状態を後で証明できるようにするプロセスのことです。Measured boot は、ハードウェア プラットフォームの Root of Trust、または不変なプラットフォーム ファームウェアに固定された、計測の Root of Trust に基づいていなければなりません。

計測プロセスには、プラットフォーム ファームウェアのバイナリ イメージ、プラットフォーム コンフィギュレーション データ、ブート パラメータの計測が含まれますが、これらに限定されるものではありません。例えば、TPM を使用する実装では、プラットフォーム構成レジスタ（PCR）ストレージを使用することができます。PCR に格納されたプラットフォーム測定値は、プラットフォーム構成を一意な数字として表し、プラットフォーム認証の目的に使用することができます。付録 A に TCG TPM の概要を示しますが、TPM は必須要件ではないため、そのような TPM が必要かどうかはシステムが判断します。

Measured Boot はプラットフォームのブートプロセスには影響しません。

プラットフォーム認証ツールは、ネットワーク上にあるリモート認証サービスだけでなく、OS サービスの一部として実装することもできます。

Measured Boot は、プラットフォームのブートプロセスには影響しません。

**図 4** は、イミュータブルなプラットフォーム ファームウェアから始まるファームウェア測定プロセスを示しています。これは、次のファームウェア イメージの測定値を使用して、PCR における現在の測定値を拡張することによって、一意の測定値を生成します。

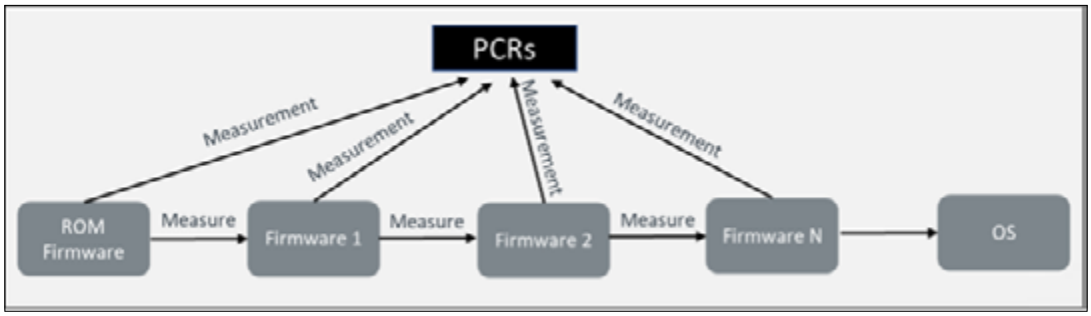


**図 4** MEASURED BOOT。各ファームウェアは次のファームウェアを測定する。

**図 5** は、イミュータブルなプラットフォーム ファームウェアから始まるファームウェア測定プロセスを示しています。すべての測定は、最初に検証されたミュータブル ファームウェア、またはファームウェアのロード プロセスを制御する信頼されたハードウェアによって行われます。これは、PCR における現在の測定値を、イメージのシーケンスの測定値で拡張（付録 A）することで、独自の測定値を生成します。

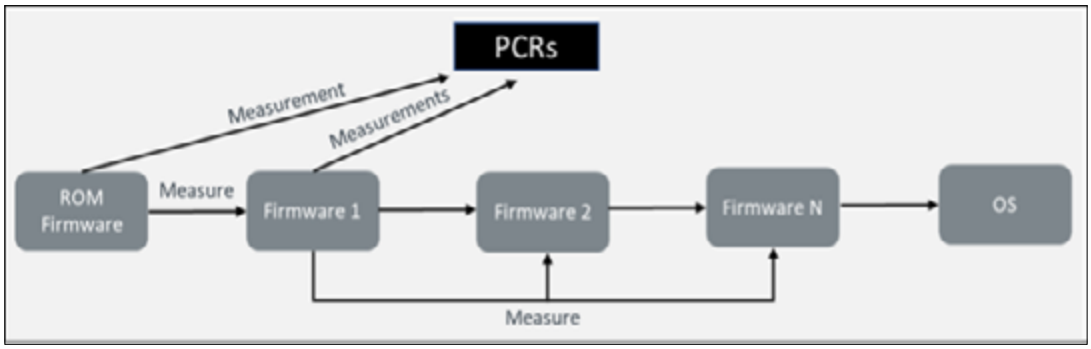


**図 5** MEASURED BOOT。信頼できる環境での測定。

プラットフォーム固有の Measured Boot の実装と Measured Boot を使用したプラットフォームのブートフローについては、付録 B と 付録 C を参照してください。

## 4.4 信頼されたプロセスのプラットフォーム分離

信頼されたプロセスの分離を実現する仕組みは、プラットフォームによって大きく異なります。多くのプラットフォームにはシステムの他の部分から分離され保護されたセキュアな実行環境を提供するシステム CPU が含まれています。他の選択肢としては、システム チップセットや専用セキュリティ チップ、例えば TPM やセキュア エレメント、オンチップ セキュア エンクレーブでの分離があります。システムは複数の分離技術を利用するのが一般的です。

検証され、測定されたブートは、Root of Trust を確立し、オプションとして他のコンポーネントを測定し、追加サービスを提供する上で重要な役割を果たすため、信頼されたプロセスである必要があります。

信頼されたプロセスの分離のプラットフォーム固有の実装については、**付録 B** と **付録 C** を参照してください。

## 4.5 プラットフォーム ブートフロー

プラットフォーム ブートフローとは、プラットフォームのブート時に実行される一連のプロセスのことです。このシーケンスは、イミュータブルなファームウェアから始まり、プラットフォーム OS のロードで終了します。プラットフォーム周辺機器やシステム内部インフラを初期化する複数のファームウェア コンポーネントが含まれることもあります。このシーケンスは、プラットフォームアーキテクチャに依存するので、実装によって異なる可能性があります。

ブートフローは、実行中の改ざんから保護され、プラットフォームのファームウェアとソフトウェアの不正な改変を検出するメカニズムを実装しなければなりません。

Akraino のブループリントは、現在、複数の種類のプラットフォーム上での実行をサポートしています。**付録 B** と **付録 C** は、サポートされるプラットフォームアーキテクチャ上のブートフローに関する情報を提供し、プラットフォームファームウェアの整合性保護のためのセキュリティメカニズムについて説明しています。

# 5 アンケート

Akraino プラットフォームセキュリティアンケートは、プラットフォームのハードウェア、ファームウェア、およびホストソフトウェアのセキュリティに関する一連の質問を提供しています。これらの質問は、プラットフォーム プロバイダーによって実装されたセキュリティレベルを評価するために使用されるべきであり、プラットフォームアーキテクチャには関係ありません。Akraino ブループリントの所有者は、Akraino ブループリントを実行するプラットフォームのセキュリティ要件を追加するためにこのアンケートを使用することができます。このアンケートは、プラットフォーム プロバイダーが回答することを目的としています。

