一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CPSY2025-53,DC2025-125(2026-03)

ユニバーサルTEEにおけるIOデバイスのメモリ防御機構

浦浪　英俊[†]　齊木　昭大[†]　内山　一秀[††,†††,††††]　五島正裕[††,††††,†††]　木村啓二[††,†]

† 早稲田大学
†† セキュアコンピュータシステム研究開発センター, ROIS-DS
††† 総合研究大学院大学
†††† 国立情報学研究所
E-mail: †{reinsirk,saiki}@kasahara.cs.waseda.ac.jp, ††{uchiyama,goshima}@nii.ac.jp, †††keiji@waseda.jp

あらまし　GPU 等の外部デバイスを用いた機密データ処理の需要が増加する中，これらのデバイスを Trusted Execution Environment（TEE）の防御対象に含める技術の研究開発が進められている．その実現には，TEE における CPU 上のプログラム実行と同様に，信頼できないシステムソフトウェアに依存することなく機密性および完全性を保証する DMA 機構に加え，TEE のメモリ空間への MMIO の安全なマッピングが必要である．本稿では，筆者等が開発を進めるユニバーサル TEE におけるメモリ防御方式である FKT/RTT 方式を対象として，ソフトウェアへの依存及び PCI における TEE サポートの標準仕様である TDISP への変更を最小化した，IOMMU によるデバイスとの TEE メモリ共有手法および MMIO マッピング手法を提案する．
キーワード　TEE，PCIe TDISP，TEE-IO，IOMMU

Memory Defence Mechanism for I/O Devices in Universal TEE

Hidetoshi URANAMI[†], Akihiro SAIKI[†], Kazuhide UCHIYAMA[††,†††,††††],
Masahiro GOSHIMA[††,††††,†††], and Keiji KIMURA[††,†]

† Waseda University
†† Center for R&D on Secure Computer Systems, ROIS-DS
††† The Graduate University for Advanced Studies
†††† National Institute of Informatics
E-mail: †{reinsirk,saiki}@kasahara.cs.waseda.ac.jp, ††{uchiyama,goshima}@nii.ac.jp, †††keiji@waseda.jp

Abstract　As demand for confidential data processing on external devices, such as GPUs, continues to grow, a variety of studies have examined extending Trusted Execution Environments (TEEs) to support these devices. Achieving this extension requires not only a secure DMA mechanism that guarantees confidentiality and integrity without relying on untrusted system software, as well as CPU execution on TEE, but also secure MMIO mapping for device control. This paper proposes an IOMMU-based TEE memory-sharing method for devices and a secure MMIO mapping scheme, while minimizing dependence on system software. They are designed for the FKT/RTT memory protection scheme within the Universal TEE architecture.
Key words　TEE, PCIe TDISP, TEE-IO, IOMMU

1. はじめに

コンピューターの利用範囲の広がりと共に OS やハイパーバイザー等のシステムソフトウェアの機能が多様化し，その規模は巨大化・複雑化している．その結果，これらのソフトウェアから脆弱性を完全に排除することは困難であり，システム全体に全面的な信頼を置くことは難しい．そのため，個人情報や暗号鍵といった機密性の高い情報を扱う際には，システムソフトウェアからも隔離された実行環境が必要である．このような背景から，ハードウェアによって強制的に分離されたプログラム実行環境である TEE（Trusted Execution Environment）が導入され，利用されている．中でもクラウド環境において，ユーザデータをベンダーや管理者から保護する VM ベースの TEE として，Intel TDX [1]，AMD SEV-SNP [2]，Arm CCA [3]，RISC-V CoVE [4] など，各 CPU ベンダーによる仕様策定および製品化が進んでいる．

一方，クラウドコンピューティングが提供するサービスも多様化し，それに伴い GPU や FPGA，高機能 NIC といった外部デバイスのクラウド環境での利用範囲も拡大している．しかしながら，このような状況はクラウドサーバー上のこれら外部デバイスが処理するデータも，物理的なバスの盗聴やサイドチャネル攻撃といった脅威にさらされ得ることを意味する．実際に，GPU におけるメモリ残留データを他プロセスから窃取可能な脆弱性 [5] が報告されるなど，従来のデバイス運用モデルでは想定されていなかったセキュリティリスクが報告されている．この課題に対するアプローチとして，TEE が提供する防御領域をこれら外部デバイスにまで拡張する手法が提案されている．

