デザインソリューション
エンベデッドセキュリティ

デジタル証明書を信頼する — オフラインの時でも

はじめに

この世界では日ごとに相互のつながりが増しています。モノのインターネット(IoT)革命は、ユーザーがコネクテッドデバイスとその基盤インフラストラクチャを信頼することができる場合にのみ成功します。かつてセキュリティは、電子決済システムのような専用アプリケーションだけの問題でした。しかし今日では、セキュリティはスマートグリッド、プロセス制御、ビルオートメーションなど、その他多くのアプリケーションの要件となっています(図1)。

コネクテッドデバイスの基本的なセキュリティ要件の1つは、受け取る情報の発信元が本物であることを確認することができるということです。これはデジタル証明書を使用して実現することができます。現場におけるコネクテッドデバイスのデジタル証明書の維持管理には、さまざまな課題があります。このデザインソリューションでは、デジタル証明書のオンライン処理について検討し、どうすればコネクテッドデバイスのライフサイクル全体にわたって(コネクテッドデバイスがオフラインの時でも)デジタル証明書をセキュアかつ最新の状態に保つことができるかについて説明します。

図1. リモートスマートホーム制御

オンライン認証

強力な認証は、暗号アルゴリズムと鍵の利用に基づきます。暗号アルゴリズムには、対称と非対称という2つの基本的なタイプがあります。対称アルゴリズム(HMACなど)では、送信者と受信者が1つの秘密情報を共有する必要があります。この秘密情報は一般に鍵と呼ばれています。秘密鍵は安全に保管し、第三者には決して開示しないようにする必要があります。送信者と受信者はどちらも同じ秘密鍵を使用してデータを認証するため、情報源に偽りがないことを確認した上で情報を交換することができます。この対称方式の短所は、鍵が配布時に外部に漏れて重大な脆弱性をもたらす可能性があることです。

もう1つの方式は、非対称(またはパブリック鍵(公開鍵)アルゴリズムを使用するもので、RSAやECDSAがその例です。非対称アルゴリズムでは2つの鍵を使用し、1つはプライベート鍵として保管され、もう1つはパブリック鍵として誰でも使用することができます。そのため、鍵の配布がはるかに容易になります。(プライベート鍵で署名されたデータは、対応するパブリック鍵によってのみ検証可能であることに注意してください)。

非対称暗号を使用したメッセージ送信の例を図2に示します。アリスが信頼性のある情報をボブに送信する場合、彼女は自分のプライベート鍵でその情報に署名します。しかし、対称方式の場合とは異なり、ボブはそのメッセージが本当にアリスによって送信されたものであることを確認する際に、アリスの秘密のプライベート鍵を必要としません。彼は、アリスのパブリック鍵を使用してメッセージを検証することができます。その名称が示唆するとおり、パブリック鍵は秘密ではなく、自由に開示可能で、その使用を望む誰にでも簡単に配布することができます。ボブはメッセージの署名を検証したら、そのメッセージがアリスから送られたものであると確信することができます。この検証の実行には、アリスのパブリック鍵しか使用することができないためです。

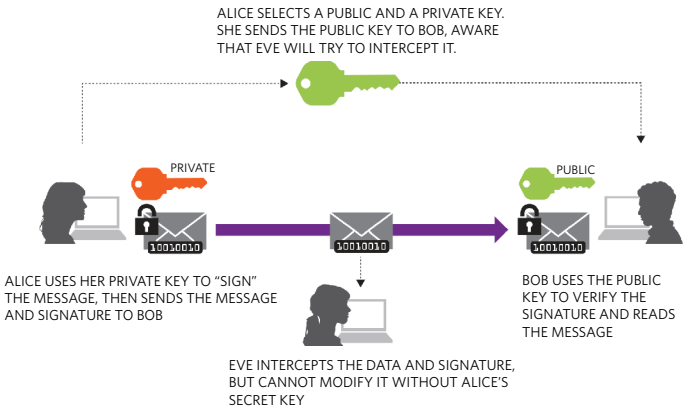

図2. 非対称暗号を使用したメッセージの送信

しかし、攻撃者がアリスのパブリック鍵を自分の鍵にすり替えるリスクは依然として存在します。そのようなシナリオ(図3)では、イブはアリスになりすまして自分のパブリック鍵をボブに送信し、それがアリスのパブリック鍵であるように見せかけます。さらに、イブは自分の鍵で署名したメッセージをボブに送信します。イブのパブリック鍵によって彼女自身のプライベート鍵を使用して署名されたメッセージを検証するため、ボブはアリスとやり取りしているものと誤認することになります。こうしたことが可能になるのは、ボブが自分にパブリック鍵を送信した人物の身元を確認する手段を持たないためです。

