信頼される安心を、社会へ。
SECOM

ISOC-JP IETF113報告会

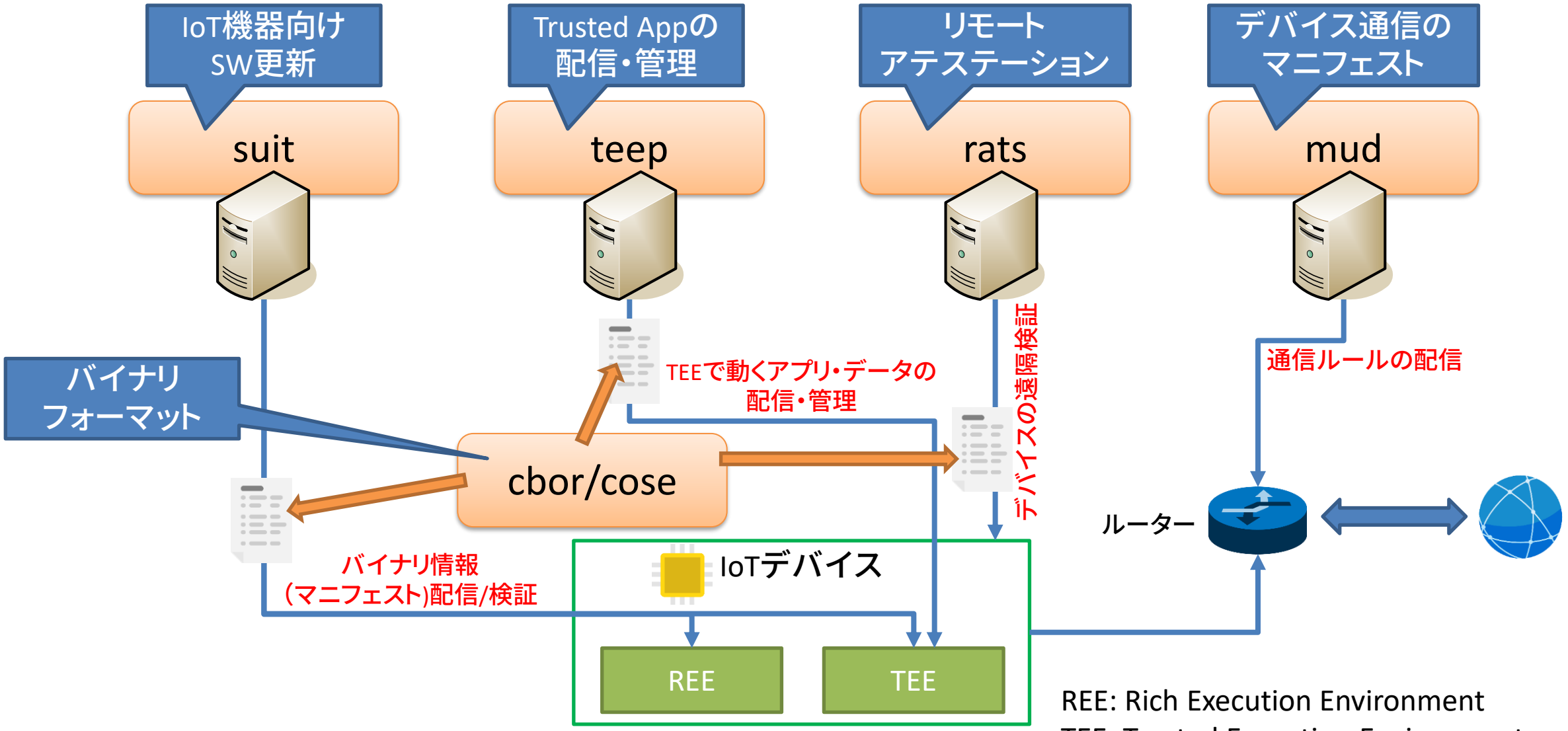
IoTデバイスマネジメント関連

セコム IS研究所 / TRASIO
磯部 光平
ko-isobe@secom.co.jp

自己紹介

- 磯部 光平
- 略歴
  - 2016年 セコム IS研究所
    コミュニケーションプラットフォームDiv. 暗号・認証基盤G.
  - 2020年 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
    IoT向けエッジセキュリティ研究に従事
- 研究領域
  - 暗号利用システム、デバイス管理システム、PKI

