GERMAN MAILGENEERING

fp-francotyp.com

FP Whitepaper

Die Vorteile eines Hardware-Sicherheitsmoduls in industriellen IoT-Anwendungen

Inhalt

Einführung 4

Ein verbreiteter Systemaufbau 6

Gefahren, die berücksichtigt werden sollten 8
- Datenmanipulation oder Integritätsverlust 8
- Eindringen und Penetration in das System 9
- Unautorisierte Softwaremodifikation 9

Integration von physischer Sicherheit in Ihre IoT-Anwendung 11

Einsatz von Hardwaresicherheit zum Schutz von TLS-Schlüsselmaterial von Clients in feindlichen Umgebungen 14
- Ein grundlegendes Authentifizierungsverfahren 14
- Restrisiken 15
- Ein alternatives Verfahren 16

Vorteile aus zusätzlichen Diensten eines HSM 18
- Validierte Algorithmen und Schlüsselstärken zur Erhöhung der Vertrauenswürdigkeit 18
- Verbesserte Kapselung 18
- Pay-per-Use 18
- Zeitstempelung 18
- Verschlüsselung und Entschlüsselung von Firmware 18
- Firmware-Update 18

Die Vorteile eines Hardware-Sicherheitsmoduls in industriellen IoT-Anwendungen

Einführung

Das „Internet der Dinge“ (Internet of Things – IoT) ist derzeit eines der großen Schlagwörter und zählt zu den Megatrends sowohl auf dem Verbrauchermarkt als auch in der Industrie. Die Anzahl der mit dem Internet verbundenen Geräte wächst exponentiell. Während dazu natürlich die Smart-Clients wie Tablets und Smartphones beitragen, erfordert die wachsende Anzahl von kleinen Konsumgütern und Industrieanlagen besondere Aufmerksamkeit. Industrieanlagen umfassen dabei einfache Sensoren und reichen von intelligenten Überwachungssystemen bis hin zu komplexen Portalen. Veröffentlichte Schätzungen gehen davon aus, dass 8 bis 20 Milliarden Geräte bis 2020 auf irgendeine Weise an die Cloud angeschlossen sein werden.[1] Die wirtschaftlichen Auswirkungen werden in Billionen von US-Dollar gemessen.

Sicherheit ist im klassischen IT-Sektor seit Langem als unabdingbar anerkannt, wird jedoch bei Systemen, die für den IoT-Markt entwickelt werden, insbesondere für das industrielle IoT, oft vernachlässigt. Die Gründe dafür sind verschieden, wobei dies möglicherweise jedoch in einem gewissen Maße darauf zurückzuführen ist, dass solche Fälle wie in der PC- und Verbraucherwelt nicht öffentlich bekannt werden; bisher wurden nur ein paar Angriffe wie Stuxnet in der industriellen IoT-Arena umfassend veröffentlicht.

Um zu sehen, woher diese Gefahren für die Sicherheit kommen können, lohnt sich ein Blick auf die am weitesten verbreiteten Arten von Angriffen, die der Öffentlichkeit bekannt sind. Dies sind (aufgelistet nach abnehmender Häufigkeit) Malware, webbasierte Angriffe, Denial of Service, bösartige Insider-Angriffe und Schadcodes. Dazu können Phishing, Social Engineering und Gerätediebstahl gehören. Ransomware ist ebenso auf dem Vormarsch wie Botnetze, die Rechenleistung von Computersystemen kapern können. Vor diesem Hintergrund können wir uns anschauen, wie industrielle Nutzer darauf reagieren.

Nicht selten lassen sich Industriekunden finden, die noch teure Maschinen, die ursprünglich in den 1950er Jahren installiert wurden, mit einem Schaltschrank voll von elektromechanischen Schützen betreiben. Zwischen den 1970er und 1990er Jahren wurden andere Anwendungen entwickelt, als die Revolution der speicherprogrammierbaren Steuerung (SPS) die früheren komplizierten Maschinensteuerungen ablöste. Jedoch arbeiten sie noch immer autonom und unabhängig, sodass die Welt der Industriemaschinen sicher abgeschirmt und offline geblieben ist, während die Informationswelt mit ihren Personal Computern und Servern Zeuge der Erfindung und des Anwachsens von bösartigen internetbasierten Angriffen wurde.

Heutzutage sind IoT-Anwendungen so populär geworden und so leicht und preiswert zu integrieren, dass die Betreiber einiger dieser Maschinen hoch motiviert – wenn nicht sogar gezwungen – sind, sie aufzurüsten. Somit werden nun Feldbus-Controller, Industrie-PCs u. ä. integriert. Natürlich laufen diese Lösungen mit gängigen Betriebssystemen wie Linux oder Windows und bieten somit volle Netzwerk-Konnektivität ähnlich wie bei den heutigen privaten Mobiltelefonen. Folglich sind industrielle IoT-Clients nun zu Schnittstellen geworden, an denen die Maschinen der Vergangenheit mit dem Internet von heute verbunden werden. Das bedeutet, dass sie die beeindruckende Menge von Angriffen im Netz bewältigen müssen. Darin besteht die Herausforderung für Entscheidungsträger und Systemarchitekten bei der Bestimmung möglicher Gefahren und Risiken für ihre Architektur.

Nicht alle Sicherheitsprobleme resultieren aus vorsätzlichen Angriffen einer Person oder einer böswilligen Gruppe. Ein Unternehmen kann einfach nur Opfer der Nebeneffekte von anderen globalen Angriffen werden und die Lösungen, die zum Zeitpunkt der Entwicklung der Architektur Standard waren, können jetzt bereits gründlich veraltet sein. Heute bekannte Gefahren waren möglicherweise zur damaligen Zeit noch nicht erkennbar. Hinzu kommt, dass verteilte IoT-Clients, die nicht nur in der Verbraucherwelt, sondern insbesondere in gewerblichen Unternehmen Einsatz finden, normalerweise nicht mit der gleichen Sorgfalt oder so häufig wie PC- oder Server-Software gewartet werden. Trotzdem benötigen sie die gleiche Aufmerksamkeit. Während Verbraucher ihre Mobiltelefone durchschnittlich alle zwei Jahre auf den neuesten Stand bringen, werden Standardbüroprodukte – Telefone, Drucker, PC-Hardware usw. – oft als gegeben angesehen und weniger häufig, wenn überhaupt, nachgerüstet.