## 5.1 ユースケース

以下のユースケースは、さまざまなブループリントの開発と展開シナリオに Akraino PSA アンケートを適用する方法のガイダンスを提供します。

1. ブループリント作成者：

- ブループリント ユーザーがブループリントのプラットフォームを提供します。ブループリント作成者は、ブループリントを実行するプラットフォームのセキュリティを確保するためにブループリント ユーザーが遵守すべき、または遵守しなければならない追加のセキュリティ要件として機能する PSA 要件を参照することができます。
- ブループリント作成者は、ブループリントのプラットフォームを提供します。ブループリント作成者は、ブループリントを実行する安全なプラットフォームであることをブループリント利用者に証明する、完成した PSA 要件への参照を提供することができます。

2. ブループリント ユーザー：

- ブループリント ユーザーは、PSA 要件を利用して、提供されたプラットフォーム上でのブループリントの特定の使用に関するプラットフォーム レベルのセキュリティ要件を満たします。ブループリント作成者が PSA を明示的に参照していない場合でも、ブループリント利用者はこの手順を実行できます。この場合、ブループリント ユーザーは、PSA 要件が適用された状態でブループリントが期待どおりに動作することを確認するためのテストに責任を負います。
- パブリッククラウドまたはプライベートクラウドの実行環境について、ブループリント利用者は、提供されるプラットフォームが必要なセキュリティレベルを実装し、安全な実行環境を提供することを保証するために、PSA のアンケートに回答するようプロバイダーに要求することができます。

3. チップ プロバイダー：

- PSA の要求事項を利用することで、自社の製品がこれらの要求事項を満たしていることを確認することができます。これは、PSA 要件を満たすチップの市場において潜在的な競争優位性を提供することができます。

4. セクション 5.2 のプラットフォーム セキュリティ要件を満たすチップを利用するハードウェア デバイス プロバイダー（サーバー、IoT デバイスなど）は、PSA 要件を満たすデバイスの市場で潜在的な競争優位性を得ることができます。

5. オープンソース ソフトウェアの使用を許可している組織は、PSA 要件を使用して、サポートするブループリントの追加プラットフォームセキュリティ要件を定義できます。PSA 要件は、組織のセキュリティ体制を強化する役割を果たします。

## 5.2 プラットフォーム セキュリティ アンケート

プラットフォーム セキュリティ アンケートは、Immutable Platform Root of Trust と Platform Security Root of Trust に基づくプラットフォーム セキュリティ コンポーネントの評価を提供します。

| # | 質問 | 詳細 |
|---|---|---|
| 1 | プラットフォームは、セキュア処理環境（SPE）を非セキュア処理環境（NSPE）から分離するためのハードウェア機構を提供しなければならない。 | 異なるセキュリティ レベルのプラットフォーム データとコードは、互いに隔離されるべきである。 |
| 2 | チップは、Immutable Platform Root of Trust のコードから起動する Verified Boot をサポートしなければならない。<br>• すべてのセキュア プロセシング環境（SPE）コード、および<br>• 非セキュア プロセッシング環境（NSPE）の最初の可変コード | プラットフォーム ブートは、hardware- based root of trust（hRoT）から開始し、次の変更可能なファームウェア イメージを検証しなければならない。 |
| 3 | Verified Boot によって開始された場合、システムは NSPE コードの検証されたブートまたは Measured Boot のいずれか（または両方）をサポートしなければならない。 | プラットフォームは、hRoT に固定された検証済みブートが完了した後、検証済みブートまたは Measured boot のいずれかを使用しなければならない。 |
| 4 | ( 該当する場合 ) プラットフォームは、セキュリティライフサイクルをサポートすること。すなわち、デバイスのライフサイクル状態に基づいて重要なセキュリティパラメータと機密データを保護し、状態間の移行ルールを実施すること。プラットフォームの認証報告書は、プラットフォームの現在のライフサイクル状態を提供すること。 | ライフサイクルをサポートするプラットフォームは、プラットフォームのライフサイクル状態の変更（例えば、本番状態から RMA/ デバッグ状態への変更）の間、重要なデータを公開してはならない。 |

| # | 質問 | 詳細 |
|---|---|---|
| 5 | プラットフォームは、以下の重要なセキュリティパラメータの最小セット（または同等のもの）の保存または生成をサポートするものとする。<br>• プラットフォーム RoT 公開鍵（ROTPK）、またはそのハッシュは、セキュアブート時に、最初の アップデート可能なファームウェア コンポーネント コードの認証に使用される。<br>プラットフォームで定義されている場合：<br>• 他のデバイスの秘密を導き出すために使用される秘密のハードウェア固有キー（HUK）。<br>• 秘密の認証キーと、認証キーを一意に識別する識別子。<br>• チップ上のプラットフォーム セキュリティ RoT を一意に識別する識別子。 | プラットフォームは、異なるプラットフォームのセキュリティ操作（検証済みブート、認証な ど）に使用される一意の鍵の保管をサポートしなければならない。 |
| 6 | プラットフォーム セキュリティ RoT は、ファームウェアとアプリケーション RoT の安全なアップデートをサポートしなければならない。アップデートは、ローカルに接続されたデバイス（リムーバブルメディアなど）またはリモート サーバーから配信されます。 | プラットフォームは、安全なファームウェア アップデート メカニズムを実装しなければならない。 |
| 7 | 更新メカニズムは、更新の不正なロールバックを防止しなければいけません。復旧のために、許可されたロールバックをサポートするメカニズムが与えられる場合もあります。ロールバック防止は強く推奨されますが、PSA では必須ではありません。 | プラットフォームは以下から保護されなければならない。ファームウェアの不正なロールバック。 |
| 8 | プラットフォームセキュリティ RoT は、プラットフォーム セキュリティ パラメータ、システムソフトウェア、およびデバイス機密データの不正な改変から保護されなければいけません。プラットフォームセキュリティ RoT は、プラットフォームセキュリティパラメータ、システムソフトウェア、およびデバイス機密データへの許可されたアクセスを強制することができます。 | プラットフォームセキュリティ RoT は、プラットフォームの機密データ（例えば、プラットフォーム RoT 公開鍵）へのアクセスを保護し、認可するために使用されなければならない。 |
| 9 | プラットフォームセキュリティ RoT は、その資産を保護するために、ベストプラクティスの暗号技術（アルゴリズム、鍵、ハッシュサイズなど）を使用するものとする。これには、ハードウェアベースの RNG による適切なランダムデータのソースを提供することが含まれます。 | プラットフォームは常にベストプラクティスの暗号を使用しなければならない。ベストプラクティスは地域や国によって異なる場合があります。 |

