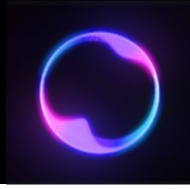
1. TEEとは何か？

Ao Sakurai

2024年度セキュリティキャンプ全国大会
S3 – TEEビルド&スクラップゼミ

本セクションの目標



- Intel SGXに触れるにあたり、SGXもその一員であるTEE技術についてその背景や考え方、性質を押さえる

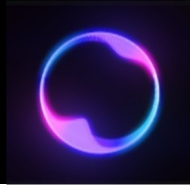
- TEEの有力な応用先の一つでもある秘密計算についてごく軽く触れ、関連する他の技術について知り比較を行う

自己紹介

- 名前：櫻井　碧（さくらい　あお）
  - Twitter: @dd_cliffford
- 所属/肩書：株式会社Acompany
  R&Dチーム テックリード、
  未踏スーパークリエータ（2019）
  seccamp全国大会講師（2023～）
- 出身大学：早稲田大学（情報工学修士）
- 分野：TEE、暗号、プライバシーテック等
- 趣味：アメ車、バイク、ゲーム、蒙古タンメン中本

「データを確実に保護しながら計算する」方法

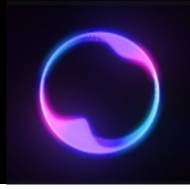
現代のITインフラに蔓延する「偽の信頼」



- 「自らのデータを他人に預けて何らかの処理をしてもらう」ユースケースは、往々にして一定のニーズが存在する
  - オンラインショッピング
  - デジタル著作権管理（DRM）処理
  - クラウドコンピューティング
  - クラウドストレージ
  - etc...

