HABANEROTSのエッジデバイス向けセキュリティー機能

Security Functions of HABANEROTS IoT Platform Service to Protect Edge Devices against Cyberattacks

南　圭祐 MINAMI Keisuke　小池　竜一 KOIKE Ryuichi　伊藤　俊夫 ITO Toshio

東芝グループは，様々なIoT（Internet of Things）システムに必要な共通機能をクラウドサービスで提供するIoT基盤サービス HABANEROTSを開発し運用することで，グループ全体のCPS（サイバーフィジカルシステム）への取り組みを推進している。HABANEROTSは，クラウドシステム上に構築されたサーバー側のソリューションであるが，日増しに重要度が高まるCPSサービスのセキュリティーは，エッジデバイスを含めたシステム全体で確立する必要がある。

そこで，今回，HABANEROTSに接続するエッジデバイスの状態を正しく認識するために必要なデバイス認証技術と，脆弱（ぜいじゃく）性のないソフトウェアに更新できるソフトウェア更新技術を開発した。

The Toshiba Group has been devoting continuous efforts to the development and operation of a cloud-based platform service called HABANEROTS so as to offer common functions necessary for various Internet of Things (IoT) systems, and promoting the expansion of its cyber-physical systems (CPS) service businesses. Although HABANEROTS is a server-side solution constructed using cloud systems for CPS services, it has become necessary to enhance the cybersecurity of overall systems including edge devices in order to establish systems that are robust against cyberattacks.

In response to these circumstances, we have now developed the following functions for HABANEROTS: (1) a device authentication scheme to correctly recognize the state of edge devices, and (2) a software update scheme to apply the latest software without cyberattack vulnerabilities to edge devices.

1. まえがき

東芝グループは，東芝サイバー戦略2019[(1)]にのっとり，CPSの共通機能を提供するプラットフォーム HABANEROTSを構築し，CPS開発運用基盤の統合による開発運用コストの低減とサービスビジネス競争力の強化を目指している。HABANEROTSの活用により，新たなCPSサービスを容易にスタートさせ，AI技術を活用したデータ処理が可能となる。

HABANEROTSは，エッジデバイス（以下，デバイスと略記）を接続するためのHTTPS（Hypertext Transfer Protocol Secure）[(2)] API（Application Programming Interface）を備え，センサーデータのアップロードや遠隔制御を可能にする。このデバイス側のセキュリティーに目を向けると，デバイス上で動作しているファームウェア（以下，広義の意味でソフトウェアと呼ぶ）にはバグが内包されており，これを完全に排除することは困難である。そのため，運用期間が長期にわたるCPSでは，将来的にこれらのバグが攻撃可能な脆弱性として顕在化する可能性が高く，適切なソフトウェア更新技術を備えることが求められる。国際的なセキュリティーガイドライン（米国国立標準技術研究所（NIST）が発行したNIST SP 800-171[(3)]やNISTIR 8259 Second Draft[(4)]など）でも，攻撃の未然防止ではなく“侵入後の検知・対応・復旧”に重点を置いて検討することが推奨されている。CPS運用におけるデバイスの復旧とは，究極的には“状態を正しく認識”して，“脆弱性のない最新ソフトウェアに更新”することの2点に集約される。

HABANEROTSでは，セキュリティーの共通プラットフォーム化・標準化の一環として，接続するデバイスの状態を正しく認識するために最初に必要となるデバイス認証技術と，ソフトウェア更新技術の2点にフォーカスして技術開発を行った。ここでは，これらのデバイス認証技術とソフトウェア更新技術について述べる。

2. デバイス認証技術

2.1 デバイス認証における問題

データを蓄積するサーバーなどでデバイスを正しく識別するには，通信プロトコルにおける認証方式が広く用いられている。認証方式には，大きく分けて共通鍵方式と公開鍵方式がある。共通鍵方式は，秘密鍵が直接通信プロトコル上で使用されるため，盗聴などのリスクにさらされることになる。一方，公開鍵方式は，秘密鍵が通信プロトコル上で使用されることはなく，より安全に認証を行うことができる。