Industrieanlagen, wie z. B. in der metallverarbeitenden Industrie, können mit gewaltigen Investitionen verbunden sein und somit wird davon ausgegangen, dass sie bis ans Ende ihrer Lebensdauer funktionieren. So werden sie funktionsfähig gehalten, während gleichzeitig ihre Steuer- und Regelungssysteme angepasst werden müssen, um den Anforderungen der sich verändernden Prozessumgebung gerecht zu werden. Anfänglich kann das ein Upgrade nach vielleicht zehn oder zwanzig Jahren bedeutet haben. Sobald sie mit dem Internet verbunden sind, könnte nun schon eine nur um zwei Monate verspätete Installation eines bestimmten Updates plötzlich kritisch werden.

Die Schäden aus Cyberkriminalität sind sprunghaft angewachsen. In der Studie „Cost of Cybercrime“ aus dem Jahr 2017 wurden ein paar Schätzungen veröffentlicht.[2] Wie dort gezeigt, erreichten die Kosten durch Cyberkriminalität in den sieben Hauptländern (US, DE, JP, UK, FR, IT, AU) im Jahr 2017 mehr als 70 Milliarden USD. Jedoch ist bei den Kosten durch Cyberkriminalität auch der Schaden für den Ruf eines Unternehmens, sein Markenimage, seine Wettbewerbsposition, seine Umsätze und seinen Börsenwert zu berücksichtigen.

? Sind Sie darauf vorbereitet?

1 Quelle: Gartner (Januar 2017) – https://www.gartner.com/newsroom/id/3598917

2 Quelle: Ponemon Institute – https://www.accenture.com/t20170926T072837Z__w__/us-en/_acnmedia/PDF-61/Accenture-2017-CostCyberCrimeStudy.pdf

Ein verbreiteter Systemaufbau

Sehen wir uns nun an, wie aus heutigen Komponenten eine IoT-Lösung typischerweise zusammengestellt sein könnte. Im nächsten Abschnitt untersuchen wir dann, mit welchen Gefahren Sie aus der daraus resultierenden Systemarchitektur rechnen sollten.

Während Smart-Clients für komplexe Anwendungen typischerweise über ihre eigenen individuellen Anwendungsprotokolle verfügen, verwenden die namhaftesten IoT Cloud Service Provider einfache und effektive Standardprotokolle für die Datenübertragung bei der Verbindung von Clients. Der Message Queue Telemetry Transport (MQTT[3]) ist ein typisches Beispiel – ein schlankes, standardisiertes Message-Protokoll. Es ermöglicht Betreibern, Topics festzulegen, die von individuellen Client-Programmen nach Wunsch gesendet oder empfangen werden können. Die zentrale Anforderung ist, dass es mindestens einen Server gibt, der als Broker fungiert und die von verschiedenen Clients übertragenen Topics empfängt und weiterleitet. Melden sich Clients für spezielle Topics an, werden sie vom Broker informiert.

Ein MQTT-Broker unterstützt bis zu drei verschiedene Servicequalitäten (Quality of Service – QoS). Level 0 bedeutet, dass eine Nachricht maximal ein Mal geliefert wird. Es wird ein bester Zustellversuch unternommen, jedoch wird keine Garantie oder Empfangsbestätigung durch den Empfänger geboten, ebenso wird kein erneuter Zustellversuch unternommen. Im QoS-Level 1 wird eine Nachricht mindestens ein Mal geliefert, was bedeutet, dass der Sender die Nachricht speichert und sie erneut sendet, bis er eine Empfangsbestätigung erhält. Das höchste QoS-Level 2 bietet die Garantie, dass die Nachricht exakt ein Mal geliefert wurde.

Neben diesen QoS-Leveln unterstützen MQTT-Broker auch die Transportschichtsicherheit (Transport Layer Security – TLS). Sie wird zur Authentifizierung der Identität des Clients und zum Schutz der Integrität und Vertraulichkeit des Inhalts der Nachricht verwendet. Hier kann ein Systemarchitekt auswählen, ob er nur die serverseitige Authentifizierung nutzen, einen Nutzernamen und ein Passwort für den Client zur gegenseitigen Authentifizierung hinzufügen oder X509-Zertifikate für den Client verwenden will.

Kommerzielle und auch teilweise kostenlose MQTT-Broker werden von den namhaften IoT Cloud Service Providern wie Amazon Web Services (AWS), Microsoft Azure, IBM Watson, Google, ThingWorx, Salesforce, CISCO Jasper, SAP HANA, BOSCH und General Electric Predix betrieben. Jedoch stehen für die Unternehmen, die es vorziehen, ihre eigenen Daten selbst zu hosten, auch Open-Source- und öffentliche Versionen zur Verfügung. Letzteres kann unter bestimmten Umständen von Vorteil sein, da die Daten lokal beim Unternehmen auf seinen eigenen Servern verbleiben, geht aber zulasten von Wartung, Schulung und Skalierfreundlichkeit.

Einer der bekanntesten öffentlich verfügbaren MQTT-Broker ist Mosquitto.[4] Er umfasst ein Client-Programm und lässt sich auf Ihrem Computersystem innerhalb von weniger als fünfzehn Minuten installieren. Fünfzehn Minuten später wären Sie dann in der Lage, Topics zu empfangen und zu senden. Für Mosquitto gibt es ebenfalls Open-Source-Bibliotheken, die Sie bei Bedarf zum Erstellen Ihrer eigenen Clients verwenden können.

Nachdem dies nun alles bekannt ist, kann ein Unternehmen problemlos einen Client erstellen: Daten von Sensoren oder anderen Akteuren auswählen, dann das geschäftliche Know-how mit den benannten Bibliotheken und Programmen zusammenführen und die Daten in die Cloud hochladen, unabhängig davon, welche Art von IoT Cloud Service Provider verwendet wird. Selbst die Prozesse für die Zugangsverwaltung und den Dashboard-Aufbau sind gut dokumentiert, sodass Sie in nur kurzer Zeit ein erstes System zum Laufen bringen sollten.