- では何をもってしてその**他人**を「**信頼できる**」と見做している？

「人間を信用してはいけない」

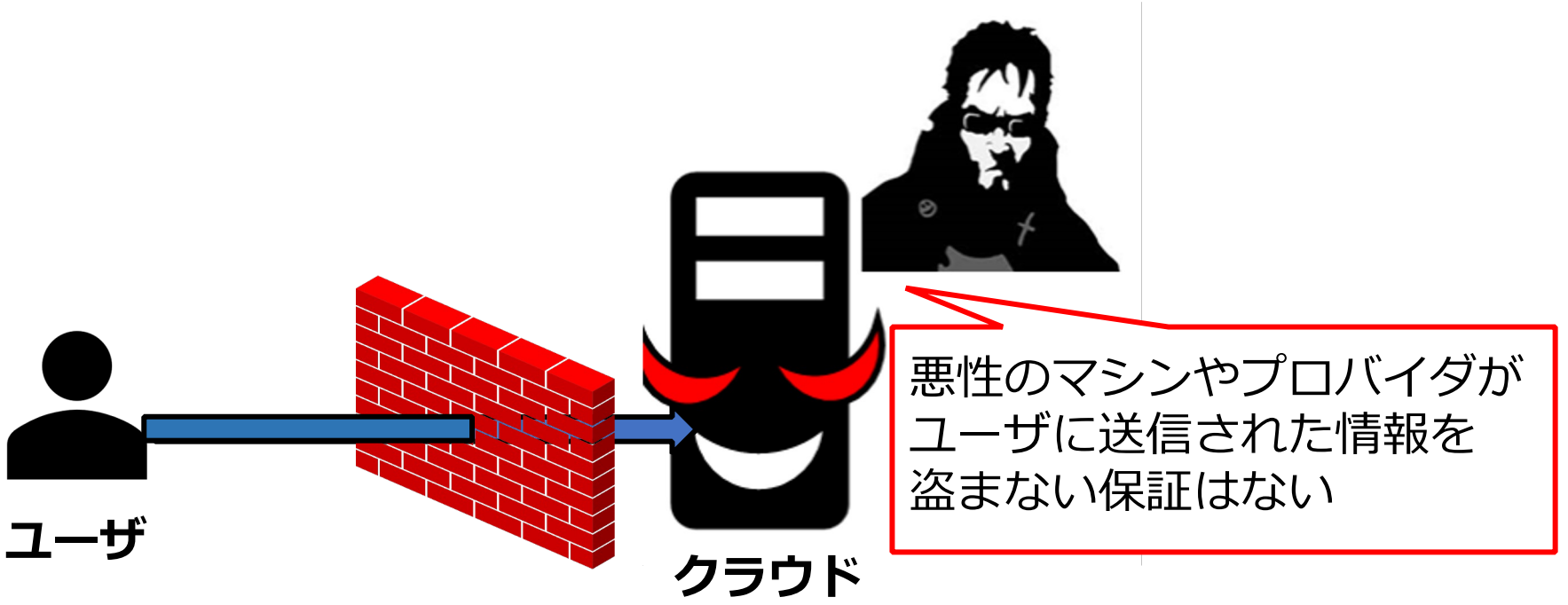


- この世には、**明らかに信用ならない**のに**世間に受け入れられてしまっている**システムが**多数存在する**

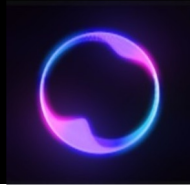
- この事実は、特に**機密性の高いデータ**を扱う場合に**大きな問題**となる
  - 医療情報
  - 生体情報
  - 機微な個人情報
  - クレジットカード番号
  - etc.

ケース①：クラウドサービス

- 現在普及しているクラウドサービスは、**クラウドプロバイダ**を**信用する事が出来ない**
  - FW等で外界との境界にて防御はしている
  - 境界の内側のセキュリティは「**ブラックボックス**」
  - 「**ハイパースケーラだから安心**」は**根拠のない神話**



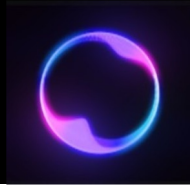
ケース②：著作物配信



- ビデオや音楽のストリーミング配信等では、**デジタル著作権保護（DRM）**によって**コンテンツを保護**する必要がある
  - いわゆる”**割れ厨**”ユーザ対策

- しかし、執念深いユーザは**DRMをバラバラに**して解析し、鍵を取り出して**暗号を解き**、**コンテンツを抽出**してしまう
  - メモリは守られていないので根本的に無意味

