2. Intel TDXの概要

Ao Sakurai

2026年度セキュリティキャンプ全国大会
L1 – 暗号・CVMビルド&スクラップゼミ

本セクションの目標

- 本ゼミでメインで取り扱う、Intel TDXの背景、概要、そして脅威モデルについて理解する

- 本資料含む一連のスライドは、引用の明記が他になければ基本的にIntel TDX Demystified論文[1]の内容に準ずるものとする
  - ただし本スライド序盤の「TDXを取り巻く背景」部分は除く

Intel TDXを取り巻く背景

Intel TDXを取り巻く背景

- 本ゼミで取り扱う**Intel TDX（Trust Domain Extensions）**の前身とも言えるTEE実装として、Intelは2015年に**Intel SGX**（Software Guard Extensions）を発表した

- SGXは**部分隔離型**TEEの一種で、Ultra HD Blu-ray再生時のコピーガード処理等、商用的にも大規模に利用された
  - 詳細は過去のセキュキャン資料を参照

- 2020年にはクライアント向けマシンからのSGXの消滅が決定した一方で、サーバ向けマシン（第3世代Xeonスケーラブルプロセッサ）以降では現在に至るまで続投している

Intel TDXを取り巻く背景

- 一方、AMDはSEV（Secure Encrypted Virtualization）という**CVM型**のTEEを2016年という比較的早い段階から発表・展開してきた[2]

- CVMは信頼可能領域内のコード量が肥大化しがちというデメリットがあるが、引き換えに既存の**非TEE向けアプリケーション**を**そのままTEE内で動かせる**（lift-and-shift）というメリットを有する
  - コード量の肥大化は、そのまま脆弱性混入量の増加に繋がるためTEEとしては好ましくない
  - しかし、部分隔離型であるSGXでは、国際論文[3]で皮肉られるレベルにその専用プログラムの実装が壊滅的に難しい

Intel TDXを取り巻く背景

- 2026年現在、日本でも**Confidential Computing**（TEEを用いた秘密計算；以下CC）という概念が一定の認知を得てきている

- そしてこのご時世のニーズとして、当然にLLMのような**超大規模ワークロード**をCC上で実行するというものが生まれる

- これを実現するには、プロセスの一部を隔離するSGXのような**部分隔離型TEEではアーキテクチャ的に限界**があり、OS丸ごと内包する前提のCVM型が時流として台頭している

Intel TDXを取り巻く背景

- そこで、IntelがAMD SEVに追随する形で2023年[4]にリリースしたCVM型TEEが、SGXの後継とも言えるIntel TDXである

- TDXが登場した事で、IntelマシンではSGX、TDXという2つのTEEを以下のように臨機応変に使い分けられるようになっている：
  - よりセキュリティを求める比較的小規模なワークロードの保護にはSGX
  - 巨大なワークロードの保護にはTDX

Intel TDXの概要

Intel TDXの概要

- TDXは、一部の第4世代Xeonスケーラブルプロセッサ（Xeon-SP）及び、全ての第5世代Xeon-SP以降で対応している

- TDXはこれまでの説明の通りCVM型に分類され、TDXにおけるCVMインスタンスを**TD（Trust Domain）**と呼んでいる

Intel TDXの概要

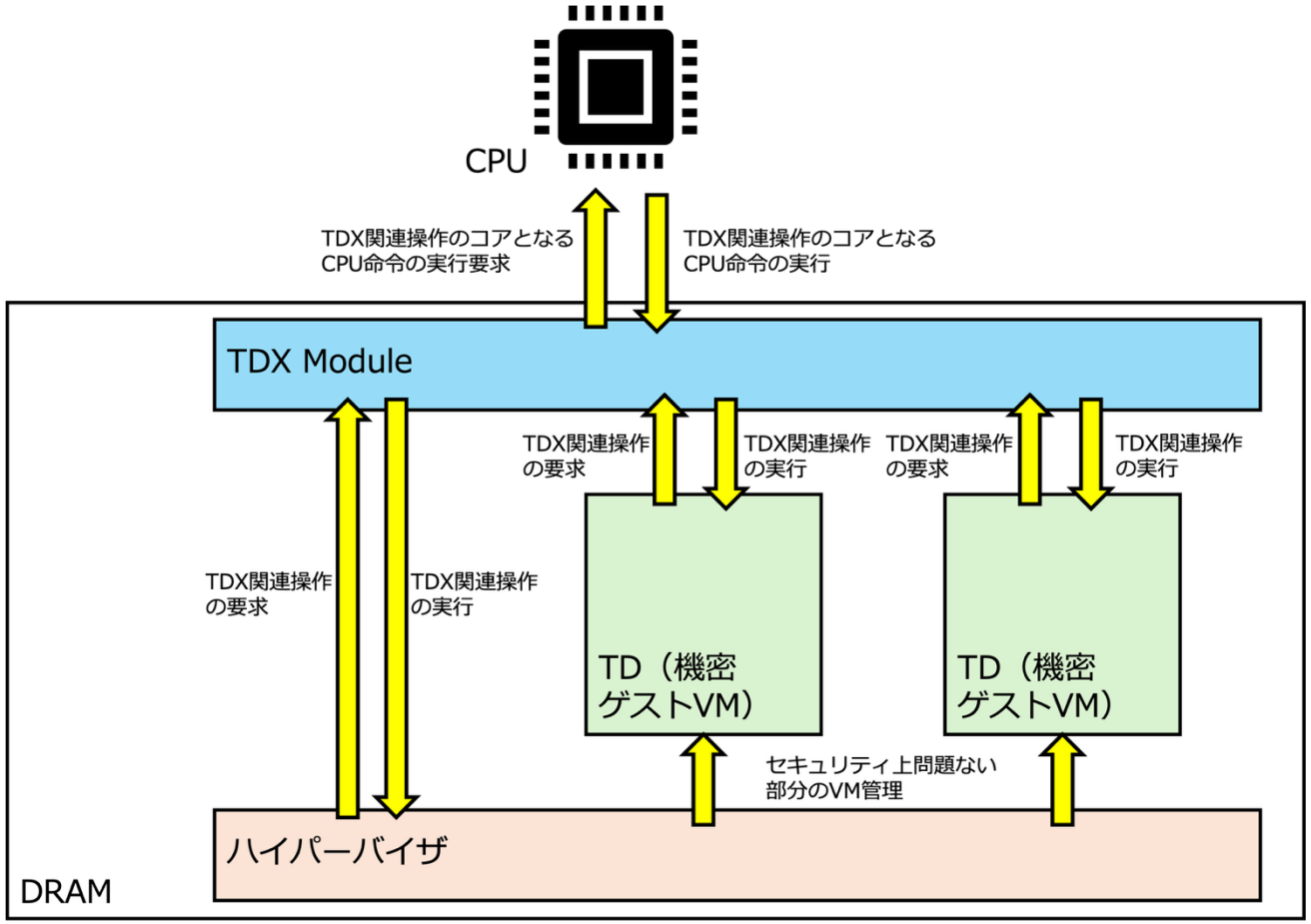
- TDXの目的は他のTEEと同様、メモリやCPUステートの機密性・完全性を保護する事が第一に挙げられる

