自動車における鍵の利用と管理

エンベデッドグループ Linuxチーム

竹内 雅紀
川崎 貴弘

1人の満足から、社会の満足へ

株式会社大阪エヌデーエス

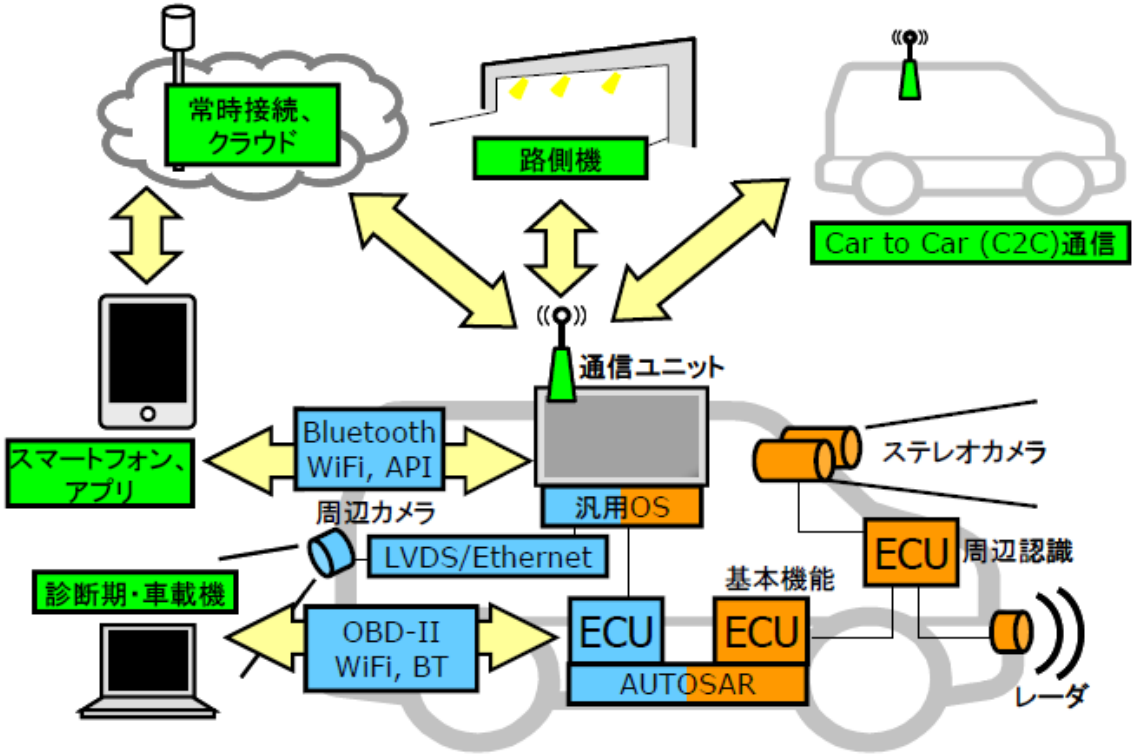
プロフィール

- 川崎 貴弘
  - 2004年 株式会社大阪エヌデーエス入社
  - 携帯電話の開発や、家電製品の組込み開発を経験
  - 最近はLinux関係の仕事を行い、今はSecurityを担当
  - 過去の趣味は車(Key words : テンロク、NA、直４、６速MT)
  - 最近の楽しみは娘と遊ぶこと

