Making Everything Easier!™

INSIDE Secure Edition

IoT Security

FOR DUMMIES®

A Wiley Brand

Learn:

- Why you need security for IoT
- Where IoT devices are vulnerable
- How to choose the right IoT security solution

Brought to you by

inside SECURE

Lawrence Miller

About INSIDE Secure

INSIDE Secure (Euronext Paris FR0010291245 – INSD) provides comprehensive embedded security solutions. World-leading companies rely on INSIDE Secure's mobile security and secure transaction offerings to protect critical assets including connected devices, content, services, identity, and transactions. Unmatched security expertise combined with a comprehensive range of IP, semiconductors, software, and associated services gives INSIDE Secure customers a single source for advanced solutions and superior investment protection.

For more information, visit **www.insidesecure.com**.

IoT Security

FOR DUMMIES®

A Wiley Brand

INSIDE Secure Edition

by Lawrence Miller

FOR DUMMIES®

A Wiley Brand

IoT Security For Dummies®, INSIDE Secure Edition

Published by: **John Wiley & Sons, Ltd.,** The Atrium, Southern Gate Chichester, West Sussex, www.wiley.com

Registered Office

John Wiley & Sons, Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

ISBN 978-1-119-21119-8 (pbk); ISBN 978-1-119-21118-1 (ebk)

Printed in Great Britain

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. For details on how to create a custom *For Dummies* book for your business or organization, contact info@dummies.biz or visit www.wiley.com/go/custompub. For details on licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

Some of the people who helped bring this book to market include the following:

Project Editor: Carrie A. Johnson

Acquisitions Editor: Katie Mohr

Editorial Manager: Rev Mengle

Business Development Representative: Felicity Whyte

Special Help: Mikael Dubreucq, Trevor Daughney, Benoit Makowka, Laurent Sustek

Production Editor: Kumar Chellappan

Table of Contents

Introduction

Today, advances in networking and semiconductor technologies, along with the ever-widening grid of interconnected and computationally capable products (or *nodes*), have ushered in the Internet of Things (IoT).

The IoT creates opportunities for business and industry. Imagine a world where you wake up and your house is already heated to a comfortable temperature and your coffee is ready — not preset to a specific time, but based on when you actually wake up. Your refrigerator reminds you that you need to pick up eggs and milk on your way home. And your vehicle alerts you of an impending problem, automatically contacts the nearest dealer to verify parts availability, and schedules your drop-off appointment!

The IoT also creates many new — and potentially more harmful, even lethal — security threats that must be proactively and continuously addressed at every step in the product life cycle and value chain. Imagine a remote attacker turning the air conditioning on in your house in the middle of winter or making your coffee machine overheat, causing an electrical fire. And if that weren't enough to ruin your day, your vehicle suddenly accelerates and your steering wheel locks up in heavy traffic!

About This Book

This book explores the power and potential of the IoT (Chapter 1), why IoT security is necessary (Chapter 2), how attackers exploit weaknesses in IoT devices (Chapter 3), the basics of cryptography (Chapter 4), how to choose and integrate IoT security solutions (Chapter 5), and what security industry best practices exist and can be adapted to the IoT (Chapter 6).

Foolish Assumptions

I assume you're a product manager (for example, in a business unit, in product marketing, project management, risk management, or IT management) that needs to understand why IoT security is needed. Or perhaps you're an engineer that needs to understand how security needs to be integrated into an IoT device, or a general manager that needs to understand the business value of including security in your IoT products and services. If any of these assumptions describe you, then this book is for you! If none of these assumptions describe you, keep reading anyway because when you finish reading it, you'll know a few things about IoT security!

Icons Used in This Book

Throughout this book, I occasionally use special icons to call attention to important information. Here's what to expect:

This icon points out information that you should commit to your non-volatile memory or your noggin' — along with anniversaries and birthdays!

You won't find a map of the human genome here, but if you seek to attain the seventh level of NERD-vana, perk up! This icon explains the jargon beneath the jargon!

Thank you for reading, hope you enjoy the book, please take care of your writers! Seriously, this icon points out helpful suggestions and useful nuggets of information.

This icon points out the stuff your mother warned you about. Okay, probably not. But you should take heed nonetheless — you might just save yourself some time and frustration!

Beyond the Book

There's only so much I can cover in 48 short pages, so if you find yourself at the end of this book, thinking "gosh, this was an amazing book, where can I learn more?" just go to `www.insidesecure.com`.

Chapter 1

Getting to Know a Few Things about the Internet of Things

In This Chapter

- Understanding what "things" make up the Internet of Things
- Recognizing the rapid growth of the IoT
- Getting an overview of the architecture of an IoT device

The Internet of Things (IoT) means different things to different people, so to ensure we're on the same page, I begin by defining exactly what the IoT is and why it's become the next big thing!

Defining the IoT

You may have heard a few things about the IoT, but what exactly is the IoT? The IoT (sometimes also referred to as the *Internet of Everything*) is a network of physical objects (or "things") embedded with electronics, software, sensors, and connectivity which enable those objects to exchange data with the operator, manufacturer, service provider, and/or other connected devices.

The IoT is based on the infrastructure of the International Telecommunication Union's (ITU) Global Standards Initiative (IoT-GSI). The IoT-GSI covers devices and objects connected over multiple communications protocols — such as personal

computing devices, laptop or desktop computers, tablets, and smartphones — as well as devices that are connected to each other through other protocols, such as Bluetooth, ZigBee (an open, global wireless standard), Long Range Wide Area Network (LoRaWAN), and SIGFOX.

IoT devices (or nodes) often operate without a screen or any user interface at all, may rely on battery power for operation, and are usually dedicated to a single task.

IoT devices are typically described as "smart objects, edge devices, or connected devices," such as

- Networked home appliances that can be monitored or controlled remotely
- "Smart home" components, such as lighting, heating, or ventilation units with remote management/monitoring access
- Wearables, or connected clothing and fashion accessories
- Sensor networks
- Connected industrial and manufacturing equipment
- Networked vehicle telematics sensors
- Other embedded devices that are network-connected and computationally capable

What makes an IoT device *smart* typically falls into one or more of the following functional areas:

- **Monitoring:** Remote sensing and reporting of operating conditions, usage, and other external environmental factors. In certain "things" (such as wearable medical devices and home security systems), monitoring may be a device's primary purpose.
- **Control:** Enables certain device functions to be managed or customized remotely. For example, a vehicle can be started remotely or a home thermostat can be adjusted from across the room.
- **Optimization:** Monitoring and control capabilities enable device manufacturers to tune performance and efficiency based on historical and real-time operating data.

Preventive maintenance or problem diagnosis can also be performed remotely.

- **Automation:** Devices can operate autonomously and adapt to environmental or operational factors with minimal human interaction required.

IoT Growth Trends

For businesses, information technology developments are driving innovation in their products and services. The IoT is enabled by numerous technology trends:

- Devices are cost effective (for example, LED TVs are affordable compared to their prohibitive cost just 10 years ago).
- Infrastructure is in place and new infrastructure is being built to support future technology needs.
- The evolution of functions is in line with user expectations (for example, smart watches that provide multiple services and applications).

In a November 2014 press release, Gartner, Inc. predicted that in 2015 there would be 4.9 billion IoT devices in use worldwide, growing to 25 billion devices by 2020, with total services spending for the IoT growing from $69.5 billion to $263 billion during that same period.