- また、リモートからそのTDを利用する際に、真正なTEEマシン上で意図するTDが動作している事を確信するための手続きである**Remote Attestation（RA）**を提供する事も目的としている

- ただし、永続ストレージの暗号化機能はTDXの機能としては提供せず、TD内で各自フルディスク暗号化（FDE）を別途行う事が要求されている
  - これは他のCVMでも同様である

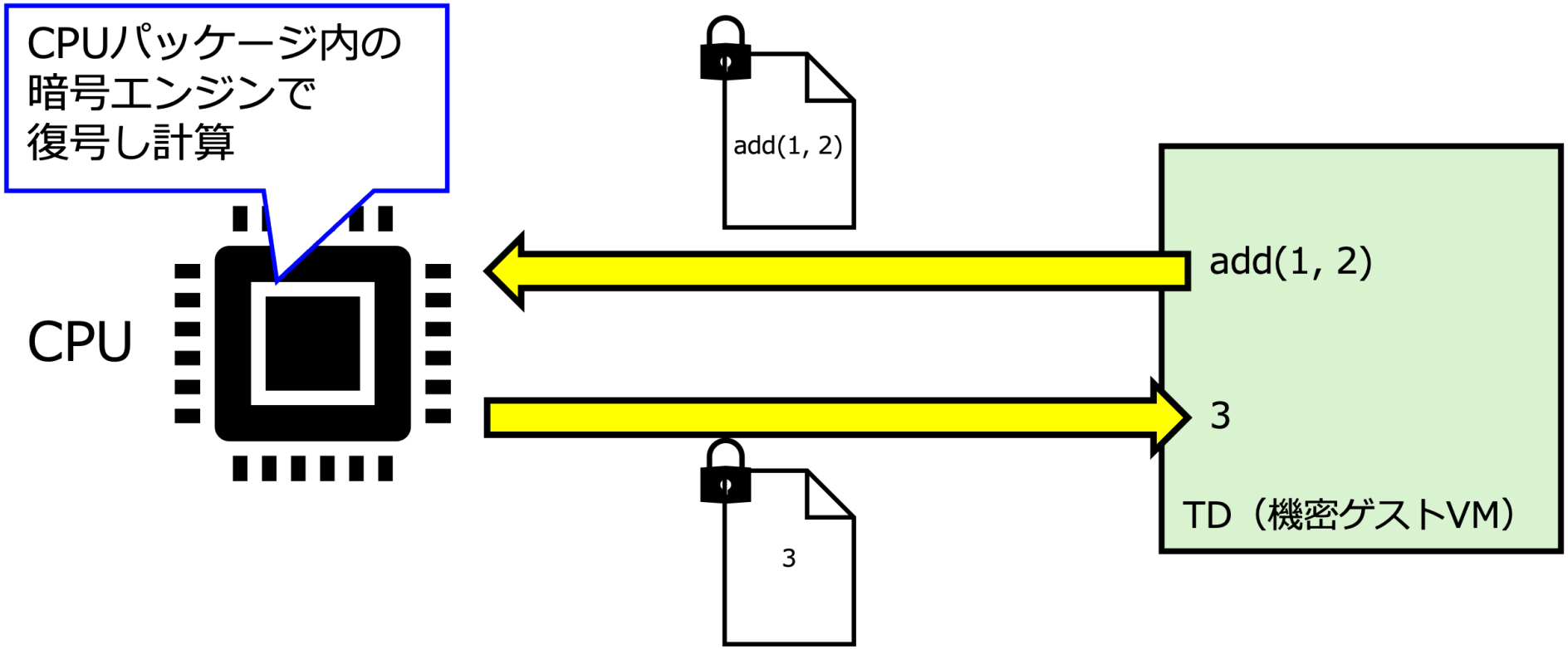
Intel TDXの概要

- TDXにおける隔離保護のイメージ図を以下に示す：



Intel TDXの概要

- TDXにおける機密計算の流れの概要図を以下に示す：



Intel TDXの概要

- TDXもTDというVMベースであるため、その管理には基本的に従来型のハイパーバイザ（以下、HV）が使用される

- しかし、HVはTEE外の主体であるため、あくまでもTDXの信頼可能範囲外の部分の操作を行う事しか許されていない

- TDの構築、実行、終了、あるいはAttestation（別スライドで説明）関係の信頼可能範囲に関わるクリティカルな処理は、HVの代わりに**TDX Module**と呼ばれるコンポネントが担当する

TDX Moduleの概要

- TDX Module：TDよりもさらに特権的なモードで動作する、TDXのTEE機能の実現の上で必要な処理を代行するソフトウェア。時にはTDX専用のCPU命令の実行も行う
  - いわば「**TD用マイクロハイパーバイザ**」とでも言える存在

- TDX ModuleはIntel TXTの一機能であるIntel ACMにより、Intel署名のついた**真正なもののみが使われる事が保証**されている
  - TXT：Trusted Execution Technology
  - ACM：Authenticated Code Module

TDX Moduleの概要

