TPMを用いたEnclaveアプリケーション実行端末の多層認証プロトコル

鎌倉 仁[1,a)]　掛井 将平[1]　白石 善明[2]　齋藤 彰一[1]

概要：信頼された実行環境である Intel SGX が生成する暗号化領域 Enclave を使うことで，第三者管理のクラウド環境においても機密データを OS や管理者から秘匿して安全に処理できる．Enclave が正しく機能していることは，リモートの実行環境を検証する Remote Attestation（RA）により確認できる．しかし，RA はハードウェア単位での検証であるため，マルチテナントの仮想環境では同一物理マシン上の異なる仮想マシン（VM）を識別できず，Enclave アプリが別テナントの VM で実行されていることを検知できない．本研究では，Trusted Platform Module（TPM）ベースの Enclave アプリ実行端末の多層認証プロトコルを提案する．提案プロトコルでは，物理 TPM により信頼された仮想 TPM の固有の署名鍵を利用して VM を一意に識別し，物理マシンと VM，Enclave アプリの組み合わせが承認された構成であることをリモートから検証可能にする．評価では，提案方式における端末のなりすまし攻撃への耐性を検証し，その安全性と限界について議論する．

キーワード：Intel SGX，Remote Attestation，仮想マシン，Trusted Platform Module

TPM-based Multi-layer Authentication Protocols for Devices Executing Enclave Applications

Hitoshi Kamakura[1,a)]　Shohei Kakei[1]　Yoshiaki Shiraishi[2]　Shoichi Saito[1]

Abstract: Trusted Execution Environment Intel SGX enables secure processing of sensitive data in third-party clouds by using an encrypted region called an Enclave, which keeps data confidential from the OS and administrators. While Remote Attestation (RA) can verify the Enclave's integrity, its hardware-level verification cannot distinguish between virtual machines (VMs) in a multi-tenant environment. This creates a security gap, as it cannot detect if an Enclave application is running on an unauthorized tenant's VM on the same physical machine. To address this, we propose a multi-layer authentication protocol for devices using the Trusted Platform Module (TPM). Our protocol leverages a virtual TPM, trusted by the physical TPM, to uniquely identify each VM via a distinct signature key. This enables remote verification of the entire authorized stack: the physical machine, the VM, and the Enclave application. We evaluate the method's resistance to device impersonation attacks and discuss its security and limitations.

Keywords: Intel SGX, Remote Attestation, Virtual Machine, Trusted Platform Module

1. はじめに

クラウドコンピューティングの普及に伴い，データ処理の性能や柔軟性が向上する一方で，機密性やプライバシの侵害に対する懸念が高まっている．保存時や転送時のデータは暗号化により保護できるが，処理時には復号されるため，処理中のデータが攻撃の対象となり得る．特に，リソースが共有されるクラウド環境では，処理中のデータが不正にアクセスされ，盗聴されるリスクがある．処理中のデータを保護する技術として，信頼された実行環境（Trusted Execution Environment，TEE）[1] が知られて

[1] 名古屋工業大学 Nagoya Institute of Technology
[2] 神戸大学 Kobe University
[a)] h.kamakura.035@nitech.jp

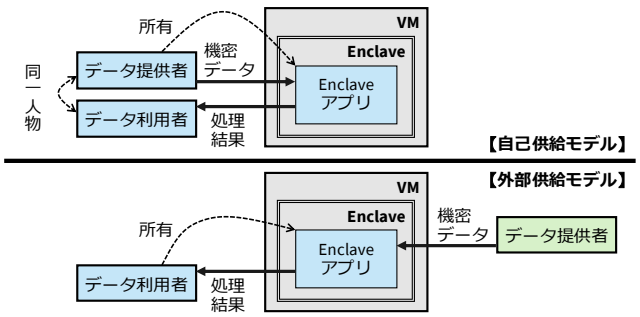

図 1 Enclave アプリの機密データ供給モデル
Fig. 1 Enclave App confidential data supply model

いる．TEE では，メイン OS とは別の独立した領域内でコードを実行し，OS に対する攻撃などからデータを保護できる．TEE の代表的な実装の一つである Intel Software Guard Extensions（Intel SGX）[2] は，メモリ内に生成された隔離実行領域（Enclave）でデータを暗号化したまま処理できる．これにより，Enclave 内部で実行されているコードや処理中データへのアクセスが遮断され，データの活用と機密性確保を両立している．

リモート環境の Enclave 内で動作するアプリケーション（Enclave アプリ）を認証する手法として，Remote Attestation（RA）[3] が知られている．RA では，ビルド時の Enclave アプリのコード情報とビルド環境の情報をもとに検証を行い，検証者は Enclave アプリのコードが改ざんされていないことや，通信先の Enclave アプリが本物であることを確認できる．