In addition to these trends, several significant technological developments over the past decade have ushered in the rapid growth of smart, connected devices and the IoT. The mobile Internet revolution introduced a number of wireless protocols that have become popular as *over-the-air* (OTA) extensions of the traditional wired Internet.

A second critical development that has paved the way for the growth of the IoT is the readily available and economically feasible computational power brought on by the present level of miniaturization of integrated circuitry. Modern chip design combined with advanced semiconductor technology provides cost-effective and dense computational capability with sufficiently low power requirements, which enables IoT devices to be capable of computing on a level that was, in the past, restricted to very large, specialized computers.

Other critical technical developments include batteries (improved power density and battery life), an increased availability of components, and predictive analytics to enable deep insights.

IoT Device Architecture

Figure 1-1 illustrates an IoT communications architecture consisting of sensors and smart objects, smart devices and gateways, and backend data centers and services.

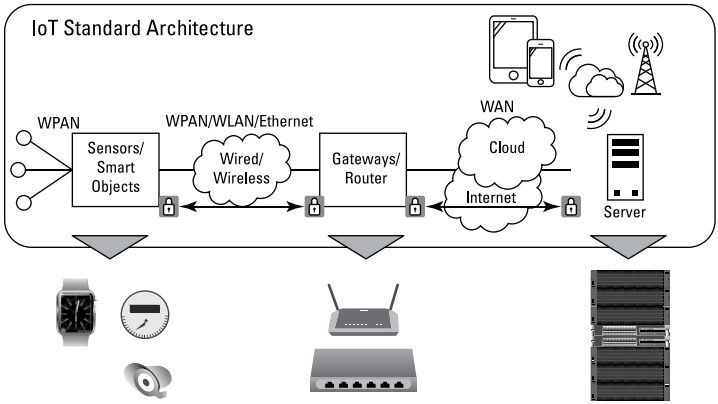

Figure 1-1: IoT system architecture and functional overview.

Figure 1-2 illustrates the device architecture of an IoT device consisting of the following components:

- **Sensors I/F (interface):** A sensor that monitors and collects data (such as a camera or thermometer)
- **Power management unit (PMU):** Switches individual device components on and off to save power
- **Host processor:** The "brain" of the device
- **Memory:** Working memory and storage memory (flash), storing data and/or application code
- **Keyboard/display:** The man–machine interface
- **Connectivity:** The communication link with the rest of the network (for example, Ethernet, WiFi, Bluetooth and others)

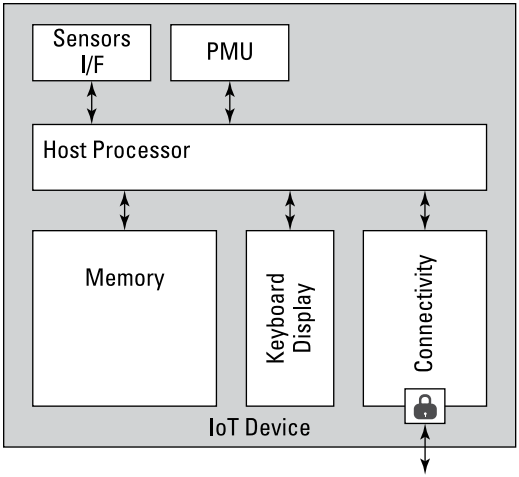

Figure 1-2: Device architecture of an IoT device.

Chapter 2

Understanding the Need for IoT Security

In This Chapter

- Protecting data
- Keeping your customers safe and secure
- Recognizing the security risk to your business reputation
- Maintaining compliance

In this chapter, you discover why securing Internet of Things (IoT) devices is important to businesses.

The Importance of Data

The privacy and security of data is a major challenge for every connected organization today. The large-scale theft of individual health and financial records from government and private sector institutions is reported in the media on a regular basis. But far greater risks — from far more personal and critical data collected by IoT devices — abound.

Although security and privacy risks to online financial and personal health information have been the subject of much attention, the vast amounts and types of data that are being collected by an unfathomable array of smart, connected devices potentially represent a far greater risk to consumers. This data, in the wrong hands, could be used for nefarious purposes. Though consumers may not yet be aware of the risk, it is the responsibility of IoT device manufacturers to provide appropriate security controls.

Protecting Your Customers

As with all things Internet, protecting your customers' sensitive data and well-being is a top priority. The IoT is no different! However, the IoT does introduce additional challenges that must be addressed to protect not only your customers, but also the operators of IoT devices. Your customers can be affected by attackers in four main ways (see Figure 2-1):

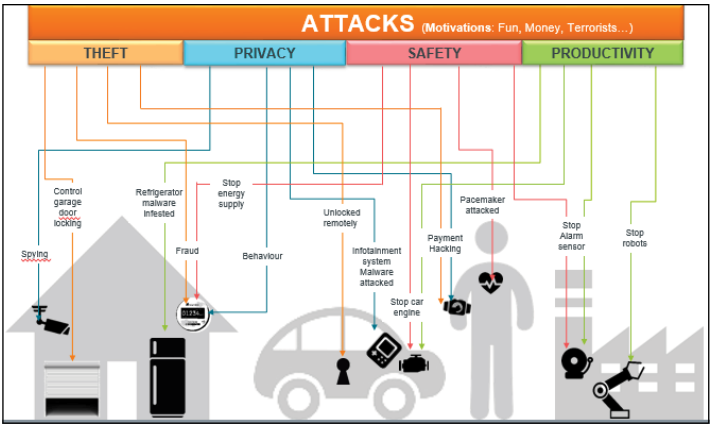

Figure 2-1: Why IoT security is essential.

- **Theft:** The risk of theft extends well beyond credit card or identity theft. For example, an attacker might steal power by tampering with a smart meter or software code from a smart device. Remote keyless entry systems potentially enable vehicle thefts and home invasions.
- **Privacy:** The volume and type of data (see the section "The Importance of Data" at the beginning of this chapter) collected by IoT devices and stored in the cloud can expose extremely sensitive information (such as real-time biometric and health information, personal behavior and eating habits, and location) in a variety of formats. For example, an attacker might extract and analyze smart meter data to detect if someone is home, or spy on someone by breaching their home security/video surveillance system.

- **Safety:** Malicious control of IoT technology in automobiles, wearable medical devices and smart meters could likewise allow an attacker to harm the end-user. More broadly, a cyberterrorist could disable a city's water system or traffic signals, or shut down the electricity grid.

- **Productivity:** For businesses, an attack can mean lost productivity. For example, a truck fleet could be rendered inoperable (navigation systems and delivery routes altered), production lines could be stopped (a robot is hacked), and offices could be made unusable (fire systems, air conditioning, and alarms could all be breached).

Next, you take a closer look at some increasingly popular IoT devices and their associated theft, privacy, safety, and productivity security risks.

Wearable technology

Wearable technology typically includes personal devices such as smart watches and health monitors, and medical devices such as wireless pacemakers and insulin pumps.

The primary security issue for wearable technology in the personal devices category is data security. These devices — and the associated IoT infrastructure (such as cloud storage and user portals) — collect and transmit volumes of information about the owner that is potentially of a very personal nature. In such cases, data theft and privacy are the primary security risks.

Similarly, medical devices collect and transmit private and detailed health information about the wearer. Like wearables, privacy is an issue, but another security risk is safety. An attacker that gains remote control of a medical device could potentially harm or kill the wearer, for example, by causing a pacemaker to malfunction or an insulin pump to administer too much or too little insulin to the wearer.