こうした背景の下，PCI-SIG は TEE とデバイス間の安全な接続を実現するための標準規格である PCIe TDISP（TEE Device Interface Security Protocol）[6] を策定した．TDISP は，PCIe バス上のリンク暗号化機能である IDE（Integrity and Data Encryption）と連携することで，ハードウェアレベルで低遅延な通信路の機密性・完全性を提供し，TEE の防御するメモリ領域への DMA アクセスを実現するためのプロトコルを提供する．また，本規格は PCIe に加え CXL（Compute Express Link）プロトコルにおいても採用されており，その適用範囲は CXL 接続されたデバイスにまで拡大している．

ただし，TDISP は TEE とデバイス間の信頼関係構築および通信路保護の制御主体として，TSM（TEE Security Manager）の存在を前提としている．TDISP に基づく TEE-IO アーキテクチャでは，セキュアモニタやセキュアプロセッサといった特権ソフトウェアが TSM として動作し，鍵管理や IOMMU の制御，MMIO マッピングの検証といった処理を一元的に担う．

一方で，筆者等が開発を進めるユニバーサル TEE [7], [8] では，メモリ防御の要件としてこのような特権ソフトウェアの排除を目標の一つとして掲げており，ユニバーサル TEE で TDISP 準拠デバイスを利用するためには，TDISP 運用の枠組みを TSM に依存しないよう再構築する必要がある．

そこで本稿では，ユニバーサル TEE において TSM を介することなく TDISP 準拠デバイスと安全に連携するための，IOMMU ベースのメモリ共有および MMIO マッピング手法を提案する．本提案は，ソフトウェアへの信頼を最小化しつつ，CPU 実行時と同等のメモリ防御保証をデバイスアクセスにも拡張する点に特徴がある．

本稿の構成は以下のとおりである．2. 節では，TDISP 以外の手法によりデバイスに対して TEE の防御を適用した既存研究と，TDISP 準拠の TEE およびデバイスの開発状況について述べる．3. 節にてユニバーサル TEE が採用するメモリ防御方式である FKT/RTT 方式について解説し，4. 節では PCI-SIG が定めた TDISP についてその仕様を概説する．それらを踏まえ，5. 節にてユニバーサル TEE におけるデバイスのメモリ防御機構について述べ，6. 節でまとめを述べる．

2. 関 連 研 究

本節では，TEE の防御対象を外部デバイスに広げた既存の研究と，現在開発が進められている TDISP 準拠の TEE およびデバイスの動向について述べる．

TDISP に準拠しない手法については，TDISP 策定以前から GPU を対象に研究・開発が行われており，例として StrongBox [9]，Graviton [10]，GhosTEE [11] などが挙げられる．中でも NVIDIA Confidential Computing（NVIDIA CC）[12] は，AMD SEV-SNP および Intel TDX 環境下で利用可能な対応 GPU が既に市販されており，実装例の一つである．しかし，これらの実装は TEE アーキテクチャとの統合が限定的であり，通信路の暗号化においてハードウェア支援が得られないことによる性能制約や，TEE の防御領域に対して直接 DMA 転送を行うことができないといった機能上の制約が存在する．

一方，TDISP 対応 TEE の整備が，VM ベースの TEE において各 CPU ベンダーによって進められている．具体的には，Intel は TDX Connect [13]，AMD は SEV-TIO [14]，Arm は Arm CCA v1.1 [15]，RISC-V は RISC-V CoVE-IO [16] として，それぞれ TDISP プロトコルを自身のメモリ防御機構と統合したアーキテクチャ定義を策定している．加えて，デバイス側においても NVIDIA が Blackwell アーキテクチャにて TDISP 準拠の製品をリリースするなど [17]，CPU とデバイスの双方において，標準規格に基づいた TEE-IO 環境の構築が進められている．

3. ユニバーサル TEE におけるメモリ防御方式

本節では，ユニバーサル TEE で提案されているメモリ防御方式である，FKT/RTT 方式 [18]〜[20] について述べる．前提として，ユニバーサル TEE ではメモリ防御の要件として以下を定めている．

（1） メモリの完全性保証
（2） ページ単位の防御
（3） 防御領域の同時・多重のネスト
（4） 防御されたページ共有
（5） セキュアモニタ・セキュアプロセッサの排除

このうち，完全性の保証はメモリ暗号化に認証付き暗号を採用することで達成される [21]．一方で，アクセス制御に関わる残りの要件は FKT/RTT 方式によって実現される．この FKT/RTT 方式によるメモリ防御の構造を図 1 に示す．