コネクティッドカー

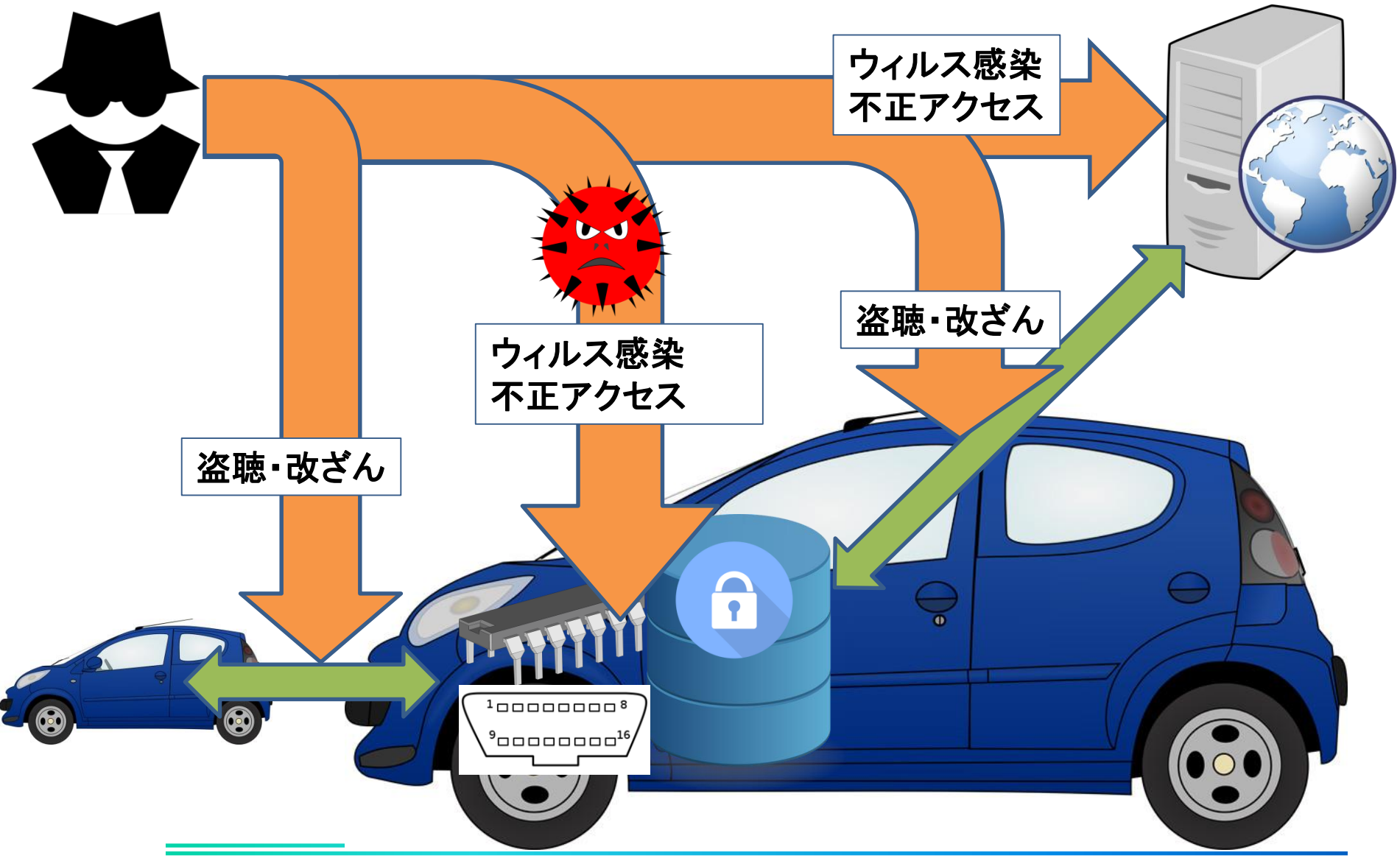
- 近年の車は様々なものと繋がっている
  - スマートフォン(Bluetoth, WiFi)
  - 車車間通信、路車間通信 (V2X通信)
  - センサ、アクチュエータ
  - データセンタ (LTE)
  - 充電ステーション
  - ダイアグ機器
- 守るべき情報資産は増えている
- ドライバー、同乗者、周囲の人の人命にも関わる

自動車のネットワーク接続



参考：https://www.ipa.go.jp/security/fy23/reports/emb_car/index.html
IPA 2011年度 自動車の情報セキュリティ動向に関する調査

自動車に想定される脅威



有効と考えられる対策

- 機器内部データの暗号化
- 不正プログラムの動作防止(ホワイトリスト制御、ソフトウェア署名)
- 機器の分解対策(耐タンパー)、データのセキュアな消去
- 通信路の暗号化
- ODB-IIポートにおける脆弱性対策、認証の強化、DoS対策
- インターネット上のサーバにおけるデータの暗号化、脆弱性対策、認証の強化

参考：https://www.ipa.go.jp/security/iot/iotguide.html
IPA IoT開発におけるセキュリティ設計の手引き

鍵を使った対策

- 鍵を使うと、データの暗号化、データの保証、ユーザ認証などができる
- データの暗号化
  - 通信データ：通信内容を隠蔽する
  - ソフトウェア：アプリ等を隠蔽する
    - セキュアな環境下でアプリを実行する
  - 情報資産データ：個人情報等を隠蔽する
- データの署名
  - 通信データ：送信元の保証
  - ソフトウェア：不正なプログラムを実行させない
    - セキュアブート
  - 情報資産データ：データが書き換えられていないことを保証
- 認証
  - アクセスできる人を限定する