Home security/automation systems

Home security and home automation systems represent another category of IoT devices that are becoming immensely popular. The primary security risk with these types of IoT devices is privacy. For example, an attacker could compromise a home security system to gain access to video surveillance inside the home. Motion sensors and electrical usage (for example, kitchen appliances) could also give an attacker valuable information about whether or not anyone is home. Thus, theft and safety are also important security risk considerations for these types of IoT devices.

Stirring up a Hornet's Nest at Black Hat USA

In 2011, Nest Labs introduced the Nest Learning Thermostat, a self-programming smart thermostat that learns the owner's temperature preferences and can save 10 to 12 percent on heating bills and up to 15 percent on cooling bills.

Recently, security researchers at Black Hat USA 2014 demonstrated how a Nest thermostat can be compromised using a micro USB cable to install a backdoor in under 15 seconds if an attacker has physical access to the device. Once compromised, an attacker can potentially spy on the homeowner and attack other connected devices, such as security cameras, remote keyless entry systems, and carbon monoxide detectors.

Although the attack method demonstrated at Black Hat requires physical access to the device, there are scenarios under which this access can be gained without actually breaking into someone's home. For example, an attacker could purchase a Nest thermostat, install the backdoor, and return it to the store for some unsuspecting customer to purchase later. At this point, there's no easy method for the average homeowner to identify a compromised device. Remote attack methods are likely to be on the horizon.

Other IoT devices will no doubt be targeted for compromise in the future as the IoT market continues to grow rapidly. For its part, Nest Labs continues to design and develop innovative new IoT devices with security in mind and says it is constantly evaluating how it can improve security in its devices and protect consumers.

With such a high level of communication and synchronization between different IoT devices in home automation, there is not only a direct risk to the security of individual IoT devices, but also indirect risks to the entire system. For example, video surveillance cameras may have relatively strong authentication security, but an attacker may be able to gain access to that system (and other systems on a home network) by attacking the much weaker security of the home's coffee machine, before exploiting other weaknesses in the network to escalate the attack to other, more secure IoT devices.

Connected cars

For many years, the automotive industry has been building vehicles equipped with *smart* devices, such as sensors capable of collecting diagnostic information and alerting car owners to potential maintenance issues, and innovative control technologies that enable a more efficient or comfortable driving experience.

Now, the next wave of technological innovation is upon us and automakers have begun building smart, *connected* cars — moving the IoT into the automotive industry. Smart, connected cars — in addition to providing a rich array of real-time, live entertainment and information services — will enable powerful new applications as well. Tesla, for example, is able remotely to push over-the-air software and feature updates for many of its vehicle models.

But these smart, connected cars present a litany of security risks for consumers, businesses and auto manufacturers including safety, theft, privacy and productivity.

Safety risks include an attacker being able to control a vehicle or certain vehicle components remotely. For example, an attacker might lock a car's steering, disable its brakes, or shut off its engine. These types of attacks are possible today (see the sidebar "Securing the smart, connected car") but are rarely carried out directly against these major control components. Instead, attackers go after more innocuous systems, such as the vehicle's entertainment or GPS navigation system.

Securing the smart, connected car

As cars become more connected, they are also becoming more vulnerable to hacks. In July 2015, *WIRED* magazine reported that two security researchers had successfully demonstrated a remote hack of a Jeep vehicle being driven on a highway from a laptop located 10 miles away.

The researchers used the Sprint cellular network — which connects Chrysler vehicles to the Internet — to exploit a security vulnerability in Chrysler's Uconnect dashboard computers, giving them control of the vehicle's entertainment system. Okay, so far not so bad — although Lady Gaga unexpectedly blaring over your car stereo may cause you suddenly to veer off the road! But the entertainment system connects to various electronic control units (ECUs) that control many less innocuous functions, such as a vehicle's steering, transmission, and brakes.

Security researchers have also discovered a vulnerability in Tesla's Model S that enables an attacker to start the car and drive it with a laptop inside the vehicle. Still more insidious (and likely), an attacker can remotely cut the vehicle's engine while someone else is driving, after physically installing a remote-access Trojan on the Model S network.

In yet another case, security researchers were able to attack a 2013 Corvette with text messages sent from a mobile phone that enabled them to activate the car's windshield wipers and apply and cut the vehicle's brakes.

These cases are vivid reminders that a compromised connected car could be fatal. Gartner estimates that by 2020, 250 million connected vehicles will be on the road. Beyond the automotive industry, there are physical safety implications across the IoT industry, which Gartner estimates will be comprised of over 25 billion connected IoT devices by 2020. Thus, it's crucial that manufacturers treat security of the connected car as a top priority. And here's how . . .

Secure sensors and communication

To ensure that the software executables for each individual component are secure and only allowed to run in the manner designed by the coder, developers should add cryptography to authenticate and protect communications both inside the component and among components.

Add remote security monitoring

Adding remote security monitoring capabilities will alert developers in the event of a software or network breach. This is more effective than creating whitelists/blacklists for known attacks and launching an "arms race" with the bad guys, which, like the ongoing battle in the anti-virus market, will never be won.

Remote keyless entry systems and ignition systems could potentially enable an attacker to steal a vehicle without having to break a window and crack open the steering column to hot wire the ignition system!

Privacy issues could arise from an attacker (or even a legitimate party, such as law enforcement or an insurance company) gaining information about your driving habits (such as your speed and braking patterns), as well as where you travel to and how often. In the future, information about when you're in your vehicle, who is with you, and possibly other health information about the vehicle's occupants (such as weight) may be available.

Finally, productivity issues are possible, for example, for businesses that operate a fleet of smart, connected vehicles. Delivery trucks could be routed along less efficient routes or caused to malfunction in order to impede timely deliveries. Or fuel-injection systems could be altered to cause higher fuel consumption.

Smart meters

Smart meters enable real-time (or near real-time) two-way communications between utilities providers (such as electricity, gas, and water companies) and their customers' buildings and homes. Beyond simple monitoring and billing of energy consumption, smart meters enable capabilities such as the following:

- Highly granular measurement of energy usage throughout the day
- Changing pricing models (for example, prepaid to postpaid)
- Reporting of events (such as blackouts and brownouts)
- Remote start-up and cutoff of services
- Direct interaction with a smart grid to enable more efficient energy delivery in accordance with real-time demand

Safety, theft, privacy and productivity are all security risks associated with smart meters. Two critical technological

issues associated with smart meters are secure communications and data management.

If attackers can successfully compromise the communications network of a smart meter IoT infrastructure, they can potentially gain access to other components in the power grid. The potential for massive, widespread attacks against critical infrastructure is significant. On a smaller scale, there is huge potential for damage to individual health and safety (such as remotely and maliciously shutting off heat during a blizzard or air conditioning during a heat wave)via a denial-of-service or other form of attack.

For example, in 2010 the U.S. Federal Bureau of Investigation (FBI) released a cyber intelligence bulletin regarding smart grid electric meters in Puerto Rico that were being exploited to under report electricity usage by consumers and businesses, with utility losses estimated at 400 million U.S. dollars annually.

Finally, an attacker that is able to take control of a business's smart meters could potentially cause productivity issues. For example, by adjusting the temperature in a data center or manufacturing facility, an attacker could cause expensive equipment to overheat and malfunction. Setting the temperature too low in an office building could likewise cause a loss of productivity by cold workers.

Protecting Your Business and Reputation

By securing IoT devices and their associated theft, privacy, safety, and productivity security risks, IoT manufacturers protect their business and their reputation. The direct cost of a security breach depends on a number of variables and is difficult to quantify.