まず，本方式における主要な構成要素について定義する．図 1 中の MgE Space は，防御対象の論理アドレス空間を表す．これは，ユニバーサル TEE が防御領域の同時・多重ネストをサポートするため，防御対象の空間を一般化して扱う必要があるためである．例えば，VM ベースの TEE であれば VM 内のプロセスごとの仮想アドレス空間が MgE Space に相当し，Enclave ベースの TEE であれば Enclave 内プロセスの仮想アドレス空間に相当する．

図 1 に示すように，FKT/RTT 方式では MgE Space 側に FKT（Forward Key Table），HPA Space 側に RTT（Reverse Tag Table）と呼ばれるデータ構造を追加する．各テーブルはページ単位のエントリを持ち，FKT エントリには共有鍵が，RTT エントリには共有タグがそれぞれ暗号化された状態で格納される．ここで共有タグは，共有鍵とアドレス（ナンスとして利用）から生成される値である．防御の確立時には，RTT 上に共有鍵から導出

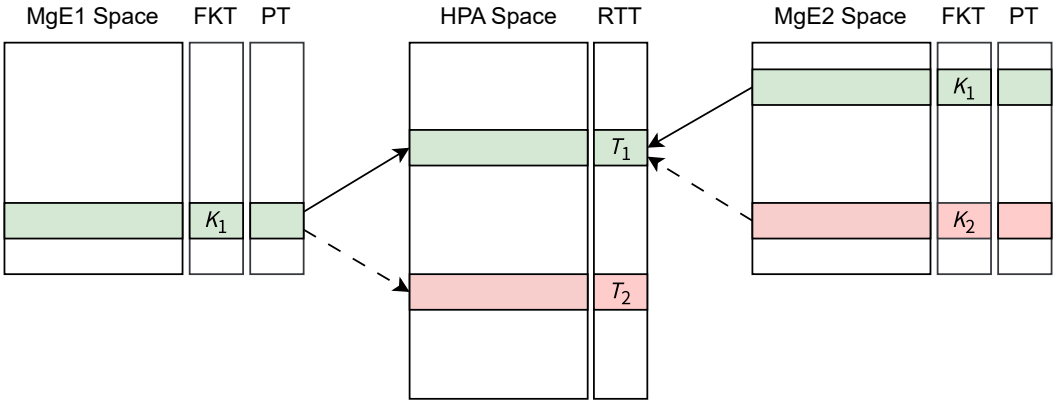

図 1　FKT/RTT 方式のメモリ防御の様子

された共有タグが配置され，メモリアクセス時にはページテーブルウォーカーがページテーブルによるアドレス変換を行った後に FKT と RTT の双方からエントリを読み出し，共有鍵とアドレスからタグを再計算して RTT 上のタグと照合を行う．これにより，防御確立時と現在のマッピングに相違がないこと，およびアクセスを試みた対象がその領域に対して正当なアクセス権を有していることをハードウェアレベルで検証する．

図 1 の例では共有鍵 K_1, K_2 から共有タグ T_1, T_2 がそれぞれ生成され，RTT に配置されている．この時，図中の実線で示したマッピングについては FKT と RTT に格納されている共有鍵，共有タグの照合に成功するためアクセスが許可される．一方，点線にて示したマッピングについては鍵から導出されるタグと RTT 上のタグが一致せず照合が失敗するため，不正なマッピングとしてアクセスが拒否される．また，図中の MgE1 と MgE2 の双方が同一の共有鍵 K_1 を保持し，同じ HPA に対するマッピングが正当であるように，FKT/RTT 方式では異なる MgE Space 間であっても適切な鍵共有を行うことで安全なメモリ共有を実現している．

さらに，この FKT/RTT 方式では，共有鍵を MgE を一意に識別する ID をナンスとしてマシン秘密鍵で暗号化し，共有タグをアドレスをナンスとしてマシン秘密鍵で暗号化することにより，FKT・RTT の管理を信頼できない OS 等に移譲可能であり、また複数の MgE 間で安全にページを共有可能な仕様となっている．

4.　PCIe TDISP

本節では，PCI-SIG によって策定された標準規格である PCIe TDISP について概説する．図 2 に，TDISP が想定するシステムアーキテクチャの概要を示す．図が表しているように，現在の TDISP 規格は主に VM ベースの TEE を保護対象として設計されている．

