ISOC-JP IETF117報告会

IoTデバイスマネジメント関連

セコムIS研究所
磯部 光平 髙山 献

発表者の紹介

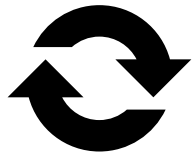
磯部 光平

- セコムIS研究所
  デジタルプラットフォームDiv.
  サイバーフィジカル
  セキュリティG. 主務研究員
- セキュアオープン
  アーキテクチャ・エッジ基盤技術
  研究組合(TRASIO)研究員
  (~2023.3)
- TEEPを中心に、
  IoTセキュリティ関連領域で活動

髙山 献

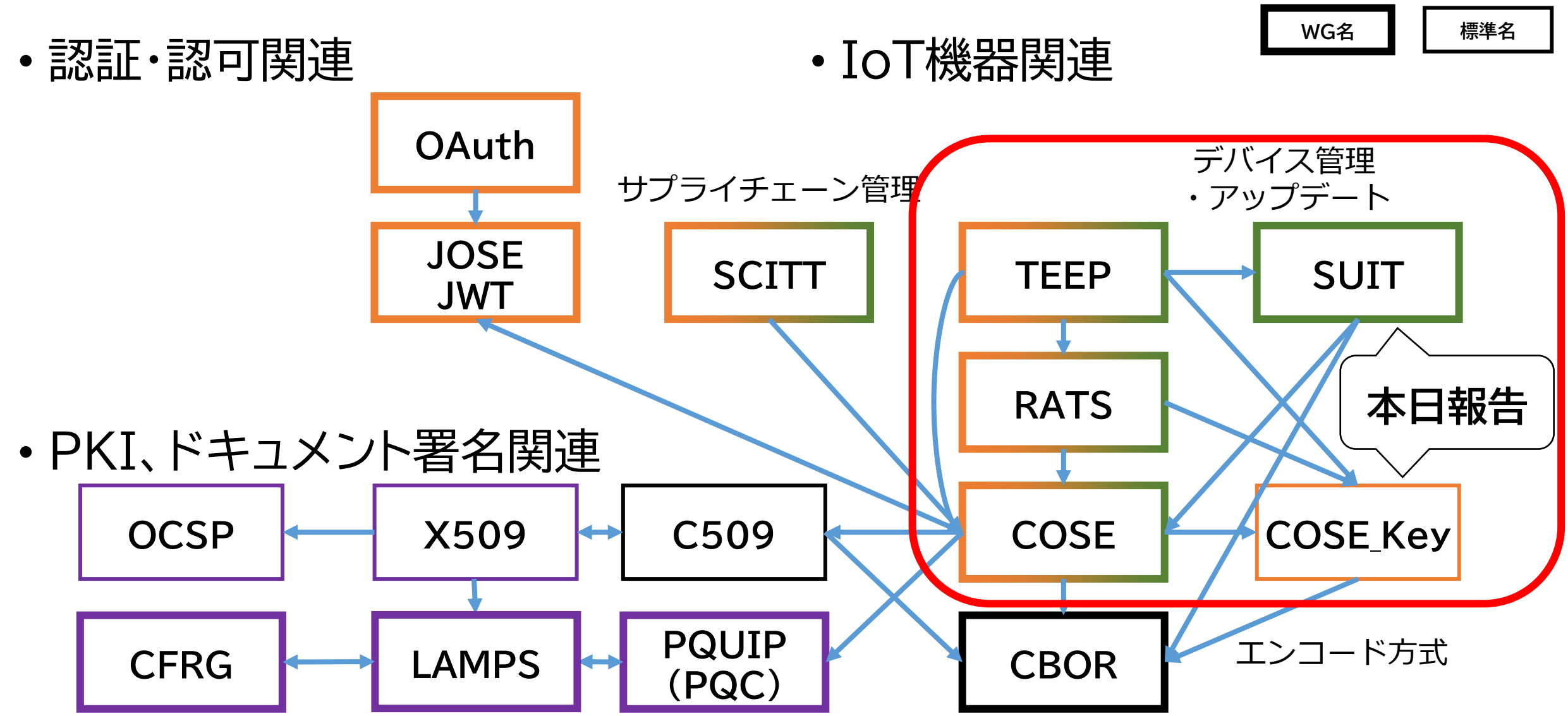
- 2019年セコムIS研究所
  デジタルプラットフォームDiv.
  デジタルシステム
  アーキテクチャG. 研究員
- 2020年よりIETFで
  IoTデバイス管理に
  関連する標準化に参画
  - OSSで提案内容を検証
  - SUITのInternet Draft共著

