IoTシステムのセキュリティーリスクに対応した組み込み機器ソフトウェアのアップデート手法

Software Update Method for Embedded Devices to Address Security Risks of IoT Systems

OSSを用いて，組み込み機器ソフトウェアのタイムリーかつ安全なアップデートを実現

IoT（Internet of Things）システムの普及に伴い，機器に組み込まれるソフトウェアの機能拡張や脆弱（ぜいじゃく）性への対応が重要性を増しています。それには，新しいソフトウェアをタイムリーかつ安全に適用するソフトウェアのアップデート手法が必要ですが，対象となる機器数が増えると，特定の機器に限定して対応する従来の手法は現実的ではなく，より標準化され適用範囲が広く，更に大量の機器にも対応可能な手法が必要になります。

東芝は，オープンソースソフトウェア（OSS）を用いて，多くの機器が接続されるシステムの更新を，タイムリーかつ安全に行うことができる手法を開発しました。

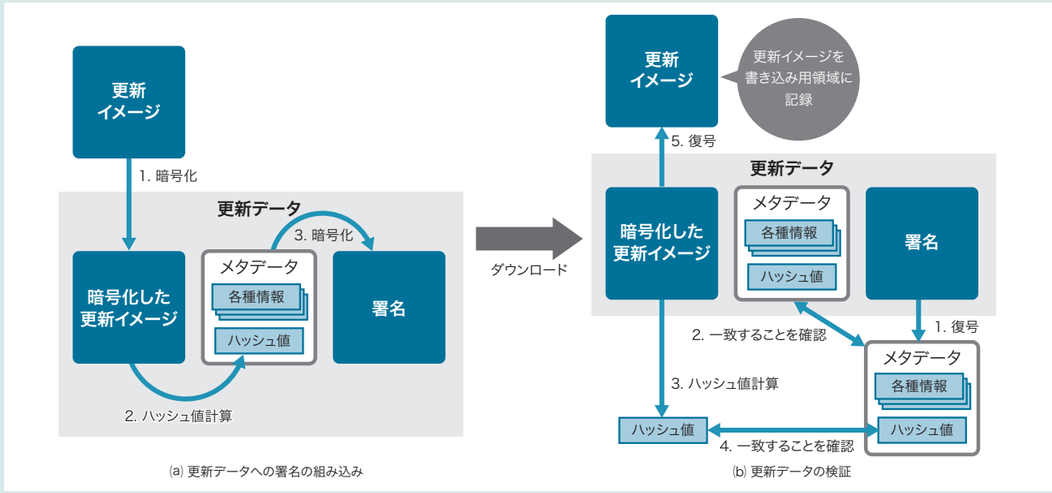

図1. 開発したアップデート手法のセキュリティー機能
更新イメージを暗号化し，そのハッシュ値を加えたメタデータから署名を作ることで，更新データの改ざん検知と窃取防止を同時に実現します。

背景

近年，IoTシステムの普及により，組み込み機器のソフトウェアは複雑な仕事を多くの機器が相互に連携して行うようになりました。これに伴い，ソフトウェアの機能拡張や脆弱性への対応に関する要求の高まりとともに，その適用方法についても様々な機能が求められています。従来は，現地に赴いて更新作業をすることが一般的でしたが，更新対象の機器が増加している昨今では現実的ではありません。また，遠隔からアップデートする仕組みを構築していても，更新失敗時はやはり現地に赴いてリカバリーする必要がありました。

東芝は，これらの問題を解決するため，より標準化され適用範囲が広く，更に大量の機器にも対応可能なソフトウェアのアップデート手法を開発しました。これにより，多くの機器のソフトウェアをタイムリーかつ安全に更新できます。

開発したアップデート手法は，Linux®をOS（基本ソフトウェア）として用いる組み込み機器を対象としています。この開発は，The Linux Foundationが主催するCivil Infrastructure Platform™プロジェクトの活動の一つとして，オープンに行われています。