公開鍵方式の認証方式としては，クライアント証明書を応用したmTLS（Mutual Transport Layer Security）などが

よく用いられている。mTLSは，その名のとおり，クライアント証明書を用いてトランスポート層で認証を行うものである。しかし，クラウドベンダーのTLS終端機能を提供するサービスにはmTLSをサポートしていないものも多く，オンプレミスを含めた多様な環境で利用できるポータビリティーに問題がある。また，トランスポート層に実装されているため，アプリケーション層の概念を使用した細粒度のセキュリティー対策が行えないという問題もある。

一方，アプリケーション層で公開鍵方式を用いるものは，様々な方式が提案されているが，デバイス認証の用途で普及した標準技術は存在していない。

2.2 アプリケーション層での公開鍵認証方式

そこで東芝グループは，mTLSが利用できない場合などに向けた代替案として，アプリケーション層にHTTP（Hypertext Transfer Protocol）を用いる公開鍵認証方式の一種である，OAuth 2.0 DPoP[5]をベースとしたデバイス認証方式を開発した。OAuth 2.0 DPoPは，JSON Web Token（JWT）[6]を用いたDPoP proof JWTと呼ばれるデータ構造によって，公開鍵方式の署名でアクセストークンの所有者を検証する。デバイス認証の文脈では，公開鍵の交換（デバイスプロビジョニング）を事前に行うことで，アクセストークンの発行が不要になる。そこで，DPoP proof JWTをベースに独自に開発した，Device Credentials Token（DCT）を用いることにした。

DCTは，Claimと呼ばれる所定の属性情報をJSON形式で記述し，デジタル署名を付与した上でテキスト形式にエンコードしたものである。**表1**は，DCTが内部に保持するClaimを示している。デバイスは，HABANEROTSサーバーにHTTPリクエストを送る際に，まずそのリクエスト情報から表1の各Claimを作成し，デバイスの秘密鍵で署名してDCTを作成する。その後，デバイスはそのDCTをHTTPリクエストの認証ヘッダーに付与し，リクエストを送信する。リクエストを受け取ったHABANEROTSサーバーは，まずDCTのsub Claimを読み取って，リクエストを送ったデバイスID（識別情報）を特定する。デバイスの公開鍵が，デバイスの所有者によって事前にHABANEROTSに登録済みの場合，サーバーは，このデバイスIDに対応するデバイス公開鍵をデータベースから読み出し，DCTの署名を検証する。この署名検証と各Claimの整合性の検証に合格すれば，サーバーは当該リクエストを正当なものとして処理する。

DCTは，HTTPリクエストに付与されるため，アプリケーション層で認証処理を実装できるという利点がある。その反面，mTLSによる認証と比べたリスクとして，署名済みのDCTが漏洩（ろうえい）すると，それを使った成り済まし（いわゆるリプレイ攻撃）が可能である点が挙げられる。表1のClaimは，そうしたリプレイ攻撃のリスクを下げるように設計されている。攻撃者が盗んだDCTによって実行可能なリクエストは，そのDCTにClaimとして含まれている特定のHTTPメソッドや，特定のURL（Uniform Resource Locator），生成されてからの経過時間などによって，ほとんどが制限される。

表1. DCTのClaim一覧

List of claims of device credentials token (DCT) developed by Toshiba Group

Claim	説　明
sub	リクエストを送信したデバイスのデバイスIDを示す。 例: "FBBVE2"
htm	HTTPリクエストのメソッドを示す。 例: "POST"
htu	HTTPリクエストのURLを示す。 例: "http://api.tenant1.habanerots.com/devices/FBBVE2/signals/timeseries"
iat	リクエストを作成した時刻を示す。 例: 1594598400
jti	このDCTを指すユニークIDを示す。 例: " 3F2504E0-4F89-11D3-9A0C-0305E82C3301"

3. ソフトウェア更新技術

3.1 更新の必要性と課題

近年，ソフトウェア更新機能自体を悪用して攻撃を成立させる事例が，報告されている。そのため，安易にソフトウェア更新機能を独自実装すると，当初の思惑に反してシステムを危険にさらすことになりかねない。

東芝グループは，十分なセキュリティー強度を持った更新方式であることと，オープンで標準規格に準拠した技術であることを条件に，ソフトウェア更新技術のベンチマークを実施した。その上で，後述するNotary®[7]を選出し，HABANEROTSに試験搭載した。