RA は CPU を含めたハードウェア単位で正当性を証明する仕組みであり，仮想環境では同一物理マシン上の個々の VM を識別できない．そのためマルチテナント環境では，Enclave アプリが別テナントの VM で実行されても検知できない．この問題は，**図 1** におけるデータ利用者と異なる主体が Enclave アプリに機密データを渡すモデル（外部供給モデルと呼ぶ）において顕在化する．例えば，異なる二者間がデータ連携を行う同モデルのフレームワーク [4] では，データ利用者がデータ処理を Enclave アプリに実装し，データ提供者はその Enclave アプリを認証して機密データを提供する．これにより，データ保有者（データ提供者）と分析技術保有者（データ利用者）が，機密性を維持しながら連携してデータを活用できる．しかし，データ提供者が Enclave アプリの実行端末を識別できなければ，同一の Enclave アプリが異なるテナントの VM で実行されても区別できない．その結果，処理データ自体は Enclave により保護されるものの，データが意図しない環境で利用されることとなる．このようなモデルでは，データ不正利用のリスクを軽減するために，Enclave アプリ自体の認証に加えて，その実行 VM および物理マシンを認証し，それらが承認された構成であることを検証できることが望ましい．

そこで本研究では，Trusted Platform Module（TPM）を利用した Enclave アプリ実行端末の多層認証プロトコルを提案する．提案プロトコルでは，VM 内の仮想 TPM で生成した固有の署名鍵と証明書を利用して VM を一意に識別する．さらに物理マシンの TPM を利用し，その固有の署名鍵で仮想 TPM の鍵を信頼することで信頼チェーンを構築する．

2. 関連技術

2.1 Intel Software Guard Extensions

Intel Software Guard Extensions（Intel SGX）は，Intel 製の CPU に搭載された，コードとデータの機密性および完全性を保護するためのハードウェアベースのセキュリティ技術であり，TEE の代表的な実装の一つである．CPU に追加された独自の命令により，アプリケーションは Enclave と呼ばれる隔離された保護領域をメモリ上に生成できる．Enclave 内にロードされたコードとデータは常に暗号化された状態で処理されるため，OS やハイパーバイザといった高い特権レベルで動作するソフトウェアからも，直接的なアクセスや内容の解読を防ぐことができる．Intel SGX を利用するアプリケーションはセキュリティ上の観点から二つの部分に分けて設計される．一つは機密性の高いデータやアルゴリズムを保護された環境内で実行する Enclave アプリケーション（Trusted 部分），もう一つは Enclave の生成・管理および Trusted 部分では実行できないシステムコールの処理を担う Host アプリケーション（Untrusted 部分）である．このような保護機構により，第三者によるデータの盗聴や改ざんのリスクを低減し，安全に計算処理を実行するための強力な基盤技術を提供している．

2.2 Local Attestation

Local Attestation（LA）は，同一のプラットフォーム上で動作する複数の Enclave 間で，互いの信頼性を検証し，安全な通信路を構築するためのプロトコルである．機能が複数の Enclave アプリに分割された Enclave アプリ間でデータを安全に交換する際などに，Enclave が同一の CPU により生成されていることを確認するために用いられる．

検証対象の Enclave は，認証情報として自身の MRENCLAVE を含む SGX Report を生成する．MRENCLAVE は Enclave をビルドした際のコードやデータなどから生成されるハッシュ値であり，SGX Report は Enclave が正しく SGX 上で実行されていることを証明するデータである．SGX Report の真正性と完全性は CPU 製造時に内部で生成されるハードウェア固有の秘密鍵 Attestation Key（AK）でメッセージ認証コード（Message Authentication Code, MAC）を計算することで保証される．検証元と検証先の Enclave が同じ CPU 上で動作していれば同じ AK にアクセスでき，SGX Report の MAC 検証により同じ CPU 上の Enclave によって生成されたものであることを確認でき

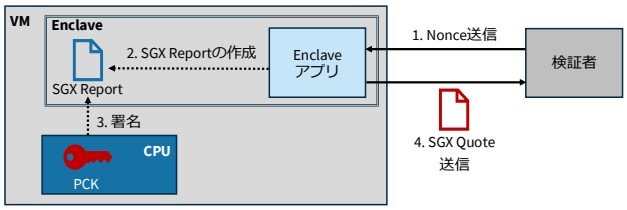

図 2 Intel DCAP 方式の Remote Attestation
Fig. 2 Remote Attestation with Intel DCAP

る．その後，SGX Report 内に含まれる MRENCLAVE の値を参照し，通信相手が期待通りの Enclave であるかを判断する．また，SGX Report 内にはアプリ利用者が任意に書き込み可能な固定長の領域（User Data）が存在し，SGX Report 作成時にデータを書き込むことができる．MAC により User Data も耐改ざん性を得るため，鍵交換に使用される鍵のハッシュ値など，SGX Report とともに送信されるデータの改ざん検知に利用され，これにより安全な通信路を構築できる．これら一連のプロセスにより，同一プラットフォーム上の Enclave 間で，ハードウェアに信頼の基点（Root of Trust，RoT）を置いた堅牢な認証が実現される．