ドキュメント執筆



OSSでの検証

IoTデバイス・トラストに関連する標準



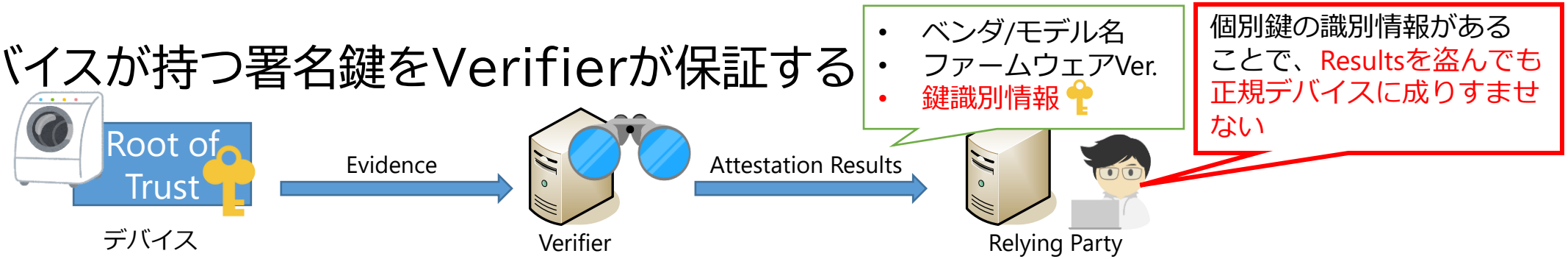
TEEP（Trusted Execution Environment Provisioning）

- 何のWG？
  - TEE(メモリ隔離によるセキュア実行環境)へのアプリ配信プロトコル
  - TEEはスマホ、IoT、クラウドなどで活用が進む
- RFC 9397 TEEP Architecture リリース
  - TEEPによる配信の概要、ユースケース、エンティティ、役割を示す
- この他のアップデート
  - TEEP ProtocolはWGLCへ
    - 配信サーバと配信先デバイス間の通信メッセージを規定
    - 配信サーバへのアテステーション拡張や、鍵識別子の参照を追加
  - Confidential VMへの適用を念頭としたI-Dが提出
    - CATSにつぐユースケース？

COSE_Key Thumbprint

（磯部共著）

- 背景
  - 暗号鍵パラメータに基づいて、識別子となる値の計算方法が欲しい
  - 例:デバイスが持つ署名鍵をVerifierが保証する



  - JWK ThumbprintやX.509のKey Identifierを流用することもできるが、JSONやX.509実装をわざわざ追加する羽目になる
- 提案
  - COSE Keyオブジェクトに対するThumbprint算出方法
    - 基本は暗号鍵パラメータを持つCOSE Key オブジェクトをハッシュ関数に入れるだけ
  - オブジェクトのブレを防ぐために、Deterministic Encodingを必須化