3 MQTT-Protokollspezifikation: http://mqtt.org/documentation

4 Eine umfassendere Übersicht findet sich unter: https://www.postscapes.com/internet-of-things-platforms/

Gefahren, die berücksichtigt werden sollten

Nehmen wir an, dass Sie sich entschieden haben, ein System auf der Grundlage der Produkte und der Architektur einzurichten, die wir im vorherigen Abschnitt beschrieben haben, so wollen wir in diesem Abschnitt auf ein paar Gefahren eingehen, die Sie kennen sollten. Falls solche Gefahren Ihr System beeinträchtigen könnten oder, was noch schlimmer ist, zu einer Beeinträchtigung anderer Systeme in Ihrem Unternehmen führen, müssen Sie sich mit ihnen auseinandersetzen und Lösungen entwickeln, um die Risiken und Schäden von Cyberangriffen zu verringern oder zu vermeiden.

Datenmanipulation oder Integritätsverlust:
Das bedeutet, dafür zu sorgen, dass sich Sender und Empfänger gegenseitig mit starken, eindeutigen Kennungen authentifizieren und somit die Datenübertragung vertraulich gehalten und ihre Integrität verifiziert wird.

Bei einem konservativen Ansatz würden Sie eine gegenseitige Authentifizierung auf der Grundlage des X509-Zertifikat-Managements wählen. Diese Option sollte von jedem renommierten IoT Cloud Service Provider mittels des von Ihrem Broker zur Verfügung gestellten TLS-Protokolls angeboten werden.

Jedoch sind die Sicherheitsprobleme hier noch nicht zu Ende. Wann immer es um Internetsicherheit und Kryptografie geht, sollten Sie an Themen in Verbindung mit der Kryptografie und der Schlüsselverwaltung denken, wie:

- Wer hat das verwendete Schlüsselmaterial generiert und signiert? Besteht ein Risiko, dass Informationen durchsickern?
- Welche Algorithmen und Schlüsselstärken wurden gewählt und wann läuft die Gültigkeit für sie ab? Nach Stand der Technik ist die aktuelle Schlüsselstärke 112 Bit und wird mit den folgenden Algorithmen und Schlüssellängen umgesetzt: ECDSA 224, RSA 2048, AES 128 und TDES 112.
- Wo sind die Schlüssel gespeichert, wie sind sie geschützt und wer hat Zugriff auf das Schlüsselmaterial?
- Sind Prozesse vorhanden, um das Schlüsselmaterial regelmäßig zu erneuern?
- Welche Art von Cipher-Suites, kryptografischen Algorithmen, Schlüsselvereinbarungsprotokollen und Zufallszahlengeneratoren wurde verwendet? Sie sollten bedenken, dass TDES – ein noch vor wenigen Jahren verbreiteter Algorithmus – heute veraltet ist.

Eindringen und Penetration in das System:
Verbindet sich Ihr Client mit dem Broker immer über TLS oder gibt es andere Ports, die weitere Dienste anbieten und daher offen gelassen wurden? Hört Ihr System auf eingehende Ereignisse, ohne dass eine Authentifizierung erforderlich ist?

Falls dies der Fall ist, muss ein Hacker z. B. die Version des installierten Betriebssystems ermitteln, um einen der im Internet vorhandenen Exploits nutzen zu können. Sie sollten nicht überrascht sein, dass diese Informationen auch gleichermaßen für Angriffe auf Ihr System genutzt werden können. Für Penetrationtests stehen Anleitungen, Open-Source-Werkzeuge und sogar ganze Systeminstallationen zur Verfügung.[5]

Unautorisierte Softwaremodifikation:
Software wird normalerweise durch einen initialen Ladeprozess gestartet, unabhängig davon, ob es sich um einen einfachen Bootloader aus Ihrer Entwicklung oder eine komplexe Variante eines Betriebssystems handelt. Am Ende muss Ihre Software von irgendeinem persistenten Speicher in einen anderen Speichertyp eingelesen werden, wo sie durch ein Verarbeitungssystem ausgeführt wird. Möglicherweise wollen Sie diesen Teil der Software schützen, sodass er nicht von unautorisiertem Personal modifiziert oder ersetzt werden kann.

Smartcard

Trusted Platform Module (TPM)

Hardware-Sicherheitsmodul (HSM)

5 https://www.gitbook.com/book/adi0x901/iot-pentesting-guide/details, https://www.usenix.org/system/files/conference/ase16/ase16-paper-chothia.pdf, https://www.kali.org/

- Enthält Ihre Software einen Satz von mehreren Dateien, wollen Sie gegebenenfalls überprüfen, ob die kombinierte Konfiguration noch gültig ist.

- Wenn Ihre Software spezielles geistiges Eigentum enthält, sind Sie vielleicht auch daran interessiert, die Software vertraulich zu speichern.

- Wird Ihr Produkt an einen Ort in einer nicht kontrollierbaren Umgebung versendet und dort betrieben oder können Sie nicht kontrollieren, ob die Betreiber Zugang zu den physischen Einrichtungen Ihres Produkts haben, wollen Sie eventuell bestimmte kritische Sicherheitsparameter vertraulich halten.

- Ist Ihr Produkt sensibel gegenüber Vervielfältigungen oder benötigt es eine eindeutige Identifizierung, die vor unautorisierter Ersetzung oder Modifikation geschützt ist, müssen Sie gegebenenfalls Gegenmaßnahmen implementieren.

Haben Sie vielleicht schon darüber nachgedacht, wie Ihre Client-Software vor vorsätzlich durch einen Angreifer ausgeführten oder zufälligen Manipulationen geschützt werden kann, wenn zum Beispiel ihre Komponenten während eines Updates der Konfiguration ausgetauscht werden.

Diese Fragen fallen bei einer Vielzahl von kryptografischen Standards unter das Thema Key-Management. Ein Beispiel ist der vom National Institute of Standards and Technology (NIST) in den USA veröffentlichte FIPS 140-2, ein Informationsverarbeitungsstandard für Bundesbehörden.