However, Verizon's 2015 Data Breach Investigations Report provides some excellent analysis and finds that the expected cost to the business of a security breach involving the loss of 100,000 records is approximately $474,600. These costs

include direct costs such as actual damages, civil fines and penalties, litigation, and remediation.

Other indirect costs can be far more significant. These include damage to your brand and reputation, and the loss of current and future customers. For example, despite Chrysler's rapid response to the Jeep hack (see the sidebar "Securing the smart, connected car") in issuing a recall of 1.4 million vehicles almost as quickly as a vulnerability was discovered, the company is still facing a potentially massive lawsuit and has undoubtedly lost possible future customers.

The IoT is a rapidly growing market that is causing many businesses to redefine their strategy. A security misstep can quickly quash these plans and, particularly for a new start-up, can be disastrous to the prosperity — even the survival — of the business.

Ensuring Compliance

Compliance with government and industry mandates is essential during the design stage of an IoT device or infrastructure. For example, various laws and regulations have recently been proposed and/or enacted in the U.K. and Germany governing smart meters. Various laws and regulations governing the security and privacy of sensitive information, such as financial and health information, exist throughout the world. IoT devices and infrastructure that collect, monitor, process, analyze, and/or store this type of sensitive information must be designed to comply with all applicable laws and regulations where they may be sold or used.

Industry standards also exist for various protocols and technologies and are being developed for new protocols and infrastructures, such as LoRaWAN and SIGFOX. Using standards-based protocols and technologies helps to ensure interoperability with other IoT devices and architectures and enables participation in the IoT ecosystem.

Chapter 3

IoT Vulnerabilities

In This Chapter

- Exploring IoT attack vectors
- Understanding the impact of attack methods
- Identifying security standards

In this chapter, I expand on Chapter 2 by looking at attacks at the software, board (or device), and chip layer, as well as areas where an attacker can exploit IoT security vulnerabilities and some of the key methods used to attack IoT security.

Assessing Security Risks at the Right Layer within the System

To ensure the appropriate level of security for their IoT devices and infrastructure, businesses must perform a risk analysis and implement appropriate safeguards. The level of safeguards implemented must be commensurate with the level of risk and the likelihood of occurrence. For the IoT, security safeguards are typically defined at the software, board, or chip layer (see Figure 3-1) and largely depend on the potential access that an attacker may have to the device or infrastructure.

Software layer attacks usually involve remote network attacks in situations where the attacker cannot gain direct physical access to the IoT device or infrastructure. These attacks are usually software-based and involve attacking

- Vulnerable communications protocols
- Weak cryptography implementations

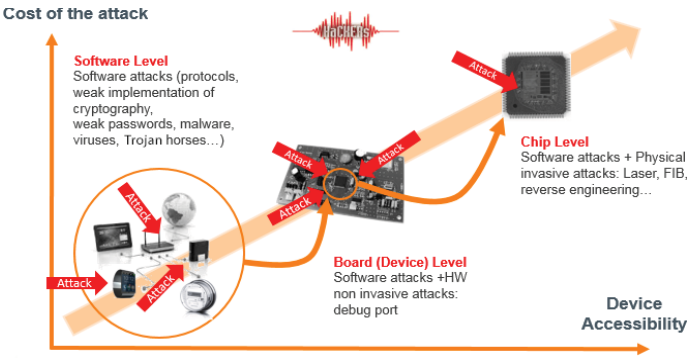

Figure 3-1: Security must be sized relative to the environment where the device is running and its accessibility to an attacker.

- Social engineering (guessing weak passwords)
- Malware (such as viruses, root kits, and Trojan horses)

Software attacks are relatively low cost to an attacker (in the range of a few hundred dollars) and may involve breaking a door lock to gain physical access to the system or bypassing WiFi security safeguards. This type of attack is very similar to the Sony PS3 exploit or an iOS jailbreak in which an attacker can cause OS execution to stop unexpectedly, allowing the attacker to install a new OS that is executed instead. The attacker can then take control of the device and steal trade secrets or other intellectual property.

Board- or device-layer attacks typically combine elements of software and non-invasive hardware attacks, such as connecting to a debug port on a device in order to upload malware or install a packet sniffer. Board-layer attacks typically require physical access to the board or device with specialized equipment that may cost an attacker several thousand dollars.

Attackers will often attempt to exploit test features in a board-layer attack. The attacker accesses the IoT device (or integrated circuit (IC) in the IoT device) in test mode to provide a basis for further attacks (such as the disclosure or corruption of memory content to retrieve user data like cryptographic keys or device configuration). Many ICs, particularly in the manufacturing industry, have well-known default passwords that are rarely changed, such as username “admin” and password “admin”.

Chip-layer attacks typically include software attacks and physically invasive attacks in which the attacker has direct access to the IoT device or infrastructure. Chip-layer attacks often require highly specialized equipment that can cost several million dollars. Chip-layer attacks include

- **Laser attacks:** The objective is to generate a fault code (code execution break/error or change memory value) by injecting a huge amount of energy (laser) onto a small area of the chip. For example, changing a Boolean value from False to True in a PIN code check condition. In banking, smart cards are protected against such attacks.
- **FIB (focus ion beam) attacks:** An FIB is usually used for chip debugging, but can be used maliciously to perform reverse engineering at the chip level in order to extract information or bypass security features.

What Control Can be Gained by Exploiting an IoT Device?

Within any IoT architecture, threats are everywhere and vulnerabilities can be exploited for a number of malicious purposes (see Figure 3-2):

- **Network:** By taking control of a gateway or router, an attacker can steal data that is being communicated between the IoT devices and the backend data management systems, or broadcast fake content to the devices or the backend infrastructure.
- **Application:** Finally, attackers can attempt to take control of the IoT application, giving them control of both the IoT devices and backend systems.
- **Device:** At the sensor or device level, an attacker can attempt to take control of the device, or insert unauthorized devices in the IoT architecture (a man-in-the-middle attack). The gateways and routers that IoT devices communicate within an IoT architecture are also potentially vulnerable to an attack.
- **Chip:** Attackers can attempt to take control of a device by targeting its microprocessor or integrated circuit (IC). Security controls, particularly in ICs, have historically been very weak.

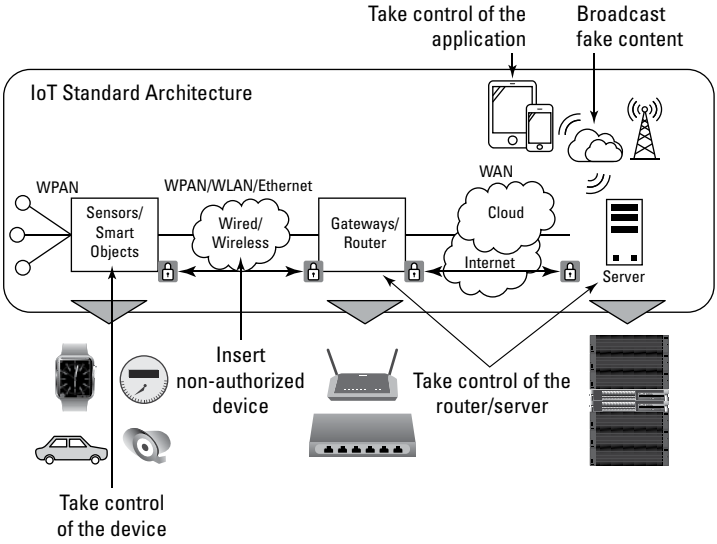

Figure 3-2: Threats are non-exhaustive and everywhere.