## 5.3 システム ソフトウェア セキュリティ アンケート

以下の表は、システムソフトウェアセキュリティに関する質問の詳細です。

| # | 質問 | 詳細 |
|---|---|---|
| 1 | システム ソフトウェアは、ローカルに接続されたデバイス（リムーバブル メディアなど）またはリモート サーバーから更新可能でなければいけません。 | システムの有効期間中にセキュリティ関連の修正が必要になることが予想されるため、システム ソフトウェアのアップデートが検証可能であることを前提としたこの要件が設定されています。 |
| 2 | ( オプション）更新メカニズムは、システムソフトウェアおよび認証データの不正なロールバ ックを防止するものとします。復旧のために、認可されたロールバックをサポートするメカニズムを提供することもできます。 | システムソフトウェアのロールバックは、禁止または許可されるべきです。 |
| 3 | システムソフトウェアは、Platform Security RoT ID のすべての問い合わせについて、Platform Security RoT のみに依存するものとします。 | 2 つの使用例：<br>• ID は直接プラットフォーム RoT に基づいています。例えば、KF Edge FDO プロジェクトでは、プラットフォームの ID を確立するために TPM を使用しています。<br>• ID は SW で定義されている。このような場合、プラットフォーム RoT を使用して、この ID への不正アクセスから保護する必要があります。例えば、TPM に格納されたキーで LUKS ディスク暗号化を使用します。 |
| 4 | システム ソフトウェアは、セキュア ストレージを使用して機密データを保護し、アプリケーション データに対してこの機能を提供するものとします。サポートされている場合、特定のデバイス インスタンスに機密データを封印し、セキュリティ ライフサイクル状態を含めなければいけません。 | 例えば、TPM に保存され、拡張認可ポリシーで保護された暗号化キーでデータシーリングを使用する場合があります。<br>もう一つの例は、TPM にキーを保存した LUKS ディスク暗号化です。<br>もうひとつの例は、信頼された環境で実行されるプラットフォーム サービスです。 |

| # | 質問 | 詳細 |
|---|---|---|
| 5 | システム ソフトウェアは、その資産を保護するために、ベストプラクティスの暗号技術（アルゴリズム、キー、ハッシュ サイズなど）を使用するものとします。これには、ハードウェアベースの RNG による適切なランダムデータ ソースの提供が含まれます。プロプライエタリな暗号アルゴリズムに依存したり、標準的な暗号アルゴリズムをカスタマイズしたりすべきではありません。 | 暗号化に関する適切な業界のベストプラクティスに従わなければなりません。こうした実践は時代とともに変化しており、地域によっても異なる場合があります。<br>適切なソースを選択し指示することは、ブループリント所有者の責任になります。 |
| 6 | システム ソフトウェアは、データの完全性、機密性、または真正性をサポートするために双方向接続を確立する際に、リモート デバイスおよび サーバーを認証する機能を提供するものとします。 | 例：<br>• mTLS<br>• TO2 プロトコルに従って確立された LF エッジ FDO チャンネル |
| 7 | システム ソフトウェアは、リモート デバイスおよび サーバーと交換するデータを暗号化し、完全性を強制する機能を提供するものとします。 | 例は上記 #6 と同じ |
| 8 | システム ソフトウェアは、認証と双方向通信の保護のために、安全なプロトコルを使用するものとします。これらのプロトコルは、デバイスの識別につながるデータを漏洩してはいけません。 | TLS は許容されるプロトコルの一例です。 |
| 9 | ( オプション）使用目的に必要のない機能パッケージはインストールしないか、インストールしないことが現実的でない場合は無効にします。 | OS または UEFI パッケージがブループリントで使用されない場合 ( そして削除するのが現実的な場合 )、誤用の可能性を防ぐためにインストールからこのパッケージを削除することをお勧めします。 |
| 10 | システムソフトウェアは、デバイスの真正性及びその完全性を証明するために使用できる証明方法をサポートしなければいけません。可能であれば、デバイスの現在のセキュリティ ライフサイクルの状態を含めるべきです。 | ソフトウェアエージェントの支援により、測定されたブートと対応するプラットフォームの認証をエンドツーエンドでサポートします。<br><br>プラットフォームはこの機能を持つべきですが、その有効化はシステムユーザーおよび / またはブループリント開発者の裁量によります。 |

| # | 質問 | 詳細 |
|---|---|---|
| 11 | システム ソフトウエアは、関連するセキュリティ イベントやエラーのログを記録します。ログには、根本原因の分析に必要な十分な詳細が含まれていなければならず、不正アクセスから保護されていなければいけません。 | セキュリティ ロギングは、インシデントの調査と監査に不可欠です。何をログに記録するかは、システム ユーザーおよび / またはブループリント開発者の裁量に任されています。<br>ディスク暗号化（LUKS など）の使用は、ログ保護の一例です。 |
| 12 | サポートされている場合、システム ソフトウェアは、意図された機能に必要な最低レベルの権限で、アプリケーション固有のソフトウェアおよびシステム ソフトウェアの実行を可能にするものとします。 | OS（Linux など）は、アカウントおよび／または特権が最小特権レベルに設定されていない状態で使用してはならず、ブループリントの所有者は、そのコンポーネントおよびユーザーに対しても最小特権レベルを使用しなければなりません。 |
| 13 | システム ソフトウェアが認証および認可パラメー タをリセットするメカニズムを持つ場合、これらのパラメータは、普遍的な工場出荷時のデフォルト値にリセット可能であってはいけません。<br>そのようなデータは、自動化された手段によって容易に決定されるものであってはならないし、一般に入手可能な情報から得られるものであってはいけません。 | システム ソフトウェアは、ユーザが認証パラメータをリセットするための安全なメカニズムを備えているべきです。<br>この仕組みの実装は、パスワードの入力を必要とするシステム リセット、電子メール 応答検証、セキュリティ質問など、さまざまな回復プロセスに基づくことができます。 |
| 14 | システム ソフトウェアがユーザ認証に暗号アルゴリズムを使用する場合、その機能に使用される暗号は、システム ソフトウェア セキュリティ セクションの要件 #5 に準拠するものとします。 | システムは、証明書（SSH アクセスなど）、スマートカード、認証子など、ユーザー認証にさまざまな方法を使うことができます。 |
| 15 | システム ソフトウェアがデータの永続的な保存を可能にする場合、不正アクセスに対する保護とアクセス権管理の仕組みをサポートしなければいけません。 | システムは、プラットフォームストレージまたは TPM 拡張認証ポリシー上のデータに対する異なるユーザーアクセス権を定義することができます。 |

# 6 結論

Platform Security は、Akraino ブループリントのセキュリティをハードウェアレベルまで拡張するために必要なツールを提供します。Akraino Platform Security はベンダーにとらわれないアプローチをとり、すべてのベンダーがサポートすべき基本的な Platform Security 分野に焦点を当てようとしています。定義されたプラットフォームセキュリティ要件を実装することで、Akraino ブループリントのユーザーは、(1) ソフトウェアを実行するハードウェア / ファームウェアが、次のようなものであることを保証することができます。(2) プラットフォーム セキュリティーに関わるキー／パスワード／情報を保存する安全な領域が提供されていること、(3) ファームウェアの安全なアップデートが保証されていること、(4) システム ファームウェアの完全性が維持されていること。