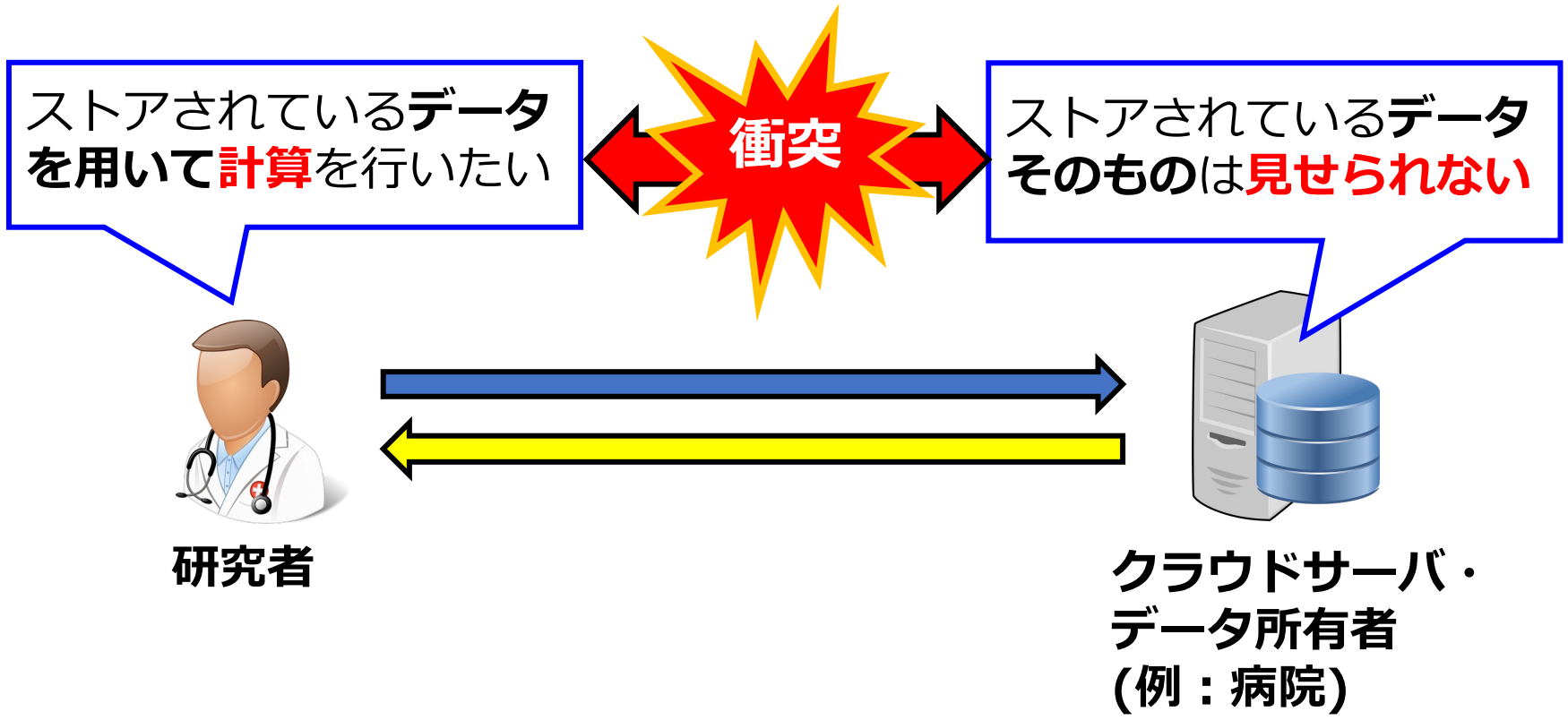
データセキュリティに必要な根本的な思想



- 大原則は、「**人間の善意**や**人間自体の信頼性**には頼らず、
  暗号学的・数理的・構造的に絶対的な安全性を保証する事」

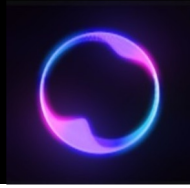
- **データが保護**されていなければならない**全てのフェーズ**で、
  絶対的にデータが保護された状態で処理を進めれば良い

「秘密計算」という新しい概念



この衝突を解決するには、**データの中身**を**見ない**まま**計算**を行い、各データを**特定できない**形の結果を得る「**秘密計算**」技術が必要

準同型暗号を使う？



- 従来手法でこの要件を満たす技術として有名なのは「**準同型暗号**」

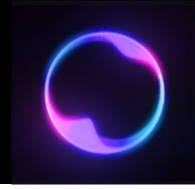
- **暗号文の状態**で**足し算**や**掛け算**が出来る種類の暗号

$$Enc(1) + Enc(4) = Enc(5)$$

暗号化　暗号化　復号

1　4　5

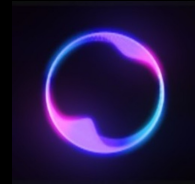
準同型暗号の現実



- 結論から述べると、**現代のコンピュータ**は**準同型暗号**に**追いついていない**
  - 必然的に**準同型暗号は現実的ではない**

メリット	デメリット
ハードウェアを**信頼しなくて良い**	**極めて遅い**
厳密な意味での**完全な保護**	**莫大なメモリ**を食い潰す
耐量子性を持つ	**精度に難がある**

準同型暗号の致命的な欠点



- とにかく**非常に重く**、到底**実用に堪えない**

```
aos@Apollyon:~/cpp$ ./a.out
Result: 100000000
Elapsed time: 113[ms]
```