図 2 に示す TSM（TEE Security Manager）は，ホスト側で TEE のセキュリティ維持の権限を持つ特権ソフトウェアであり，セキュアモニタやセキュアプロセッサ等が該当する．デバイス側には，TSM に対応する DSM（Device Security Manager）が存在し，デバイス内部のセキュリティ維持の権限を持つ論理コンポーネントとして機能する．IOMMU（Input-Output Memory Management Unit）はホストとデバイスの間に位置し，DMA リマッピングおよび割り込みリマッピング機能によってデバイスからのメモリアクセスを制御するハードウェアコンポーネントである．また，TDI（TEE Device Interface）は各 TEE へのデバイス機能の割当単位であり，デバイスパススルーにより TEE に割り当てられる．TDI の実体は一般に，SR-IOV によって作成される VF（Virtual Function），あるいは SIOV における SDI（Scalable Device Interface）として実装される．SPDM（Security Protocol and Data Model）[22] および IDE（Integrity and Data Encryption）[6] は，いずれも通信の機密性・完全性を保証するための規格である．

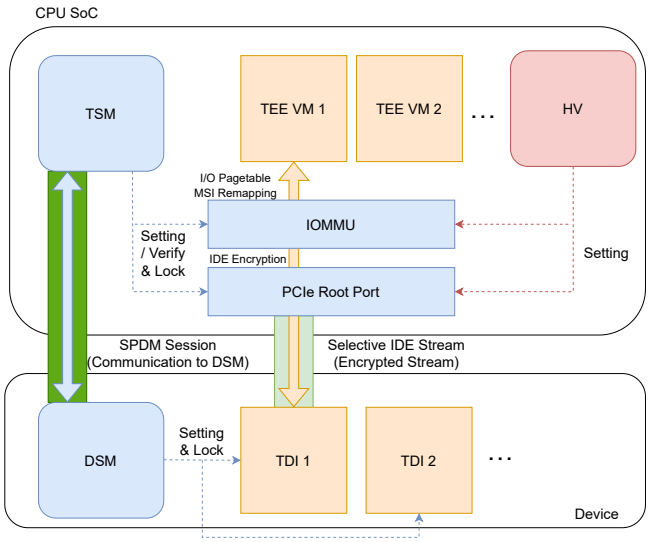

図 2　TDISP のリファレンスアーキテクチャ [6], [13]

TSM は CPU 側で IDE 通信に用いるセッション鍵の生成，および IDE の暗号・復号を担う PCIe Root Port の設定を行う．さらに，IOMMU に対してページテーブルやメモリ防御機構の設定を行う．TEE のアーキテクチャによっては，信頼できないハイパーバイザーがこれらの設定を実施し得るため，その場合にはハイパーバイザーによる設定内容の検証を TSM が担う．IOMMU は TSM の管理下にて，特定の TDI が許可された TEE

のメモリ領域以外へアクセスすることを防ぐアクセス制御を行う．具体的には，トランザクションの RID（Requester ID）に紐付く，TSM が設定または検証したページテーブルによるアドレス変換を行うことで，許可されないアクセスを遮断する．DSM は TSM からの指示に基づいて TDI を管理し，デバイスの Measurement Report を TSM へ提供する役割を担う．データ転送時には，PCIe Root Port がデバイスから受信した IDE 暗号化パケットを復号し，整合性検証を行うことで，物理インターコネクト上での盗聴・改竄を検知・遮断する．正当性が確認されたトランザクションのみが後段の IOMMU やメモリコントローラへ転送されるため，中間経路の攻撃者による不正パケット注入やデータすり替えはハードウェアレベルで防止される．

TDISP は TSM の存在を前提とした規格となっており，メモリ共有の確立プロセスも TSM による集中管理に依存している．しかしながら，我々が提案するユニバーサル TEE では，メモリ防御の要件としてセキュアモニタ・セキュアプロセッサといった特権ソフトウェアの排除を掲げており，このような TSM 依存の仕組みをそのまま適用することは困難である．そこで次節では，TSM に頼ることなくデバイスとの安全なメモリ共有および MMIO を実現するための手法について述べる．

5. ユニバーサル TEE におけるデバイスとのメモリ共有方法

本節では，TSM のような特権的ソフトウェアへの依存及び TDISP 仕様への変更を最小化しつつ，ユニバーサル TEE がデバイスと安全にメモリ共有する方式を提案する．

5.1 脅威モデルとその対処

本節では，本提案における脅威モデルを定義し，それに対する対処方針を述べる．本提案は TDISP に基づくため，TDISP が想定する脅威に対処可能であることを目標とする．