デバイス管理に関連するプロトコル



REE: Rich Execution Environment
TEE: Trusted Execution Environment

TEEP

https://github.com/ietf-teep

信頼される安心を、社会へ。
SECOM

- TEE(Trusted Execution Environment)環境への
  アプリ・データ配信方法を規定
  - TEEP architecture – TEE搭載デバイスやTAM(配信サーバ)などを規定
  - TEEP Protocol – TAMとTEE搭載デバイス間のメッセージングプロトコル
- TEEP Architecture
  - AD Review(Ben)へ対応
  - End User/ Trust Model
    - Privacy Consideration / Security Considerationの兼ね合い(Double Edge Sword)
    - デバイスの前にいるのは”エンドユーザ“(RFC8890)なのか、管理者なのか。
      デバイスオーナーとは?
    - エンドユーザに対する透明性/アテステーション
  - REEからのDoS攻撃や
    バイナリ配布パターンを網羅する記述追加
  - Transport Draftも上記に合わせ記述追加

TEEP

- Hackathon Report
  - RATS Background Modelのサポート
  - EATのMessage Typeを表現できない問題
  - SUIT/EATとの必須・推奨暗号化アルゴリズムでの互換確保
- TEEP Protocol
  - Ciphersuiteの簡潔化。HSS-LMSの追加
  - TEEPにおけるEAT Profileの提案
  - EvidenceをコピーしてAttestation Resultを作るという記述...
    Verifierが精査したのかどうかわからないとまずい
- Confidential Computing in Computing Aware Network
  - CAN BoFに関連した提案
  - (提案者の)TEEの使い方が???

RATS

- リモートアテステーションに係る標準化
  - Rats Architecture – エンティティ/モデルなどを整理
    Attester=>Verifier=>Relying Party
  - EAT – Entity Attestation Token
    EvidenceやAttestation Resultの伝送フォーマット
- Recharter
  - Evidenceに加え、Attestation Resultの伝送プロトコルも
    スコープに追加
    - 既存プロトコルの採用を前提に
  - Endorsement, Reference Valueのフォーマットも
    スコープに追加

RATS

- EAT (Attestation Result(AR) Framing)
  - ARはシステム全体の安全性を表現すべきではないか
    - ARは検証専用HWから、SWのみのアプリまでカバーできている
    - →システムの安全性などの意味付けは検証者のポリシーに依存する。
      ARはあくまでEvidenceに対応するものでしかない
    - 絶対的なセキュリティレベルの規定もできない
  - Device Identifierの標準化
    - （絶対的な）Device Identifierを定めるべきではないか
    - →UEIDやIMSIなど現状のIdentifierは収容済み。分野ごとにID文化は違うし、
      現状の拡張可能な形式でIdentityを表現できる。
  - Claimsのパススルー
    - 機械学習ベースのVerifierではすべてのEvidenceの提供が有用。
      ARでこれを可能とすべき
    - →Attester/RPのネスト関係は許容されている。
      Attester, RPはRoleであり、実体ではない。
      AttesterとRPは同居していいし、その範疇で表現できる

SUIT

- IoT機器向けソフトウェア更新スキーム
  - Suit manifestによる更新方法やリソースのありかの定義
  - Suit Report：更新結果の報告

- Hackathon
  - ARM HannesがEncrypted Firmwareの実装を実施
- SUIT Manifest Format
  - PQC(耐量子計算機暗号)対応としてHSS-LMSのMTI(Mandatory to Implement)追加。
    - vs Falcon?　比較は継続に
    - ブートローダにも現時点でPQC実装を強制すべきか→厳しいのでは
  - Crypto Agility
    - MTIや暗号アルゴリズムの指定はManifestのドラフトから分離すべきでは

SUIT

信頼される安心を、社会へ。
SECOM

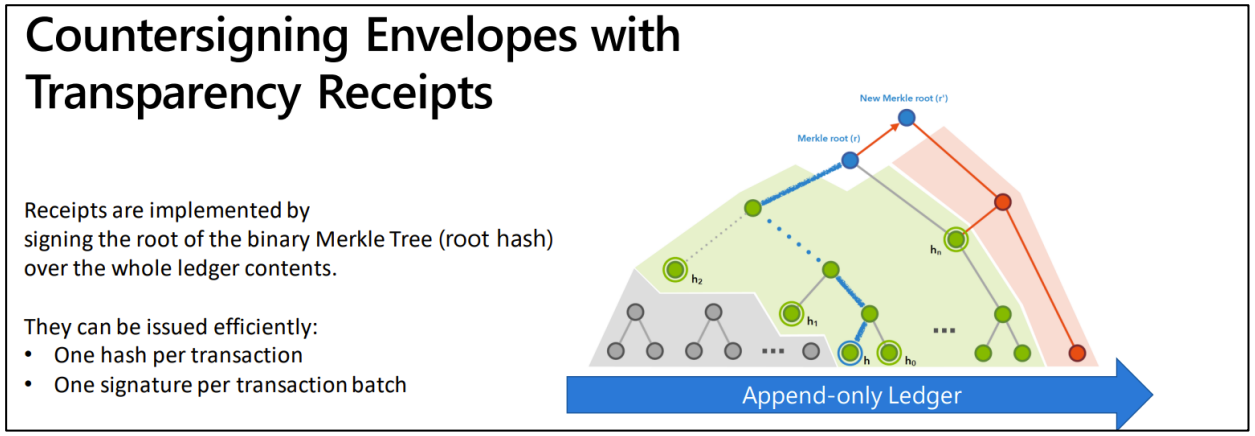
- SUIT Trust Domain(Dependencyの整理)
  - 実装フィードバック待ち
- Firmware Encryption(AES-KW+HPKEでFWの暗号化)
  - HPKEのRFC化は完了。COSE-HPKEの更新待ち
  - 認証付暗号用の拡張が提案されるも、COSEでの対応に寄せるべきとの意見
- SUIT-related Claims(FWアップデート状況をアテステーションクレームとする)
  - RATSへの取り込みは失敗。RATSで規定済みのものと重複があるため
    - 実装上の理由でSUIT寄りの構造を採用していると主張
  - SUIT ReportをEvidenceとしVerifierがAttestation Resultに変換できるようにする提案
  - EATに対し埋め込み、変換、そのまま使う、分けて使うなどをMSのDaveが提案
    - EATへの埋め込みなどはTEEPでのアテステーションタイミングに影響があるとも発言