- ちなみにAMD SEV-SNPでは、AMD-SPというコプロセッサ上のSEV FWというファームウェアが似た役割を持つ

- TDX Moduleの詳細については後続の専用スライドにて詳細に解説する

TDXにおける隔離保護の概要

- TDはTDXの機能により、他のTDや非TDエンティティから**アクセスベース制御で隔離**される
  - さらに言えばTDX ModuleもTDのプライベートデータを見る事はできない
- さらに、TDのメモリ自体も**CPUハードウェア（HW）により暗号化**される
- この暗号化には**TME-MK**（Total Memory Encryption - Multi-Key）という技術が使用され、CPUのメモリコントローラ内の**MK-TMEエンジン**がその暗号処理を執り行う
  - 128bit AES-XTSにより暗号化される

TDXにおける隔離保護の概要

- どのメモリ暗号化機能を持つTEEにも共通の話であるが、MK-TMEエンジンのようなメモリ暗号化エンジンで復号された先の、**CPU内部ではデータは平文で存在**する
  - 例：キャッシュ階層、レジスタ上、その他マイクロアーキテクチャバッファ

- 逆に、CPUの外に出る際は必ずメモリ暗号化エンジンで暗号化されるため、**平文でCPU外に出る事は無い**

- 現代の最新CPUの内部を何らかの方法（例：電子顕微鏡観察、物理プロービング）で観測するのは技術的に不可能であると考えられるため、**TEEはCPU内部は安全である前提とする**

TDXのセキュリティ原則

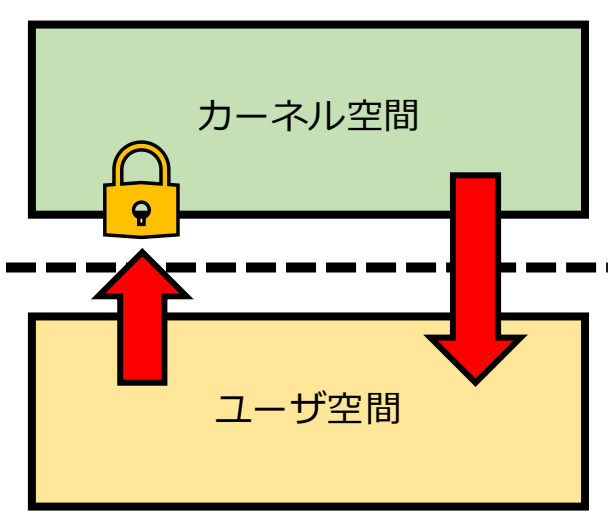
TDXのセキュリティ保証

- 他のCVMと同様、TDXはセキュリティドメイン間の暗号学的分離により、TDを以下のような主体から隔離保護する事ができる：
  - クラウドサービスプロバイダ（CSP）等が管理するHV
  - ホストOS
  - 同居する他のVM（TD・非TD双方を含む）
  - その他そのTD外のアプリケーション