プロフィール

- 竹内 雅紀
  - 1974年大阪市で生まれる
  - その後、幼稚園、小学校、中学校、高校、大学と大阪市内の学校へ通う
  - 1996年 株式会社大阪エヌデーエス入社
  - ほとんどの期間を組み込み関係の開発に従事
  - 主に家電製品、携帯電話、車載製品など、組み込み以外では官庁のシステム開発
  - 現在はLinux関係の仕事を行い、Securityを担当
  - 趣味は食べ歩き(ラーメン、肉)、ゲーム、ギャンブル少々(競馬⇒パチスロ)

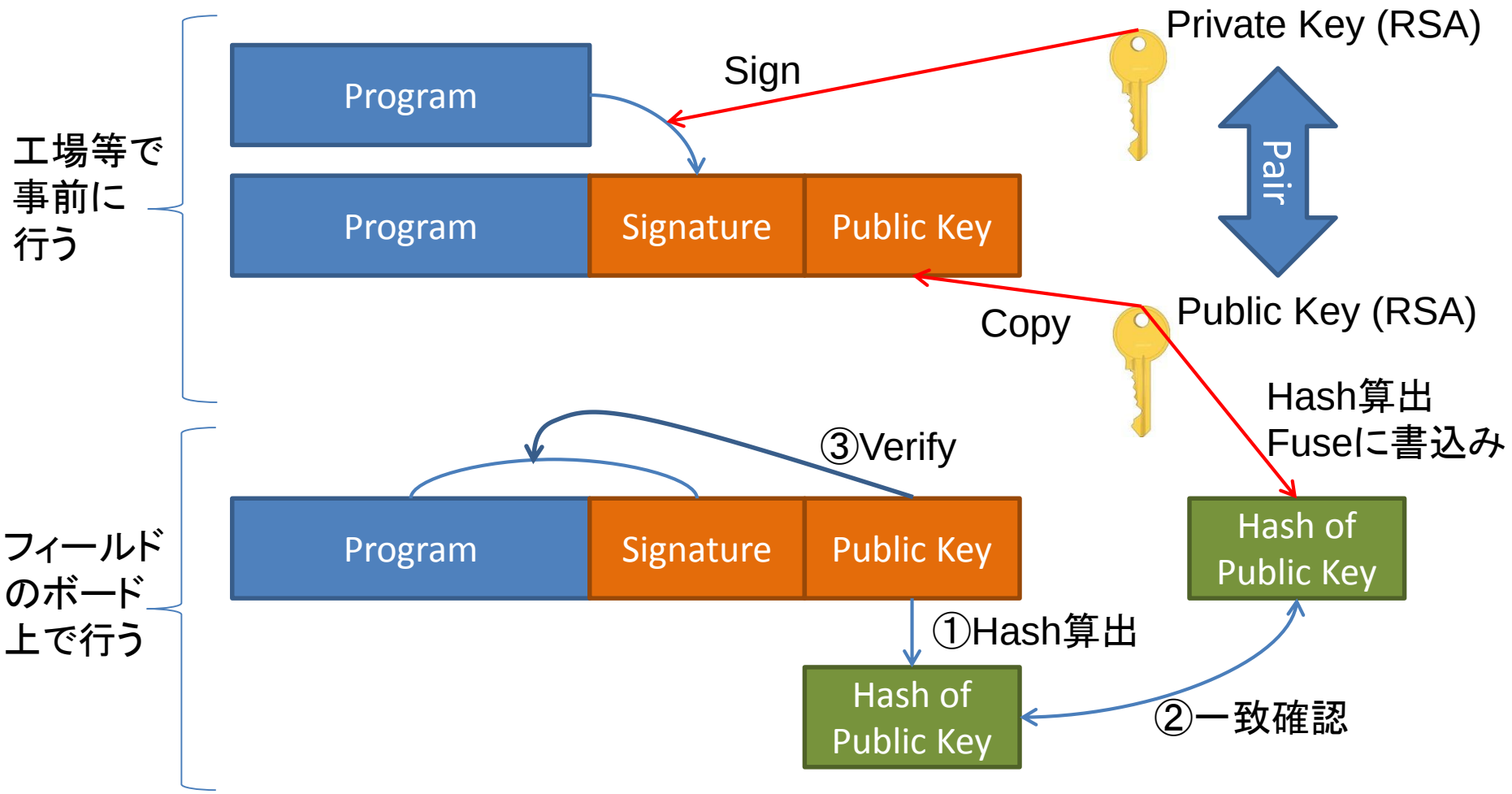
セキュアな環境

- セキュアな環境とは？
  - 不審者がアクセスできない
  - 不審者による情報の読み書きができない
  - 不正なプログラムが実行できない
- セキュアなアーキテクチャ
  - 代表的なものはARMのTrustZone
  - ARM以外でも当然存在します。