2.3 Remote Attestation

Remote Attestation（RA）は，LA によって確立されたプラットフォーム内部の信頼性を，ネットワークを介してリモートに拡張するプロトコルである．これにより，リモート検証者は物理的に離れたプラットフォーム上で実行されている Enclave が正当かつ改ざんされていないことを検証できる．Intel SGX の RA がサポートする Data Center Attestation Primitives（DCAP）方式 [5] の基本的なフロー（DCAP-RA）を**図 2** に示す．DCAP-RA はローカルで生成された SGX Report をリモートで検証可能な SGX Quote に変換することで実現される．まず，リモートの検証者は RA のリプレイ攻撃を防ぐために，検証対象の Enclave に対して Nonce を送信する（Step 1）．検証対象の Enclave は，2.2 節で述べた LA のプロセスを経て，MRENCLAVE などの測定値と検証者から受信した Nonce を含む SGX Report を生成する（Step 2）．次に，Provisioning Certification Key（PCK）と呼ばれるプラットフォーム固有の非対称鍵で SGX Report を署名し，この署名と SGX Report を含む SGX Quote を検証者に送信する（Step 3–4）．PCK は Intel のルート認証局（CA）を頂点とする証明書チェーンによってその信頼性が担保されている．最後に，リモート検証者は受信した SGX Quote に対して，PCK 証明書を用いて SGX Quote の署名を検証する．これにより，リモート検証者は対象の Enclave アプリが正規の Intel SGX プラットフォームで生成され，SGX Quote 内の MRENCLAVE によって表される正規のアプリケーションであることを確認できる．

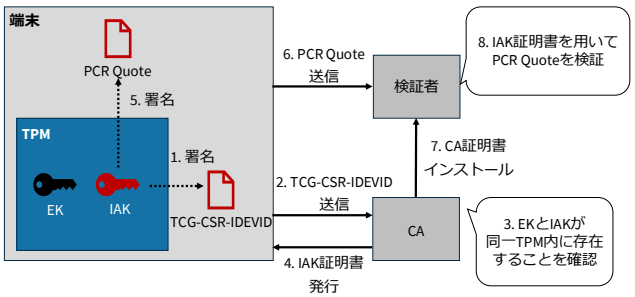

図 3 TPM 鍵を利用した端末認証
Fig. 3 Device Authentication using TPM keys

2.4 Trusted Platform Module

Trusted Platform Module（TPM）は，Trusted Computing Group（TCG）によって標準化されたハードウェアセキュリティチップである．TPM は鍵生成，暗号化・復号，署名生成・検証などの機能を有し，プラットフォームのハードウェアレベルの RoT として機能する．また，鍵管理や端末の整合性検証 [6] の機能も提供する．TPM の鍵は Name と呼ばれるハッシュ値で識別され，扱うデータを TPM 内に限定する Restricted，署名専用の Sign，暗号化・復号専用の Decrypt などの属性を設定できる．さらに，TPM は Platform Configuration Register（PCR）を備え，ファームウェアや OS カーネルのロードごとにハッシュ値を順次記録できる．PCR の最終的な値はシステムの状態を表し，ブートプロセスの改変の検出に利用できる．リモートの検証者は PCR 値を確認することで，端末が信頼できる状態で起動したことを検証できる．本検証では TPM 内の署名鍵で署名された PCR Quote が用いられる．PCR Quote には検証者が発行した Nonce，署名鍵の Name，現在の PCR 値などが含まれる．検証者は PCR Quote の署名を検証することで，PCR 値がその TPM により発行されたことを確認できる．

TPM は端末の識別や認証にも利用でき，TCG はそのための鍵管理について定義している [7]．認証鍵には，TPM 固有の暗号鍵 Endorsement Key（EK）と，端末認証用の署名鍵 Initial Attestation Key（IAK）がある．EK は TPM 製造時，IAK は端末製造時に生成され，それぞれの公開鍵証明書とともに TPM に保存される．IAK 証明書の証明書署名申請（Certificate Signing Request, CSR）は，TCG-CSR-IDEVID と呼ばれ，端末のモデル名，シリアル番号，EK 証明書，IAK 公開鍵などを含む．

TPM の鍵を利用した端末認証の概要を**図 3** に示す．まず，TCG-CSR-IDEVID を IAK で署名し，その完全性を保証する（Step 1）．次に，CA に TCG-CSR-IDEVID を送信する（Step 2）．CA は TCG-CSR-IDEVID を IAK 公

開鍵で検証し，そこに含まれる EK 証明書を確認することで，IAK が EK に紐づく TPM 内に存在することを確認する（Step 3）．その後，端末に IAK 証明書が発行される（Step 4）．IAK 証明書には，端末のモデル名やシリアル番号など，端末を特定する情報が記載されている．IAK 証明書の取得後，検証者による鍵の真正性確認と端末の整合性検証が行わる．まず，端末は検証者からランダムな Nonce を受け取り，Nonce と PCR 値，IAK Name を含む PCR Quote を作成して，IAK で署名する（Step 5）．次に，PCR Quote を検証者に送信し，検証者は CA 証明書で IAK 証明書を検証し，IAK 証明書内の IAK 公開鍵で PCR Quote の署名を検証する（Step 6–8）．以上のプロセスにより，検証者は TPM を介して TPM が組み込まれた端末を認証できる．

仮想 TPM（vTPM）は，物理 TPM（pTPM）の機能をソフトウェアで模倣したもので，仮想環境で利用される．vTPM の内部状態は物理マシン側が保有しており，暗号化などの保護対策により TPM データの窃盗や古い状態へのロールバック攻撃から保護する必要がある．vTPM により，各 VM は独立した PCR や鍵ストレージを持ち，これにより VM 単位での整合性検証や ID に基づいた認証を実現できる．ただし，vTPM はハードウェアの RoT を持たず，EK 証明書を得られないため，pTPM を RoT とした信頼関係の構築が必要である [8]．その方法の一つとして，pTPM が vTPM の EK 公開鍵ハッシュ値を含む PCR Quote を生成し，これを EK 証明書として利用する方法がある．