要約すると、Akraino Platform Security を実装することで、ブループリントの所有者は、利用可能なプラットフォームセキュリティツールを活用して、Akraino ブループリントが安全なプラットフォーム上で実行されていることを確認できます。

# 付録 A

この付録では、プラットフォームに依存しないセキュリティ モジュールと デバイスの実装について説明します。

## A.1 測定と報告の Root of Trust としての TPM

TCG（Trusted Computing Group）は、TPM（Trusted Platform Module）を定義する仕様を発表した業界団体です。現在定義されているバージョン（またはファミリー）は TPM 2.0 です。TPM の使用は各実装に依存し、Akraino プラットフォームの必須要件ではありません。

TPM 仕様には複数の機能が定義されています。そのうちの 2 つが、今回の Measured boot の主な対象です。

**TPM はストレージの Root of Trust（ROTS）として機能します。**

- 測定値はプラットフォーム コンフィギュレーション レジスタ（PCR）に格納されます。
- TPM ブート毎に、全ての PCR は初期値にリセットされます。
- PCR の値を変更する唯一の方法は「拡張」することです。拡張とは、TPM が現在の PCR 値に新しい測定値（通常はコードまたはデータのハッシュ）を連結したハッシュ値を計算し、新しいハッシュ値を PCR に格納することを意味します。以下の記号 "||" は連結を意味します。

  PCR(new) = HASH (PCR(old) || measurement(new))
- TPM の仕様では、複数のハッシュアルゴリズムが定義されています。
- プラットフォームが PCR を複数回拡張する場合、すべてのデータは PCR にハッシュされます。異なる順序で測定を拡張すると、異なる PCR 結果値が得られます。
- Measured boot が完了した後、各 PCR は、測定されたブートの間にこの PCR に対して TPM によって実行されたすべての拡張操作の積である最終値を保持します。

• TPM は、Root of Trust for Reporting（ROTR）としての役割も果たします。TPM はブートベリファイアからの応答で署名された見積もり（別名、Measured boot レポート）を生成することができます。他の多くのパラメータの中の見積もりは PCR 値を含んでいます。また、TCG SCRTM (Static Core Root of Trust for Measurement) イベントログがあり、すべての完了した拡張操作のログを記録しています。検証者はこのログを要求し、必要に応じて " 再生 " することができます。以下は引用署名キーチェーンです。

- TPM には、その製造過程で用意された非対称のエンドースメント鍵（EK）が付属しています。EK の秘密部分は TPM だけが知っています。EK 公開部分は、TPM 製造元が発行する証明書に含まれています。証明書は製造 CA の署名鍵によって署名されます。
- プラットフォームの所有者は、TPM に非対称認証鍵ペア（AK）の生成を指示します。AK の秘密部分は TPM のみが知っており、公開部分は EK によって署名されます。

AK は TPM 見積書に署名するために使用されます。EK は、プライバシーへの配慮から、TPM 見積もりへの署名には直接使用されません。

# 付録 B

この付録では、インテル固有のセキュリティモデルとブートフローについて説明します。

## B.1 インテル プラットフォーム セキュリティ モデル

### B.1.1 x86 プラットフォームでの Measured boot サポート

下の図は、検証されたブートと測定されたブートの概念を並べて示したものです。

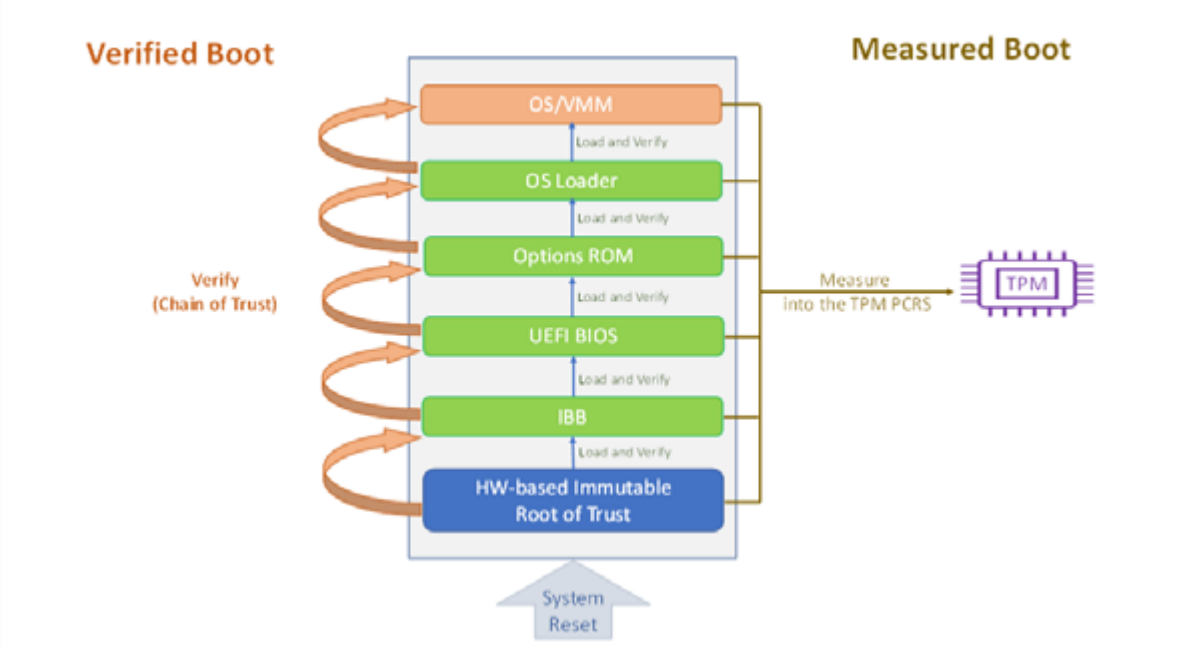


**図 6** VERIFIED BOOT と MEASURED BOOT を比較。

### B.1.2 x86 プラットフォームでの Immutable RoT サポート

歴史的に、x86 ベースのプラットフォームは、ファームウェアの検証、測定、更新、およびリカバリのために、としてブート ROM/FLASH を利用することが多くありました。このアプローチは、Immutable Root of Trust を提供せず、結果として、Akraino プラット フォームのセキュリティ要件を満たしません。

幸いなことに、最近の x86 ベースのシステムには、Akraino のプラットフォーム セキュリティ要件を満たす HW ベースのセキュリティ機能が搭載されています。OEM は適切な CPU/SoC 機能を利用することで、x86 ベースのシステムにエンドツーエンドのセキュリティソリューションを提供することができます。

| Root of Trust | インテル ® ブートガード | インテル ® Bios ガード | インテル ® PFR |
|---|---|---|---|
| 検証 | Yes<br>(CPU/ACM) | - | Yes<br>(PFR CPLD) |
| 測定 | Yes<br>(CPU/ACM) | - | Yes<br>(PFR CPLD) |
| ストレージ | Yes<br>(TPM/PCR) | - | Yes<br>(TPM/PCR) |
| レポーティング | Yes<br>(TPM/EK and AIK) | - | Yes<br>(TPM/EK and AIK) |
| ファームウェア<br>アップデート | - | Yes<br>(SMM and ACM) | Yes<br>(PFR CPLD) |
| ファームウェア<br>リカバリ | - | - | Yes<br>(PFR CPLD) |