Security Standards

Due to the diversity of IoT devices and the rapid emergence of new devices on the market, there is currently no single standard that comprehensively describes potential IoT attacks or vulnerabilities. However, some standards do exist and can provide a good reference to IoT vulnerabilities, including

- **Common Weakness Enumeration (CWE)** (`https://cwe.mitre.org`): Describes software weaknesses in architecture and design.
- **Common Attack Pattern Enumeration and Classification (CAPEC)** (`https://capec.mitre.org`): Provides a resource for identifying and understanding attack methods.
- **Federal Information Processing Standard (FIPS) 140-2** (`http://csrc.nist.gov`): U.S. Government computer security standard for accrediting cryptographic modules.
- **Common Criteria** (`https://www.commoncriteriaportal.org`): International standard (ISO/IEC 15408) for computer security certification developed for the smart card industry.

Chapter 4

Understanding the Role of Cryptography

In This Chapter

- Decrypting cryptography
- Understanding symmetric algorithms
- Learning about asymmetric algorithms

Cryptography is the foundation of IoT security and is implemented by using hardware and software technologies. Cryptography is the science of encrypting and decrypting data communications in order to protect information. The three main functions of cryptography are

- **Confidentiality.** Roughly equivalent to privacy, confidentiality prevents sensitive information from being obtained by an unauthorized user or device, while ensuring that it was received by the correct user or device. Data encryption is a frequently used method of ensuring confidentiality. A common example is Transport Layer Security (TLS), which was previously Secure Sockets Layer (SSL), a security protocol for communications sent over the Internet and compatible with a large number of Internet protocols.
- **Integrity.** Integrity involves maintaining the consistency, accuracy, and trustworthiness of data during its entire life cycle to ensure it is not altered by unauthorized people. A common method of protecting data integrity is to create a hash (a cryptographic representation) of received data and compare it with the hash of the original message.

- **Authentication.** The process that confirms the identity of a remote device or a device on the network to ensure that only authorized devices are connected to the network. Public key infrastructure (PKI) authentication is one of the most common solutions used. PKI uses digital certificates to prove a device's identity.

Cryptographic algorithms have two major types: symmetric and asymmetric.

Symmetric Algorithms

Symmetric algorithms share a single secret key between communicating peers to encrypt and decrypt information. Symmetric algorithms are relatively efficient in terms of speed and computing power, but key management (securely exchanging and storing keys) is critical. If an unauthorized party gains access to the key, the communication is no longer protected. Symmetric cryptosystems are typically used to provide confidentiality.

A cryptographic *key* (also known as a *cryptovariable*) is a secret value (essentially, a password) applied to a cryptographic algorithm. The strength and effectiveness of the cryptographic algorithm are largely dependent on the secrecy and strength (or length) of the key.

Examples of symmetric algorithms include the Triple Data Encryption Standard (3DES), the Advanced Encryption Standard (AES), the International Data Encryption Algorithm (IDEA), and Rivest Cipher 5 (RC5).

Asymmetric Algorithms

Asymmetric algorithms use a public and private key pair combination. Only the public key needs to be exchanged between communicating peers and it does not need to be kept secret (hence, it's a *public* key). Communications are encrypted using the public key, but can only be decrypted with the private key. Asymmetric algorithms are computationally more intensive than symmetric algorithms but do not have the

same key management issues. Asymmetric cryptosystems are typically used to provide authentication and for *key exchange* in symmetric cryptosystems (discussed in Chapter 5).

Some asymmetric algorithms, like RSA, require a larger cryptographic key to provide the same level of security as symmetric algorithms. Generally speaking, for the same key size, asymmetric algorithms are typically less secure than symmetric algorithms by several orders of magnitude. For example, a 1024-bit asymmetric key might provide security equivalent to an 80-bit symmetric key.

Key exchange is a protocol used to share a secret key between a client and server without transmitting the key (for example, the Diffie–Hellman protocol is commonly used for key exchange in transport layer security, or TLS). This secret key is then used in symmetric cryptosystems for confidentiality purposes (encrypting the communications link between the client and server).

Examples of asymmetric algorithms include the Rivest–Shamir–Adelman (RSA) and the Elliptic Curve (EC).

A *cryptographic hash function* is a one-way operation used to digitally sign and verify the integrity of information. Analogous to a hash recipe that consists of diced meat, potatoes, and spices mixed together, you can't simply separate the ingredients of the cooked hash and recreate the cow or pig, potatoes, and spices used to create the hash! Typical uses for a hash include digital signatures and password verification (a hash is a shorter representation of a long password that provides a higher level of security because it doesn't require disclosing all of the bits in a long password). In order to be effective, a hash function must be

- **Easy to compute** (any given message can be represented by a fixed-length hash)
- **Non-reversible** (practically impossible to generate the original message from the hash)
- **Unalterable** (practically impossible to modify the original message without causing the hash also to be modified)
- **Unique** (practically impossible for two different messages to generate the same hash)

Examples of hash functions approved by the U.S. National Institute of Standards and Technology (NIST) include SHA-1 (secure hash algorithm), SHA-2, and SHA-3.

See Table 4-1 for a comparison of the advantages and disadvantages of symmetric and asymmetric encryption.

Table 4-1 Symmetric and Asymmetric Encryption

	Advantages	***Disadvantages***
Symmetric	Strong security	Secure key exchange
	Faster performance	Key management
	Fewer computing resources	
Asymmetric	No key exchange/management issues	Requires larger keys for equivalent security to symmetric
	Highly scalable	Slower performance
	Multiple uses (authentication, access control, confidentiality and privacy, data integrity, non-repudiation	Computationally intensive

Chapter 5

Choosing the Right IoT Security Solutions

In This Chapter

- Authenticating devices
- Making secure connections
- Executing code securely
- Storing data securely

Internet of Things (IoT) devices share common security needs which stand as the four elements of IoT security. In this chapter, you find out more about these elements.

The Four Elements of IoT Security

Security is like a chain that is only as strong as the weakest link. To secure an IoT device effectively, you need to secure all of the following four elements (see Figure 5-1):

- **Device authentication:** Confirming the true and unique identity of communicating devices on a network
- **Secure connections:** Protecting "data in motion" by maintaining the confidentiality and integrity of connections between peers

- **Secure code execution:** Protecting "data in use" by ensuring the device (host) runs the software in the way it was intended at original boot and after secure updates without information leakage
- **Secure storage:** Protecting "data at rest" by encrypting the data and by storing it in a secure location

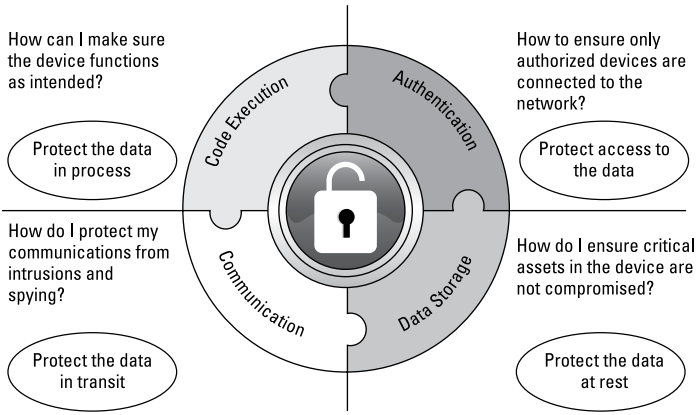

Figure 5-1: Four elements of IoT security.

