Vol.2019-CSEC-86 No.43
Vol.2019-SPT-34 No.43
2019/7/24

ブロックチェーンを用いたIoTシステム向け証明サービス基盤の提案

大久保 隆夫[1,a)]　田嶋 健[2]　上原 敏幸[2]　牧野 進二[3]

概要：本稿では，IoT システムなどを対象とした，ブロックチェーン技術に基づく証明サービス基盤 kusabi を提案する．従来の証明書サービスでは，公開鍵の真正性の保証のために証明局 (CA) が発行した証明書を必要とし，コストや期限切れによる更新などの作業が必要であり，低コストや M2M による自動的な機器間のインタラクションを前提とする IoT システムの導入には障害となることが多かった．kusabi では，CA の代わりにブロックチェーンを用いて公開鍵を保管することで，証明書そのものを不要にした鍵管理を可能にする．筆者らは脅威分析により網羅的に脅威を抽出し，提案する kusabi 基盤が，CA 証明書を用いた PKI と同等の安全性 (鍵の真正性，完全性) であることを示す．また筆者らは，機器の設置，ファームウェア更新にも kusabi の PKI を導入した．これにより，耐タンパー技術や TPM を実装できない非力な IoT 機器で構成されたシステムにおいても，一定の安全性を確保できることを示す．

Certificate Service Infrastructure using Blockchain for IoT Systems

1. はじめに

近年，組込み機器や Internet of Things (IoT) のセキュリティが問題となっている．問題の中には，ハードウェアなど組込み特有の対策が必要なものもあるが，その多くは従来の IT などで使われてきたセキュリティ要素技術の転用が可能である．ネットワーク通信における中間者攻撃対策としての公開鍵暗号基盤 (PKI) もその一つであり，PKI を採用することで，認証，暗号化，および署名による改ざん検出が可能となる．しかし，現実には PKI が IoT システムに十分に浸透しているとは言い難い．その原因の一つは，従来の PKI が公開鍵証明書認証局 (Certificate Authority, CA) が発行する証明書を必要とする点と考えられる．CA を用いる認証では，機器ごとに個別に認証や暗号化，署名を行う場合，機器ごとに証明書を用意する必要がある．IoT システムの中には，これらの機器のハードウェア価格は低く抑える必要がある場合があり，これらの場合において機器ごとの証明書の購入は負担となることが考えられる．また，証明書には有効期限があり，無効化した証明書の更新や管理などのコストも必要になる．

本稿では，これらの問題を回避し，主に機器の認証と暗号化，署名を行う場合に，ブロックチェーン基盤に公開鍵を格納する PKI「kusabi」を提案する．kusabi では，機器ごとに用いる PKI の公開鍵を CA の証明書ではなくブロックチェーン基盤に格納し，必要に応じて API 経由で取り出せるようにする．kusabi では証明書が不要になるため，証明書の購入や期限切れなどに伴う更新などの管理が不要になる．本稿では，kusabi のセキュリティ的な安全性を，脅威分析を用いて検証した．その結果，プロトコルレベルにおいては，CA 証明書による PKI と同等の安全性が確保されることを確認した．また，プロトコル以外の，機器が攻撃された場合を考慮し，機器ビルド，設置/納品時および更新時のプロセスを定義した．これらのプロセスの導入により，鍵など管理に人を介さずに自動で設置、納品から更新までの自動化が可能になった．

また，提案した kusabi をプロトタイプとして実装し，PKI としての安全を確保しつつ実用的な性能で動作することを確認した．

[1] 情報セキュリティ大学院大学
IISEC, Tsuruyamachi 2-14-1, Kanagawa-ku, Yokohama, 221–0835, Japan
[2] アイビーシー株式会社
IBC Co.,Ltd.
[3] 一般社団法人組込みシステム技術協会
Japan Embedded Systems Technology Association
[a)] okubo@iisec.ac.jp