普通のプログラム：0.113秒

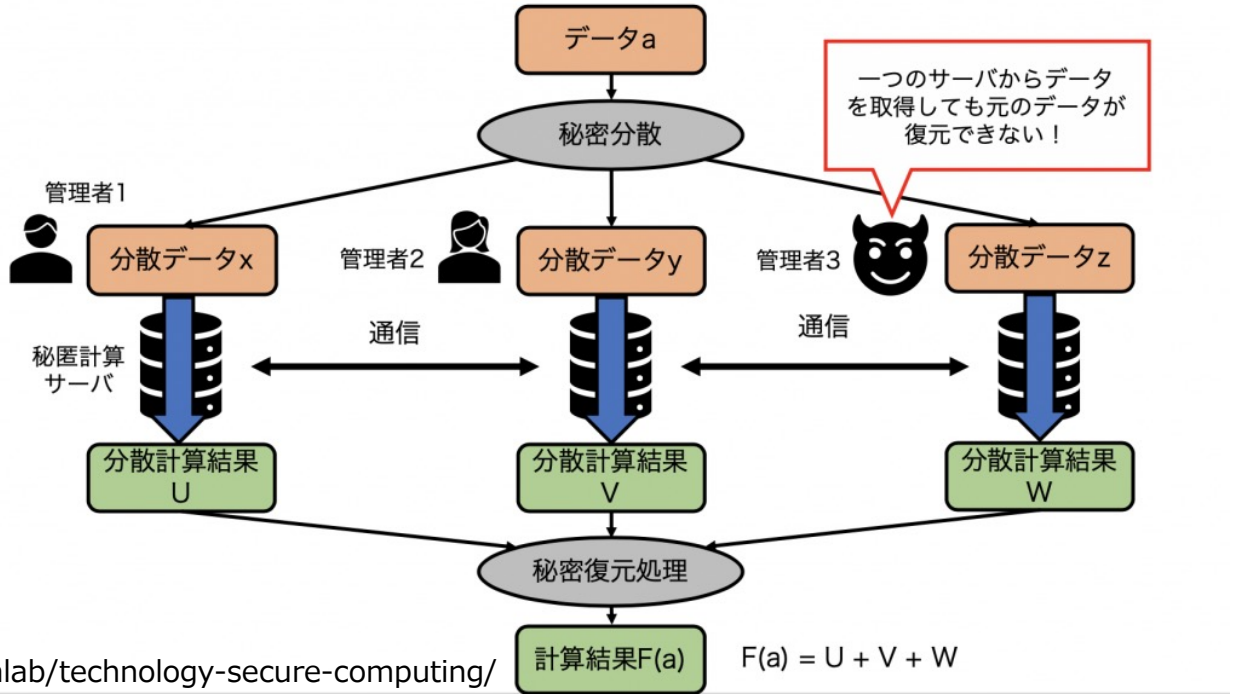
```
m = 32109, p = 4999, phi(m) = 16560
  ord(p)=690
  normBnd=2.32723
  polyNormBnd=58.2464
  factors=[3 7 11 139]
  generator 320 has order (== Z_m^*) of 6
  generator 3893 has order (== Z_m^*) of 2
  generator 14596 has order (== Z_m^*) of 2
  T = [1 14596 3893 21407 320 14915 25618 11023 6073 20
668 9965 27479 16820 31415 10009 27523 20197 2683 24089
 9494 9131 23726 2320 19834 ]

Security: 127.626
Creating secret key...
Generating key-switching matrices...
Number of slots: 24
Initial Ptxt1: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2]
Initial Ptxt2: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1]

Elapsed time for initialization: 9600[ms]

WARNING: decrypting with too much noise
Decrypted Ptxt: [4844 2247 319 4883 3790 2401 3966 1092
 1438 2549 320 1139 3046 1921 3095 1123 832 1055 703 20
09 4243 2354 886 4665]

Elapsed time for calculation: 100825[ms]
```