図3. 侵入者によるパブリック鍵のすり替え

公開鍵基盤(PKI)を使用すると、この種の攻撃を防止することができます。PKIでは、パブリック鍵の有効性が信頼性のある第三者の認証局(CA)によって保証されます。CAは、第三者のパブリック鍵のデジタル証明書に署名し、そのデジタル証明書を発行します。デジタル証明書とは、パブリック鍵とその鍵の所有権の確認に使用することができるID情報を含んだファイルです。この証明書は、さらにCAによってCA自身のプライベート鍵を使用して署名されます。

たとえば、図4では、アリスが自分のパブリック鍵にID情報を付加し、信頼性のあるCAが自分のプライベート鍵を使用してこの証明書に署名することによって、アリスの信頼性のあるデジタル証明書を作成しています。アリスはこのデジタル証明書のコピーをボブに送信します。ボブはCAのパブリック鍵を使用して、この証明書が信頼性のあるものであることを確認します(この証明書はその信頼性のあるCAのプライベート鍵によって署名されているためです)。そして、ボブはアリスの証明書を自分の信頼性のある証明書のリストに追加します。その後、彼はアリスのパブリック鍵を使用して、将来アリスから受け取るメッセージが本当に彼女から送られたものであることを確認することができます。ボブがCAによって検証することができない添付証明書を受け取った場合、その証明書は廃棄され、それ以降、その証明書が添付されたメッセージが届いてもすべて無視されます。

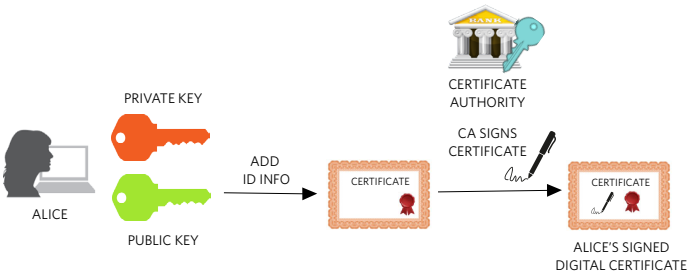

図4. CAを使用したデジタル証明書の作成

注目すべきことは、CAの有効性は、より高レベルのCAのパブリック鍵によって保証される場合があるという点です。この場合、階層構造の最上位には、安全に保管されたパブリック鍵である「ルート鍵」があります。この鍵は、権限のない者が改変したり干渉したりすることはできません。

PKIは、ウェブサイトの認証に広く使用されています。たとえば、ホームバンキングは、PKIとCAへの接続に大きく依存しています。通常、ブラウザのアドレス欄でhttps://の隣に表示されるグリーンの鍵は、そのウェブページが本物であることが証明書によって確認されていることを意味します。このオンラインの検証プロセスは、ウェブサイトのユーザーに対して概ね透過的です。

オフライン認証

コネクテッドデバイスとは、ネットワークまたはインターネットを通じてリモートサーバと通信するデバイスです。その簡単な一例として、産業用の環境でネットワーク接続されたプログラマブルロジックコントローラ(PLC)を挙げることができます(図5)。産業用の環境では、ネットワークは通常、プライベートなもので、インターネットに直接接続することはできず、信頼性のある第三者のCAに接続することもできません。ネットワークがインターネットに接続されている場合でも、ネットワーク管理者は費用やその他の理由により、第三者のCAのサービスを使用しないことを選択する場合があります。

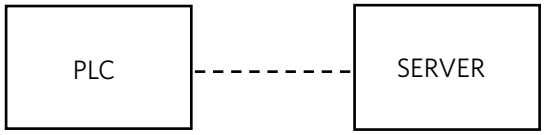

図5. ネットワーク接続されたPLC

時には、新しいソフトウェアや設定情報をPLCにアップロードすることが必要になる場合もあります。ネットワークがプライベートである場合でも、PLCにおいてデータソースが本物であることを確認することができることは依然として重要です。そうでなければ、侵入者がネットワークへのアクセス権を取得した場合、ソフトウェアや偽の設定データをPLCにダウンロードしようとする可能性があります。これを防止するため、デジタル証明書を、PLCに送られた情報が本物であることを確認するために使用することは可能です。しかし、証明書を検証する手段がない場合(たとえば、インターネット接続がない場合)、侵入者は偽のデジタル証明書をPLC上の信頼性のある証明書のリストにアップロードし、PLCとの通信を開始する可能性があります。