2. 従来の PKI の課題と研究の動機

CA の証明書を用いる PKI を**図 1** に示す．公開鍵は証明書に格納する形で提供される．相手を認証する場合，認証する側は相手の証明書を受けとり，証明書の正当性を CA に確認することで正当な相手であることを確認し，TLS などの暗号化通信を開始する．また，電子署名を行う際は，送信側が秘密鍵を用いて署名し，受信側は証明書の公開鍵を用いて署名検証を行う．どの利用においても，CA がその鍵の正当性を保証する仕組みになっている．また，証明書には有効期限があり，期限の切れた証明書は無効となり，再び利用を可能とするには証明書の更新が必要となる．

ネットワーク監視カメラのように複数の機器を持つシステムにおいて，同一の鍵を使用する場合，1 つの鍵が漏洩することによりシステムのすべての情報が危険にさらされる．このため，機器ごとに鍵を設定することが望ましいが，CA 証明書を用いる PKI においては，鍵ごとに証明書の購入，管理が必要になり，コストが増加する．

コストを節約するため，コストのかからない自己発行証明書を使う手段もあるが，この場合は証明書の正当性について第三者の保証が得られないことになる．また，証明書の価格については，数万から数十万と価格差があるが，価格の差は認証の信頼度の差によるものと言われる．

また，証明書による PKI を IoT 機器で安全に利用するためには，機器からの鍵の情報漏えいを防ぐため，機器に耐タンパー性のハードウェアの実装が必要になる．

著者らは，CA を用いる場合と同等の信頼性を確保しつつ，CA 証明書の購入や管理，耐タンパーのハードウェアを不要とするために，公開鍵をブロックチェーンに格納する方式を提案する．

3. 関連研究

本節では，PKI に関連する研究・技術および，ブロックチェーンに関連する研究，技術について述べる．

櫻井ら [1] は，証明書形式に基づく鍵管理の問題について整理し，証明書を適切に変更するための設計を提案している．櫻井らは，証明書変更時に生じる認証について，オンライン，オフラインによる認証を挙げているが，オンライン認証については，ユーザ自身の証明書を用いる方式，他の証明書を用いる方式，Kerberos 認証を用いる方式などいくつかを挙げている．このうちのいくつかの手法では，変更を自動化できるとしているが，認証局そのものは存在する前提で提案されている．

PKI についてはこのほか，主にエンドポイントにおける証明書保管のための技術として，ハードウェアとして証明書や鍵データを保護する耐タンパー技術 [2][3] が挙げられる．また，耐タンパーに関連する技術として，セキュアな領域と通常の領域を隔離する TrustZone[5] などの技術や，エンドポイントの信頼性を確保するために，起動時に信頼性の拠点 (root of trust) を設け，拠点から信頼のチェーンを確立することで安全を確保する TPM[4] などの技術も，PKI を補強する技術として挙げられる．

ブロックチェーンを応用した技術については，Bitcoin などの仮想通貨取引への応用が代表的であるが，分散データベースとしての応用 [7][8] や、改ざんが困難であるという特徴を利用した研究，技術 [10]，機密性の維持に応用する研究 [9] などが提案されている．

大橋ら [6] は，ブロックチェーンにコンテンツのハッシュ値と電子署名を登録することで信頼性を検証するコンテンツ管理システムを提案している．大橋らの手法はブロックチェーンを改ざんの検証に用いているが，PKI 基盤の提案ではない．

4. 提案方式

提案方式「kusabi」の構成を**図 2** に示す．kusabi では，証明書ペアを生成後，秘密鍵を各機器に，公開鍵をコンソーシアム型ブロックチェーンサーバ (BS) に格納する．そして各機器には秘密鍵とともに，公開鍵を取得するための ID「kusabi-ID」を発行し配布する．公開鍵の取得はプロビジョニングサーバ (PS) に対する API 呼出しによって行う．なお，PS 上で鍵ペアを生成して，公開鍵を登録後は PS から鍵は削除する．また，機器に秘密鍵を配布後は PS から秘密鍵も削除し，PS 上ではブロックチェーンの証跡以外は管理しないものとする．