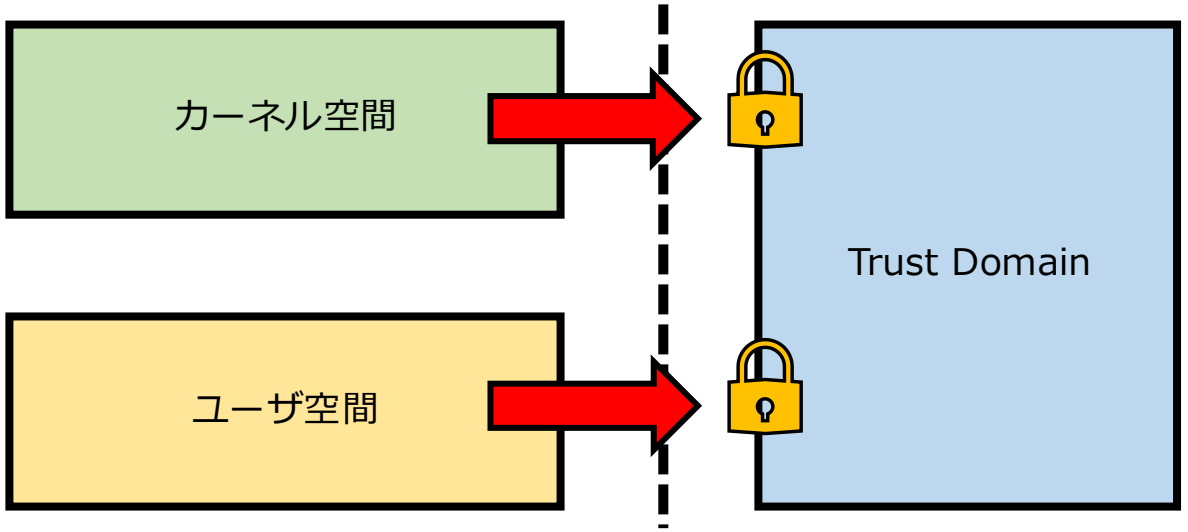
- **従来型のVM**ではHVが握る権限が強く、**そのVMのメモリ内容を見られてしまう**状況であったため、このTDX（CVM）のセキュリティ保証は**非常に強力で魅力的**である

非階層型分離モデル

- CVMに限らずTEE全般は、「非階層型分離モデル」を実現できるという特徴を備えている[5]



従来の分離モデル



TEEの非階層型分離モデル

非階層型分離モデル

- 従来のゲストVMとHV間、あるいはユーザとOS（カーネル）間では、**後者が前者よりも強い**という**階層構造**が前提として置かれている

- これは、前者→後者のアクセスはできないが、その逆はできてしまう事を意味する

- これは、弱い権限である前者（VMやユーザプロセス等）が**仮にプライバシー性の高い情報**を有していても、より強い後者（カーネル、HV等）から**それを守れない事**を意味する

非階層型分離モデル

- 一方でTDX含むTEEでは、そのTEEで保護される隔離領域は、**どれほどに強い外部の特権主体**でも**そのTEE隔離領域の中身を見る事はできない**

- 逆に、TEE側から外へのインタフェースも厳格に制限されているため、TEEから外の何かを不当に閲覧できてしまう事もない
  - TEE外含めて余程恣意的に設計するか、バグでも無い限りはまずない

- このように、TDX含むTEEは従来の階層型分離モデルよりも強力な**非階層型分離モデル**を実現する技術であると言える

TDXの隔離保護を支える要素

- TDXの提供する、メモリとvCPUの機密性・完全性保証や、他のセキュリティドメインからのアクセス・改竄の阻止は、主に以下の3つの要素によって実現される：
  - メモリアクセス制御
  - ランタイムメモリ暗号化
  - TDX Module

- **メモリアクセス制御**は、言い換えれば権限的な分離とも表現でき、Arm TrustZoneのようなメモリ暗号化を備えないTEEであってもTEEであれば必ず備えている機能である

TDXの隔離保護を支える要素

- 一方、**ランタイムメモリ暗号化**は暗号学的な分離で、TrustZoneのように一部TEEでは実装されていないが、TDXではしっかりと実装されている

- また、セキュリティクリティカルな処理は**TDX Module**が実施する事でTDX全体の安全な動作を担保する

- そして、真正なTDX対応Intel製プロセッサ上で、上記のようなセキュリティ機能を備えているTDXの下、目当てのTDが動作している事をリモートから検証する手段が**RA**である

TDXの隔離保護を支える要素

- ただし、これらは後述の脅威モデルを前提としているため、完全なる万能ではない点には注意が必要
  - 例えば物理的アクセスを持つ非常に強力な攻撃者からの攻撃は、TDXを含む現行のTEEでは完全に防ぐ事はできない

- 次ページ以降では、TDXがメモリ機密性、CPUステート機密性、実行の完全性、I/O保護の観点でどのような保護を実現できるかを説明する

TDXにおけるメモリ機密性・完全性

- 前述の通り、レジスタやキャッシュメモリのようなCPU内では TDのデータは平文で存在する
  - これは先行のTEEであるSGX含めどのTEEでも同様な話である
- そして、そのデータがメインメモリにストアされるとなった時に、 MK-TMEエンジンがその**TDに一意に割り当てた暗号鍵**で 暗号化し、CPUの外に出してメインメモリにストアする
  - この暗号化はキャッシュラインサイズ（大抵は64B）単位で実施される
- 周辺デバイスやホストOS等の権限のない主体が、TDのプライベート メモリを**検出される事なく読み取り・改竄する事はできない**
  - プライベートメモリ：外部への共有にも使用しない、そのTDに完全に プライベートであるメモリ

CPUステートの機密性

- TDXは、セキュリティドメイン間の全てのコンテキストスイッチ時にvCPU （仮想CPU）ステートを管理する事で、**TDのレジスタコンテキストの保護**も行う
  - TDのレジスタコンテキストには、**TD内の秘密情報に依存する重要な情報**も**数多くある**ため、**野晒しにするのは危険**である
- SGXでも、例外等によるEnclave緊急脱出時には、SSAと呼ばれる安全な領域にレジスタコンテキストを退避させる
  - 意図通りの通常の脱出（EEXIT）時にも、tRTSという信頼可能ランタイムがレジスタコンテキストの安全なクリアを実施する
- AMDの初代SEVではこの**退避したレジスタコンテキストを保護しておらず問題になった**ため、SEV-ES（Encrypted State）という次の世代のSEVが開発・提供された