TDISP では，想定される脅威に対し，以下の機構のいずれか，またはそれらの組合せによって対処する．

- PCIe IDE
- SPDM Secure Session
- Device Attestation
- TSM
- DSM

本提案は TSM への依存を最小化することを目的とするため，従来 TSM が担っていた機能（および脅威対処）を他の機構へ再配置する必要がある．以下，本節ではまず TSM が従来対処していた脅威とその方法を整理し，その上で，TSM を介さずに運用することで新たに生じる脅威を明確化する．

a） SPDM Secure Session と PCIe IDE

TSM は，SPDM Secure Session および PCIe IDE の運用を通じて，デバイスとの安全な通信路の確立とデータ防御を実現する．SPDM Secure Session については，TSM の代替として TEE 自身が DSM とセキュアセッションを確立することで，同等の機密性および完全性を確保できる．

一方，PCIe IDE に関しては，TSM が IDE 設定の正当性確認および鍵管理を担う．TSM を介さずにハイパーバイザー（HV）が IDE 設定を行う構成を採用すると仮定すると，以下の追加脅威が生じる．

- **IDE 設定の不正構成による保護の無効化：**IDE ストリーム選択や暗号化設定を不正に構成することで，TEE 由来の MMIO/DMA が IDE により保護されない状態を誘発する攻撃.
- **鍵設定権限の濫用：**IDE 鍵の設定権限を有する HV が，鍵を悪用して通信内容の漏えい，または改ざんを行う攻撃.

これらは本提案において対処すべき脅威である．

以下では，TSM が対処していた脅威を DMA および MMIO に分類し，その内容と本提案における対処方針を述べる．

b） DMA に関する脅威

TSM は DMA に関連して，主として以下の脅威に対処している．

- **悪意あるデバイスによる TEE メモリへの DMA Read/Write：**悪意あるデバイスが TEE のプライベートメモリ領域へ DMA アクセスを行い，機密情報の漏えい，ならびにコード／データの改ざんを引き起こす攻撃.
- **TEE 防御領域外への TDI のアクセス：**TEE によって防御されない領域へ TDI をアクセスさせることにより，機密情報の漏えいや意図しない動作を誘発する攻撃.

PCIe TDISP では，TSM が CPU によるアクセス制御と同等の隔離を DMA にも強制することで，TEE メモリの機密性・完全性を担保していた．本提案では，IOMMU 側に FKT/RTT 方式によるアクセス制御を導入し，TSM に依存せずに同等の隔離を実現する．

c） MMIO に関する脅威

TSM は，ホストが CPU を介して MMIO アクセスを実行可能である状況を想定し，デバイス状態の改ざんや隔離設定の破壊につながる以下の脅威に対処する．

- **所有者以外の MMIO アクセス：**攻撃者が TDI の MMIO 領域に対して読み書きを行い，機密情報の漏えい，または不正なデバイス動作を誘発する攻撃.
- **物理メモリ上へのマッピング：**TEE の仮想アドレスから物理アドレスへのマッピングを，MMIO ではなく物理メモリに配置し，HV が仲介者となって観測・改変可能とする攻撃.
- **MMIO のシャッフル：**MMIO のページ順序を入れ替えることで，デバイスの予期しない動作を誘発する，あるいは別デバイスへの誤マッピングにより機密情報を漏えいさせる攻撃.
- **MMIO デコード範囲改変による経路のすり替え：**PCIe Root Complex ／ Root Port における MMIO ルーティング設定を変更することにより，本来 TEE のみに許可されるべき MMIO 領域をホストから到達可能にする，または暗号化されない経路で別デバイスへ誤配送する攻撃.

TSM は MMIO マッピングの構成，あるいはマッピングの検証とロックにより，これらの脅威に対処する．本提案では，正当性検証の手段としてチャレンジレスポンス方式を用いる．ただし，TOCTTOU（Time-of-Check to Time-of-Use）攻撃を防ぐためには，検証結果と設定状態の結び付けを保証する追加の対策が必要となる．