Device Authentication

IoT deployments consist of a vast number of interconnected and distributed endpoints that need to communicate with each other, thus the strength, reliability, and scalability of the authentication methods used are critically important. Each endpoint must be controlled and properly authenticated to ensure it is genuine and to prevent fake or unauthorized devices from being installed on the network.

Authentication is the process in which communicating peers (endpoints or nodes) identify each other and verify their identity to each other. In machine-to-machine communications, authentication is obtained through cryptographic protocols by verifying that each party directly or indirectly shares the same secret. Authentication protocols must be strong enough to be resilient against many different attacks, such as eavesdropping (spying), replay attacks (an attacker keeps

the result of a previous authentication made by an authorized user and uses it again), man-in-the-middle attacks (an attacker secretly relays and possibly alters the communication between two parties), dictionary attacks (an attacker tries the most probable secret keys: such as names and birth dates for a password) or brute-force attacks (an attacker tries all possible password combinations).

The most common (and simplest) form of authentication is based on sharing a secret key. 3DES and AES are typical examples of symmetric algorithms (discussed in Chapter 4) used for authentication. However, anyone that has the secret key can decrypt the information that is being protected, thus if the secret key is compromised, the entire security mechanism collapses.

Also, symmetric key distribution within a network is difficult. It can be centralized, but must be trusted and robust enough to protect the keys because the key storage area is a preferred target for attackers. Also, because keys must be shared between every communicating peer (that is, every device), the risk of key compromise becomes significantly greater as the number of communicating devices increases. Compromising just a single key potentially compromises the entire system.

A better authentication method relies on the use of asymmetric (or *public key*) cryptography. Public key authentication is commonly used to provide strong authentication for servers and devices connected to the Internet. The Elliptic Curve Digital Signature Algorithm (ECDSA) is an example of an asymmetric algorithm used for authentication.

The use of public key cryptography requires access to a public key infrastructure (PKI). PKI is applicable to a wide variety of standards-based, secure communications protocols such as IPsec, Transport Layer Security/Secure Sockets Layer (TLS/SSL), and Datagram Transport Layer Security (DTLS). A secure implementation of PKI for authentication requires the following:

- On-chip key pair generation with a key generator using a high-quality TRNG (true random number generator) and secure key storage inside the security module

- Execution of cryptographic operations (such as signing, signature verification, encryption and decryption) within a controlled environment

Software-layer attacks on a random number generator (RNG) are sometimes attempted. Random numbers are at the basis of cryptography. Reducing the entropy of an RNG significantly reduces the security level of the system.

The importance of quality random number generation cannot be overstressed in the context of embedded device security. Some very prominent cryptosystem failures have been traced back to poor random number generation. True random number generation on embedded systems is a genuinely difficult task, yet true randomness is vital for high-quality generation of encryption keys. The strength and quality of the random number generation relate directly to the strength and quality of the cryptographic security system.

Authentication and root keys

The challenge for initiating secure communication between peer devices is to ensure that the public key that is received by a device belongs to the intended communication peer — the device it wants to talk to — and that it can be trusted.

This challenge invariably requires that a public key is stored securely on the device. Although the public key does not need to be kept secret, it must be immutable — it must not be possible for an attacker to modify the key or cause the device to use another (fake) key instead. A device's or user's public key is sometimes combined with other information to identify the device (such as a unique device identifier, IP address, domain name, or real name and address). Digital certificates are commonly used to provide authentication of public keys.

Key exchange mechanism

The process for provisioning and loading keys in a device (the *key exchange* or *key establishment mechanism*) is critical to the overall security of the device. The main challenge for a key exchange mechanism is to exchange keys in a secure manner so that no one but the authorized peers can obtain a copy of the keys.

One secure key exchange method is to create a secret key that is shared between two peers. The shared secret key can then be used as a root (or master) key to create subsequent derivative keys using a key derivation function (KDF). The generated keys can then be used by the peer devices as temporary keys to communicate securely using an asymmetric cryptographic algorithm.

A KDF creates one or more secret keys using a pseudo-random function and a master key.

The Diffie–Hellman (DH) key exchange protocol is an example of a cryptographic protocol used to exchange keys securely. The station-to-station (STS) protocol is another well-known key exchange mechanism based on Diffie–Hellman.

Both DH and STS must be combined with other methods to address device authentication (verifying the identity of the peer at the other end of the communication channel). For this purpose, the identity can either be built in to the device (if all devices only need to talk to a single server), or the identity can be signed with a single root key that is used by a trusted third party to sign all valid public keys (if devices need to be able to set up secure connections to multiple other devices).

Adding security into an IoT architecture entails the secure management of cryptographic keys and others assets. For example, personalization and provisioning services for applications and devices are needed to support customers for the management of assets at any steps during and after manufacturing – throughout the entire device life cycle.

Digital signatures for device authentication

A digital signature uses asymmetric cryptography to guarantee authenticity and integrity. This is accomplished by creating a one-way (or irreversible) hash of the data (or a secret key) to be signed (or verified), then encrypting the one-way hash with the sender's private key (see 'Chapter 4'). The recipient (user or device) then uses the sender's public key (which is known to everyone) to decrypt the hash and verify that it was created with the sender's private key (authentication) and that the data hasn't been tampered with (integrity).

Digital signatures are used to verify that communicating peer devices have or know a particular secret key (either symmetric or asymmetric) and for device authentication. Digital signatures can be used for offline authentication (for example, by signing boot code) and for online authentication (for example, by signing challenges).

A *challenge* is a question that one authenticating peer presents to the other peer, to which the other peer must provide the correct response.

Secure Communication

IoT devices leverage existing Internet technologies and protocols which provide proven secure communication solutions to protect "data in motion". The actual use case of the device or node typically dictates the criteria for selecting the communication protocol. For example, in smart grid deployments that typically follow the client–server model, virtual private networks (VPNs) based on protocols such as SSL/TLS (Secure Sockets Layer/Transport Layer Security), MACsec (Media Access Control Security) or DTLS (Datagram Transport Layer Security) are used, while in "full mesh" networks a network-level security protocol such as IPsec is more applicable.

Regardless of the communication protocol used, public key infrastructure (PKI) can be used for device authentication. This guarantees the seamless integration of IoT nodes into, and interoperability with, the existing network infrastructure.

Data transfer rates to and from an individual device or node are typically limited in IoT deployments, which can pose implementation challenges for cryptographically secured communications. In order to limit the processing overhead on the general purpose processing unit and to reduce power consumption on the device during cryptographic operations, it is often beneficial to offload cryptographic functions to specialized hardware on the device. When executed on dedicated hardware, symmetric encryption and asymmetric algorithms are faster and more power-efficient (which is important in power-constrained and battery-powered devices) than software implementations (discussed in the next section).

Secure Code Execution

The application code executed by an IoT device must run in a secure manner and perform as expected: it should not be modified or corrupted, and should not "leak" sensitive data. This is particularly important when the code uses sensitive data ("data in use"), such as cryptographic keys or functions, payment applications, and health information.

A *secure boot process* (which isn't available in a software-only solution) is another important aspect of protecting data in use and ensuring that the device only runs the intended software in the way its manufacturer or deploying organization intended.

Secure boot ensures that only kernel or software images that are endorsed or signed by the device manufacturer (or a trusted third party) are allowed to boot on the device. This is accomplished using a hardware-assisted boot process in which kernel and software images are verified by hashes (that is, a data integrity check) and digital signatures prior to executing them at boot time.