Der FIPS 140-2 gilt für kryptografische Module und definiert vier Level für die Anforderungen an software- und hardwarebasierte Lösungen. Die höheren Level gelten normalerweise nur für kryptografische Hardwaremodule, die für den physischen Schutz von Firmware und kritischen Sicherheitsparametern sorgen. Es gibt auch noch andere Standards für kryptografische Algorithmen, Modi und Protokolle. Sie sind bei Designs zu berücksichtigen, die den Standard FIPS 140-2 erfüllen. Das NIST veröffentlicht Implementierungsrichtlinien (Implementation Guidance – IG), die zweimal pro Jahr aktualisiert werden. Soll ein Modul validiert werden, muss es allen Bestimmungen dieser Richtlinien genügen. Das Ziel besteht in der Gewährleistung einer Sicherheit nach dem Stand der Technik.

Es wurden einige kommerzielle und Open-Source-Software-Bibliotheken, wie z. B. Microsoft oder OpenSSL, auf der Grundlage von FIPS 140-2 Level 1 validiert, wobei es sich aber um Ausnahmen handelt. Für den physischen Schutz müssen Sie natürlich zumindest ein paar physisch sichere Elemente in Ihr Konzept einbauen.

Wir hoffen, dass Sie hier ein paar Anregungen finden, wie Sie von der Integration physischer Sicherheit in Ihre Lösungen profitieren können!

Integration von physischer Sicherheit in Ihre IoT-Anwendung

Während einige der vorstehend genannten Gefahren Aspekte einer sorgfältigen Systemkonfiguration betreffen, erfordern andere grundlegende konzeptionelle Entscheidungen. Es gibt einen großen Unterschied zwischen der Implementierung Ihres Clients in einer Software und der Speicherung der notwendigen kritischen Sicherheitsparameter im ungeschützten Speicher oder der Wahl einer eingebetteten Lösung mit einer höheren Datensicherheit.

Die Integration von physischer Sicherheit in Ihre IoT-Anwendung kann bei der Reduzierung von Risiken hinsichtlich der Cybersicherheit helfen. Es gibt verschiedene sichere Elemente, unter denen ausgewählt werden kann:

- *Einfache sichere Elemente:* Im Wesentlichen handelt es sich dabei um einen kryptografischen integrierten Schaltkreis (IC), der eine eindeutige Identifizierung ermöglicht, mit der Sie sicher eine Stelle authentifizieren können, mit der der IC verbunden ist. Einige können auch zum Austausch eines Sitzungsschlüssels (Session Key) auf der Grundlage ihrer Identität verwendet werden. Wenn diese Komponente ohne Autorisierung – entweder unbeabsichtigt oder in böswilliger Absicht – ausgetauscht wird, würde dies auf ähnliche Weise erkannt werden, als wenn Sie versuchen würden, sich mit einem falschen Benutzernamen oder einem falschen Passwort auf einer Internetseite einzuloggen. Manchmal werden auch zusätzliche Merkmale angeboten. Die Funktionalität dieser Geräte, die preisgünstig sind, ist normalerweise auf die Authentifizierung reduziert: Zum Beispiel bieten sie keine kryptografischen Primitive zur Generierung und Nutzung von Schlüsselmaterial für andere Zwecke.

- *Trusted Platform Module (TPM):* Im Wesentlichen ist ein TPM ein sicherer Einzelchip-Coprozessor, der kryptografische Schlüssel speichern und kryptografische Primitive bereitstellen kann, die mit diesen Schlüsseln verwendet werden. Die Idee wurde ursprünglich von Microsoft, Intel und HP konzipiert. Kernstück eines TPM sind der „Endorsement Key“ und der „Storage Root Key“. Der„Endorsement Key“ wird während der Produktion in die TPM-Hardware gebrannt, während der „Storage Root Key“ verwendet wird, um anderes, vom TPM generiertes Schlüsselmaterial zu schützen, das jedoch vom TPM nach der Initialisierung generiert wird. Ein TPM enthält auch Firmware zur Bereitstellung einer Reihe von kryptografischen Primitiven.

TPM werden gewöhnlich auf die Hauptplatinen gelötet und lassen sich einfach austauschen. Ihr Hauptziel besteht im Schutz der Integrität der Hostplattform, jedoch können sie auch sichere Speicher- und Verschlüsselungs-/Entschlüsselungs-Primitive für Anwendungen bereitstellen. Moderne PCs und Server sind normalerweise mit TPM ausgestattet, die von den Bootloadern verwendet werden, um den authentifizierten sicheren Boot-Prozess zu verifizieren, der das Hauptbetriebssystem (z. B. Windows oder Linux) startet. Ein TPM implementiert selbst keine Art von Schlüsselverwaltung, sondern verlässt sich diesbezüglich auf externe Software.

- *Sichere Smartcard:* Smartcards sind integrierte Einzelchip-Schaltkreise, die wie ein TPM eine begrenzte sichere Speicherung für Schlüsselmaterial und einen primitiven Satz kryptografischer Funktionen bieten. Sie schützen das Schlüsselmaterial vor der Weitergabe oder Substitution auf einem höheren physischen Level, d. h. durch Einbeziehung von Manipulationserkennung und Reaktionsmechanismen. Ihre kryptografischen Operationen werden gewöhnlich auf Performance optimiert. Anders als TPM gibt es Smartcard-Versionen mit eigenem Betriebssystem, die somit auch Dienste für die Schlüsselverwaltung unterstützen. Insbesondere ermöglicht diese Technologie die sichere Identifizierung von Nutzern und gestattet auch die Aktualisierung von Daten und Firmware, ohne dass die installierten Karten ersetzt werden müssen. Noch ausgefeiltere, zugelassene Betriebssysteme ermöglichen den Betrieb von einzelnen und mehreren Anwendungen auf Java-basierten virtuellen Maschinen auf den Karten.

  Wie TPM sind Smartcards Einzelchip-Lösungen, die Strom von außen und ein Taktsignal benötigen. Diese Leitungen sind oft für sogenannte Seitenkanalattacken anfällig. Diese Attacken sind so angelegt, dass geheime Daten von einem System durch Ausspähen von Faktoren wie Signal-Timing oder Leistungsverbrauch des Geräts beim Rechnen gestohlen werden. Eine weitere Einschränkung von Smartcards besteht in ihrem begrenzten Speicherplatz für Daten und Programme.