ACM – 認証されたコンピュート モジュールは、（CPU マイクロコードと連動して）Immutable Root of Trust の一部として特別なモードで実行される。

CPLD – Complex Programmable Logic Device の略。CPLD は、インテル ® PFR に基づいて設計されたシステムにおける Root of Trust です。

TPM – トラステッド プラットフォーム モジュール

「インテル ® Boot Guard を使用したプラットフォーム ブート フロー」では、インテル ® Boot Guard を使用したブートの計測と安全性について詳しく説明します。

# B.2 インテル プラットフォームのブートフロー

## B.2.1 インテル ® Boot Guard を使用したプラットフォーム ブートフロー

インテル ® Boot Guard は、Verified boot の場合は HW ベースの Root of Trust の一例であり、Measured boot の場合は Root of Trust for Measurements の一例です。プラットフォームブートは、以下から始まります。

- チップセットで実行され、OEM ブート ポリシーを読み取るインテル サーバー プラットフォーム サービス
- インテル ® 認証コード モジュール (ACM) バイナリを認証する CPU マイクロコード。

- OEM が提供するイニシャル ブート ブロック (IBB) を検証するインテル ® ACM。

ブートガードは、PCH（プラットフォーム コントローラー ハブ）フィールド プログラミング ヒューズ（FPF）に依存し、OEM 公開鍵のハッシュ値とブートガードのブートポリシーを保存します。FPF は一度だけプログラムすることができ、OEM はプラットフォームの製造プロセス中にこれを行います。

ACM は、ブートガード ソリューションにおいて重要なタスクを実行します。ACM はインテルによってデジタル署名され、BIOS やその他のファームウェア コンポーネントとともにフラッシュ上に保存されます。ACM の署名を検証するための公開鍵は、インテルの CPU にハードコードされています。CPU のオペコードは、ACRAM（Authenticated Code RAM）と呼ばれる内部で保護された L2 キャッシュに ACM をロードし、ACRAM が ACM を検証し、検証に成功した場合のみ、CPU は ACM の実行を許可する。ACM の認証に失敗すると、CPU がシャットダウンします。

ACM は、OEM によって提供され署名された初期ブートブロック（IBB）をロードします。IBB は OEM のブートガード固有の要件です。通常、これにはシリコンとメモリの初期化 FW が含まれます。その理由は、内部で保護されたメモリ内で ACM が素早く測定できる小さなモジュールを提供するためです。ACM は IBB を最終レベルのキャッシュにロードし、No-Eviction Memory を有効にすることで、BIOS コードが実行できる安全で隔離された環境を作ります。次に ACM は、FPF 内の OEM Boot Guard キーストアを使用して IBB の署名を検証します。検証に成功した場合のみ、ACM は IBB に制御を移します。IBB の検証に失敗すると、SPS の実施により CPU がシャットダウンされます。

以下の図は、Boot Guard ACM と IBB の実行フローを表しています。

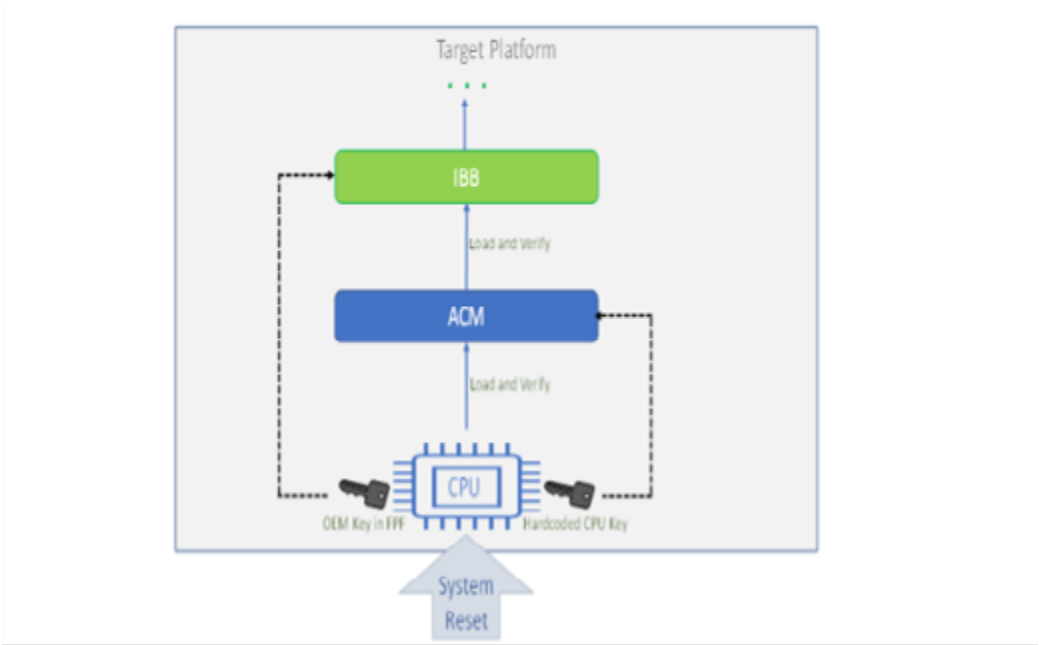


**図 7** インテル ® BOOT GUARD のコンセプト

下図は、インテル ® Boot Guard が Immutable な HW ベースの検証 Root of Trust として機能する、簡略化された Verified boot フローを示しています。

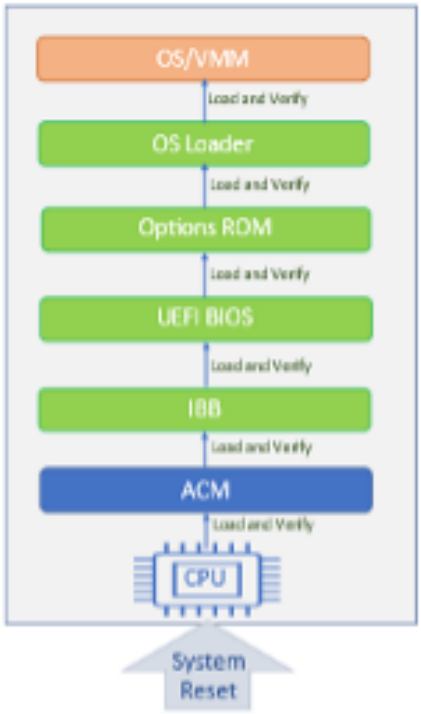


**図 8** VERIFIED BOOT 例

システムがリセットされた後、CPU は ACM をロードし検証します。ACM は前節で説明したように IBB をロードし検証します。続いてロードされるコンポーネントの完全性は、Root of Trust によって実行されます。以下のコンポーネント ( 緑色 ) は OEM によって提供され、各コンポーネントは、後続のコンポーネントに制御を移す前に、後続のコンポーネントをロードして検証しなければなりません。ハッシュまたはデジタル署名を使用することで、ロードと検証の過程でデータの整合性を確保することができます。検証に失敗した場合、ブート プロセスは終了します。図中の各矢印は、この "load-and-verify-before-execute " パターンを表しています。最終的に、システムは OSV が提供する OS/VMM をロードし検証します。