COSE_Key Thumbprint

（磯部共著）

- WGでの議論
  - WG Adoptionは合意
  - 対称鍵やCWTオブジェクトのサポートがリクエストされる

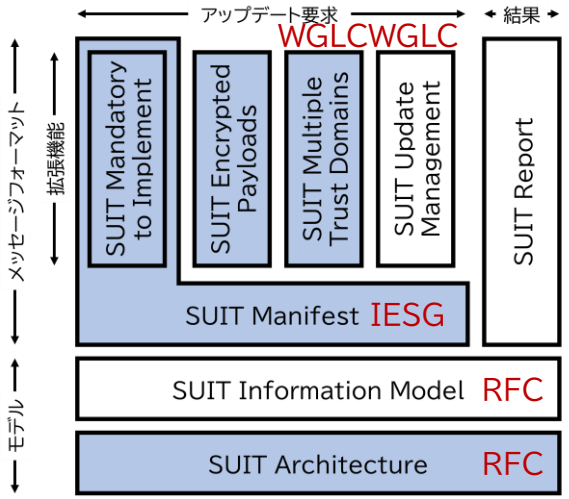
draft-ietf-cose-key-thumbprint-01 - COSE Key Thumbprint

RATS （Remote ATtestation procedureS）

- 何のWG？
  - リモートアテステーション(ネットワーク経由での端末の健全性検証)の標準化
  - 伝送用メッセージフォーマットEAT(Entity Attestation Token)など規定
- 今回のアップデート
  - EndorsementのWG Adoption
    - Endorsement: 例:デバイスメーカーが提供するデバイス情報
    - RecharterしてRATSの標準化スコープに入った
  - -02版にアップデート
    - Conditional Endorsement
    - Endorsing Identity
    - Multiple Endorsement
  - Endorsing Identityはさらなる改版が必要という結論に至り、今回はWGadoptせず。
    - Identity ≠ Key Materialな点は整理すべき

SUIT（Software Updates for Internet of Things）

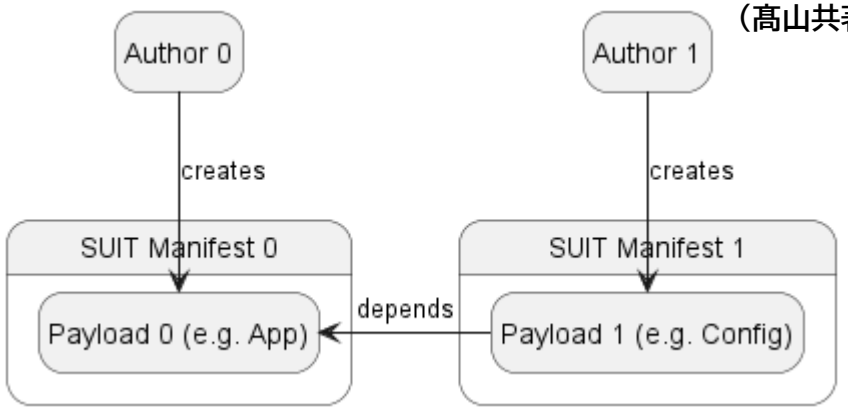
- 何のWG？
  - IoTデバイスでも利用可能なファームウェアアップデート方法の標準化
    - 特にRAM 10KB 、FLASH 100KB程度のハードウェア制約の強いデバイス
  - TEEのアプリケーション配信にも利用可能
- SUIT Manifestドキュメントは IESGの出版待ち
  - ファームウェア配信のためデータフォーマット仕様を規定（コア機能）
- 拡張するドキュメントも一部落ち着きつつある
  - WGLC中：SUIT Multiple Trust Domains, SUIT Update Management, SUIT MUD
  - WGLC間近？：SUIT Encrypted Payloads, SUIT MTI
  - まだまだ仕様策定中：SUIT Report



SUIT Multiple Trust Domains

- 背景
  - 複数の人/組織で1つのデバイス内を管理したい
- 提案
  - 複数のSUIT Manifestの
    依存関係を記述できるようにする
  - 公開鍵のチェーンを作ることで他者の公開鍵も追加可能
- 状態：SUIT WG内でLast Call済
  - Last Callで受けたコメントを反映　#6, #9
  - 表記揺れ、記述漏れ、誤記などを修正　#4, #7, #8, #10, #11, #12, #13, #14
  - 文書だけだったドキュメントに、髙山が作成したサンプルを追加　#5