- *Hardware-Sicherheitsmodul (HSM):* Hier meinen wir ein hardwarebasiertes kryptografisches Modul, das offiziell gemäß dem FIPS-140-2-Standard validiert wurde. HSM haben verschiedene Formen: z. B. Einzelchip-, autonome Multichip- oder eingebettete Multichip-Module. Wie TPM und Smartcards bieten HSM sichere Speicherung und verschiedene Dienste für die Generierung, Speicherung, Nutzung und Pflege von kritischen Sicherheitsparametern wie Schlüsseln, Passwörtern und anderen vertraulichen Daten. Sie werden normalerweise als kryptografische Coprozessoren verwendet und ihre Multichip-Versionen unterstützen im Allgemeinen einen umfassenden Bereich von Diensten und die Speicherung. Sie können auch ihre eigene batteriebetriebene Schaltung und Spannungsüberwachung enthalten, die die Integration einer Echtzeituhr zur korrekten Zeiterfassung und Zeitstempelung ermöglicht, um zu gewährleisten, dass das abgelaufene Schlüsselmaterial nicht länger genutzt werden kann. Auch können sie einen redundanten Speicher enthalten, der mehrere Technologien zur zusätzlichen Datensicherheit verwendet.

  Normalerweise werden HSM-Anwendungen ebenfalls gemäß FIPS 140-2 validiert; somit können nur zugelassene Anwendungen in das HSM zur Erweiterung des Umfangs seiner Dienste geladen werden. In Abhängigkeit vom Level werden Dienste mit Rollen verlinkt, sodass zum Beispiel die Administration nur durch autorisierte Nutzer erfolgen kann. Für Level 3 ist eine identitätsbasierte Authentifizierung erforderlich, sodass eine Nutzung nur durch autorisierte Nutzer möglich ist. Ausfallschutz und Prüfung von Betriebsparametern (Spannung undTemperatur), Manipulationserkennung und Reaktionsmechanismen sind vorhanden und schützen die speziellen Anwendungen des Kunden. HSM verfügen normalerweise über einen sicheren Update-Mechanismus, der die Erweiterung oder den Ersatz der Sicherheitsmerkmale ermöglicht, um somit den sich ständig verändernden Bedürfnissen der Sicherheitslandschaft zu entsprechen.

  Um ein FIPS-140-2-Zertifikat zu erhalten, muss der Anbieter sein Gerät zur intensiven Prüfung bei einem akkreditierten Labor einreichen. Dieser formale Zulassungsprozess ist als Cryptographic Module Validation Program (CMVP) bekannt.

  FP bewertet das Niveau der physischen Sicherheit in der Reihenfolge, wie in der Abbildung unten aufgeführt.

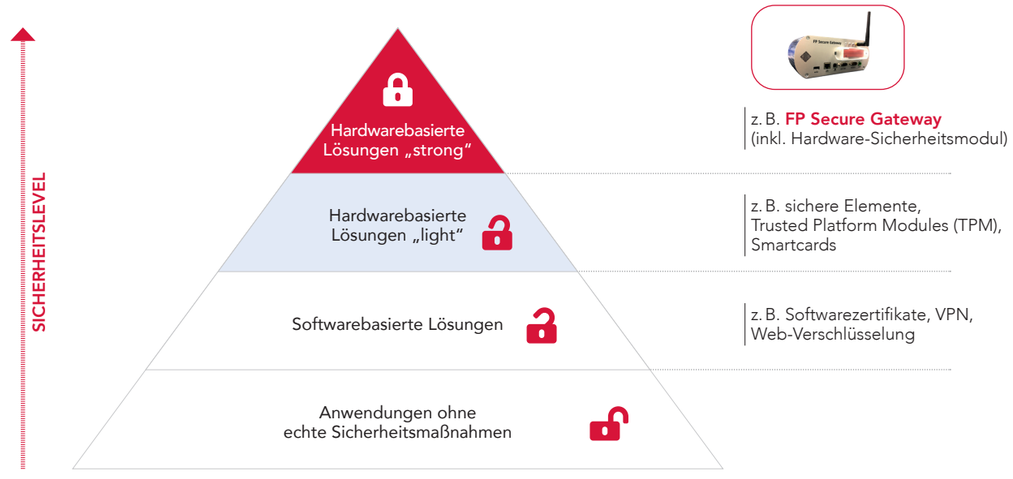

Abbildung 1: Sicherheitspyramide

Einsatz von Hardwaresicherheit zum Schutz von TLS-Schlüsselmaterial von Clients in feindlichen Umgebungen

In diesem Abschnitt wollen wir alle vorstehenden Fakten zusammenführen und zeigen, wie eine Konfiguration, einschließlich IoT Secure Gateway von **FP**, Mosquitto und einer Schnittstelle zu einem IoT Cloud Service Provider (der nachstehend als ICSP bezeichnet wird), wie z. B. Amazon Web Services, aussehen kann. Das in den nachstehenden Abbildungen gezeigte IoT Secure Gateway verwendet eine CAN-Bus-Schnittstelle zum Auslesen von Daten aus Sensoren einer Industrieanlage und nutzt dabei ein Hardware-Sicherheitsmodul zum Schutz der kritischen Sicherheitsparameter des Betreibers.

Normalerweise verfügt der ISCP über eine Public Key Infrastructure unter seiner eigenen Root-Zertifizierungsstelle (Root Certificate Authority – CA), wofür wir in der nachfolgenden Abbildung die Bezeichnung ICSP RootCA verwenden. Sie kann ebenfalls von einer anerkannten internationalen, vertrauenswürdigen Stelle wie VeriSign, D-TRUST, Deutsche Telekom, Microsoft oder GlobalSign signiert sein.

Ein Schlüsselpaar (hier z. B. das MQTT-Broker-Schlüsselpaar) besteht aus einem privaten Schlüssel (Private Key) und einem öffentlichen Schlüssel (Public Key), dessen Gültigkeit durch ein Zertifikat vorgeschrieben ist. Das Zertifikat enthält Informationen über den Typ und den Zweck des Schlüssels und seine Lebensdauer. Abbildung 2 zeigt eine einfache Schlüsselhierarchie für unseren ICSP und seinen MQTT-Broker.