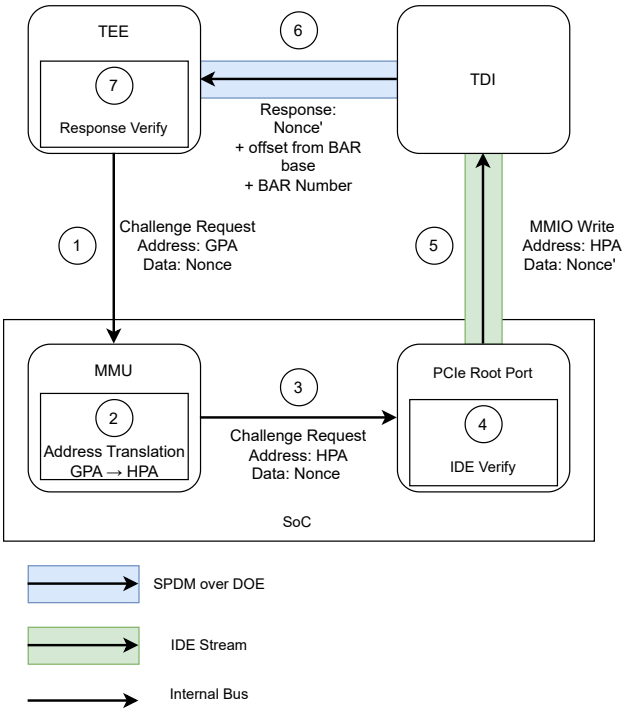

図 3　MMIO と IDE 整合性検証のためのチャレンジレスポンス

5.2　提 案 方 式

本方式では，以下の手順により TEE が I/O デバイスを使用可能とする．

（1） **リソースの割り当て：**HV は対象 TDI を特定の TEE に割り当てる．このとき，IOMMU に IOVA（GPA）→HPA のマッピングを設定し，TEE に対して TDI の MMIO をマッピングし，さらに IDE Stream を設定する．

（2） **SPDM セッションの確立：**従来 TSM が担っていた DSM との SPDM セッション確立を，各 TEE が主体となって実行する．

（3） **デバイスの検証：**TEE は SPDM セッション上で DSM から測定値（Measurement Report）を受領し，デバイスの正当性を検証する．以降，検証に成功したデバイスのみを信頼対象とする．

（4） **IDE 鍵の生成・設定：**デバイスの正当性確認後，TEE は IDE 鍵を生成し，DSM へ SPDM セッション経由で安全に配送する．PCIe Root Port 側の鍵設定は FKT/RTT の設定と同様に，TEE がハードウェア提供の暗号化機能を用いて暗号化した状態で実施する．この暗号化には TOCTTOU 対策として，PCIe Root Complex の設定変更時にインクリメントされるカウンタ値を用いる．PCIe Root Complex は SoC 内の保護機構により復号し，鍵をインストールする．併せて，インストール対象 Stream の設定レジスタ群をロックする．

（5） **MMIO マッピングと IDE 設定の検証：**図 3 に示すチャレンジレスポンスにより，MMIO マッピングおよび IDE 設定の正当性を検証する．TOCTTOU 攻撃を防ぐため，検証対象となる MMIO マッピングは事前に FKT/RTT 方式により保護する．

TEE は MMIO 領域に対してページ単位の MMIO Write（チャレンジ）を発行する（図 3 の (1)）．書き込みデータはナンスとし，PCIe Root Port にチャレンジレスポンスであることを識別させるための専用命令（または専用トランザクション種別）を用いる．当該要求は MMU に到達し，通常の MMIO アクセスと同様にアドレス変換が行われる（図 3 の (2)）．変換後の HPA が MMIO 領域であれば，要求は PCIe Root Port へ到達する（図 3 の (3)）．

要求が PCIe Root Port に到達すると，Root Port はアドレスに基づき対象 IDE Stream を導出する．いずれの有効 Stream にもヒットしない場合，要求を拒否する．これは，TEE 由来のアクセスに対して IDE 暗号化を必須とするためである．Stream が導出された場合，Root Port は当該 Stream に対応するレジスタ群（アドレス範囲，RID 対応，鍵設定等）がロック済みであることを確認し，さらに IDE 鍵が設定済みであることを検査する．

これらの検査に成功した場合に限り，Root Port は IDE により暗号化されたチャレンジ TLP をデバイスへ送信する（図 3 の (5)）．このとき送信ナンスはハードウェアにより別値へ変換され（図 3 の Nonce’），変換後ナンスを用いて検証可能であれば具体方式は任意とする．

TDI は受信したナンスを SPDM 経由で TEE へエコーし，併せて BAR 番号および BAR base からのオフセットを返す（図 3 の (6)）．この動作は TDISP 標準に定義されないため，本提案におけるデバイス側拡張と位置付ける．TEE は返送ナンスおよび BAR 情報を照合し，意図した BAR 領域への到達を確認する（図 3 の (7)）．