これを防止するには、PLC自体に検証手段を内蔵することが必要です。セキュアな自己認証の仕組みを確立することは、明らかに容易な仕事ではありません。実際には、多くのPLC(およびその他のコネクテッドデバイス)は、デジタル証明書が本物であることを確認するために必要なセキュリティ機能を備えていません。そのため、ネットワーク内に重大なセキュリティ侵害の可能性が生じます。

セキュアマイクロコントローラ

この問題に対するシンプルなソリューションは、認証機能を内蔵した追加のセキュアマイクロコントローラを使用することです。DeepCover®暗号コントローラのMAXQ1061は、コネクテッドデバイスメーカーが(将来のファームウェア更新などのために)自社製のすべてのコネクテッドデバイス用に独自のCAを確立することを可能にします。デバイスメーカーは、プライベート鍵/パブリック鍵のペアを作成した上、MAXQ1061のEEPROMにルートパブリック鍵を安全に保存します。パブリック鍵は、Maximがサービスとしてお客様の代わりにプログラムすることもできます。MAXQ1061のハードウェアセキュリティ機能とセキュアファイルシステムは、権限のないユーザーが鍵を改変することを防止します。このルート鍵は、デバイスメーカーのCA階層の最上位に置かれ、デバイスが将来受け取るすべての「子」証明書に適用されます。

図6は、MAXQ1061が新たに発行された子証明書を検証する手順を示しています。まず、コネクテッドデバイスとの通信を望むユーザーのパブリック鍵とIDを含んだ新しい子証明書が作成されます。次に、その子証明書はコネクテッドデバイスメーカーによってシステムプライベート鍵を使用してオフラインで署名され、ユーザーのデジタル証明書が作成されます。さらに、この証明書を受信したら、MAXQ1061は保存されているルートパブリック鍵を使用して署名を検証します。最後に、検証が成功した場合、MAXQ1061はそれを「証明書オブジェクト」としてEEPROMに保存します。検証済みの証明書は、破棄されるまで、このユーザーによるその後のすべての通信において信頼性があるものとみなされます。このようにして、新しい子証明書をいつでも安全にロードすることができ、既存の証明書を簡単に更新または破棄することができます。したがって、デバイスメーカーは、現場でデバイスを使用する信頼性のあるユーザーにとって、事実上オフラインCAの役割を果たします。

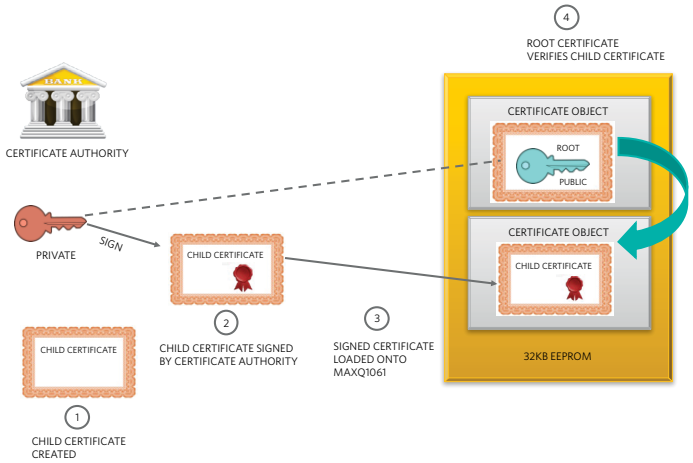

図6. MAXQ1061による証明書の検証

MAXQ1061の固有の利点

MAXQ1061では、複数の証明書ツリーとCAを同時に利用することができます。前述のPLCの例を取り上げると、PLCでは、次のように2つの異なる発信元からの情報を認証することが必要になる可能性があります。