検証されたブートは、測定されたブートと組み合わせることができます。検証されたブートがどのように全体像に適合するかについては、B.2.3 節を参照してください。

## B.2.2 インテル ® Boot Guard と TPM による Measured boot フロー

Measured boot は Verified Boot とは異なります。Verified Boot では、各コンポーネントは次のコンポーネントを認証し、暗号的に検証します。認証や検証に失敗した場合、検証済みブートは終了します。Measured boot では、あるコンポーネントが次のコンポーネントをロードする前に、次のコンポーネントを TPM PCR に検証します。Measured boot では、ブート中に検証は実行されないため、失敗することはありません。このブートシーケンスは Chain of

Trust (COT) と呼ばれます。このセクションで後述するように、ブートフローの最初の 2 つの FW コンポーネント（ACM と IBB）の実行に関連する 1 つの例外があります。

下図は、インテル ブート ガードを計測の Root of Trust として使用し、TPM をストレージとレポートの Root of Trust として使用する簡略化された Measured boot フローを示しています。

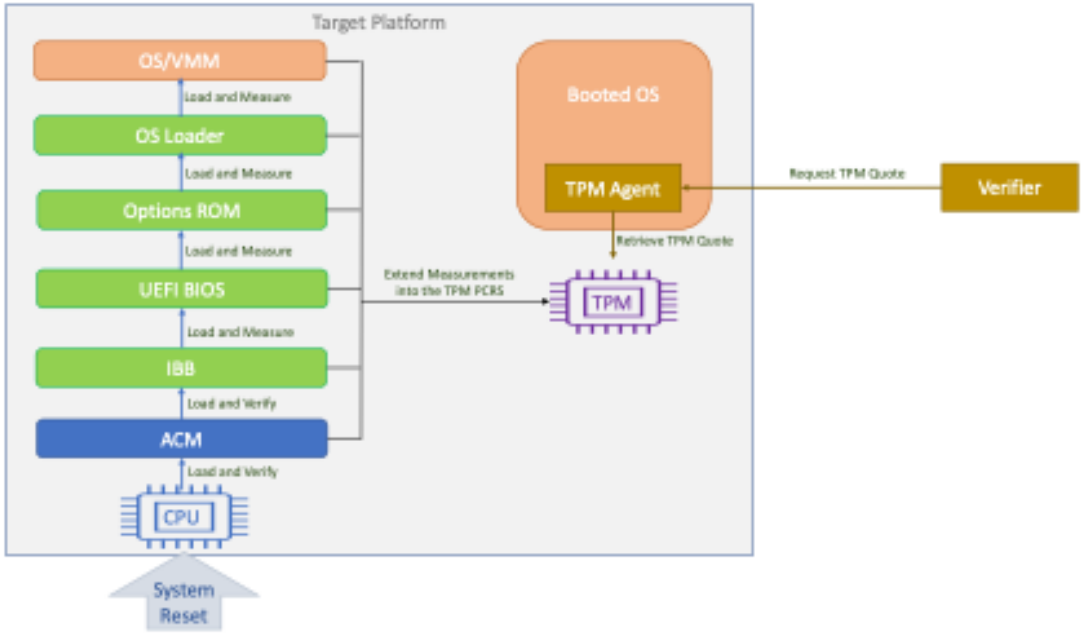


**図 9** MEASURED BOOT の例

Measured boot は、セクション B.2.1 で定義された 、ブートガード ACM と IBB を実行することによって開始します。CPU のオペコードによる ACM 検証、または ACM による IBB 検証のいずれかが失敗した場合、ブートは終了します。CPU と ACM、ACM と IBB の間の矢印は、（残りの）プラットフォームが Measured boot 用に構成されているにもかかわらず、Verified Boot の動作を表しています（他の青い矢印については、「ロードと測定」ではなく「ロードと検証」のテキストに注目してください）。さらに、ACM は、IBB の測定に加えて、自己測定を TPM PCR0 に拡張します。次に、IBB と Chain of Trust 内の後続の各コンポーネント（最後のコンポーネントを除く）が次のコンポーネントをロードし測定する。Chain of Trust は、TCG プラットフォームファームウェアプロファイル仕様に従って、すべての測定を対応する TPM PCR に拡張します。

リセット時にはメモリが利用できないため、ACM は、ブートガードが有効かどうかを判断し、TPM イベントログ用の最初のイベントを生成するために、検証済みの BIOS に依存します。イベントログの再生が成功するように、BIOS によってさらにイベントが記録される前に、イベントログのヘッダー、ローカリティスタートアップイベント、および SCRTM イベントを作成する必要があります。

Measured boot が完了した後、プラットフォームは信頼されているとみなされますが、必ずしも安全ではありません。例えば、PCR 値はコンポーネントの 1 つに既知の脆弱性があることを示すかもしれません。TPM の見積もりと、任意で対応するブートログに基づいてセキュリティ判断を下すのは、他のソフトウェア（上図ではベリファイアと呼ばれる）次第です。ベリファイアは、

TPM（引用）のみを信頼しなければなりません。ターゲット プラットフォーム上の TPM エージェントは、TPM へのアクセスを容易にしますが、システムの信頼ステータスに関係なく、また知られる前に通常の OS プロセスまたはシステムサービスとして実行されるため、信頼されません。例えば、その意味するところの一つは、ベリファイア自身が、可能性のあるクォート リプレイ攻撃を軽減するために、クォートに含めるために TPM に送信されるワンタイムノンスを生成しなければならないということです。

## B.2.3 Measured and Verified Boot

前の節では、Verified boot と Measured boot を切り離して説明しましたが、現実の世界では、これらを組み合わせて Measured and Verified Boot にすることができます。このような組み合わせには正当な理由があります。

- 例えば、プラットフォームは署名された OS カーネルで起動されるが、カーネルのバージョンが正しくない。
- 脆弱性を持つ可能性のある、古いがまだ正式には有効で署名されたファームウェアの起動を検出するため。
- Verified boot が失敗した場合にリカバリーモードでブートする機能がプラットフォームにある場合、通常ブートモードとリカバリーブートモードを区別するため。

Verified boot および / または Measured boot は、プラットフォームを保護するために単独で使用されるのではなく、例えば適切なポリシーを適用するための追加ソフトウェアと組み合わせて使用されます。