Secure Boot

- 出どころのわからないプログラムでは起動させない仕組み
- 主に非対称のRSA鍵が使用される
- 製造者が非対称鍵の秘密鍵を持っていて、作成したプログラムに署名を行う
- ボード上では対応する公開鍵を使って、署名を確認してからプログラムを起動する
- 公開鍵のハッシュ値は、ボードのFUSEなどに焼かれていて、秘密鍵と公開鍵がペアであることを保証する

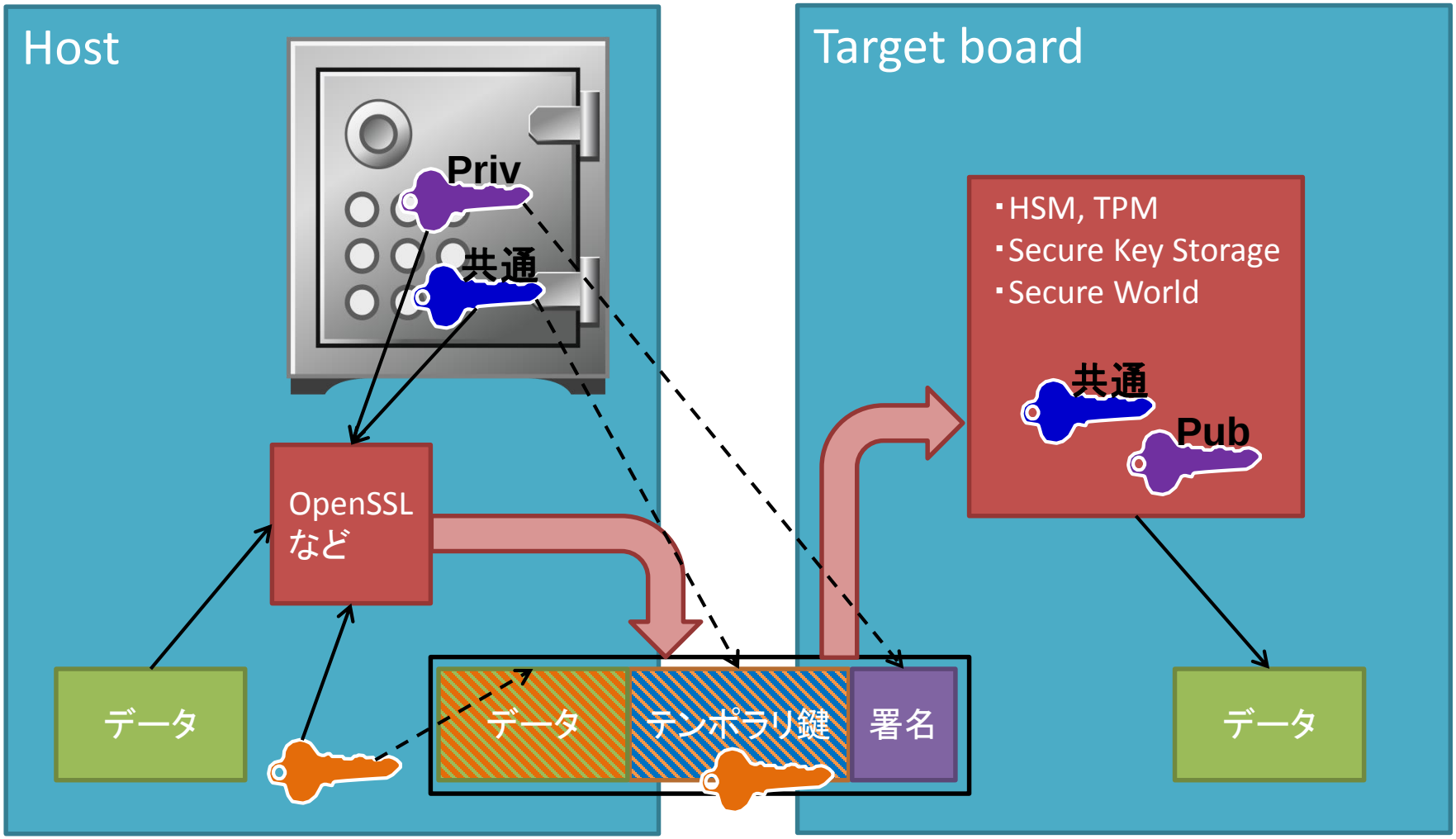