3. 関連研究

Chen らは，Intel SGX の RA が抱えるプライバシーとパフォーマンスの問題を指摘し，その解決策として OPERA [9] を提案した．RA のたびに，検証者は SGX Quote を Intel の検証サービスに送信する必要があるため，クラウドサービスのような多数のノードによる頻繁な RA が求められるシナリオでは，Intel 依存の RA はパフォーマンスとスケーラビリティのボトルネックとなる．OPERA はこの問題を解決するオープンな Attestation プラットフォームである．Intel の検証サービスを介さずに RA を行うことで，検証プロセスの高速化・分散化を行なっており，同時に開発者の情報が Intel に漏洩することも防ぐ．

Galanou らは，TEE における RA の脆弱性に対応する MATEE [10] を提案した．Intel SGX などの TEE は単一のハードウェア RoT に依存するため，サイドチャネル攻撃などにより AK が漏洩するリスクが存在する [11, 12]．MATEE は，Intel SGX ベースの RA に加えて TPM を第二の RoT として導入し，TPM の鍵を利用して TEE の Attestation を強化する仕組みを提供する．これにより，Intel SGX の AK が漏洩した場合でも，TPM に基づく Attestation により TEE の正当性が保証される．しかし，検証者がリモート環境の被検証者アプリに導入された MATEE の実装を信頼することが前提であり，被検証者が Enclave アプリを実装するような我々が想定するモデルには適用できない．

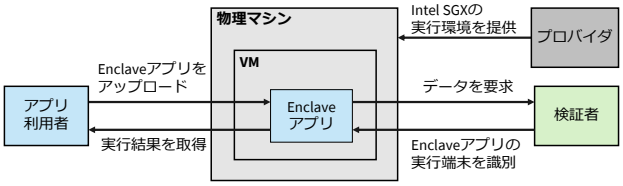

図 4 アプリ利用者・検証者・プロバイダの三者モデル
Fig. 4 Three-party model consisting of App User, Verifier, and Provider

4. 問題設定とアプローチ

4.1 問題設定

本研究では，アプリ利用者，検証者，プロバイダの三者モデル（**図 4**）を想定する．プロバイダはアプリ利用者と検証者から信頼された主体で，Intel SGX が有効な物理マシン上で Enclave アプリが実行可能な VM を構築し，これをアプリ利用者に提供する．アプリ利用者は VM を直接操作する権限は持たず，VM にアップロードしたビルド済みの Enclave アプリのみを実行できる．本 Enclave アプリは外部供給モデルのアプリケーションであり，検証者から受け取ったデータを Enclave アプリで処理し，その結果を出力する．なお，検証者は事前に Enclave アプリの MRENCLAVE を知っており，Enclave アプリに対する RA が成功した場合のみデータをに提供するものとする．

ここで，第三者が Enclave アプリの複製を取得して，その Enclave アプリを実行して検証者にデータを要求する状況を考える．このような状況は，例えば，アプリ利用者が悪意を持って機密データの利用手段を他者に渡す可能性や不正に Enclave アプリが摂取される可能性が考えられる．このとき，従来の RA のみでは，Enclave アプリはアプリ利用者の VM の外で実行されているが，Enclave アプリの真正性は確認できるためデータの提供を安全と検証者は判断する．

4.2 アプローチ

提案アプローチの概要を**図 5** に示す．本研究では，TPM と Intel SGX を用いて，どの Enclave アプリがどの物理マシンのどの VM で実行されているかを認証する多層認証手法を提案する．Intel SGX ベースの RA によるアプリ利用者の Enclave アプリ（User Enclave App）の完全性保証に加え，User Enclave App が実行されている VM を vTPM で，VM が動作している物理マシンを pTPM で多層的に認証する．外部供給モデルの Enclave アプリでは，User

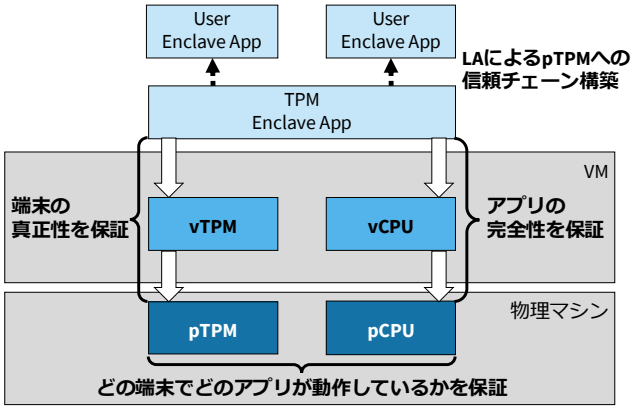

図 5 提案アプローチの概要
Fig. 5 Approach Overview

Enclave App をアプリ利用者が実装するため，ネットワークを介してリモートの vTPM を利用していない保証がない．そこで，Use Enclave App をローカルの vTPM に紐づける TPM Enclave App を導入する．TPM Enclave App は提案する多層認証手法の信頼性を支えるコアモジュールで，LA により同一マシン上に存在することを確認した User Enclave App に対して，vTPM による実行端末の証明を与える．TPM Enclave App の MRENCLAVE は統一された実装により固有の値を有し，検証者はその MRENCLAVE を確認することで，正規の多層認証が実施されたと判断できる．VM 構築時に vTPM を pTPM へ紐づけておくことで，User Enclave App は TPM Enclave App を介して pTPM までの信頼チェーンを得る．