（6） **DMA 用メモリ共有の有効化：**DMA 許可は，TEE が FKT に共有鍵を設定することで実現する．ただし設定操作は FKT/RTT 方式に則り HV へ移譲する．この際，FKT に格納される共有鍵は固有値をナンスとして暗号化される必要があるため，まず固有値を設定する．本提案では IDE Stream と TDI が 1 対 1 で対応するため，固有値も同様に紐付けて管理する．具体的方法は後述する IDE 鍵更新手順に従う．

以上により，本方式は TSM に依存せず，TEE とデバイス間の相互認証，MMIO の整合性検証，およびセキュアなメモリ共有を実現する．

5.3　鍵更新とデバイス再利用時の機密消去

IDE Stream で用いられる暗号方式は AES-GCM であり，初期化ベクトル（IV）の枯渇時には鍵更新が必要となる．鍵更新に際しては，「鍵を設定したコンポーネント以外による更新を許容しない」ことが要件となる．そこで本提案では，IDE 送信に用いる IV のうち低位側の一部を鍵更新用に予約し，新鍵は予約された IV を用いて旧鍵で暗号化された状態で設定されなければならないとする．これにより，鍵更新手順も HV へ安全に移譲可能となる．

また，TDI の使用終了後に別 TEE または HV が再利用する場合には，IDE Stream 鍵および FKT アクセス用固有値を削除する．削除は以下の事象をトリガーとして実行する．

- MMIO 領域に対する RTT の変更
- IDE Stream の初期化

5.4　脅威モデルへの対処の要約

IDE 設定に関する脅威は，チャレンジレスポンスにより不正

設定を検出できる．また，HV は暗号化された鍵のみを観測し平文鍵を得られないため，鍵濫用による漏えい・改ざんは抑止される．

DMA に関する脅威については，IOMMU に FKT/RTT 方式を導入することで隔離を実現する．さらに FKT 参照には IDE Stream に紐付く固有値を要するため，非所有者による読み出しを防止できる．

MMIO に関しては, 所有者以外の MMIO アクセスは FKT/RTT によるアクセス制御で防止される．また，物理メモリ上へのマッピングおよび MMIO シャッフルはチャレンジレスポンスにより検出される．MMIO デコード範囲改変による経路すり替えは，TDI 運用中に Root Complex 内のデコード範囲変更をハードウェアでロックすることにより対処される．

6. ま と め

本稿では，ユニバーサル TEE アーキテクチャにおいて TSM を介することなく，CPU 実行時と同等のメモリ防御保証をデバイスアクセスに拡張し，PCIe TDISP 準拠のデバイスに最小限の改変で安全なメモリ共有およびデバイス制御を実現する手法を提案した．

今後は，提案した IOMMU および PCIe Root Complex 拡張の実装を進め，FPGA 等を用いたプロトタイプ実装による面積・性能評価，ならびに実際の TDISP 対応デバイスを用いた実証実験を行う予定である．

謝辞　本研究は、JST 経済安全保障重要技術育成プログラム【JPMJKP24U4】の支援を受けたものです。

文　献

[1] Intel, “Intel® Trust Domain Extensions,” Feb. 2023. https://cdrdv2.intel.com/v1/dl/getContent/690419

[2] AMD, “AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More,” Jan. 2020. https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/solution-briefs/amd-secure-encrypted-virtualization-solution-brief.pdf

[3] X. Li, C. Dall, R. Gu, J. Nieh, Y. Sait, and G. Stockwell, “Design and verification of the arm confidential compute architecture,” 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), pp.465–484, USENIX Association, Carlsbad, CA, July 2022. https://www.usenix.org/conference/osdi22/presentation/li

[4] R. Sahita, V. Shanbhogue, A. Bresticker, A. Khare, A. Patra, S. Ortiz, D. Reid, and R. Kanwal, “CoVE: Towards Confidential Computing on RISC-V Platforms,” Proceedings of the 20th ACM International Conference on Computing Frontiers, pp.315–321, CF ’23, Association for Computing Machinery, New York, NY, USA, 2023. https://doi.org/10.1145/3587135.3592168

[5] T. Sorensen and H. Khlaaf, “Leftoverlocals: Listening to llm responses through leaked gpu local memory,” 2024. https://doi.org/10.48550/arXiv.2401.16603

[6] PCI SIG, “PCI Express Base Specification Revision 7.0,” June 2025.