本提案においては，以下のセキュリティ的な前提をおく．

- kusabi-ID は鍵ペアにユニークな ID として発行され，kusabi-ID 自体の推測，および kusabi-ID から公開鍵や秘密鍵の推測は困難である．
- 公開鍵の取得 API は安全である (機密性，完全性やなりすましがないことが保証される)
- ブロックチェーンへの公開鍵の格納や取り出しは，コンソーシアム型を用いるため，コンソーシアム参加者以外による不正アクセスは困難である．この安全性は用いるブロックチェーンサーバの安全性に依存し，本稿の提案の対象外とする．

5. 提案方式に対する脅威の検証

筆者らは，提案方式 (kusabi) のモデルをもとに，脅威モデリング [11] を用いた脅威分析を行った．kusabi のデータフローを脅威モデリングの DFD でモデル化したものを**図 3** に示す．

図 2 において，前項の前提から，プロビジョニングサーバ (PS) とブロックチェーンサーバ (BS) 間には信頼境界を置かないものとする．また，機器と PS 間には境界を置くが，通信は API 呼びだしのため，いわゆる中間者攻撃

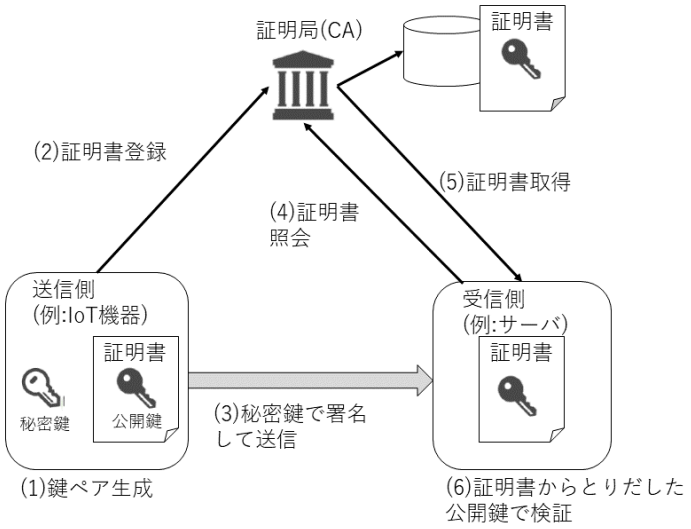

図 1 証明局 (CA) を用いた PKI の流れの例
Fig. 1 Example of PKI flow using CA.

(MITM) の脅威は想定しないものとする．また，各機器には鍵と kusabi-ID を保存するストアがあるが，PS は一方がら送られた鍵や ID を他方に渡し，送信後は削除するため，ブロックチェーンの証跡以外の鍵や ID のストアは持たないものとする．

次に，識別されたエントリポイントに対し STRIDE の 6 種類の脅威をあてはめるが，本稿においては電子証明に関わる安全性に着目するため，否認 (R)，およびサービス妨害 (D) についての攻撃可能性については対象外とする．

データフロー図を用いてエントリポイントと脅威，攻撃可能性をまとめたものを**表 1** に示す．

上記前提により，機器およびプロビジョニングサーバ (PS) における脅威が攻撃可能性ありとして残る．このうち，機器を対象とする対策として，ファームウェアのビルドと機器への設置，およびファームウェア更新時のプロセスを新たに導入することにする．ファームウェアの更新は，脆弱性を除去するために必要であり，かつ総務省の省令においても IoT 機器における更新が必須と規定され [12]，安全性が重視される点である．

6. ビルド，設置，更新時のプロセス

機器のファームウェアのビルド，機器への設置/出荷，アクティベート，更新時にそれぞれ別個の鍵ペアを用意し，公開鍵をブロックチェーンに登録，管理する．ビルド/設置/出荷時およびアクティベート時のシーケンスを**図 4** に示す．