5. 提案プロトコル

5.1 全体概要

提案プロトコルの構成を**図 6** に示し，各エンティティの役割を以下に説明する．

プロバイダ Intel SGX 対応の VM を各テナントに提供する主体である．各 VM には，提案する多層認証プロトコルの中核となる TPM Enclave App が事前にインストールされている．プロバイダは各 VM に割り当てられた vTPM の安全性に対して責任を持ち，pTPM と vTPM の信頼関係を維持する．これにより，物理マシン・VM・Enclave アプリの組み合わせが承認された構成であることを保証する．

アプリ利用者 プロバイダが提供する VM にビルド済みの Enclave アプリをアップロードし，機密データの処理を行う主体である．アプリ利用者の権限はアプリケーションレベルに制限されており，VM 自体や物理マシン，および vTPM を直接操作する権限は持たない．これにより，システムの信頼性と他テナントとの分離が保たれる．

検証者 後述する TPM Enclave App を介した User Enclave App から pTPM までの信頼チェーンにより，物理マシン・VM・Enclave アプリの構成をリモートから確認する主体である．承認されていない構成での User Enclave App の実行や実行環境の不正な変更を検知することで，処理中のデータ保護に加えて，データが処理される場所を保証する．

TPM Enclave App User Enclave App による vTPM へのアクセスを仲介する Enclave アプリである．vTPM を操作し，提案する多層認証に必要な vIAK 証明書や PCR Quote を取得するほか，物理マシンと通信し，pTPM から取得した pIAK 証明書と PCR Quote を合わせて User Enclave App に提供する．User Enclave App と同一端末での動作を保証する LA により，操作する vTPM を信頼でき，pTPM から User Enclave App までの信頼チェーンを構築できる．TPM Enclave App の実装は統一されており，その固有の MRENCLAVE は公開されているものとする．

User Enclave App アプリ利用者が実装する Enclave アプリであり，VM にアップロードされて実行される．多層認証プロトコルにおいて，TPM Enclave App と連携し，検証者に対してどの VM・どの物理マシン上で自身が実行されているかを証明する．

Privacy-CA 提案手法において，pTPM と vTPM に対してそれぞれ IAK 証明書を発行する認証局である．物理マシンには pIAK 証明書を，各 VM には vIAK 証明書を発行し，pTPM と vTPM の信頼関係を証明書レベルで保証する．検証者に CA 証明書を提供し，発行した IAK 証明書の検証を可能にする．これにより，検証者は受信した IAK 証明書の正当性を確認できる．

pTPM 物理マシン上の物理 TPM であり，多層認証に必要な TPM データを TPM Enclave App を介して検証者に提供する．また，pIAK 証明書の取得に必要なデータを Privacy-CA に提供する．さらに，pIAK を利用して vEK の公開鍵ハッシュ値を含めた PCR Quote を作成することで，vTPM の物理 RoT を保証する．

vTPM ソフトウェア実装の TPM であり，各 VM に搭載され，TPM Enclave App を介して多層認証に必要な TPM データを検証者に提供する．また，vIAK 証明書の取得に必要なデータを Privacy-CA に提供する．

5.2 提案プロトコルを構成する処理フェーズ

本節では，提案プロトコルを構成する三つのフェーズについて説明する．

5.2.1 Provider Setup フェーズ

本フェーズは，VM Setup フェーズの前に一度だけ実行され，pIAK の作成と pIAK 証明書のインストールを行う．Provider Setup フェーズの流れを**図 7** に示す．