完全準同型暗号 (HElib):
合計110秒、しかも解が破損

秘密分散を使う？

- 秘密計算手法でよく使われる他の手法に「**秘密分散**」がある

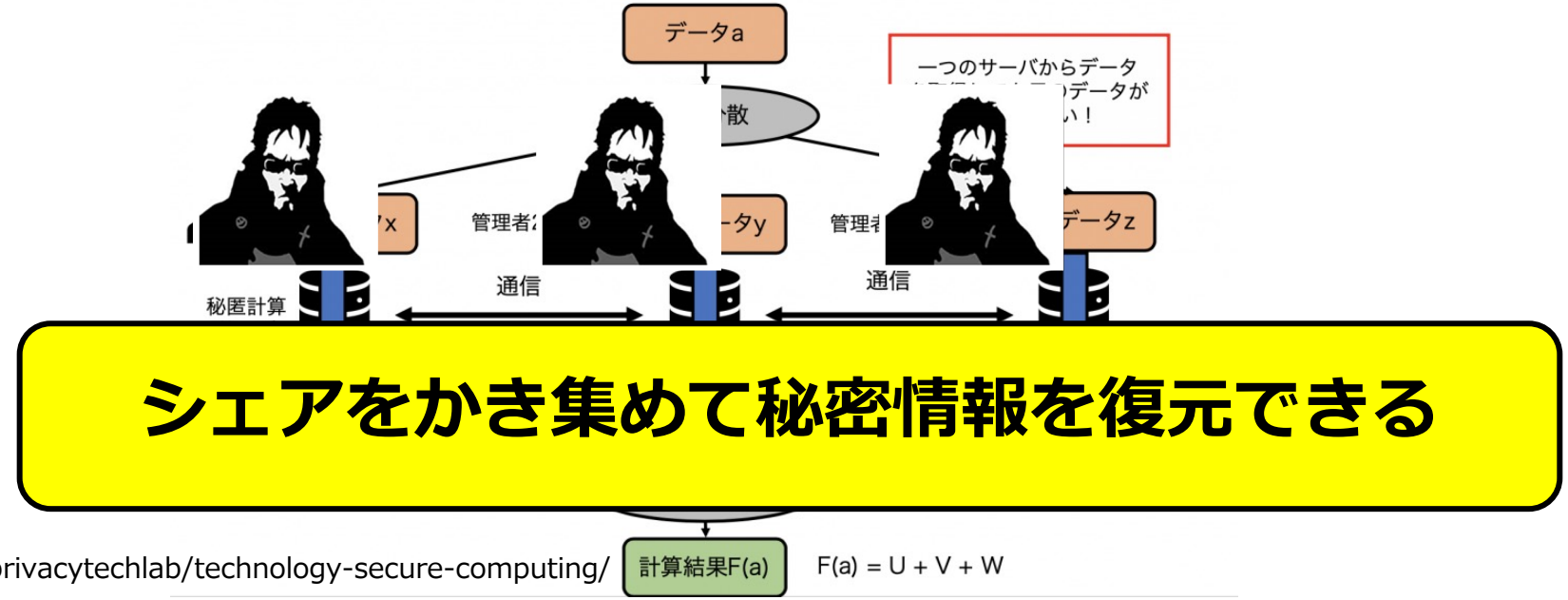
- 「シェア」という無意味化された断片に分割して計算を行う、**情報理論的安全性**を保証できる技術
  - **シェアを全て揃えない限り**はどう足掻いても秘密を解読できない



出典:https://acompany.tech/privacytechlab/technology-secure-computing/

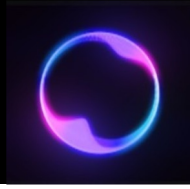
秘密分散は「陰謀論」に弱い

- 逆に言えば、分散先のサーバが**全て危殆化**している場合、
  一瞬で秘密情報が解読されてしまう
  - 全サーバが同一事業者のクラウドだったら、事業者は攻撃できるのでは？
  - 外部の攻撃者が全サーバを侵害していたらどうするのか？
    ⇒**限りなく陰謀論だが言い返せない**
  - **処理速度も決して速くはない**（通信のオーバヘッドが顕著）



出典:https://acompany.tech/privacytechlab/technology-secure-computing/

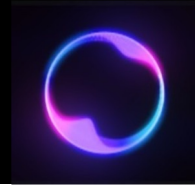
秘密計算界のダークホース



- そんな中、比較的最近登場した技術が**TEE**
  （Trusted Execution Environment; 信頼可能な実行環境）

- ハードウェアの力を借りる事で、秘密情報を保護したまま
  計算に使用できる保護領域を実現できる技術
  - 秘密計算に使えそう

TEE（信頼可能な実行環境）（1/3）

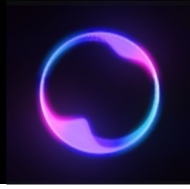


- **TEEの思想**：コンピュータリソースを、**信頼可能な領域**と**信頼できない領域**に分ける
  - **信頼可能領域でデータを扱う**事で、**データを保護しながら**の**プログラム実行**を可能とする