（髙山共著）



SUIT Encrypted Payloads

（髙山共著）

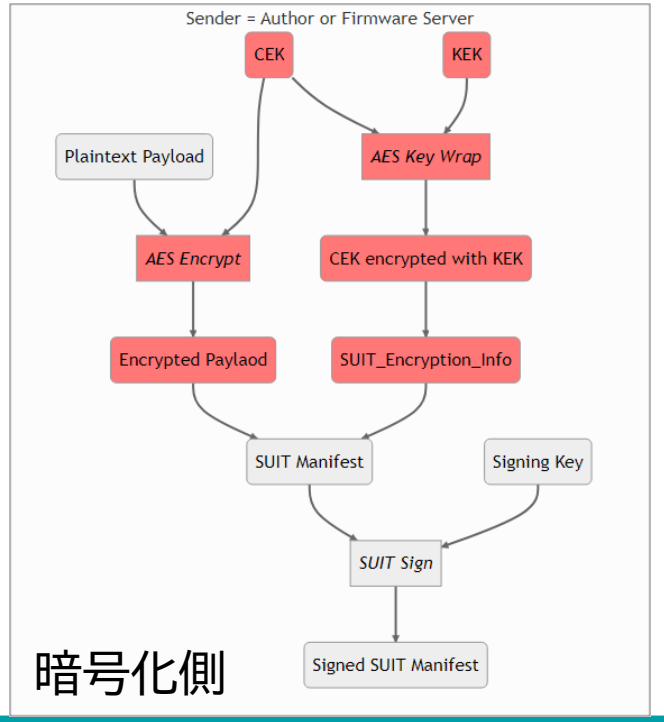
- 背景
  - 暗号化したままのペイロード(ファームウェア等)を送りたい
  - しかし復号鍵を平文で送るべきではない、しかしHTTPSやVPNは重い
- 提案
  - CEK (Content-Encryption Key) を送信者・受信者で鍵共有する
  - そのためのパラメータ等を SUIT_Encryption_Info に封入
    - 共通鍵暗号を使うAES-KW、公開鍵暗号を使って鍵交換するECDHの2方式を利用
    - 将来的にはHPKEも利用可能にしたい(が時間がかかりそうなので削除)
- 状態:執筆中、実装による検証中
  - IETF117ハッカソンでECDHを実装(AES-KWは実装済みだった)