まず，pTPM は pIAK を作成し，pIAK 公開鍵および

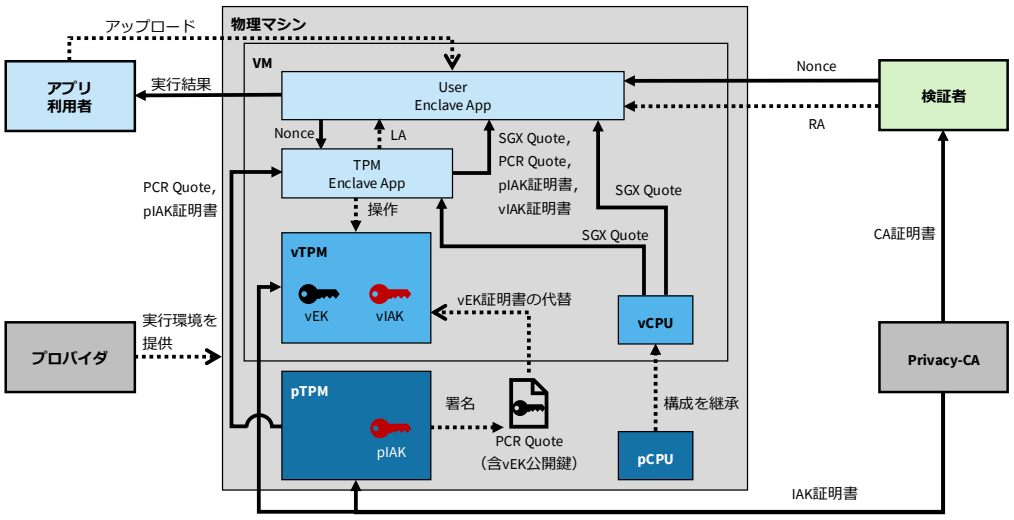

図 6　提案プロトコルの構成
Fig. 6　Overall structure of the proposal protocol

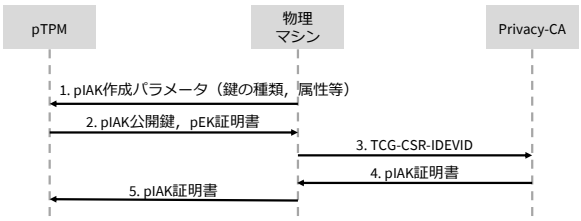

図 7　Provider Setup フェーズの処理行程
Fig. 7　Provider Setup Phase process flow

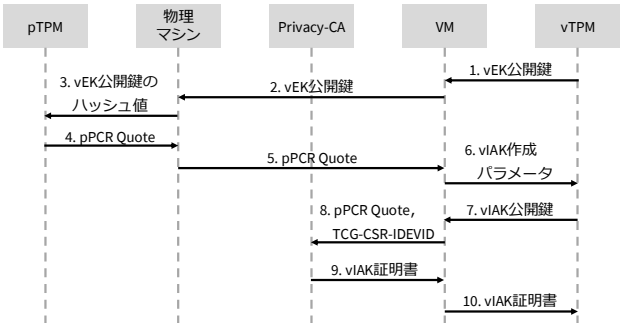

図 8　VM Setup フェーズの処理行程
Fig. 8　VM Setup Phase process flow

pEK 証明書を取り出す（Step 1–2）．次に，取得した TPM データをもとに TCG-CSR-IDEVID を作成し，pIAK を用いて署名した後，署名データとともに Privacy-CA に送信する（Step 3）．Privacy-CA は 2.4 節で述べた IAK 証明書発行プロセスを経て，TCG-CSR-IDEVID を検証し，物理マシンに対して pIAK 証明書を発行する（Step 4）．最後に，pIAK 証明書を pTPM 内に保存する（Step 5）．

5.2.2　VM Setup フェーズ

本フェーズは，Provider Setup フェーズの後に，VM を作成するたびに実行され，各 vTPM に対して vIAK の作成と vIAK 証明書のインストールを行う．VM Setup フェーズの流れを**図 8** に示す．

まず，VM は vTPM から vEK 公開鍵を取り出し，物理マシンに送信する（Step 1–2）．次に，物理マシンは pTPM の pIAK を利用し，vEK 公開鍵のハッシュ値を含めた pPCR Quote を作成し，VM に送信する（Step 3–5）．続いて，VM は vTPM で vIAK を作成し，vIAK 公開鍵を取り出す（Step 6–7）．pPCR Quote を vEK 証明書の代替として TCG-CSR-IDEVID を作成した後，vIAK を用いて署名し，署名データとともに Privacy-CA に送信する（Step 8）．Privacy-CA は 2.4 節で述べた IAK 証明書発行プロセスを経て，TCG-CSR-IDEVID を検証し，VM に対して vIAK 証明書を発行する（Step 9）．vIAK 証明書には，端末モデル名とシリアル番号の代わりにプロバイダと VM の識別情報が入っており，さらに pIAK Name が記載されている．最後に，vIAK 証明書を vTPM 内に保存する（Step 10）．