Secure Boot 図解



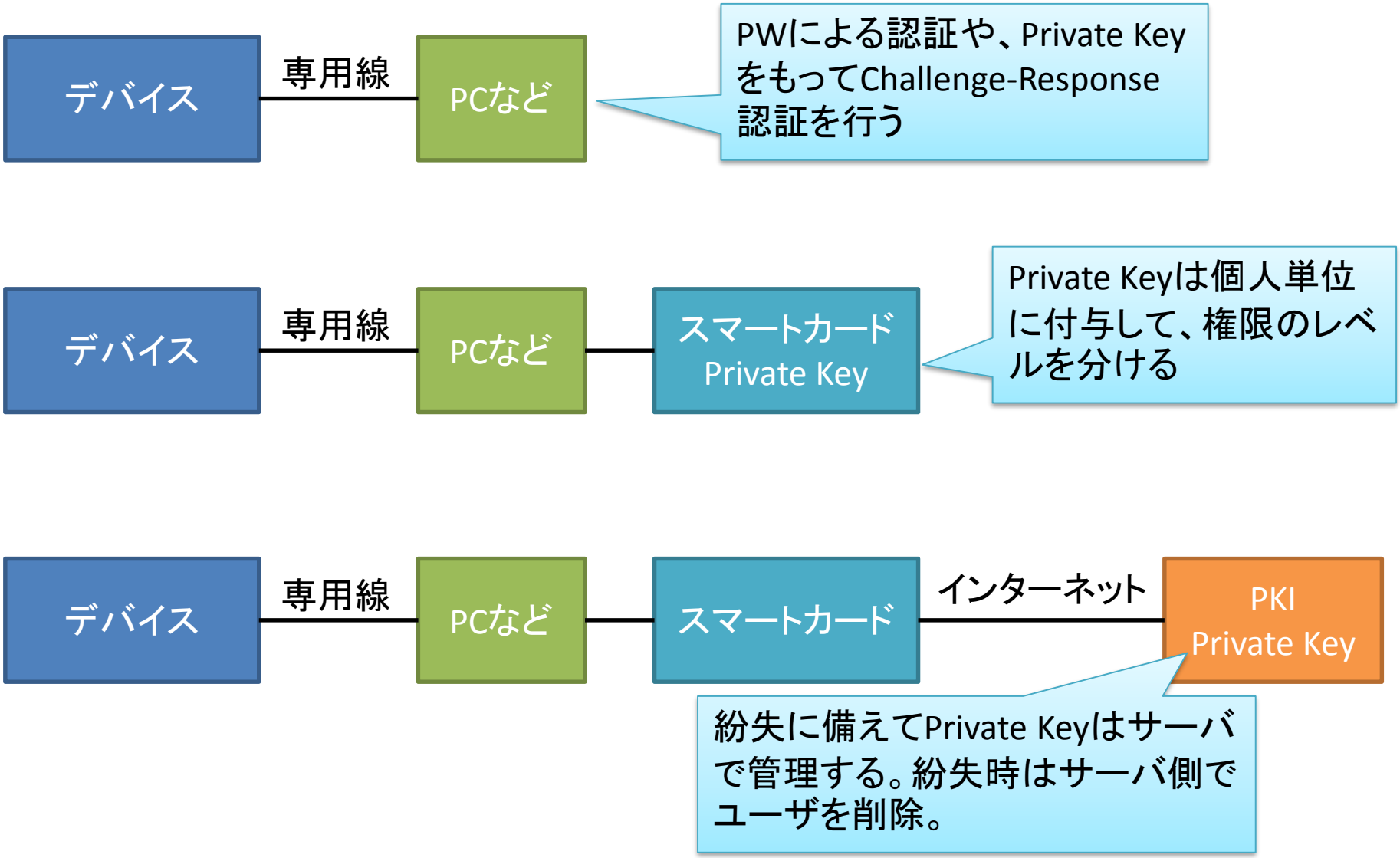
Secure Software Update

- ホスト側で準備された更新データをターゲットに渡すには、データの暗号化と署名は必要
- データの復号と署名確認をする鍵は、あらかじめターゲットの中にセキュアに保存しておく
- 仮に鍵が漏えいした時の被害を最小にする仕組みも必要
- ターゲット毎、セッション毎に鍵を変えるとセキュリティは高くなるが、運用コストは上がる