Ein grundlegendes Authentifizierungsverfahren

Es gibt verschiedene Optionen für die Generierung des Schlüsselmaterials, um Ihr IoT-Gerät eindeutig zu identifizieren. Eine unkomplizierte Methode ist, den ICSP das Schlüsselpaar für Sie generieren zu lassen. In diesem Fall generiert der ICSP ein einmaliges Schlüsselpaar und signiert den öffentlichen Schlüssel mittels des Root Certificate.

Sie erhalten drei Dateien: den privaten Geräteschlüssel für das IoT-Gerät, den entsprechenden öffentlichen Geräteschlüssel in einem Zertifikat und ein Zertifikat für die Root-Zertifizierungsstelle des ICSP. Die Dateien sind in Abbildung 3 dargestellt. Diese Dateien können in Ihrer Client-Anwendung verwendet werden. Open-Source- und kommerziell verfügbare MQTT-Implementierungen unterstützen dieses Schlüsselmaterial, wobei die daraus resultierende Konfiguration so ähnlich aussieht wie in Abbildung 4 dargestellt.

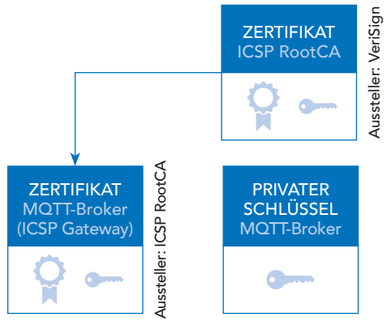

Abbildung 2: Schlüsselhierarchie für einen beispielhaften IoT Cloud Service Provider (ICSP)

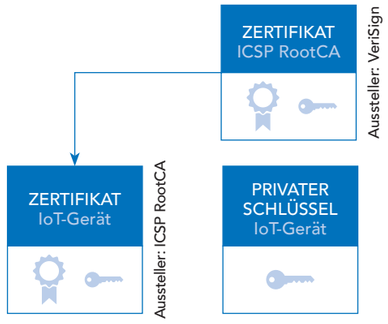

Abbildung 3: Schlüsselmaterial, das von einem IoT Cloud Service Provider (ICSP) bereitgestellt wird

Zu beachten ist, dass in diesem Fall das Schlüsselmaterial von Ihrem ICSP generiert wird. Das ICSP Gateway in den folgenden Abbildungen wird als MQTT-Broker bezeichnet.

Wenn sich Ihre Client-Anwendung mit dem Cloud Service Provider verbindet, verwendet sie den Geräteschlüssel, um sich beim MQTT-Broker des ICSP zu authentifizieren. Der Broker ist in der Lage, die Anfrage zu verifizieren, da er seine eigene Root-Zertifizierungsstelle vom Gerätezertifikat erkennt. Gleichermaßen authentifiziert sich der MQTT-Broker unter Verwendung seines eigenen Serverzertifikats gegenüber dem IoT-Gerät. Da dieses Zertifikat auch von derselben Zertifizierungsstelle ausgestellt wird und das Gerät über eine Kopie dieses Zertifikats verfügt, kann das IoT-Gerät ebenfalls den Broker verifizieren. Auf diese Weise sind die Seiten in der Lage, sich gegenseitig selbst zu authentifizieren, sodass eine sichere, authentifizierte Transportschichtsitzung aufgebaut werden kann. So weit, so gut.

Restrisiken

Diese Art der Konfiguration beinhaltet jedoch ein Restrisiko für den Betreiber des Geräts. Das Problem besteht darin, dass der private Schlüssel des IoT-Geräts extern für das Unternehmen, das das Gerät betreibt, generiert wurde. Auf irgendeine Weise muss dieser Schlüssel an den Betreiber des Geräts übertragen werden, was normalerweise über eine andere Art sichere Sitzung erfolgt. Der private Schlüssel geht zwangsweise durch eine Anzahl von Händen, wie z. B. durch die eines Administrators im Unternehmen, und könnte unterwegs unter Umständen auf unsichere Weise gespeichert werden. All dies stellt ein Sicherheitsrisiko dar und schafft Möglichkeiten, wie in einer feindlichen oder Wettbewerbsumgebung ein externer oder interner Hacker die Sicherheit des Systems gefährden könnte.