- **信頼可能な領域**：**信頼可能なハードウェア**及びそれにより**メモリ上に生成された保護領域**

- **信頼できない領域**：**それ以外のすべて**
  - 脅威モデル次第では、OSやVMMすら非信頼領域として扱う

TEE（信頼可能な実行環境）（2/3）

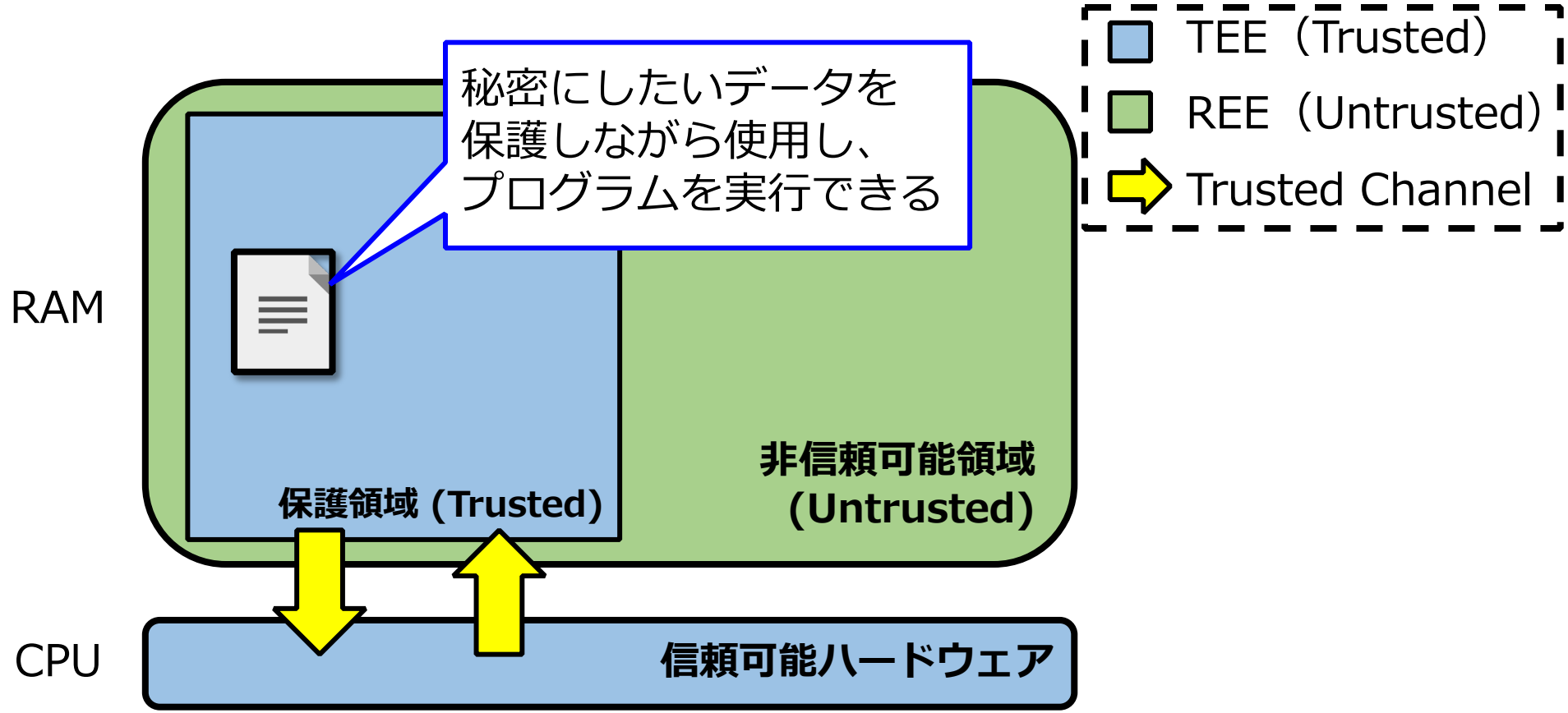


- 大分類としては、TEEは**HIEE**（Hardware-Assisted Isolated Execution Environment）と呼ばれる技術に分類される