- ビルド/設置/出荷時
  ビルド時に，ビルドする環境で認証前の鍵ペアを生成し公開鍵を BS に送り，認証前鍵に対応する kusabi-ID(pre) を取得し，秘密鍵 (pre) とともに機器に設定情報として保存する．これを各機器に対して行い，設定，出荷する．
- アクティベート時

  (1) 機器は PS に kusabi-ID(pre) を渡して共通鍵を要求する．
  (2) PS は BS に kusabi-ID を渡して対応する公開鍵 (pre) を取得し，公開鍵をもとに AES 共通鍵を生成する．生成した共通鍵をセッション ID とともに公開鍵 (pre) で暗号化し機器に送信する．
  (3) 機器は秘密鍵 (pre) で復号し，共通鍵，セッション ID を得る．PS にセッション ID を渡しアクティベート用の秘密鍵 (act) を PS に要求する．
  (4) PS はアクティベート用の鍵ペア (act) を生成し，公開鍵 (act) を BS に登録し kusabi-ID(act) を得る．kusabi-ID(act) および秘密鍵 (act) を共通鍵

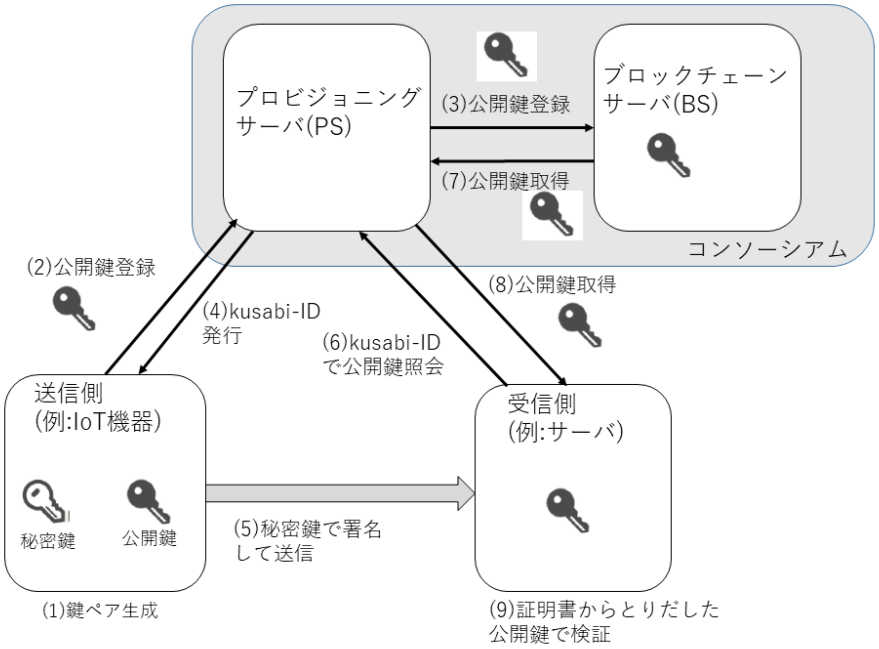

図 2 提案方式の構成
Fig. 2 Structure of the proposed method.

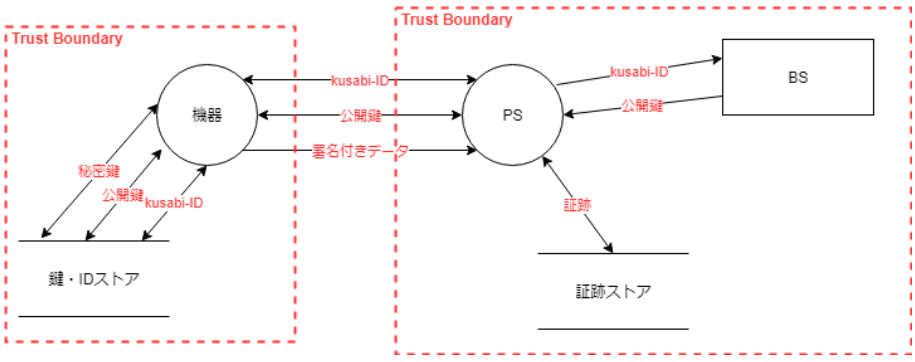

図 3 kusabi の DFD モデル
Fig. 3 DFD model of kusabi.