- ディスク暗号化ソフトウェアは、TPM に封印されたディスク暗号化キーを使用することができます。TPM PCR 値の組み合わせは、TPM 拡張認証ポリシーとして構成されます。鍵は、現在の TPM PCR の状態がポリシーに一致する場合にのみ解放されます。例えば、LUKS はこのように設定できます。
- Kubernetes (K8S) POD 実行ポリシーの実施。外部のハードウェア検証サービス（HVS）がプラットフォームの TPM クォートを取得しそれを検証します。HVS の検証結果は、プラットフォームが K8S ポッドの実行を許可されるかどうかを決定するために使用されます。
- プラットフォームの測定値に基づいて、ネットワークデバイスやストレージなどの外部リソースへのアクセスを制限します。

# 付録 C

## C.1 Arm プラットフォーム リファレンス ブートフロー

以下のセクションでは、Arm 社が提供するプラットフォーム ファームウェア リファレンス実装 [6] のブート フローとプラットフォーム ファームウェア検証シーケンスについて説明します。ファームウェアのブートフローは、プラットフォームの構成や要件に応じて、OEM/ODM や SiP によって調整されるかもしれません。しかし、プラットフォームベンダのブートフローの調整によって、プラット フォームファームウェアのセキュリティ要件が影響を受けるべきではありません。

### C.1.1 Trusted Boot

The Trusted Board Boot（TBB）[1] 機能は、通常世界のブートローダを含む、全てのファームウェア イメージを認証することによって、プラットフォーム上で悪意のあるファームウェアが実行されるのを防ぎます。これは、公開鍵暗号化標準（PKCS）を使用して、信頼の連鎖（Chain of Trust）を確立することによって実現されます。

### C.1.2 信頼の連鎖

信頼の連鎖（CoT）は、暗黙的に信頼されたコンポーネントのセットから始まる。Arm のリファレンス実装では、これらのコンポーネントは以下のとおりです。

- Root of Trust 公開鍵 (ROTPK) の SHA-256 ハッシュ。 [3] および [5] を参照のこと。
- BL1 のイメージは、ROM に保存されているため、改ざんされないという前提で。

CoT の残りのコンポーネントは、証明書またはブートローダ イメージです。証明書は X.509 v3 標準に従っています。この規格は、CoT を確立するために不可欠な情報を保存するために使用される証明書に、カスタム拡張を追加することを可能にします。

証明書は、「鍵証明書」と「コンテンツ証明書」に分類されます。鍵証明書は、コンテンツ証明書に署名するために使用された公開鍵を検証するために使用されます。コンテンツ証明書は、ブートローダ イメージのハッシュを保存するために使用されます。イメージのハッシュを計算し、コンテンツ証明書から抽出したハッシュと照合することで、イメージを認証することができます。

### C.1.3 実行とセキュリティのステート

Armv8-A または Armv9-A プロセッサの現在のステートは、例外レベル [2] と現在の実行ステートによって決まります。現在の実行状態は、汎用レジスタの標準幅と使用可能な命令セットを定義します。

実行状態は、メモリモデルや例外の管理方法にも影響を与えます。

現在のセキュリティ ステートは、どの例外レベルが現在有効か、どのメモリ領域にアクセス可能か、そしてそれらのアクセスがシステム メモリ バス上でどのように表現されるかを制御します。

**図 10** は、異なる実行ステートが使用されている場合の、例外レベルとセキュリティステートを示しています。

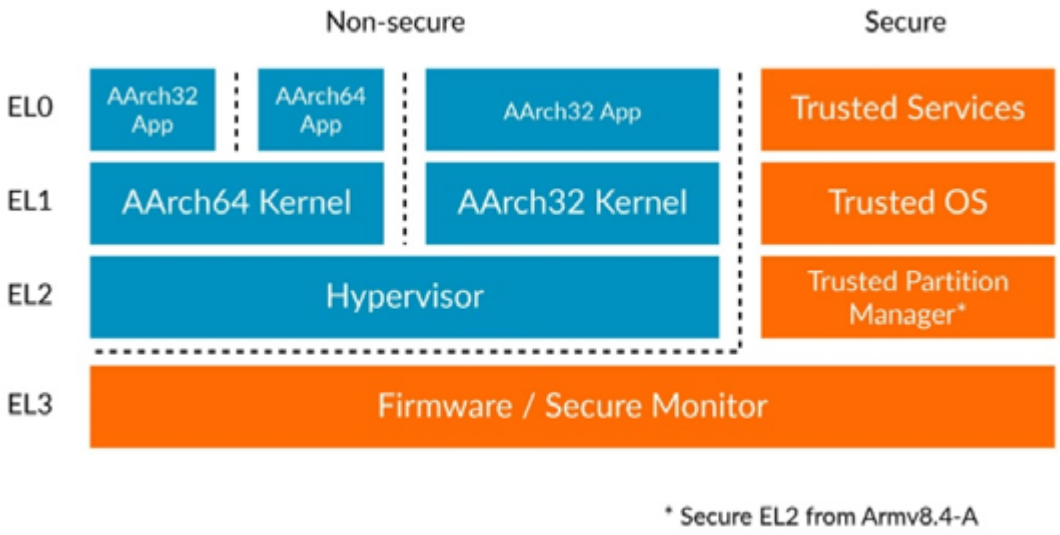


**図 10** ARMV8-A および ARMV9-A 例外モデル。

## C.1.4 トラステッド ボード ブート シークエンス

ブートシーケンスは、複数の段階を含み、ファームウェアの複数のイメージをロードするかもしれません。Arm リファレンス実装 [6] では、ブート ローダ（BL）と呼ばれる、特定のファームウェアをロードする 5 つのブート ステージが定義されています。

- ブートローダステージ 1 (BL1) AP トラステッド ROM。
- ブートローダステージ 2 (BL2) トラステッド ブート ファームウェア。
- ブートローダステージ 3-1 (BL3-1) EL3 ランタイムファームウェア。
- ブートローダステージ 3-2 (BL3-2) Secure-EL1 ペイロード（オプション）。
- ブートローダステージ 3-3 (BL3-3) 非信頼ファームウェア。

CoT は、**図 11** に示される以下の一連のステップを経て検証されます。ステップのいずれかに失敗した場合、システムは起動を停止するか、リカバリ モードに切り替わります。

- BL1 が BL2 のコンテンツ証明書をロードし、検証します。検証された証明書から発行者の公開鍵が読み出される。その鍵のハッシュが計算され、信頼されるルート鍵格納レジスタから読み出される ROTPK のハッシュと比較される。両者が一致する場合、証明書から BL2 ハッシュが読み取られる。
- BL1 は BL2 イメージをロードします。BL2 ハッシュが計算され、BL2 コンテンツ証明書から読み取られたハッシュと比較されます。すべての比較が成功した場合、制御は BL2 イメージに移されます。