Secure Software Update



鍵を使った認証



AES, RSAの強度(将来性)

Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Elliptic Curve	Date
80	2TDEA	1024	160	Legacy
112	3TDEA	2048	224	2016-2030
128	AES-128	3072	256	2016-2030 & beyond
192	AES-192	7680	384	2016-2030 & beyond
256	AES-256	15360	512	2016-2030 & beyond

参考：https://www.keylength.com/en/4/
Cryptographic Key Length Recommendation(NIST)

Secure Key Storage

- 使用する鍵は改ざんされないように保管する必要がある
- 鍵の保管
  - 共通鍵暗号(Symmetric Cryptography)を使用している場合
    - 共通鍵は完全性及び機密性を満たす条件の下で保管すること
    - 永続的に使用する共通鍵は、耐タンパー性を有するHW(例：暗号専用回路を持つIC)の内部で保管することが望ましい
  - 公開鍵暗号(Asymmetric Cryptograph)を使用している場合
    - 公開鍵は完全性を、秘密鍵は完全性及び機密性を満たす条件下で保管すること
    - 秘密鍵は、耐タンパー性を有するHWの内部で保管することが望ましい
- 鍵のアルゴリズムと鍵長
  - 暗号化アルゴリズム(共通鍵暗号)を使用している場合
    - 128bit以上の暗号鍵を選択すること
    - 安全なアルゴリズムを選択すること
    - CRYPTREC暗号リストの「電子政府推奨暗号リスト」に掲載された暗号利用モードを採用することが望ましい
    - 共通鍵暗号としてブロック暗号を採用する場合、CRYPTREC暗号リストに掲載された暗号利用モードを採用することが望ましい

Secure Key Storage

❖ 鍵のアルゴリズムと鍵長(続き)

❑ 電子署名アルゴリズム(公開鍵暗号)を使用している場合

- 安全なアルゴリズムを選択すること
- CRYPTREC暗号リストの「電子政府推奨暗号リスト」に掲載された公開鍵暗号を採用することが望ましい
- 有限体上の離散対数問題または素因数分解問題に基づく公開鍵暗号の場合、鍵長2048ビット以上(西暦2030年まで)、または鍵長3072ビット以上(西暦2031年以降)の暗号鍵を選択すること
- 楕円曲線上の離散対数問題に基づく公開鍵暗号の場合、鍵長256ビット以上の暗号鍵を選択すること

参考https://www.ipa.go.jp/security/iot/iotguide.html
IPA付録C：IoTにおける暗号技術利用チェックリスト
参考 http://www.cryptrec.go.jp/list.html
電子政府推奨暗号リスト

実際に困ったこと

- パフォーマンスの問題
  - 実行時のパフォーマンスが要求を満たさない
- セキュアな方法を考えても、現実的にその環境が準備できない、運用までつなげられない
  - ユースケースに合わない
  - PKI(Public Key Infrastructure)サーバなどが必要
- 製造工程で事前にデバイスに行うことが煩雑になる
  - 鍵の生成やインストール
  - 鍵を間違う等、何か1つ足りないと動かないので問い合わせが増える
- セキュリティを高めるほど、パフォーマンスは低下し、手順や運用コストが増えるので、後回しになりがち
- 日本語と英語で表現が違う
  - 共通鍵暗号方式と公開鍵暗号方式は、Symmetric CryptographyとAsymmetric Cryptography。Common KeyとかPublic Keyというと、非対称鍵のPublic KeyとPrivate Keyと勘違いされる

まとめ

- データの漏えいや改ざんを防ぐには、鍵の有効利用が重要となる
- 使用される鍵の漏えいや改ざんを防ぐ仕組みも重要である
- 現在解読が難しい鍵アルゴリズムであっても、将来解読される可能性があるので、用途に合わせて、より安全なものを選択する必要がある
- ターゲットの種類により、HSM(Hardware Security Module)やセキュアな実行環境の提供方法が異なるので、ボードに合わせてどのように鍵を管理・運用するか考える必要がある