CPUステートの機密性

- 同時に、コンテキストスイッチの際にはTLBエントリや分岐予測バッファといったプロセッサ内部のレジスタやバッファからTD固有のステートをクリアまたは分離する
  - TLB：Translation Lookaside Buffer。いわばページテーブルのキャッシュ
- このTLBや内部バッファのクリア措置はサイドチャネル攻撃対策の側面も強いと推測される
  - 例えばSGXでは、Enclave脱出時に分岐予測バッファを適切にクリアしていなかったが故にBranch Shadowing Attack[8]という攻撃が可能となっていた

実行の完全性

- TDXはホスト等のTD外の主体からの、**TDの実行完全性の保護**も**提供**する

- 例えば、もしTDに対して割り込みが発生した場合、再開時には確実にその割り込み発生地点から再開される事が保証される

- また、外部からの悪性操作によりvCPUステートの改竄、プライベートメモリへの命令注入・変更・削除が行われた場合、それらを検知できる仕組みとなっている

実行の完全性

- ただし、TD内で実行されるコードの正しさまではTDXは保証しない

- これはどのTEEも同じで、コードの**機密性と完全性の保護は行う**が、そのコードが持つ**脆弱性等を直してくれたりは当然一切しない**

- 例えばBoF攻撃を突いたRoP攻撃に場合、その脆弱性を全て潰しておくか、Intel CETのような対策技術を使用して**TD所有者**が自前で対策する必要がある。他の（一般的な）脆弱性も同様
  - CET：Control flow Enforcement Technology
  - TD所有者（TD Owner）：そのTDイメージを用意しクラウド等のTDXマシンにデプロイする主体。ただし、現状のクラウドではクラウドが自前で用意し自己デプロイする事も多く、機能不全感も否めない

I/Oの保護

- 周辺デバイスやGPU等はTDの信頼境界外であるため、これらはTDのプライベートメモリにアクセスする事はできない

- 代わりに、TDがこれら外部の主体とのデータ共有用に明示的に切り出す**共有メモリ**を通す事で、TDと外部主体とでデータのやり取りを行う事ができる
  - 勿論、ホストOS等デバイス以外の外部主体との公開情報のやり取りに共有メモリを使う事もできる

- 共有メモリはTDのセキュリティ保証外であるため、**TDXによりデータの機密性や完全性が保護される事は無い**点に注意
  - よって、TLS等でデバイスとの暗号通信路を張る事等による転送データの保護はTD所有者（TDワークロード開発者）の責任となる

I/Oの保護

- 例外として、TDX 2.0では**TDX Connect**というTrusted I/O機能が実装されている

- これを利用すると、そのデバイスをAttestation（安全性・真正性を確認）した上で、「**信頼可能デバイス**」として**TDの信頼境界内に組み入れる**事ができる
  - つまりは**信頼可能デバイスはTDのプライベートメモリに直接アクセス可能**となる
  - TDとデバイスの通信の際にはIOMMUを通して実施する

- TDX ConnectはPCI SIGの**TDISP**という規格に則って実装されている
  - TDISP：TEE Device Interface Protocol
  - AMD SEVではSEV-TIOという同様の実装が開発されている

TDXの脅威モデル

TDXの脅威モデル

- これまでに説明した「TDXにできる事」は、あくまでもTDXが想定する脅威モデル下でのみ十全に機能するため、このTDXの脅威モデルについて説明する

- TDXでは、攻撃者がマシンに物理的またはリモートでアクセスし、以下の要素を制御できる前提を想定する：
  - ブートファームウェア
  - システム管理モード（SMM）
  - ホストOS
  - ハイパーバイザ（HV）
  - 周辺デバイス

TDXの脅威モデル

- 要するに、特権・非特権に関わらずCPUやTEEの外のいかなる主体からの攻撃も想定している事になる
  - これはTEE特有の強い脅威モデルに一致している
- 攻撃者の目的は、機密データの窃取を行う事であり、この防御をTDXはメインとして据えている
- 対して、可用性に対する攻撃は他のTEE同様、基本的には考慮しない
  - 例えば、攻撃者がHVを制御してTDに一切のリソースを渡さないDoS攻撃が可能だが、TEEは可用性保護は行わないのでスコープ外として見る

TDXの脅威モデル

- この脅威モデルにおいて攻撃者による実行が想定されている攻撃を以下の表に列挙する：

攻撃タイプ	具体的な攻撃例
正規インタフェースを通した攻撃	TDX Moduleのホスト側インタフェースを通じて TDX Moduleと対話を行う事による、TDのビルド、初期化、（同一性のハッシュ）測定、破棄。 このインタフェースへの入力は、意味論または文法的に有効でも無効でも良く、かつ任意の順序で呼び出せる。
計算リソース制御	物理メモリページ、プロセッサ時間、物理/仮想デバイス等の、TDに割り当てられた計算リソースの制御
割り込み	任意のタイミングでのTDへの割り込み
メモリ読み書き	任意のメモリ位置への読み書き。注意点として、これはTDのメモリ暗号化や分離を無条件に突破できるという意味ではない。他の項目にも言える事だが、あくまでもTDによる保護機能の影響は受ける