で暗号化し機器に送る (送信後、

(5) 機器で秘密鍵 (act) と kusabi-ID(act) を復号し，秘密鍵 (pre) と kusabi-ID(pre) を削除後，秘密鍵 (act) と kusabi-ID(act) を登録する．その後認証を得て，アクティベートが実行される．

- 更新時
  各機器は定期的に PS に対しファームウェア更新のチェックを行い，更新がある場合は秘密鍵 (art) で署名されたデータをダウンロードする．その後，各機器は PS 経由で BS から公開鍵 (art) を更新データの真正性を検証後，更新，再起動を行う．自身のインベントリ情報を収集し，収集結果に対し秘密鍵で署名を行い，PS に送信する．PS は事前に登録されたインベントリ情報と比較し，相違のないことを確認する．
- ファームウェア更新の際は，PS がファームウェアを機器に送信し強制的に更新 (OTA) と再起動を行う．

表 1 脅威分析の結果
Table 1 Result of threat analysis.

エントリポイント	脅威	攻撃可能性
機器	機器からの秘密鍵/kusabi-ID の漏えい	あり
	機器の特権昇格	あり
機器-PS 間	機器のなりすまし	あり
	MITM によるリクエストの改ざん/偽造	対象外
	MITM によるレスポンス (公開鍵) の改ざん	対象外
	機器による送信の否認	対象外
	MITM による kusabi-ID の漏えい	対象
	機器から PS に対する DoS 攻撃	対象外
PS	PS からの kusabi-ID，機器情報などの情報漏えい	あり
	PS に対する特権昇格	あり
PS-BS 間	PS のなりすまし	あり
	MITM によるリクエストの偽造	対象外
	MITM によるレスポンス (公開鍵) の改ざん	対象外
	MITM によるリクエストの改ざん/偽造	対象外
	PS による送信の否認	対象外
	MITM による kusabi-ID の漏えい	対象外
	PS に対する DoS 攻撃	対象外

7. プロトタイプ実装

提案方式 kusabi が初期設定，更新も含めて実際に動作することを確認するため，プロトタイプの実装を行った．IoT の各機器には，Yocto Linux 2.6.2(thud) と OTA として Mender 1.8[13] を用いた．Mender は，機器を稼動させたまま，ファームウェア更新を行い，再起動させることができる．これらのソフトウェアを，下記 2 種類の市販されている安価なハードウェアにて動作させた．

- Raspberry Pi3 (動作クロック：1.2GHz，メモリ：1GB)
- Up Board (CPU：Intel x5-Z8350(Atom)，動作クロック：1.44GHz，メモリ：4GB)

を用いた．また，プロビジョニングサーバには AWS，ブロックチェーンは HyperLedger Fabric[14] ベースの IBM Blockchain Platform(IBP) を用いた．結果の確認は PS 上に実装した管理画面で行った．管理画面を**図 5** に示す．

プロトタイプを稼動させた結果，署名と検証，初期設定とファームウェア更新を確実に行えることを確認した．プロトタイプ環境においては，機器の認証と更新を 30 分ごとに自動でポーリングさせた．その結果，認証 (Kusabi 側との ID/証明書/鍵の交換) は，数百 ms〜数秒で完了した．また，更新に要する時間は，数 GB のファームウェア更新に約 10〜15 分を要した．ただし，大半が PS からの更新データのダウンロード時間で，機器の再認証は，数十 ms〜数秒で完了する．したがって，更新全体に要する時間は更新データのサイズやネットワーク環境に依存し，より少ない更新データでは時間が短縮される可能性が高いと考えられる．

8. 議論

脅威分析の結果，複数機器で構成されたシステムで利用する場合においては，従来の CA による電子証明と比較して，提案方式 (kusabi) は同等の安全性を有していると言える．また，6 節に上げたプロセスを実現することにより，下記の脅威についてはある程度検出，抑止が可能となることがわかった．