Implementing secure boot as a part of an IoT device architecture is closely connected to other device management and maintenance tasks, and should be considered an integral part of the larger security strategy. For example, the secure booting procedure needs to be compatible with remote and possibly over-the-air (OTA) device firmware upgrade mechanisms, which allow device software images to be modified or updated post-deployment.

IoT devices aren't likely to be static. In most cases, the software running on an IoT device needs to be upgraded at some point during its life cycle (for example, bug fixes, updates, and new features). The management of software upgrades is another important aspect of IoT security. Secure software updates usually use a combination of digital signature verification (software has been signed at the source by a trusted party and can be verified by the device with a public key), and data integrity checks (using a hashing function) to verify that the software code has not been modified between the source and the device.

Application code can be protected in the security architecture in three primary ways:

- Software mechanisms
- Secure module chips or cryptographic security modules
- Hardware IP cores

All three of these security mechanisms can be deployed separately or can be combined within the trusted computing (or secure execution) environment to offer additional protection in the IoT device architecture.

A major difference between the three options is the ease of adding security to the device. Software mechanisms are easiest to update. Adding a secure module is a bit more intensive because it involves reconfiguring the board. Adding a hardware IP core involves updating the design of the chip itself, which typically involves a long planning and manufacturing process.

A *trusted computing* environment refers to a computing platform that operates in a known (trusted) state that is ensured by hardware and software mechanisms in the platform.

Software mechanisms

Code execution code can be protected with software security mechanisms, such as a software development toolkit, to improve resistance against several types of attacks. The secure software resides in the memory and is activated by the host processor. This is similar to an operating system, which also resides in the memory and is activated by the host processor. For example, software obfuscation tools provide protection against debuggers, memory dumpers, and reverse engineering, and provide an additional layer of security by hiding, partitioning, and re-arranging data as it is handled by software (see Figure 5-2).

The advantages of software security mechanisms in an IoT device architecture include the following:

- High flexibility
- Can be implemented without hardware changes in existing systems
- Fast time to market
- Cost efficiency

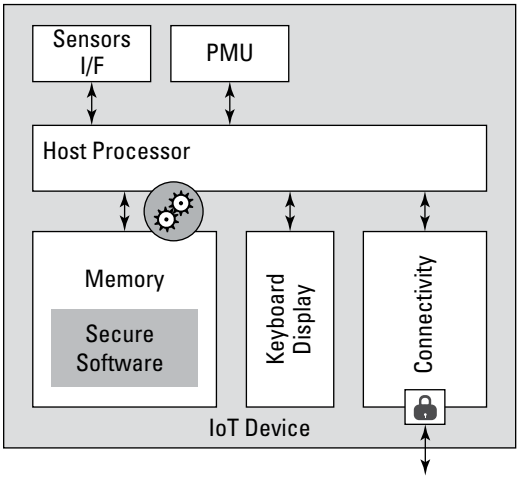

Figure 5-2: Software security mechanisms in an IoT device architecture.

Secure module chips

Secure module chips (also called *security chips*) provide a secure or trusted execution environment that can be inserted as a companion chip into an IoT device architecture. This chip can be seen in Figure 5-3.

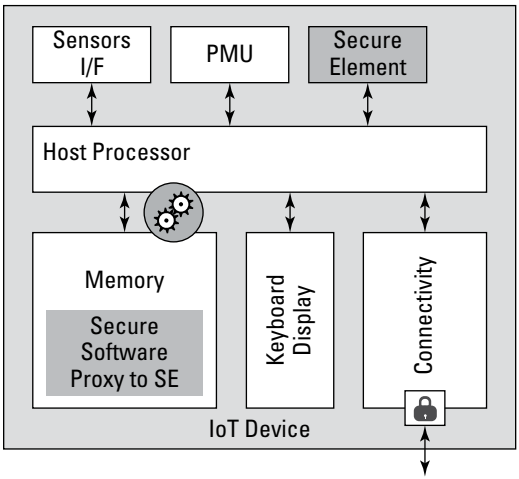

Figure 5-3: Software and secure element chip or security module in an IoT device architecture.

Advantages of this approach include

- Chip-level security increases the cost to attack
- Provides root of trust (a separate processor in a trusted computing module that performs a set of functions which are always trusted by the operating system)
- Common Criteria/Federal Information Processing Standard (FIPS) certified solutions

Common Criteria (CC) is an international standard (ISO/IEC 15408) computer security framework frequently used in government agencies and critical infrastructure. FIPS is a U.S. Government computer security standard used to accredit cryptographic systems. Both define standards-based cryptographic and security requirements or recommendations. They also provide a third-party certification framework in which evaluation is delegated to an accredited lab and the government entity delivers the certificate based on the evaluation results.

- No need to have a custom system on chip (SoC). The secure element can accompany a standard memory controller unit (MCU) in the device
- Excellent solution for low–mid volume products; doesn't require development of custom integrated circuits (ICs)

There are two basic types of secure element modules. The first type of chip only executes security functions (such as cryptography and data storage) within the secure environment and is therefore optimized in terms of power consumption and cost. The second type of module offers additional computing power and memory and can execute a part of the device application within a secure or trusted environment.

Because cryptographic functions are executed by an external component to the main CPU on the device, secure element implementations require a secure channel between the secure element and the host CPU.

Hardware IP cores

A secure or trusted execution environment can also be built directly into the IoT device by adding a security hardware IP core into the SoC or CPU of the device (see Figure 5-4).

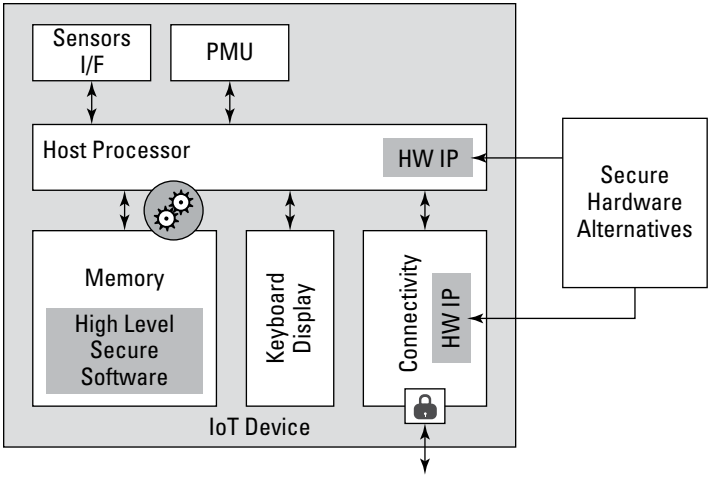

Figure 5-4: Hardware IP cores in an IoT device architecture.

This solution offers an excellent option for high-volume devices in terms of cost. It also offers the best solution when power constraint is a key issue in the IoT device and when cryptographic functions are executed in dedicated hardware, such as a separate cryptographic processor or module. In this case, the need to create a secure bridge between the host processor and the secure element no longer exists.

The advantages of a hardware IP core implementation include the following:

- Provides the best compromise for cost, speed, power consumption, and security level (it combines both software and hardware security in the same execution environment)
- High security at system and board level against software attacks
- Intrinsically binds security to the device

Secure Storage

Stored data (or data at rest) must be protected. The two main types of data that must be protected in storage are

- **Encryption keys and unique device identifiers:** Such data is used as the trust anchor (an authoritative entity for which trust is assumed rather than derived from another entity) for a secure system. From this root key, derivative and session keys will be generated to authenticate and securely communicate between peers. If these keys are compromised, the device can be cloned, and communications traffic can be decrypted by malicious parties.
- **Sensitive application data:** The confidentiality, privacy, and integrity of application data, such as financial and health information, must be safeguarded.