- オペレーティングシステムのアップグレードのためにPLCメーカーを認証する
- プログラムやユーザーデータの更新のためにPLCプログラマを認証する

このシナリオでは、PLCメーカーとPLCプログラマに1つずつ、別個のルート鍵が必要になります。MAXQ1061は、独立した各子証明書が互いに干渉しないようにルート鍵を管理します。MAXQ1061は、証明書の配布と更新を可能にするだけでなく、大量のリソースを要する署名検証処理を実行するために必要なコンピューティングパワーも備えています。これは、特にコネクテッドデバイスのメインマイクロコントローラが暗号機能や、これらの大量の演算を要する機能を処理するためのコンピューティングパワーを備えていない時に有用です。

MAXQ1061のもう1つの注目すべき利点は、各種のオブジェクトを定義し、内蔵の32KB EEPROMに保存することができるという点です。次のような例が挙げられます。

- CAによって署名されたデバイス自体の証明書、および対応するプライベート鍵。これらを使用してピアデバイスに認証を提供することができます。各証明書は改変されないように保護され、各プライベート鍵は開示されないように保護されます。
- CAのルートパブリック鍵。CAのパブリック鍵は改変することができないため、ピア証明書チェーンの検証に使用することができます。
- 暗号アルゴリズムで使用される任意の対称鍵、パブリック鍵、およびプライベート鍵。
- 管理者の認証鍵とホストの認証鍵(オプションのセキュアチャネル用)。
- その他のあらゆる任意データ。最大ストレージ容量が上限になるまで。

オブジェクトはハードウェア保護された不揮発性メモリに保存され、オブジェクトの変更はアトミック操作であるため、書込み処理が(停電などのために)中断された場合、そのオブジェクトは以前の既知の値を維持します。オブジェクトは必要に応じて割当ておよび割当て解除可能で、解除すれば空きスペースが回復されます。保存されたオブジェクトは、デバイスのライフサイクル状態(初期化、運用中など)に応じて、個別のアクセス条件(読取り、書込み、実行など)を有します。

結論

このデザインソリューションでは、コネクテッドデバイスによる証明書を通じたセキュアな認証の要件について検討しました。また、CAの検証を受けないデバイスについて攻撃への脆弱性を指摘しました。そのようなデバイスには、着信する情報の発信元が本物であることを確認する手段もありません。

この問題に対するシンプルなソリューションとして、DeepCover暗号コントローラのMAXQ1061を紹介しました。証明書の配布と管理に加えて、MAXQ1061の固有の利点を適用することによって、アクセス制御(ホームオートメーション用など)、電子署名の作成、およびスマートグリッドなどの重要なインフラストラクチャのサイバーセキュリティを確保することができます。また、MAXQ1061は、第三者のCAのサービスに依存することを望まず、デジタル証明書の自己認証を実装したいと考えているIoTデバイスのメーカーにとって最適なソリューションです。

用語集

IoT: モノのインターネット。設定や制御用にインターネットへのアクセスを備えた、リモートのマイクロコントローラ駆動デバイスです。

CA: 認証局。パブリック鍵が本物であることを確認する責任を負う独立した信頼性のある組織です。

PKI: 公開鍵基盤。デジタル証明書を作成、管理、配布、使用、保存、および破棄し、パブリック鍵の暗号化を管理するために必要な一連のロール、ポリシー、および手順です。

PLC: プログラマブルロジックコントローラ。産業用プロセス制御専用のコンピュータです。

SHA: セキュアハッシュアルゴリズム

RSA: リベスト-シャミア-エーデルマン(Rivest Shamir Adleman)。パブリック鍵暗号化アルゴリズムの1つです。

ECDSA: 楕円曲線デジタル署名アルゴリズム。パブリック鍵暗号化アルゴリズムの1つです。

さらに詳しく:

MAXQ1061:DeepCover暗号コントローラ、エンベデッドデバイス用

デザインソリューション No. 56

Rev 0; May 2017

設計サポートが必要な場合は、Eメールにてお問い合わせください。
https://www.maximintegrated.com/jp/support/overview.html/TechSupportFormJapan

その他のデザインソリューションを探す

マキシム・ジャパン株式会社
〒141-0032 東京都品川区大崎1-6-4 大崎ニューシティ4号館20F maximintegrated.com/jp

© 2019 Maxim Integrated Products, Inc. All rights reserved. Maxim IntegratedおよびMaxim Integratedのロゴは、米国およびその他の国の管轄域におけるMaxim Integrated Products, Inc.の登録商標です。その他、記載されている会社名、製品名は各社の登録商標、または商標です。