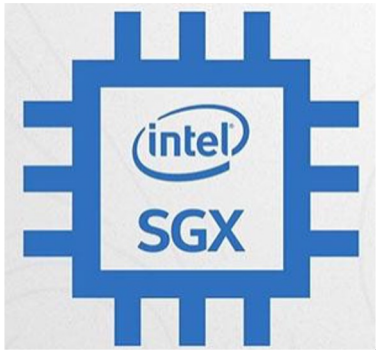
- 他のHIEE技術（Intel TPM等）と比較した場合のTEEの明確な特徴は、**保護領域内での動作**を**ユーザが定義**できるという点
  - 今回のセキュリティキャンプでのコード実装の大部分はこれ

TEE（信頼可能な実行環境）（3/3）

- より具体的な実例に落とし込むと、**信頼可能領域**は**CPU内**、CPUによって生成された**RAM上の保護領域**及びその間の**通信路**がTEEとなる



メジャーなTEE技術



Intel SGX

arm
TRUSTZONE

ARM TrustZone

ARM CCA

Keystone

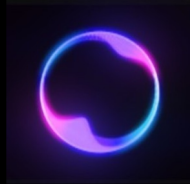
RISC-V Keystone

AMD

AMD SEV

Intel TDX

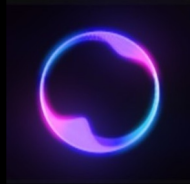
2タイプのTEE



- **部分隔離型**
  - 例：SGX、TrustZone、KeyStone
  - メモリの**特定の区画を保護**する、**本来の文脈でのTEE**
  - 保護領域内の動作定義の実装に**独特のスキルが必要**
  - **OSやVMMすら信頼しない**モデルが多い

- **VM型（全体保護型）**
  - 例：SEV、TDX、CCA
  - メモリ（あるいはVM）を**丸ごと保護**する、**TEEとしては割と異端**
  - TEE実行を想定していないプログラムも**そのまま実行可能**（**OS含む**）
  - **OSやクラウドベンダを信頼**する必要がある（脅威モデルが弱い）

TEEの暗号化

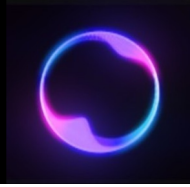


- 保護領域が**暗号化**されるかは**TEEの技術次第**

- 例えば、TrustZoneやKeyStoneは、**超特権ソフトウェア**による極めて強力な**アクセス制御**により保護領域が守られる
  - 超特権ソフトウェアの名前は前者では「Secure Monitor」、後者では「Security Monitor」
  - ここで**専用のCPU命令**により保護領域内外の切り替えが発生する

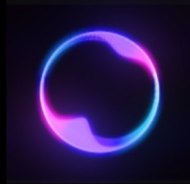
- 暗号化されていない場合、**コールドブート攻撃**等には**無力**
  - 大体は暗号化プラグインが用意されている

TEEの盲点



- Q1. TEEは準同型暗号や秘密分散より速い？
  - **YES**。CPU内での演算は**平文を用いた普通の処理**であり、秘密分散に顕著な多量の通信も発生しないため、関連技術の中では**極めて速い**
- Q2. TEEは本当に誰も信頼しなくていいのか？
  - **NO**。明らかにそのCPUの**ハードウェアベンダ**（SGXならIntel）を**無条件に信頼する必要がある**
  - 何ならハードウェアベンダによるTEEの実装に不具合があると致命的な脆弱性になり得る　⇒**本ゼミの攻撃実践パートで説明**
  - さらに言えばVM型（SEV、TDX等）に関しては**OSやクラウドベンダすら信頼する前提**となっている
    （例：悪性のOSによる悪さを覆い隠すつもりか？）

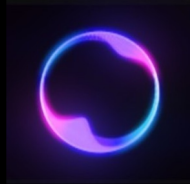
その他TEEについての議論



- 未対策では**サイドチャネル攻撃に弱い**
  - **サイドチャネル攻撃**：直接秘密情報を解読・取得するのではなく、周辺の情報（例：実行時間）から秘密情報を推測する攻撃