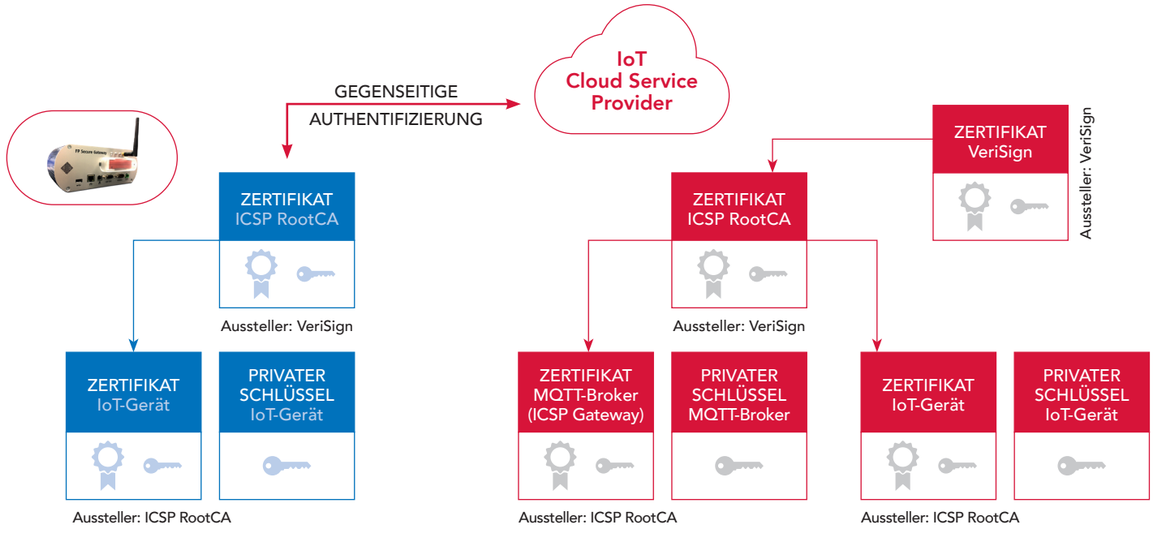

Abbildung 4: Verwendung von Schlüsselmaterial, das vom ICSP generiert wurde

Ein alternatives Verfahren

In Anerkennung dieses Problems unterstützen große IoT Cloud Service Provider gewöhnlich ein alternatives Verfahren, das von **FP** zum Wirksamkeitsnachweis (Proof of Concept) mit Amazon Web Services (AWS) geprüft wurde. Bei diesem Verfahren ermöglicht AWS Kunden, ihre eigene Sub-Zertifizierungsstelle zur Ausstellung von Zertifikaten zu registrieren, die im Zusammenhang mit einem HSM verwendet werden können. In unserem Beispiel ist die zertifizierende Sub-Zertifizierungsstelle das **FP** IoT Data Center.

FP betreibt sein eigenes Trust Center, d. h. eine Hochsicherheitsumgebung, in der **FP** sein eigenes Root Certificate – das **FP** RootCA – generiert hat. **FP** hat auch andere Schlüssel und Zertifikate für seine eigenen Rechenzentren, einschließlich das IoT Data Center, dort generiert. Das Rechenzentrum speichert diese Schlüssel sicher in seiner Serverumgebung.

Die Produktion von HSM von **FP** beinhaltet, dass jedes mit einer Kette von Zertifikaten vorab geladen wird, wozu sowohl die für das **FP** RootCA als auch die für das IoT Data Center gehören. Dies erfolgt in einer Hochsicherheitsumgebung und ermöglicht dem HSM die Identifizierung der Server von **FP** in der Zukunft. Des Weiteren generiert das HSM seinen eigenen privaten und öffentlichen Geräteschlüssel. Der private Schlüssel verlässt nie das Gerät, jedoch wird der öffentliche Schlüssel an die **FP** Server zur dortigen Signierung weitergegeben. Im Gegenzug erhält das HSM sein personalisiertes, von **FP** signiertes IoT-Gerätezertifikat. Sobald dies erfolgt ist, ist jeder Server, der RootCA oder Data-Center-Zertifikate von **FP** erkennt, ständig in der Lage, dieses HSM als echtes **FP** Gerät zu authentifizieren und dafür auch eine eindeutige Kennung vorzuhalten. Somit lassen sich leicht gegenseitig authentifizierte TLS-Verbindungen aufbauen.

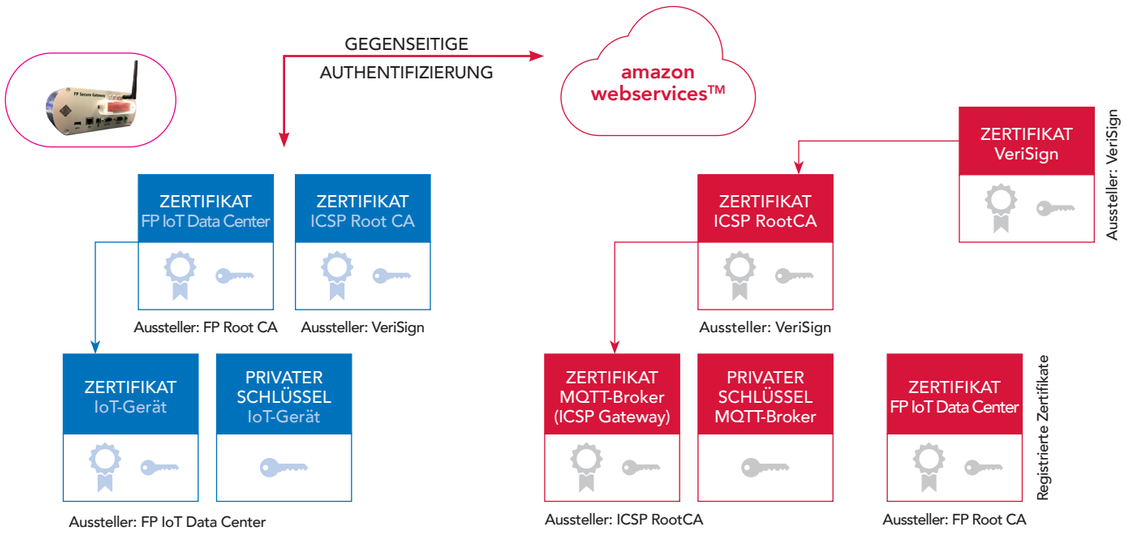

Abbildung 5: Registrierte Sub-Zertifizierungsstelle unter Verwendung von selbst generiertem Schlüsselmaterial

FP muss lediglich noch sein IoT Data Center als Zertifikat ausstellende Sub-Zertifizierungsstelle bei AWS registrieren. Die sich daraus ergebende Konfiguration sieht der in Abbildung 5 gezeigten ähnlich.

Dieses Verfahren bietet eine Reihe von Vorteilen:

- *Vertraulichkeit:* Der private Schlüssel des IoT-Geräts verlässt nie sein HSM.
- *Verwaltung:* Neue Schlüssel und die Einführung von anderem Schlüsselmaterial sind bei diesem Verfahren ziemlich einfach, indem Standardverfahren des Key Management zur Anwendung kommen.
- *Skalierbarkeit:* Wenn Sie mehrere Clients betreiben, müssen Sie nicht jeden Client manuell bei Ihrem ICSP registrieren, da der ICSP die registrierten Root-Zertifikate erkennt und automatisch Ihre Geräte anmelden kann, wenn sie sich zum ersten Mal verbinden.

Diese Architektur ist gleichermaßen für Clients geeignet, die ein TPM, eine Smartcard oder wie im **FP** Secure Gateway ein HSM nutzen. Die HSM von **FP** generieren während der Fertigung intern ihr erstes Schlüsselmaterial und erhalten signierte Zertifikate vom IoT Data Center von **FP**. Da das **FP** IoT Data Center bei AWS registriert wurde, kann dies verwendet werden, um das IoT-Gerät gegenüber dem MQTT-Broker von AWS zu authentifizieren. Der MQTT-Broker bleibt noch beim IoT-Gerät authentifiziert, wenn das IoT-Gerät die Root-Zertifizierungsstelle von AWS speichert.