ソフトウェアのアップデートに求められる機能

ソフトウェアのアップデートには様々な機能が求められていますが，代表的なものを以下に示します。

(1) アップデートの管理や操作が可能なこと
(2) 遠隔から多くの機器をアップデートできること
(3) アップデート失敗時にリカバリーできること
(4) アップデート中の電源断に対応できること
(5) 正しい更新データであることが確認できること
(6) 更新データの内容を盗み見られないこと

(1)は，アップデートの進捗状況の確認や実行指示の操作などをしやすくするために，備えておくべき機能です。(2)，(3)は，アップデートの過程で発生する想定外の事象を考慮した機能で，実運用には欠かせません。(4)は，更新対象の機器が増加している昨今では必須の機能です。(5)，(6)は，アップデートの仕組みを悪用したシステムへの侵入や情報の窃取を防ぐもので，安全なアップデートを実現するために必要となります。

開発したアップデート手法の構成と機能

今回開発したソフトウェアのアップデート手法は，更新対象の機器にSWUpdate，サーバーにEclipse hawkBit™と呼ばれるOSSを用いる構成になっています。SWUpdateは，多くのアップデートに必要な機能を標準で備えているだけでなく，汎用性が高いという特長を持っています。Eclipse hawkBit™は，更新データの管理や，機器の登録，アップデートの実行指示などを提供するとともに，(1)の管理画面を備えています。また，SWUpdateとEclipse hawkBit™は連携することも可能で，これによって(2)の遠隔からのアップデートを実現しています。

(3)，(4)は，更新対象の領域を二重に用意し，切り替えて更新するA/Bアップデートの手法で実現しています。開発した手法は，二重化した領域の一方を起動用，もう一方を更新イメージの書き込み用として使用し，領域を切り替えることで更新を完了します。そのため，更新中に電源断が起きても起動用の領域は影響を受けず，更新イメージの書き込みを再開できます。また，更新後に問題が起きても，再度領域の切り替えを行うことで更新前の状態に戻すことができます。ここで，更新後のシステムで一定時間内に所定の動作ができなかった場合は，ウオッチドッグタイマーでシステムをリセットして領域の切り替えを行います。

現在，ここまでの実装を基に，(5)，(6)で求められるセキュリティー機能の検証を進めています。この機能を**図1**に示します。SWUpdateは，更新対象の機器に書き込む更新イメージと付帯情報（メタデータ）から構成される更新データを使用します。(5)で求められるデータの正確性を検証するために，まず更新イメージのハッシュ値（定められた計算手順によって元データから得られる固有の値）を求めます。この値をメタデータに追加し，更にメタデータを暗号化したものを署名として，更新データに組み込みます（図1(a)）。この手順により，後にSWUpdateが更新データをダウンロードする際に，その署名と更新イメージの両方について，ハッシュ値などの情報が一致しているかを検証することで改ざんを検知できます（図1(b)）。(6)は，更新データ自体を暗号化することで，内容を盗み見られることを回避できます。

このような汎用性の高い手法やOSSにより，少しのカスタマイズで様々な機器への適用が可能になります。

今後の展望

組み込み機器は，利用できるリソースが限られるケースが多いため，アップデートに利用できるネットワーク通信容量や更新データ量が制限される状況にも対応できるように，開発した手法の機能を拡充する予定です。また，現在検証を進めているセキュリティー機能は，更新データのセキュリティーを向上させることに特化しているため，万一外部から更新対象の機器への侵入があった場合でも，不正なアップデートを防ぐことができる機能の開発も進めていきます。

今回用いたSWUpdate，Eclipse hawkBit™以外にも，ソフトウェアのアップデート機能を提供しているOSSは多数存在します。組み込み機器ソフトウェアにより適したアップデート手法を構築できるように，様々なOSSの機能や性能の評価も同時に進めていきます。

鈴木 章浩
技術企画部 ソフトウェア技術センター
共創ソフトウェア開発技術部