3.2 更新に対する攻撃

ソフトウェア更新は，**図1**に示すように“配布中ソフトウェアのバージョン検知”，“ソフトウェアの入手”，及び“インストール”の3ステップを踏む。このとき，各ステップには様々な攻撃の余地がある。

例えば，デバイスがソフトウェア更新を行う際には，新規ソフトウェアが配布されていることを認識する必要がある。このとき，攻撃者が常に小さなバージョン番号をデバイスに通知すると，デバイスは自身のソフトウェアよりも新しいソフトウェアが配布されていることに気付けなくなる（バージョン固定攻撃）。また，異常に大きなバージョン番号をデバイスに通知し，それが受け入れられたとすると，より小さなバー

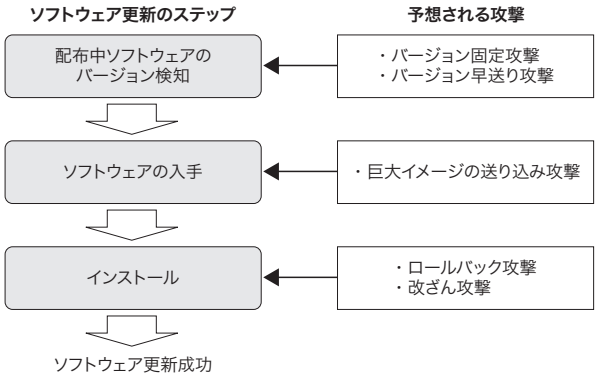


図1. ソフトウェア更新における各処理への攻撃例

ソフトウェア更新成功に至るまでの各ステップにおいて，様々な攻撃の余地がある。

Examples of cyberattacks on software updating processes

ジョン番号を持つ正規ソフトウェアが古いものと認識されて，更新に利用されなくなる（バージョン早送り攻撃）。

次に，インストール時の処理に目を向けると，バージョン番号の大小などを考慮しないと，脆弱性のある古いソフトウェアをインストールされるおそれがある（ロールバック攻撃）。また，完全性を検証しないと，改ざんされた悪意のあるソフトウェアを導入されるおそれもある。

3.3 Notary® を用いたクラウド型更新システム

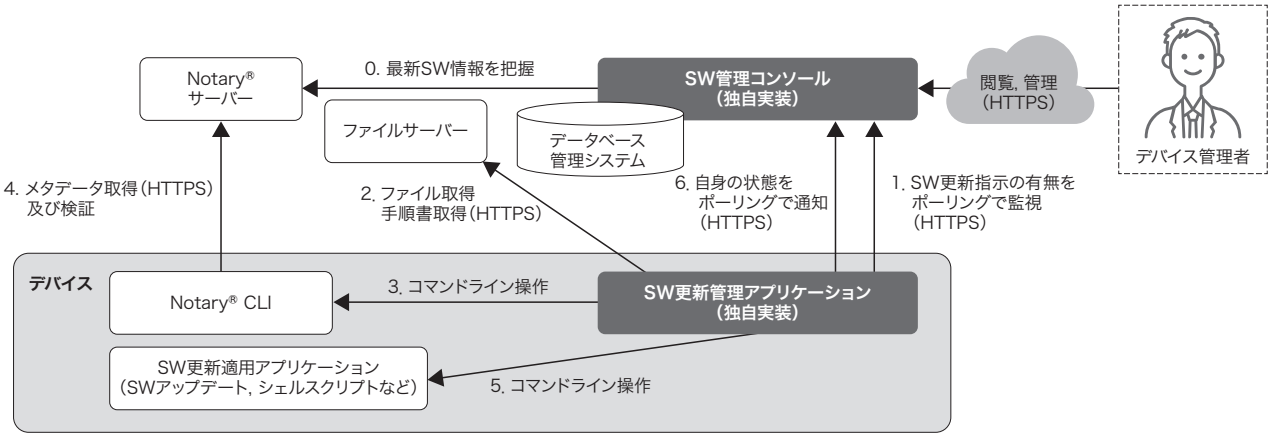
東芝グループは，3.1節で述べたように，HABANEROTSでNotary®を活用している（**図2**）。Notary®は，Cloud Native Computing Foundation（CNCF®）[8]のプロジェクトであるThe Update Framework（TUF）[9]規格のオープンソースソフトウェア実装である。Notary®は，信頼できるコンテンツのコレクションを公開して管理するための汎用的なフレームワークで，クライアントとサーバーとのペアを提供する。サーバーは，バージョン情報などを付加した署名付きメタデータをホストしており，クライアントが任意のタイミングでメタデータを参照することで，3.2節で述べたような攻撃を，網羅的に検出可能な機構となっている。