- 機器上の秘密鍵の改ざん
  機器が署名したインベントリ情報の署名検証を受信側で行うことにより検出可能．
- 設置時における不正ソフトウェアの混入
  安全な環境で実施することで，事前に排除することが可能．
- 設置後の不正ソフトウェアの混入
  機器のインベントリ情報の確認を行うことで，未知のソフトウェアのインストールなどは検出可能になる．仮に検出できなかったとしても，また，定期的なファームウェアの強制更新により修復が可能となる．

ただし，脅威分析で識別された脅威のうち，機器からの秘密鍵，kusabi-ID の漏えいについては，ハードウェア上で秘密鍵を安全に保持する耐タンパーのチップによる保護がなければ，対策はむずかしい．また，機器に不正ソフトウェアをインストールされた場合においても，不正ソフトウェアが秘密鍵にアクセスが可能で，かつインベントリ情報を改ざんして送信することが可能であるとすると，その情報によって機器の異常を検出することは困難である．また，このような場合，提案方式で行っているファームウェアの強制更新と機器の再起動も不正ソフトウェアにより回

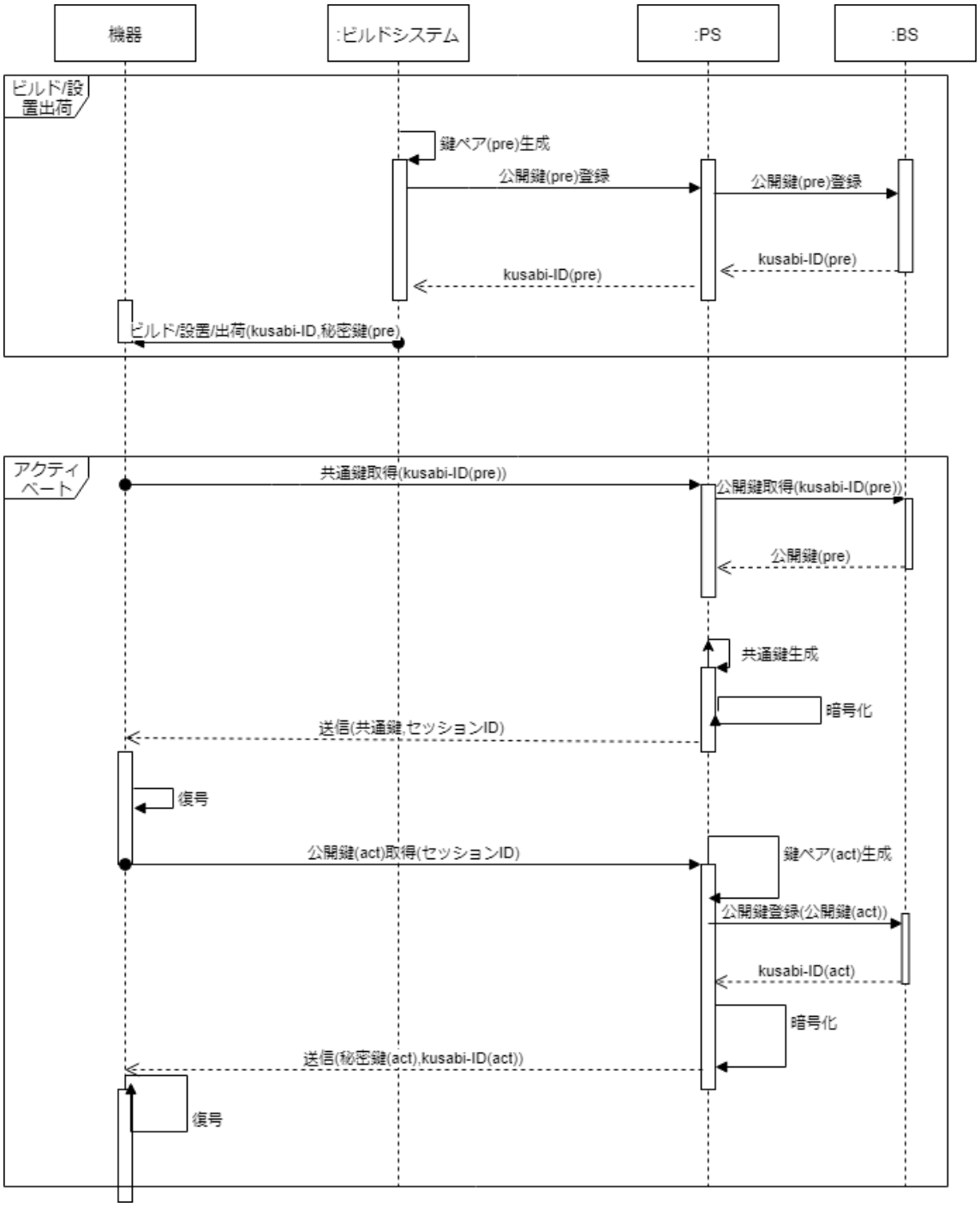

図 4 ビルド/設置/出荷時およびアクティベート時のシーケンス