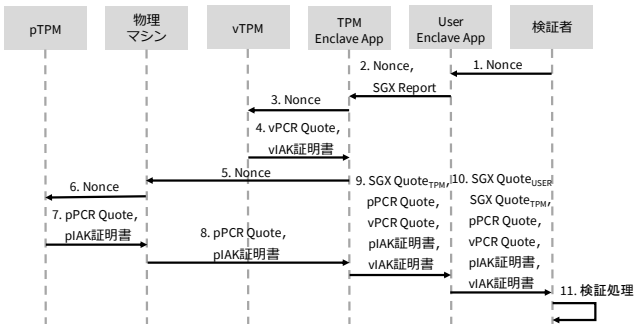

図 9　Attestation フェーズの処理行程
Fig. 9　Attestation Phase process flow

5.2.3　Attestation フェーズ

本フェーズでは，検証者により Enclave App および実行端末の認証が行われる．Attestation フェーズの流れを**図 9** に示す．

検証者は User Enclave App に Nonce を送信し，認証を

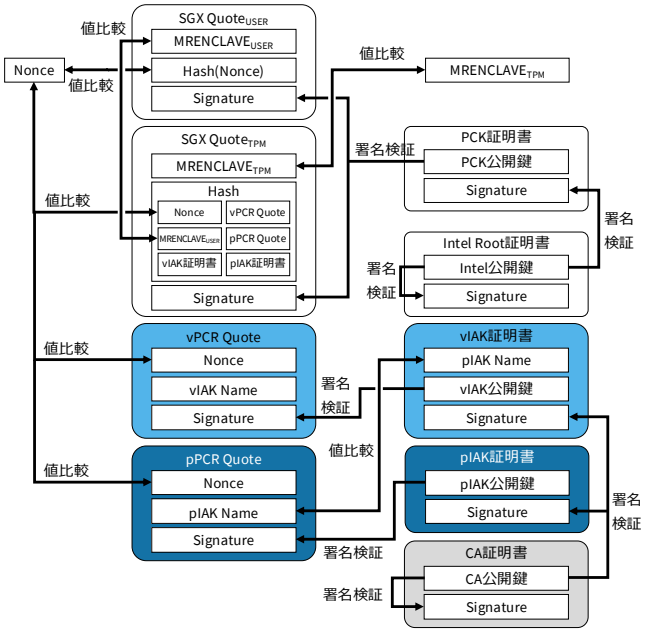

図 10 検証者によるデータ検証
Fig. 10 Data verification by Verifier

開始する（Step 1）．User Enclave App は Nonce と生成した SGX Report を TPM Enclave App に送信する（Step 2）．TPM Enclave App は受信後 SGX Report を検証することで，User Enclave App に対して LA を行う．LA の成功後，TPM Enclave App は vTPM を操作し，Nonce が含まれる vPCR Quote を作成し，vIAK で署名した後，vTPM 内から vIAK 証明書を取り出す（Step 3–4）．また，TPM Enclave App は Nonce を物理マシンに送信し，pTPM が作成した Nonce の含まれる pPCR Quote と pIAK 証明書を受信する（Step 5–8）．次に，TPM Enclave App は SGX Quote を作成し，二種類の PCR Quote および pIAK 証明書，vIAK 証明書とともに User Enclave App に送信する（Step 9）．User Enclave App は受信データに加え，自身の SGX Quote を作成し，検証者に送信する（Step 10）．最後に，検証者は受信したデータに対して検証処理を行う（Step 11）．

検証者が検証するデータを**図 10** に示す．検証者は受信した SGX Quote を検証し，MRENCLAVE が事前に知らされたものと一致するかを確認することで User Enclave App と TPM Enclave App に対して RA を行う．RA に成功後，CA 証明書を用いて pIAK 証明書と vIAK 証明書を検証し，それぞれの IAK 証明書から pIAK 公開鍵と vIAK 公開鍵を抽出した後，対応する PCR Quote の署名を検証する．さらに，vIAK 証明書内に記載されている pIAK Name が pIAK 公開鍵のものと一致するかを確認する．検証者が PCR Quote の検証を終え，端末認証に成功した時点で Attestation フェーズは終了する．これにより，検証者は User Enclave App の実行端末である VM と物理マシンを一意に特定できる．

6. 安全性評価

本章では，提案する多層認証の評価として，実行端末を偽る攻撃に対する安全性を議論する．なお，本評価では，TPM および Intel SGX の実装を信頼できるものとし，その実装に脆弱性はないもとする．

6.1 実行端末の変更検知

検証者は 5.2.3 節で述べた Attestation フェーズで User Enclave App 実行端末の認証を行ったうえで，User Enclave App の MRENCLAVE と実行端末の PCR Quote 内に記載されている pIAK Name と vIAK Name の情報を保存しておくことで，User Enclave App の実行端末の変更を検知できる．User Enclave App の MRENCLAVE は，アプリの変更や異なる環境での再ビルドがない限り変化しない．また，pIAK を再生成せず，物理マシンが変更されない限り，pIAK 証明書の内容や pIAK Name も変化しない．vIAK Name も同様である．そのため，アプリ利用者が同じ User Enclave App を同じ端末で実行し続ける限り，検証者が保存している MRENCLAVE と IAK Name の対応付けデータに変化は生じない．したがって検証者は，Attestation フェーズで取得した MRENCLAVE・vIAK Name・pIAK Name の組み合わせが保存されているものと一致するかどうかで実行端末が変わったことを検知できる．

6.2 実行端末の偽装検知

提案手法では，VM レベルでの実行端末の偽装と，物理マシンレベルでの実行端末の偽装の二種類の攻撃に大別できる．

6.2.1 VM レベルでの実行端末の偽装

攻撃者は User Enclave App が他の VM の vTPM に紐づくことを検証者に証明できれば，実行端末を偽ることができる．具体的には，User Enclave App が外部の VM の TPM Enclave App に対して Attestation フェーズの Step 2 を実行することが考えられる．しかし，TPM Enclave App は LA により，User Enclave App の実行環境を検証するため，外部の User Enclave App からの要求を拒否できる．

これを回避するために，攻撃者が vIAK 証明書を偽装対象の vIAK 証明書に差し替えることが考えられる．物理マシンレベルでは偽装していないため，差し替えた vIAK 証明書と pIAK 証明書の対応関係は維持されている．しかし，TPM Enclave App は自身が動作する VM の vIAK 証明書から計算されたハッシュ値を含む SGX Quote を生成するので，vIAK 証明書の差し替えは SGX Quote の検証エラーとして Attestaion フェーズの Step 11 で検知できる．