TDXの脅威モデル

- 前ページの続き

攻撃タイプ	具体的な攻撃例
IOMMUの再構成	IOMMUの構成や設定の変更
TDへの入力の操作・改竄	以下のようなTDへの入力として機能する要素の操作： - 電源やHW構成管理を行うACPIの各種情報が記述されたACPIテーブル - PCI（Peripheral Component Interconnect）構成設定 - MSR（モデル固有レジスタ）。MSRとは、CPU内部の特殊な制御を行うためのCPU固有のレジスタの事 - DMA（Direct Memory Access） - エミュレートされたデバイス - ホストによって処理されるハイパーコール - 乱数生成源 - 時間生成源（time notion）
一部の物理攻撃	- 例えば悪意のあるDMAを通じてのメインメモリアクセスにより、TDを侵害する事はできない。 - 別途説明するCryptographic Integrity（Ci）という機能を有効化すると、RowHammerのような電荷氾濫によるメモリのビット反転攻撃も防げると主張されている。 - マシン終了後にDRAMを取り出して冷却し、電荷揮発を遅らせつつ解析器で内容を盗聴するようなコールドブート攻撃は対策可能。 - バスのプロービング（介入・探知しての盗聴）は、その試行単体では暗号文が読み取れるだけであるため、脅威にならない。しかし、複数回試行による暗号文サイドチャネル攻撃の実例は存在する。

TDXの脅威モデル

- 表を見ると複雑だが、要は**ソフトウェア（SW）制御で実現できる機密性・完全性攻撃**はどのようなものであっても**TDに対しては無力**であるという事である
  - TDXのRAに対する攻撃も、次ページ以降説明する強力な物理攻撃を使われない限り勿論保護できる

- 一方で、**物理的な攻撃**に関しては現行のTEEの例に漏れず、正直**不十分な側面があると言わざるを得ない**

- 前述の表の通り、コールドブート攻撃、RowHammer攻撃、バスの単純なプロービング攻撃程度であれば防げる

TDXの脅威モデル

- 反面、TDXもRAにおいて依存しているSGX Enclaveは、TDメモリ同様メモリ暗号化の調整値として物理アドレスのみに依存するため、SGX Enclave経由で攻撃を仕掛ける事を考えられる
  - 物理アドレスを合わせれば決定性であり、かつTDXよりもSGXの方が経験的に攻撃がしやすい事を悪用した攻撃

- このRA専用SGX Enclaveに対して物理装置で暗号文を読み取り、その内容からRAにおいてクリティカルな鍵を抽出し正当なTDの身元を偽造し放題にする、**TEE.fail**[9]という攻撃が存在する

TDXの脅威モデル

- また、DIMM（メインメモリ）のアドレス線のアース（0化）を物理装置で動的に切り替え、メモリ上の特定の物理位置とのエイリアスを作り秘密を抽出する**BatteringRAM**攻撃[10]がある
  - DDR4特化であるため、DDR5が必須なTDXには効かず、SGXやAMD SEV-SNPを攻撃対象としている

- このような強力な物理攻撃からの保護を提供するのは、今では廃止された古いクライアント向けのSGX（**Legacy-SGX**）**のみ**で、**現行のTEEではことごとく対策する事ができない**

- これらの物理攻撃については本ゼミの攻撃編で解説予定

TDXでは対策できない攻撃

- 以上も踏まえ、TDXにおいて明示的に対策されない攻撃を以下の表に示す：

攻撃タイプ	具体的な攻撃例
メモリロールバックを行う物理攻撃	物理的な攻撃装置を用いて、特定のメモリ位置のメモリページを過去のメモリページで上書き（ロールバック）するような攻撃。ある過去の時点t'では秘密情報が載っていたが、その後の時点tでは返送用のデータが載っているような場合、このロールバックにより秘密情報を返送してしまう事になる。TDXにおける完全性保護機能はメインメモリ上で完結するSWレベルでしか無いため、そのようなSWレベル完全性保護に使われるメタデータすら、この種の攻撃によりロールバックされてしまう。
CPU内からの秘密奪取	プロセッサチップのOTP（One-Time Programmable）ヒューズに格納されている秘密鍵材料（SGXでも使用されているRoot Provisioning Key; RPK等）の抽出。 しかし、極めて古いSGXマシンにおける太古の昔の脆弱性を通した例でしか抽出例が無く、現実的にTDXマシンからこれらを抽出する事は不可能と言って良い。
インターポーザ攻撃	前述の通り、TEE.failのような実際の攻撃成功例がある。
サイドチャネル攻撃	故障注入、電力グリッチ、タイミング攻撃、電力解析、ページフォールト攻撃等。 SGXで言えば、それぞれ故障注入：Plundervolt、Load Value Injection（LVI）；電力グリッチ：Plundervolt；タイミング攻撃：オーソドックスなキャッシュサイドチャネル攻撃；電力解析：PLATYPUS；ページフォールト攻撃：Controlled-Channel Attacks 等が存在する。

TDXでは対策できない攻撃

- これまた表が複雑であるが、要は**一定以上に強力な物理攻撃に対しては無力**であり、かつ**サイドチャネル攻撃**に対して何らかの**追加の保護は行わない**という事である