Fig. 4 Sequence diagram on build/deploy/ship/activate.

避されるリスクがある．これらのリスクを回避する手法としては，関連研究で述べたように TPM により起動時に信頼の拠点 (root of trust) を確保する技術が知られている．本提案方式は TPM のような起動時から信頼の拠点を確保する技術と同等の安全性の確保は困難であるが，TPM のような技術と組合せることて，TPM と CA 証明書の組み合わせによる PKI と同等の安全性を確保することは可能と考えられる．

脅威分析で識別された上記以外の脅威としては，PS のなりすましやのっとりの脅威がある．この脅威は，CA 証明書において，CA のなりすまし，のっとりに対する脅威が存在することと同等であり，これに対して CA ののっとり，なりすまし対策と同様に PS においてもなりすまし，のっとり対策が必要になるということである．したがって，kusabi の安全性は PS の安全性確保が前提であると言えるものの，対策としては既知の対策を導入することで対応が可能であると考えられる．

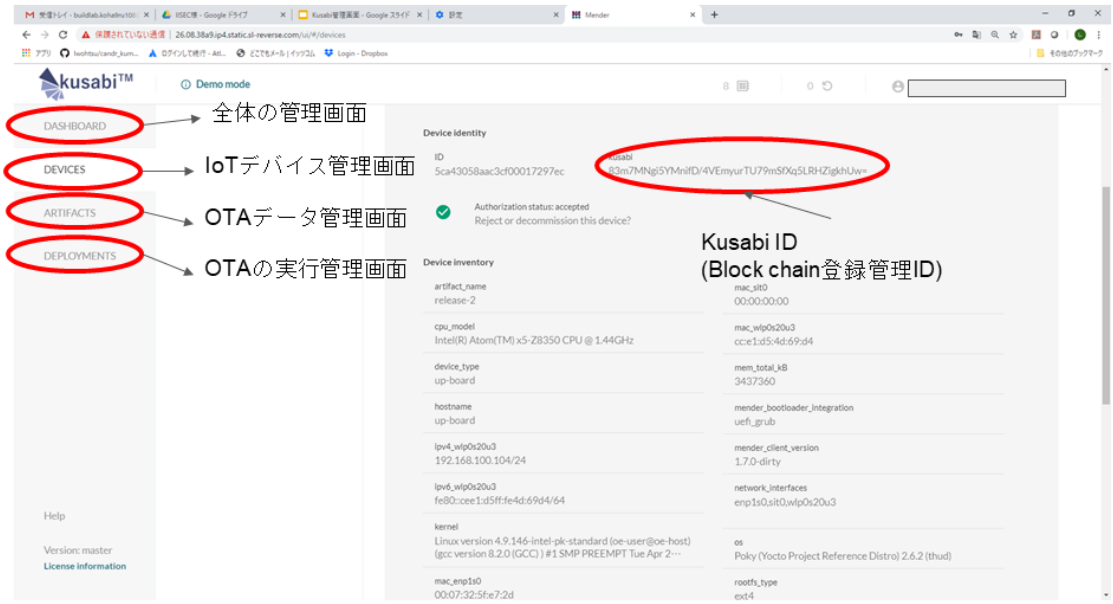

図 5 PS の管理画面
Fig. 5 Management screen on the provisioning server.