6.2.2 物理マシンレベルでの実行端末の偽装

攻撃者は User Enclave App が他の物理マシンの pTPM に紐づくことを検証者に証明できれば，実行端末を偽ることができる．TPM Enclave App はローカルの pIAK 証明書を取得するため，プロトコルの途中で User Enclave App が偽装対象の pIAK 証明書に差し替えることが考えられる．具体的には，Attestation フェーズの Step 9 において，偽装対象の物理マシンの pIAK 証明書に差し替える．しかし，vIAK 証明書には当該 VM が動作する物理マシンの pIAK の Name が記載されているため，Attestaion フェーズの Step 11 における pIAK の Name 検証で pIAK 証明書の差し替えを検知できる．

7. 議論

アプリ利用者が実装する Enclave アプリの開発・ビルド環境と実行環境が異なるため，アプリケーションのアップロードのみでは正確な実行結果を得られない可能性がある．この場合，VM の操作権限を利用者に与えることによって，開発・ビルド環境と実行環境が一致し，アプリ利用者はより柔軟に Enclave アプリの開発をすることができる．しかし，Enclave アプリの Untrusted 部分の変更は MRENCLAVE に反映されないため，ライブラリの悪用と合わせることで，操作する TPM の対象を外部にリダイレクトさせるリスクが存在する．本提案プロトコルはこのような攻撃を想定していないため，Untrusted 部分の変更や別 TPM データの取得を検知できる仕組みが必要である．

実際のクラウドサービスでは，プロバイダが物理マシンのメンテナンス等で VM を他のマシンにマイグレーションさせることが起こりうる．このとき，vTPM も VM とともに別の物理マシンにマイグレーションされるため，移行元物理マシンの pTPM と VM の vTPM 間で構築した信頼関係が無効化され，再度移行先マシンで vTPM の RoT を構築する必要がある．また，移行前後の pIAK が異なり，移行前の pIAK Name が登録されている検証者により誤検知が発生する可能性があるため，マイグレーションへの対応が重要となる．

8. おわりに

Intel SGX における RA は，Enclave アプリが特定の端末で動作することを保証できず，外部供給モデルに対応できない．そのため，Enclave アプリの実行端末を特定できないことに着目し，TPM を活用した多層認証プロトコルを提案した．本提案プロトコルでは，vTPM 内で生成される IAK を利用し，Enclave アプリと VM への紐付けを可能にした．また，物理マシンと VM，Enclave アプリの組み合わせをリモート検証者が検証できるようにし，Enclave アプリの実行端末の変更を検知可能にした．安全性評価では，RA における端末識別の問題点を解決し，TPM データの偽造に対する耐性を確認した．今後の課題として，外部 TPM へのリダイレクト攻撃の対策と，VM マイグレーションへの対応などが挙げられる．

謝辞 本研究の一部は，JSPS 科研費 JP25K21199，JP23K03847 の助成を受けたものです．

参考文献

[1] Sabt, M., Achemlal, M. and Bouabdallah, A.: Trusted Execution Environment: What It is, and What It is Not, *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1, pp. 57–64 (2015).

[2] Costan, V. and Devadas, S.: Intel SGX Explained, *IACR Cryptol. ePrint Arch.*, Vol. 2016, p. 86 (2016).

[3] Anati, I., Gueron, S., Johnson, S. and Scarlata, V.: Innovative Technology for CPU Based Attestation and Sealing (2013).

[4] Tokuda, S., Kakei, S., Shiraishi, Y. and Saito, S.: Decentralized Data Usage Control with Confidential Data Processing on Trusted Execution Environment and Distributed Ledger Technology, *International Conference on Network and System Security* (2024).

[5] : ECDSA Attestation with Intel Software Guard Extensions Data Center Attestation Primitives (Intel(R) SGX DCAP). [Online; accessed 2025-08-12].

[6] : Remote Attestation — tpm2-software community (2019). [Online; accessed 2025-08-12].

[7] : TPM 2.0 Keys for Device Identity and Attestation — Trusted Computing Group. [Online; accessed 2025-08-12].

[8] Berger, S., Cáceres, R., Goldman, K. A., Perez, R., Sailer, R. and van Doorn, L.: vTPM: Virtualizing the Trusted Platform Module, *USENIX Security Symposium* (2006).

[9] Chen, G., Zhang, Y. and Lai, T.-H.: OPERA: Open Remote Attestation for Intel's Secure Enclaves, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019).

[10] Galanou, A., Gregor, F., Kapitza, R. and Fetzer, C.: MATEE: multimodal attestation for trusted execution environments, *Proceedings of the 23rd ACM/IFIP International Middleware Conference* (2022).

[11] Bulck, J. V., Minkin, M., Weisse, O., Genkin, D., Kasikci, B., Piessens, F., Silberstein, M., Wenisch, T. F., Yarom, Y. and Strackx, R.: Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution, *USENIX Security Symposium* (2018).

[12] van Schaik, S., Kwong, A., Genkin, D. and Yarom, Y.: SGAxe: How SGX Fails in Practice (2020).