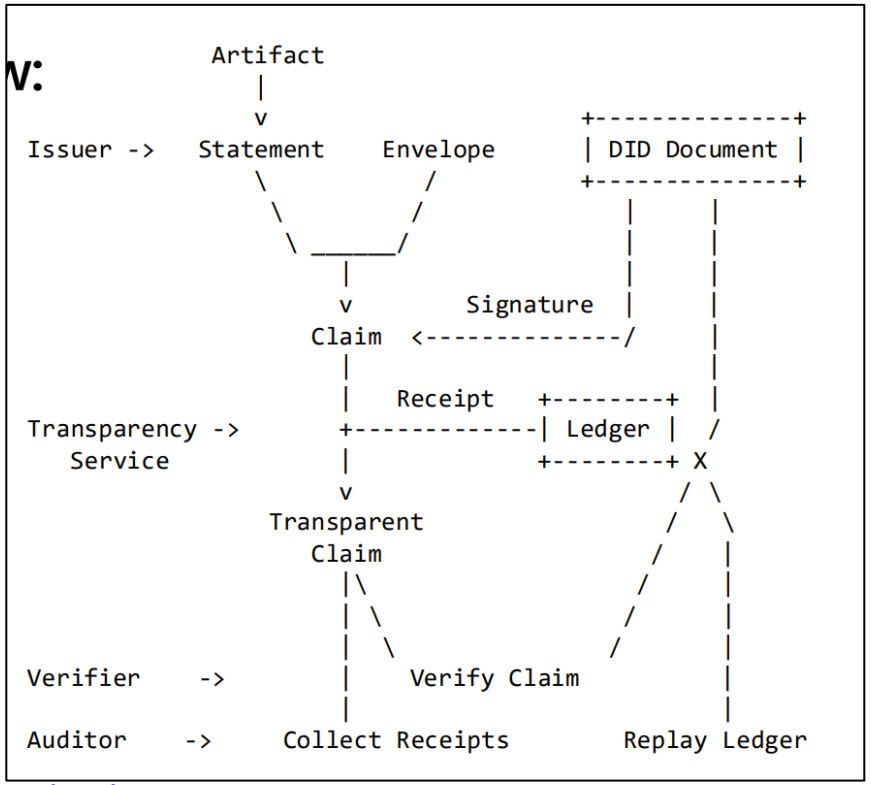
Open Issues

- No running code for:
  - Delegation
  - Multiple Manifest Processors
  - Integrated Dependencies
- Is the use of CWTs correct?
- Contributors welcome!
  - Especially from TEEP!

secdispatch

- Trustworthy and Transparent Digital Supply Chains
  - Verifiable Claimsを使い、出元を検証可能にするアーキテクチャの提案
  - Certificate Transparencyと違う点：
    ・Transparency LedgerにClaimのReceiptを保存
    ・Receipt：Claimに対するカウンター署名
    ・COSE・DIDベース
    ・スケーラビリティのためReceiptはMerkle-Treeを使い、Ledgerに格納





https://datatracker.ietf.org/meeting/113/materials/slides-113-secdispatch-trustworthy-digital-supply-chain-transparency-services-00

secdispatch

信頼される安心を、社会へ。
SECOM

- Trustworthy and Transparent Digital Supply Chains
  - MSのConfidential Computing Frameworkを使った実装例
  - リアクション
    対処が必要な課題ではある
    IETFで取り扱うには大きい
    スコープの絞り込みが必要
    BoFはやったほうがいい