- ただし前回スライドでも説明の通り、TDX（TEE）の使用により**サイドチャネル攻撃に弱くなるわけではない**という点に注意
  - 弱いコードはTEEで動かしても弱く、逆に定数時間実装のような強いコードはTEEで動かしても強い
  - 稀に「TEEを使うとサイドチャネル攻撃に滅茶苦茶弱くなる」という誤解があるがこれは**明確に誤り**

TDXでは対策できない攻撃

- これまた表が複雑であるが、要は**一定以上に強力な物理攻撃に対しては無力**であり、かつ**サイドチャネル攻撃**に対して何らかの**追加の保護は行わない**という事である

- ただし前回スライドでも説明の通り、TDX（TEE）の使用により**サイドチャネル攻撃に弱くなるわけではない**という点に注意
  - 弱いコードはTEEで動かしても弱く、逆に定数時間実装のような強いコードはTEEで動かしても強い
  - 稀に「TEEを使うとサイドチャネル攻撃に滅茶苦茶弱くなる」という誤解があるがこれは明確に誤り

TCB

- TEEにおいては、**TCB（Trusted Computing Base）**という用語がよく登場する

- しかし、このTCBという用語は使用者により**その意味する所が大きく変わってくる**

- 本ゼミ資料では、**狭義のTCB**と**広義のTCB**という2つのTCBを用語として定義して使用する

TCB

- 狭義のTCBとは、そのTEEが**安全に動作する上で正しく動作**し、**悪意や危殆化が存在しない**事が求められるコンポネント一式の事を表す
  - 語弊を恐れずに言えば、そのTEEを使う上では**無条件に信頼しなければならないコンポネント**の事を指す
  - TEEインスタンス（TD）は狭義のTCBが正しい前提で提供されるものであるため、TD自体は狭義のTCBには入らない

- 広義のTCBとは、**そのTEEにおいて保護されている領域全般**を指す
  - 狭義のTCBに加え、そのTEEの機能により保護されているTD（TEEインスタンス）等も「保護領域」であるため、広義のTCBには含まれる

TCB

- **狭義のTCB**は、主に**TEE自体の設計**におけるアタックサーフェスの議論をする際に用語として使われる事が多い
  - 例：「SGXはマイクロコード実装はじめハードウェアTCBの割合が多い」「Arm CCAはTF-Aはじめソフトウェア TCBの割合が多い」

- **広義のTCB**は、その**TEEに載るコードベースの大きさに起因**するアタックサーフェスの大きさを議論する際に用語として使われる
  - 例：「CVMはLinuxが丸ごと載っているのでTCBが大きすぎる」「SGX Enclaveは特定コードのみ隔離するのでTCBが小さい」

TDXにおける狭義のTCB

- TDXにおける狭義の**SW-TCB**を実現するためには、以下のような技術（狭義の**HW-TCB**及びその機能）が使用されている：
  - TDX対応プロセッサ及びその中のマイクロコード
  - **Intel VT**（Intel CPUに搭載されている仮想化支援機能）
  - Intel TME-MK
  - Intel TDX
- マイクロコード：特にCISCプロセッサにおいてCPU命令を実行するためにプロセッサ内で使用される、より単純な命令
  - 語弊を恐れずに言えば、分子に対する原子のような命令
- VTやTME-MK等の基盤技術及びその使われ方の説明は次回のスライドで実施する