[7] 石川　裕，木村啓二，河野健二，光来健一，五島正裕，塩谷亮太，須崎有康，関山太朗，高前田伸也，竹房あつ子，中條拓伯，古川潤，宮澤慎一，“ハードウェア・ソフトウェア・理論の連携によるユニバーサル TEE アーキテクチャの実現に向けて—システムソフトウェアの観点から—,” 情処研報，vol.2025-OS-167，no.4，pp.1–9，May 2025． https://ipsj.ixsq.nii.ac.jp/records/2002115

[8] 石川　裕，木村啓二，河野健二，光来健一，五島正裕，塩谷亮太，須崎有康，関山太朗，高前田伸也，竹房あつ子，中條拓伯，古川潤，宮澤慎一，“ハードウェア・ソフトウェア・理論の連携によるユニバーサル TEE アーキテクチャの実現に向けて—ハードウェアの観点から—,” 情処研報，vol.2025-ARC-261，no.5，pp.1–7，June 2025． https://ipsj.ixsq.nii.ac.jp/records/2002241

[9] Y. Deng, C. Wang, S. Yu, S. Liu, Z. Ning, K. Leach, J. Li, S. Yan, Z. He, J. Cao, and F. Zhang, “Strongbox: A gpu tee on arm endpoints,” Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp.769–783, CCS ’22, Association for Computing Machinery, New York, NY, USA, 2022. https://doi.org/10.1145/3548606.3560627

[10] S. Volos, K. Vaswani, and R. Bruno, “Graviton: trusted execution environments on gpus,” Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation, pp.681–696, OSDI’18, USENIX Association, USA, 2018.

[11] A.-C. Aprodu, H. Meyer zumFelde, D. Kowatsch, and K. Böttinger, “Ghostee: An approach to solving the gpu-privacy trade-off for machine learning inference,” Proceedings of the 18th ACM Workshop on Artificial Intelligence and Security, pp.1–12, AISec ’25, Association for Computing Machinery, New York, NY, USA, 2026. https://doi.org/10.1145/3733799.3762962

[12] G. Dhanuskodi, S. Guha, V. Krishnan, A. Manjunatha, R. Nertney, M. O’Connor, and P. Rogers, “Creating the first confidential gpus,” Commun. ACM, vol.67, no.1, p.60–67, Dec. 2023. https://doi.org/10.1145/3626827

[13] Intel, “Intel® TDX Connect Architecture Specification,” Aug. 2025. https://www.intel.co.jp/content/www/jp/ja/content-details/862706/intel-tdx-connect-architecture-specification.html

[14] AMD, “AMD SEV-TIO: Trusted I/O for Secure Encrypted Virtualization,” March 2023. https://www.amd.com/content/dam/amd/en/documents/developer/sev-tio-whitepaper.pdf

[15] Suzuki Kuruppassery Poulose, “Arm CCA - IOMMU & Trusted Devices,” Sept. 2024. Linux Plumbers Conference.

[16] RISC-V International, “Confidential VM Extension I/O (CoVE-IO) for Confidential Computing on RISC-V platforms,” May 2024. https://github.com/riscv-non-isa/riscv-ap-tee-io/releases/download/v0.2.0/riscv-cove-io-v0.2.0.pdf

[17] NVIDIA, “NVIDIA Secure AI with Blackwell and Hopper GPUs,” Aug. 2025. https://docs.nvidia.com/nvidia-secure-ai-with-blackwell-and-hopper-gpus-whitepaper.pdf

[18] 内山一秀，五島正裕，“ユニバーサル TEE アーキテクチャのためのメモリ保護方式,” 情処研報，vol.2025-ARC-260，no.9，pp.1–8，March 2025． https://ipsj.ixsq.nii.ac.jp/records/2001060

[19] 内山一秀，松見湧斗，五島正裕，“ページ共有可能な TEE のメモリ保護機構における認証暗号によるテーブル管理の委譲,” 情処研報，vol.2025-ARC-261，no.6，pp.1–9，June 2025． https://ipsj.ixsq.nii.ac.jp/records/2002242

[20] 内山一秀，五島正裕，“ユニバーサル TEE アーキテクチャのためのメモリ保護方式,” Aug. 2025． cross-disciplinary workshop on Systems, Infrastructures, and programminG, xSIG.

[21] 松見湧斗，内山一秀，大畑幸矢，高前田伸也，古川　潤，五島正裕，“高度に並列動作する認証メモリ暗号アーキテクチャ,” 情処研報，vol.2025-ARC-262，no.1，pp.1–13，July 2025． https://ipsj.ixsq.nii.ac.jp/records/2002242

[22] Distributed Management Task Force, “Security Protocol and Data Model (SPDM) Specification Version: 1.4.0,” May 2025.