- BL2 が信頼済み鍵証明書をロードし、検証する。検証された証明書から発行者の公開鍵が読み出される。その鍵のハッシュが計算され、信頼されるルート鍵保存レジスタから読み出される ROTPK のハッシュと比較される。比較に成功した場合、BL2 は検証済み証明書から信頼される公開鍵と信頼されない公開鍵を読み出し保存します。

次の 2 つのステップは、SCP_BL30、BL31、BL32 の各イメージに対して実行されます。オプションの SCP_BL30 および BL32 イメージのステップは、これらのイメージが存在しない場合はスキップされます。

- BL2 は BL3x 鍵証明書をロードし検証します。証明書の署名は、Trusted Key 証明書の Trusted World 公開鍵を使用して検証されます。署名検証に成功した場合、BL2 は証明書から BL3x 公開鍵を読み出し、保存します。
- BL2 は BL3x コンテンツ証明書をロードし、検証します。署名は BL3x 公開鍵を用いて検証されます。署名検証が成功した場合、BL2 は証明書から BL3x イメージハッシュを読み込み保存します。

次の 2 つのステップは、BL33 画像に対してのみ実行される。

- BL2 は BL33 鍵証明書を読み込み、検証します。署名検証に成功した場合、BL2 は証明書から BL33 公開鍵を読み込み保存します。
- BL2 は BL33 コンテンツ証明書を読み込み、検証します。署名検証が成功した場合、BL2 は証明書から BL33 イメージハッシュを読み込み保存します。

次のステップは、すべてのブートローダー イメージに対して実行されます。

- BL2 は各画像のハッシュを計算する。このハッシュを、対応するコンテンツ証明書から取得したハッシュと比較する。ハッシュが一致すれば、画像認証は成功する。

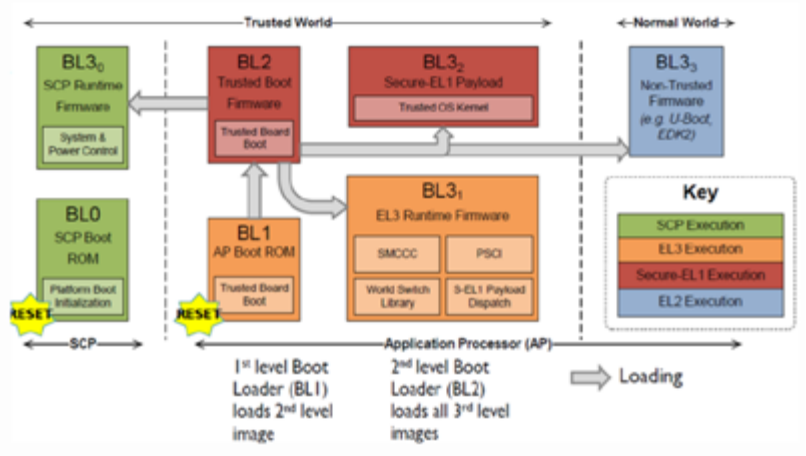


**図 11** 信頼されたブートフロー

## C.1.5 Measured Boot と認証

SoC は、そのソフトウェアの完全性をリモートパーティや同じボード上のローカルシステムに対して証明する必要があるかもしれません。プラットフォームの状態を証明するための前提条件は、各起動時にロードされたコードとデータの測定値を作成することです。測定値は、信頼できるサブシステムに安全に保存されます。これは測定されたブートとして知られています。ローカルまたはリモートの認証メカニズムに提供される測定レポートは、そのようなファームウェアの完全性を評価するために使用することができ、全体的な信頼の連鎖の一部となります。

信頼の連鎖の各段階では、ロードされるすべての重要なコードとデータを正確かつ堅牢に測定します。これには以下も含まれます。

- ロード可能なモジュール（ダイナミックパッチやペリフェラルからロードされたコードを含む）
- ブートの動作に影響を与えるパラメータ（例えば、ロードされたプログラムの制御フローを変更する可能性のあるフラグや変数）。

信頼の連鎖の各段階は、測定値をローカルな Root of Trust に格納します。測定値は、セキュリティモジュールまたは他のタイプの信頼されたサブシステム（TPM など、付録 A）に保持することができます。リモートパーティは、測定値のリストを使用して、プラットフォームの特定のソフトウェア ID を検証することができます。

Immutable なブートローダは、RoT ランタイムサービスがアクセス可能なブートステートを保存できます。ブート状態の一部には、ブートシードと呼ばれる新しく生成された番号が含まれています。ブートシードは、例えば検証エンティティが、異なる認証エンドポイントの認証が同じブートセッションで生成されたことを確認するために、後のサービスによって使用されるかもしれません。ブートシードは、グローバルな衝突が統計的に起こり得ないように十分な大きさでなければなりません。

システムは、署名されたファームウェア イメージ パッケージ、あるいは単一のイメージの形 で、複数のイメージを提供することが可能です。ファームウェア イメージ パッケージは、ブートローダ イメージ（および潜在的に他の ペイロード）を、ロード可能な単一のアーカイブにまとめることを可能にします。とはいえ、各コンポーネントは独立して測定されなければなりません。これは、リモートパーティがリモート認証を容易に検証するために必要です。

# 参考文献

[1] Trusted Board Boot Requirements: https://developer.arm.com/documentation/den0006/latest

[2] AArch64 Exception Model: https://developer.arm.com/documentation/102412/0102

[3] Platform Security Model: https://www.psacertified.org/app/uploads/2021/12/JSADEN014_PSA_Certified_SM_V1.1_BET0.pdf

[4] Platform Threat Model and Security Goals: https://www.psacertified.org/development-resources/building-in-security/platform-threat-model-and-security-goals

[5] Platform Security Boot Guide: https://developer.arm.com/documentation/den0072/0101/

[6] Arm Trusted Firmware for A-class processors: https://www.trustedfirmware.org/projects/tf-a/

[7] PSA Certified Level 1 Questionnaire: https://www.psacertified.org/app/uploads/2022/06/JSADEN001-PSA_Certified_Level_1-2.2-REL-01.pdf

[8] Intel HW Shield (including Intel(R) Boot Guard): https://www.intel.com/content/dam/www/central-libraries/us/en/documents/below-the-os-security-white-paper.pdf

[9] Secure Boot with Intel(R) Boot Guard in context of Network Infrastructure: https://builders.intel.com/docs/networkbuilders/secure-the-network-infrastructure-secure-boot-methodologies.pdf

[10] TCG Glossary: https://trustedcomputinggroup.org/wp-content/uploads/TCG-Glossary-V1.1-Rev-1.0.pdf

## 著者

Daniil Egranov (Arm Ltd.)
Eugene Yarmosh (Intel)
Randy Stricklin (AT&T)

## 貢献者

Catharine West (Intel)
Don Banks (Arm Ltd.)
Rob Smart (Arm Ltd.)

**このレポートは以下の文書の参考訳です。**
Akraino Platform Security Architecture

翻訳協力：吉田行男


AKRAINO
LF EDGE