9. おわりに

本稿では，主に複数の IoT 機器を管理するようなシステムにおいて，従来のように CA の証明書を機器ごとに購入する方式が証明書購入，人による管理が障害となっている問題について，CA の証明書の代わりに，公開鍵をコンソーシアム型ブロックチェーンで保管することで電子証明を実現する方式 kusabi を提案した．また，提案方式の脅威分析を行い，プロトコルとして CA 証明書による PKI と同等の安全性があることを確認した．また，プロトコル以外の，機器の安全確保については，ファームウェア設置，更新時のプロセスを導入することにより，TPM の導入が困難な機器について一定の安全を確保可能にしただけでなく，M2M の環境において自動的にファームウェアの更新の実現を可能にした．また，提案方式を実装し，電子署名と検証，ファームウェア設置，更新が可能であることを示した．

今後の課題としては，TPM 技術など，起動時からの信頼拠点確保の技術と組み合わせることによる一層の安全性向上が挙げられる．

参考文献

[1] 櫻井三子, 佐野晋: 公開鍵暗号系を利用した証明書の変更を考慮した管理方式の設計. 情報処理学会マルチメディアと分散処理ワークショップ, pp.93-100 (1995).
[2] 松本勉, “ 耐タンパー技術： 物理と論理のはざま ”, 電子情報通信学会基礎・境界ソサイエティ大会予稿集, pp.296-297 (1997).
[3] 平成 14 年度耐タンパー性調査研究委員会報告書, 日本規格協会情報技術標準化研究センター (2003).
[4] *TPM 2.0 Library Specification Approved as an ISO/IEC International Standard*, Trusted Computing Group (2015).
[5] *ARM Security Technology Building a Secure System using TrustZone Technology*, ARM Developer, available from `https://developer.arm.com/docs/genc009492/latest/preface` (2019-6-19 参照).
[6] 大橋盛徳 他.: デジタルコンテンツのスマートプロパティ化に向けた情報登録方法の提案と実装. 信学技報, vol. 116, no. 23, LOIS2016-3, pp. 13-18 (2016).
[7] Racin Nygaard, Hein Meling and Leander Jehl: *Distributed storage system based on permissioned blockchain*, SAC ’19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (2019).
[8] 堀真寿美他: ブロックチェーンを用いた非集中型学習支援システムの提案, 情報処理学会インターネットと運用技術（IOT）研究報告, 2019-IOT-46(5), pp.1-8 (2019).
[9] 高木誠也他: ブロックチェーンを使用したクラウド上でのソフトウェア著作権保護システムの提案, 情報処理学会インターネットと運用技術（IOT）研究報告, 2019-IOT-

44(35), pp.1-6 (2019).
[10] Rong Wang et al.: *A Video Surveillance System Based on Permissioned Blockchains and Edge Computing*, 2019 IEEE International Conference on Big Data and Smart Computing (BigComp) (2019).
[11] Adam Shostack: Threat Modeling: Designing for Security, Wiley, 2014.
[12] 端末設備等規則等の一部を改正する省令案に対する意見募集の結果及び情報通信行政・郵政行政審議会からの答申（IoT の普及に対応した電気通信設備の技術基準等に関する制度整備）, 総務省 (2019).
[13] *Documentation for Mender 2.0*, Mender, available from `https://docs.mender.io/` (2019-6-19 参照).
[14] *Hyperledger Architecture, Volume 1: Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus*, Linux Foundation, 2018.