Will ein Unternehmen **FP** als Sub-Zertifizierungsstelle bei seinem AWS Account anmelden, muss es seinen Registrierungscode von AWS bei **FP** angeben. In einem einmaligen, sicheren Prozess generiert **FP** ein Verifizierungsschlüsselpaar, das für den Registrierungsprozess benötigt wird, sowie ein entsprechendes Certificate Signing Request (CSR) mit dem Registrierungscode des Unternehmens als Common Name im Zertifikat. Zum Schluss verwendet **FP** das Schlüsselmaterial seines IoT Data Center und stellt auf diesen unternehmensspezifischen öffentlichen Verifizierungsschlüssel ein Zertifikat aus.

Sowohl das Zertifikat des IoT Data Center von **FP** als auch das Zertifikat für den unternehmensspezifischen Verifizierungsschlüssel werden dem Unternehmen übergeben, das nun diese Zertifikate zur Anmeldung des IoT-Data Center von **FP** bei seinem AWS Account verwenden und aktivieren kann, wie z. B. über die AWS Kommandozeile (Command Line Interface – CLI). Das Zertifikat für den unternehmensspezifischen Verifizierungsschlüssel wird von AWS nur ein Mal im Rahmen der Registrierung zum Nachweis benötigt, dass der Aussteller des Zertifikats den zu registrierenden privaten Schlüssel der Zertifikate besitzt. Wenn Sie noch mehr über die Feinheiten dieses Verfahrens erfahren wollen, steht Ihnen eine Dokumentation online zur Verfügung. [6]

6 https://docs.aws.amazon.com/iot/latest/developerguide/device-certs-your-own.html

Vorteile aus zusätzlichen Diensten eines HSM

Nach Prüfung Ihrer betrieblichen Anforderungen können Sie sich, wenn Sie sich für ein gemäß FIPS 140-2 Level 3 zugelassenes HSM entschieden haben, sicher sein, dass das Gerät und seine Firmware von einem unabhängigen, akkreditierten Labor zugelassen wurden. Im Fall von Francotyp-Postalia wird die HSM-Produktion auch regelmäßig durch die Regulierungsbehörden der Post auditiert.

Im Gegensatz zu anderen Hardwareansätzen bietet die Verwendung von HSM Möglichkeiten für die Integration kundenspezifischer Dienstleistungen, die in einigen nachfolgend aufgeführten Beispielen beschrieben sind. Und sollte es erforderlich sein, einen Dienst oder einen anderen Algorithmus hinzuzufügen, kann das HSM von **FP** stets über Fernzugriff aktualisiert werden.

Beispiele für zusätzliche Dienste sind:

Validierte Algorithmen und Schlüsselstärken zur Erhöhung der Vertrauenswürdigkeit:

HSM sind gemäß FIPS 140-2 validiert. Dies erfordert, dass alle Algorithmen und gewählten Schlüsselstärken zum Zeitpunkt der Evaluierung dem Stand der Technik gemäß den Empfehlungen und Implementierungsrichtlinien des NIST entsprechen. Derzeit gehören dazu die Verwendung von symmetrischen Algorithmen wie TDEA, AES und Hashing und deren Kombinationen wie HMAC, verschiedene Blockverschlüsselungsmodi wie CBC und CTR, asymmetrische Algorithmen wie RSA und ECDSA, Signatur-schemata, Schlüsselableitungsfunktionen, Schlüsselvereinbarungsfunktionen, Key-Wrapping-Funktionen, Deterministic Random Bit Generators (DRBG) sowie True Random Number Generators (TRNG) und deren Entropie-Aussagen.

Verbesserte Kapselung:

HSM schützen Schlüsselmaterial physisch vor Modifikationen oder Ersetzungen. Die Verwendung validierter Firmware beschränkt das Risiko einer Einflussnahme durch Malware oder Ransomware. Jedes Gerät bietet eine eindeutige Identifikationsnummer, mit der sich die Identität jedes Geräts und sein physischer Standort unterscheiden lassen.

Pay-per-Use:

Die HSM von **FP** bieten eine Palette von Diensten an, die zum Herunterladen von Geldbeträgen von einem Kundenkonto im Data Center von **FP** genutzt werden können. Das Herunterladen von Geld kann mit anderen Ereignissen und Triggern, die im IoT Gateway definiert sind, kombiniert werden. Sensorereignisse können verwendet werden, um Micro-Accounting-Transaktionen auszulösen. Für jede Transaktion ist das HSM zum Beispiel in der Lage, einen kryptografisch signierten Zahlungsbeleg bereitzustellen.

Zeitstempelung:

Durch die Verwendung einer Echtzeituhr, wie es in den HSM von **FP** Standard ist, ist in Kombination mit Unterschriften die Generierung eines fälschungssicheren Zeitstempels auf einem Datensegment, in einem Protokoll oder in einem Dokument möglich.

Verschlüsselung und Entschlüsselung von Firmware:

Durch die Verwendung von im HSM enthaltenen symmetrischen Schlüsseln kann das HSM Anwendungs-Firmware für Dritte empfangen und sie für die Speicherung auf einem anderen Gerät verschlüsseln. Vor der Ausführung müssten diese Dritten das HSM besitzen, um die Firmware zu entschlüsseln und auszuführen. In diesem Fall dient das HSM als ein Schlüssel-Dongle.

Firmware-Update:

Vom Kunden können andere Dienste festgelegt und sicher in das HSM heruntergeladen werden. Dies ist durch den Firmware Load Service in den HSM von **FP** möglich. Dieser Service kann verwendet werden, um die HSM-Dienste sicher zu nutzen und zu erweitern. Nur signierte Firmware kann in das Gerät geladen werden.

FP GERMAN MAILGENEERING

FP ist der Spezialist für sicheres Mail-Business und sichere digitale Kommunikationsprozesse.

Dirk Rosenau
FP InovoLabs GmbH
Prenzlauer Promenade 28
13089 Berlin
Deutschland
www.inovolabs.com

fp-francotyp.com