- 量子コンピュータ耐性はある？
  ⇒現時点の推定では恐らく当面は安全
  - AES 256bit相当以上であればまず間違いなく当面は大丈夫そう
  - SGXの保護領域はAES 128bit相当であるが、最近のNISTの見解[3]によれば128bitも同様に当面は安全そうであるため、こちらも当面は問題ないと考えられる

本ゼミでやる内容

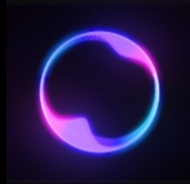


- TEEの中でも最も実世界での普及に成功している**Intel SGX**に着目し、その**基本知識**や**応用的な議論**を解説する

- SGXを用いて**秘密情報を安全に取り扱いながら処理を行う**アプリケーションを開発し、**TEEの恩恵を体感する**

- SGX、ひいてはTEEに対する**おびただしい数の攻撃**について詳細に解説し、その一端を**実践的に体験する**

本ゼミの全体像（1/4）



🔥：難易度

■§1 – TEEとは何か？ 🔥

本資料

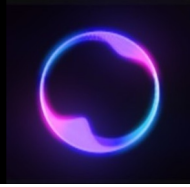
■§2 – Intel SGXの基礎 🔥🔥

SGXについてのごく基本的な概念や仕様について、比較的網羅的に解説する。

■§3 – SGXプログラミングの基礎 🔥🔥

その難易度から悪名を轟かせている、SGXSDKを用いた実際のSGXプログラミングについて、簡単な基礎部分を解説し実践する。

本ゼミの全体像（2/4）



🔥：難易度

■§4 – Sealing 🔥🔥

揮発性領域であるEnclave内の秘密情報を永続化する機能である「シーリング」について解説し実践する。

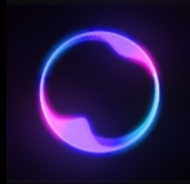
■§5 – Local Attestation 🔥🔥🔥

SGXを確実に信頼可能な状態で使用するために不可欠な検証処理であるAttestationの内、同一マシン上のEnclave同士での検証処理であるLocal Attestationについて解説する。

■§6 – EPID Remote Attestation 🔥🔥🔥🔥🔥

リモートのSGXマシンとEnclaveの検証を行うRemote Attestation（RA）の内、EPID方式と呼ばれるタイプのRAについて解説する。

本ゼミの全体像（3/4）



🔥：難易度

■§7 – DCAP Remote Attestation 🔥🔥🔥🔥🔥

同じくRAの内、DCAP方式と呼ばれるタイプのRAについて解説し、
既存フレームワークをベースとして実際に実装を行う。

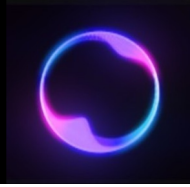
■§8 – SGX Fail 🔥🔥🔥

SGXのクソ仕様について解説し、SGXの抱える運用上の難しさに
より引き起こされた悲劇の実例を紹介する。

■§9 – SGX攻撃編① 🔥🔥🔥

SGXに対する攻撃を概観し、いくつかの比較的簡単な攻撃について
説明する。

本ゼミの全体像（4/4）



■§10 – SGX攻撃編② 🔥🔥🔥🔥🔥

他の攻撃の要素技術としても使用される攻撃や、過渡的実行攻撃と呼ばれる極めて難易度の高い攻撃を紹介し、一部については攻撃を実践する。

■§11 – SGX攻撃編③ 🔥🔥🔥🔥🔥🔥

SGXに対する攻撃の中でも極限の難易度を誇るものや、比較的最新の攻撃について解説し、SGXへの理解を極致に昇華させる。

参考文献

[1] ”【技術】TEE（Trusted Execution Environment）とは？”, 自己引用, https://acompany.tech/privacytechlab/trusted-execution-environment/

[2] “TEE (Trusted Execution Environment)は第二の仮想化技術になるか?” by Kuniyasu Suzaki, http://www.ipsj.or.jp/sig/os/index.php?plugin=attach&refer=ComSys2020&openfile=ComSys2020-Suzaki.pdf

[3]”Post-Quantum Cryptography”, NIST, 2023/7/27閲覧, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)