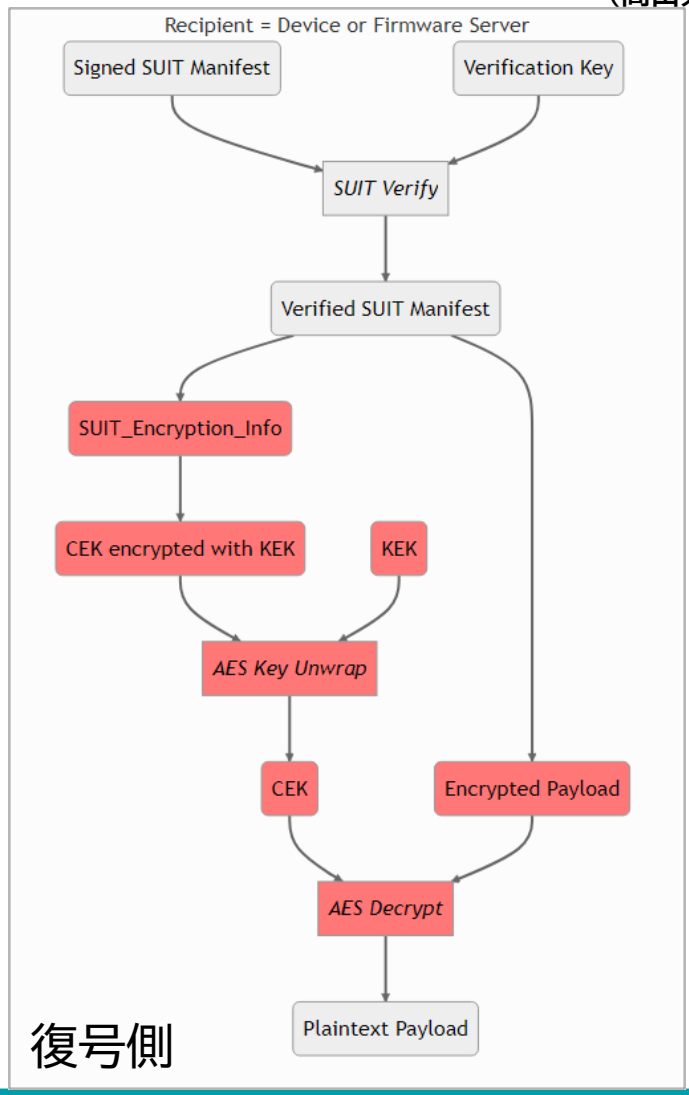
SUIT Manifestの作り方

（髙山共著）

- 通常のSUIT Manifestの作り方に以下を追加
  - CEKとKEKからSUIT_Encryption_Info作成
  - CEKとPlaintext PayloadからEncrypted Payload作成

Encrypted Payloadの作成と復号処理のために赤背景部分を追加



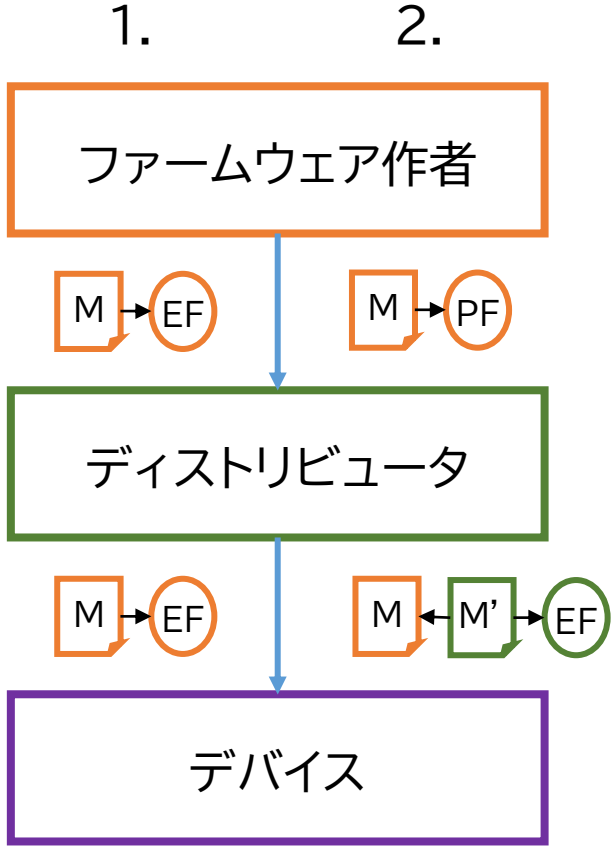


議論：誰が暗号化するべき？

1. ファームウェア作者（Author）
   - 👍ディストリビュータから平文ファームウェアを隠せる
   - 👎デバイスに合わせて動的に暗号化することは困難
     - 機種ごとに同じ暗号鍵を使うのが現実的？
2. ディストリビュータ（Firmware Server）
   - 👍デバイスに合わせて暗号化も可能
   - 👎ディストリビュータには平文ファームウェアが見える
     - TEEPの配信サーバーの場合はPersonalization Dataも見える

➢どちらのやり方にもメリット・デメリットがあるので
読者の選択を補助するために明記することに

（髙山共著）



まとめ

- 磯部と髙山でIoT機器関連の標準を見ています
- 不足しているものを提案する標準ドキュメントに共著で入りました
  - 磯部：COSE_Key Thumbprint
  - 髙山：SUIT Multiple Trust Domains, SUIT Encrypted Payloads
- TEEP・SUITともにドキュメントの執筆が落ち着きつつある
  - アーキテクチャなど用語のコアになるドキュメントはそれぞれRFC化された
  - 通信プロトコル、メッセージフォーマットなどのドキュメントはRFC化間近