東芝グループは，独自の管理コンソールを開発してNotary®を拡張している。管理コンソールは，デバイス管理者向けの操作画面を提供しており，配下デバイスのバージョン把握や更新指示が行える。更新指示に関しては，特定のデバイスだけや特定のグループだけを更新するなど，トポロジーを考慮した処理を可能としている。

前述の更新は，Notary®サーバーが配布しているメタデータを管理コンソールが代理取得した後，デバイスのトポロジーを考慮して再配布する動作により実現している。これまで，Notary®単体では，インターネットに直接接続されていない配下デバイスの状態などを加味したきめ細かい更新制御は難しかったが，この技術により，一元的なソフトウェア更新管理とNotary®本来の網羅的なセキュリティー機能とを結合できる。

4. あとがき

HABANEROTSのエッジデバイス向けセキュリティー機能として，デバイス認証技術とソフトウェア更新技術について検討・開発を行った。今後は，更なる安全性・レジリエンス（危機対応能力）向上のため，遠隔からデバイスの状態を



SW：ソフトウェア

図2. Notary® を用いたクラウドシステム型更新システム

Notary® サーバー，Notary® CLI（Command Line Interface），及び独自開発の管理ソフトウェアを組み合わせることで，一元的な更新管理と網羅的なセキュリティー機能を結合している。

Cloud-based updating system using Notary® based on The Update Framework (TUF) specification

正しく認識するための技術であるリモートアテステーションなどの応用も検討していく。

文　献

(1) 山本 宏．東芝Cyber戦略2019 〜世界有数のCPSカンパニーを目指して〜．東芝，2019，35p．<https://www.toshiba.co.jp/about/ir/jp/pr/pdf/tpr20191128_2.pdf>，（参照 2021-05-17）．
(2) Fielding, R.; J. Reschke, Eds. RFC 7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. The Internet Engineering Task Force (IETF), 2014, 101p. <https://www.rfc-editor.org/rfc/pdfrfc/rfc7231.txt.pdf>, (accessed 2021-05-20).
(3) National Institute of Standards and Technology (NIST). Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations. SP 800-171 Revision 2, NIST, 2020, 111p. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r2.pdf>, (accessed 2021-05-17).
(4) Fagan, M. et al. Draft (2nd) NISTIR 8259 Recommendations for IoT Device Manufactures. NIST, 2020, 39p. <https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259-draft2.pdf>, (accessed 2021-05-17).
(5) Fett, D. et al. OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPoP) draft-ietf-oauth-dpop-03. <https://tools.ietf.org/pdf/draft-ietf-oauth-dpop-03.pdf>, IETF, 2021, 32p. (accessed 2021-05-20).
(6) Jones, M. et al. JSON Web Token (JWT). IETF, 2015, 30p. <https://rfc-editor.org/rfc/pdfrfc/rfc7519.txt.pdf>, (accessed 2021-05-28).
(7) GitHubDocker, Inc. "Notary". GitHub - theupdateframework/notary: Notary is a project that allows anyone to have trust over arbitrary collections of data". <https://github.com/theupdateframework/notary>,（accessed 2021-05-20）.
(8) The Linux Foundation. "Cloud Native Computing Foundation". <https://www.cncf.io/>, (accessed 2021-05-20).
(9) The Update Framework authors. "The Update Framework". <https://theupdateframework.io/>, (accessed 2021-05-20).

・Notary，CNCFは，The Linux Foundationの登録商標。

南　　圭祐　MINAMI Keisuke
デジタルイノベーションテクノロジーセンター
第二技術開発部
Digital Innovation Technology Development Dept. 2

小池　竜一　KOIKE Ryuichi, Ph.D.
研究開発センター
サイバーセキュリティ技術センター セキュリティ基盤研究部
博士（工学）　電子情報通信学会・情報処理学会会員
Security Research Dept.

伊藤　俊夫　ITO Toshio
研究開発センター 情報通信プラットフォーム研究所
コンピュータ&ネットワークシステムラボラトリー
情報処理学会会員
Computer and Network Systems Lab.