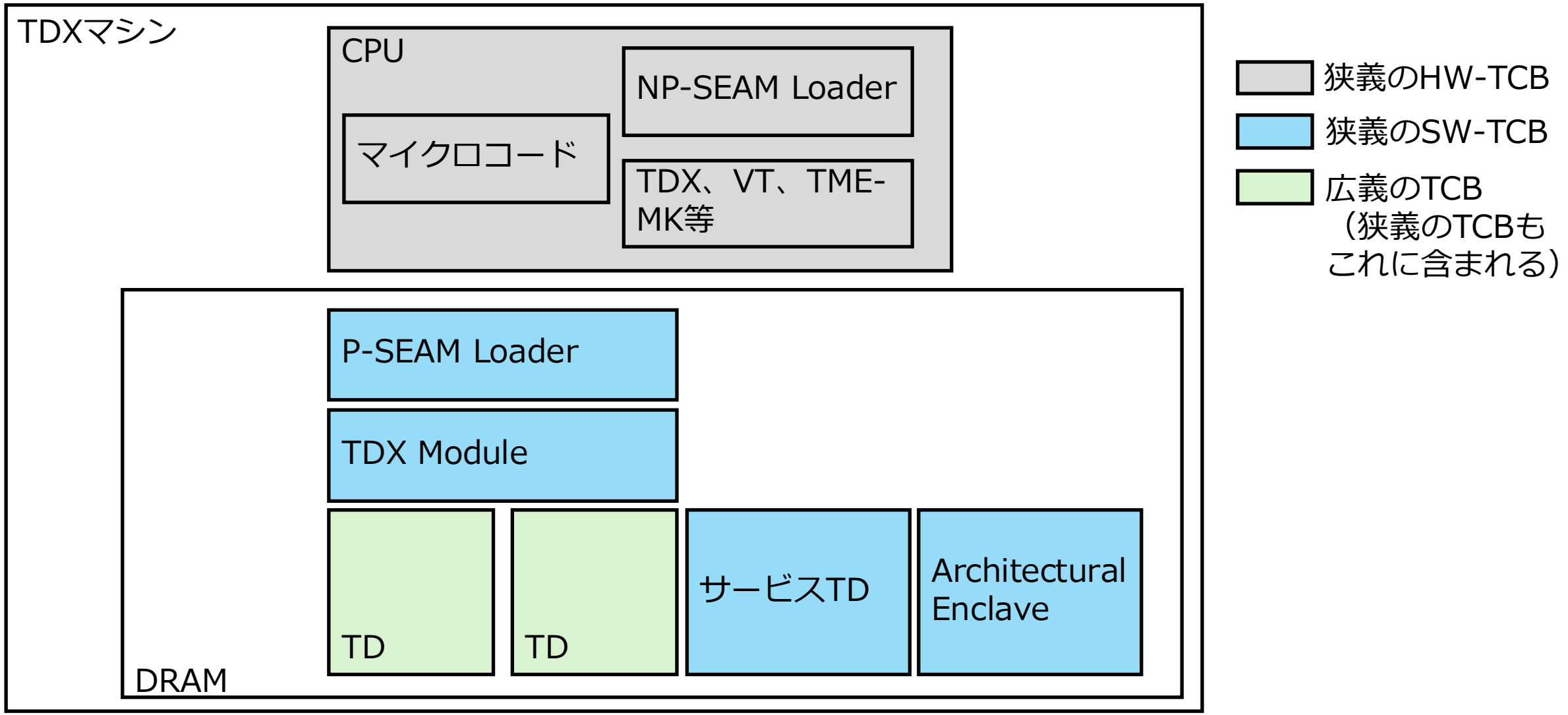
TDXにおける狭義のTCB

- そして、TDXの狭義のSW-TCBには以下のようなコンポネントが含まれる：
  - TDX Module
  - **NP-SEAM Loader**（プロセッサ内で動作するのでHW-TCBに括る方が適切かもしれない）、**P-SEAM Loader**
  - RAにおいて使用されるSGXのArchitectural Enclave（AE）。原則としてIntel署名のついた専用Enclaveの事である

- TDX ModuleやP/NP-SEAM LoaderについてはTDX Moduleについてのスライドにて説明する

TDXにおける狭義のTCB

- 狭義のTCBと広義のTCBの関係性のイメージ図は以下の通り：



TDXにおける信頼の前提

- TDXで使用されるHW/SWコンポネントの開発、製造、構築、署名の全てにおいて、**全面的にIntelを信頼する必要がある**
  - これはTEEベンダさえ信用すれば高性能なData-in-Use保護を行えるというトレードオフを飲んだ**TEEであれば宿命と言える性質**である
- また、TDXで使用される暗号プリミティブは、乱数生成やサイドチャネル攻撃耐性含め、健全かつその実装がIntelにより安全に行われているという前提を取る
- かつ、RAにおいてトラストアンカー及びルートCAとして機能する**Intel PCS**も信頼する必要がある
  - PCS：Provisioning Certification Service

TDXにおける信頼の前提

- TDX Module、P/NP-SEAM Loader、RA基盤となるDCAPのソースコードは**監査用に公開されている**ため、その内容を見て信頼性の評価を行う事はできる
  - DCAP：DataCenter Attestation Primitives

- ただし、Intelが真にそのソースからビルドしているかの確認は、測定値（ハッシュ）の確認はRAで可能ではあれど限度があり、やはり**基本的にIntelは信頼する前提**である

- 逆に言えば、Intel純正のものを使用する限りは、**Intel以外は信頼しなくて良いと割り切れる**とも言える

まとめ

- Intel TDXについての概要について説明した

- TDXによって守れる事、守る上で前提とする脅威モデル、
  そしてTDXを使う上で信頼の前提とする要素について説明した

参考文献

[1] Pau-Chen Cheng, Wojciech Ozga, Enriquillo Valdez, Salman Ahmed, Zhongshu Gu, Hani Jamjoom, Hubertus Franke, and James Bottomley. 2024. Intel TDX Demystified: A Top-Down Approach. ACM Comput. Surv. 56, 9, Article 238 (September 2024), 33 pages. https://doi.org/10.1145/3652597

[2] “Configuring Confidential Computing with AMD SEV-SNP and VMware ESXi 9.0”, Lenovo, https://lenovopress.lenovo.com/lp2349-configuring-confidential-computing-with-amd-sev-snp-and-vmware-esxi-90

[3] ”SgxElide: Enabling Enclave Code Secrecy via Self-Modification”, Erick Bauman et al., https://dl.acm.org/doi/pdf/10.1145/3168833

[4] “Intel TDX”, Edgeless Systems, https://edgelesssys.github.io/edgeless.systems-wiki/hardware/intel-tdx/

[5] ”Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution”, Jo Van Bulck et al., https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-van_bulck.pdf

[6] “INTEL SOFTWARE GUARD EXTENSIONS (INTEL SGX)”, Intel, https://www.intel.com/content/dam/develop/external/us/en/documents/332680-001-720907.pdf

参考文献

[8] “Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing”, Sangho Lee et al., https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-lee-sangho.pdf

[9] “TEE.fail: Breaking Trusted Execution Environments via DDR5 Memory Bus Interposition”, Jalen Chuang et al., https://tee.fail/files/paper.pdf

[10] “Battering RAM: Low-Cost Interposer Attacks on Confidential Computing via Dynamic Memory Aliasing”, Jesse De Meulemeester et al., https://batteringram.eu/batteringram.pdf