In IoT devices, data is stored in a variety of media. Data can be protected with encryption, by being stored in a tamper-resistant device (such as a device that is rendered unusable if there is evidence of tampering), or a combination of both. In any case, the access to such data must be carefully controlled and restricted to authorized persons, machines, and processes.

Chapter 6

Ten IoT Security Best Practices

In This Chapter

- Minimizing the attack surface
- Authenticating connected devices
- Protecting data in motion, data in use, and data at rest

In this chapter, I briefly describe a few security best practices you should consider for your Internet of Things (IoT) infrastructure to ensure end-to-end security.

Understand the Risks

The enormous number of IoT devices connected through heterogeneous infrastructures increases the risk of attacks. A potential attack against IoT infrastructure — networks or end devices — creates enormous risks to consumers and IoT device operators, including

- **Safety:** Serious personal injury or death may occur if an attacker successfully exploits IoT security weaknesses, for example, in connected cars and wearable medical devices. On a larger scale, cyberterrorists could shut down an electrical grid or disable a city's water system.
- **Privacy:** With IoT, the potential for data theft extends well beyond health and financial information. For example, an attacker can breach a home video surveillance system to spy on someone, or monitor smart meter data to know when someone is home.

- **Theft:** Remote keyless entry systems could be compromised to gain entry into homes and vehicles, and smart meters could be hacked to steal electricity.
- **Productivity:** Highly automated manufacturing production lines (for example, utilizing robots) could be shut down and offices could be made unusable (for example, setting off a fire sprinkler system).

Additionally, IoT device manufacturers risk liability issues, damage to their reputation, and loss of profits. See Chapter 3 for a complete discussion of IoT security risks.

In most cases, it is faster and more economically feasible to implement the right security level when designing a system, rather than trying to implement security in an existing system after it has already been deployed. That being said, security can still be added later, when necessary.

Security is a chain that is only as strong as its weakest link.

Never Underestimate Your Enemy

Attackers and other malicious users are motivated by various factors including financial gain, terrorism, political hacktivism, thrill seeking, and technical hubris. Increasingly, attackers are backed by large criminal organizations or rogue nation-states with significant computing and financial resources necessary to carry out stealthy, sophisticated attacks against targets over extended periods of time (sometimes years).

Never underestimate your enemy. Today's cybercriminals aren't all stereotypical basement hackers seeking a cheap thrill and a little notoriety.

Minimize the Attack Surface

When designing a new IoT device, it may be tempting to load it up with all the bells and whistles. While such a design

philosophy is likely music to your marketing team's ears, it is hated by your security team because it increases the potential number of attack vectors and vulnerabilities in your device. For example, Chapter 2 explains how the entertainment system of a Jeep was hacked to gain remote control of the vehicle's steering, transmission, and brakes. That's not to say you should design a vehicle without an entertainment system! Just remember, attackers don't necessarily target their final objective directly but instead target other, less secure connected devices or systems to gain entry and establish a foothold in the IoT architecture.

Implement Security at the Right Layer

Appropriate security safeguards need to be installed at the right component layer — system, board, or chip — based on the environment where the device will be running and the potential accessibility to an attacker (see Chapter 3).

Authenticate Connected Devices

All connected devices in an IoT infrastructure must be uniquely identified and able to verify the identity and authenticity of every other communicating device in the infrastructure.

Asymmetric cryptography employing a public key infrastructure (PKI) is the best solution for device authentication using the largest technically feasible keys.

Larger cryptographic keys are required in asymmetric cryptography to provide the same level of security as symmetric cryptography. See Chapter 4 to learn more about symmetric and asymmetric cryptography and their uses.

Use Standards-based Protocols and Algorithms

Standards help to ensure interoperability and promote cooperation throughout the IoT ecosystem. When a flaw or vulnerability is discovered in a standard protocol or algorithm, for example, the entire community works together to fix the issue and improve the entire standard. Although it could be argued that attackers have access to those same standards and a protocol or algorithm is therefore easier to attack, the premise is false.

For example, cryptographic algorithms are published and well-known. It is the mathematical complexity and astronomically large number of possible solutions rather than the secrecy of the algorithm that makes the cryptosystem secure. Likewise, communications protocols and other standards-based technologies benefit from the expertise of the entire ecosystem.

Rather than trying to reinvent the wheel, use proven, standards-based solutions. Security by obscurity isn't an effective strategy.

Protect Data in Motion

All communications in an IoT infrastructure should be encrypted from end to end to ensure the confidentiality and integrity of data in motion. Secure network protocols such as Secure Sockets Layer/Transport Layer Security (SSL/TLS), Datagram Transport Layer Security (DTLS), Media Access Control Security (MACsec) and IPsec are commonly used to secure network communications through a virtual private network (VPN) in an IoT infrastructure. See Chapter 5 to learn more.

Protect Data in Use

Applications running on IoT devices and the data they access and collect must be protected at all times. A secure boot

process implemented as part of a trusted computing environment ensures that only kernel and software images in a known good state (that is, no unauthorized modifications) that are endorsed or signed by the device manufacturer (or trusted third party) can run on the device.

Application code in an IoT infrastructure can be protected in the following ways:

- Software mechanisms
- Secure element chips
- Hardware IP cores

These areas are covered in more detail in Chapter 5.

Protect Data at Rest

IoT devices collect vast amounts and types of data that can be used for malicious purposes by an attacker (see Chapter 2 for several examples). For this reason, you must protect not only encryption keys and unique device identifiers that perform other critical IoT security and authentication functions, but also the data that is collected and stored on IoT devices and the underlying infrastructure.

Data should be encrypted and/or stored in a tamper-resistant device (see Chapter 5).

Choose the Right Security Vendor/Partner

To secure your IoT infrastructure properly, you must protect the entire system. Choose security partners who understand how to integrate security for all four pillars of security: device authentication, secure connections, secure code execution and secure storage.

Likewise, you must ensure that your entire ecosystem is protected. To protect your IoT device architecture, make sure you only work with trusted partners throughout your value chain that are equally committed to the security and integrity of your IoT infrastructure. And if your devices leverage other manufacturers' devices or communications infrastructures, you must likewise commit to the security of their IoT infrastructures.

If you are an application developer, device manufacturer or a semiconductor company looking to add security to your IoT solution, INSIDE Secure can help.

INSIDE Secure protects IoT applications and devices from the core to the cloud to protect your intellectual property, reputation and customers. To date, we have secured over 2 billion digital products to protect data at-rest, during processing and in-transit. Our software toolkits, hardware IP and secure microcontrollers are used by the world's leading companies to foil hackers, meet industry standards, and comply with government regulations and certifications.

Featured products:

Application Protection for IoT: an easy to use software toolkit to integrate code obfuscation, anti-tampering and whitebox encryption into your device applications and mobile apps

MatrixSSL: a software toolkit to provide a secure SSL/TLS connection between your device and the cloud

VaultIP-130: a FIPS 140-2 level 2 certified semiconductor IP core used to secure the hardware execution environment of IoT devices

www.insidesecure.com

Security is the #1 Internet of Things (IoT) challenge

No IoT device is safe from attack — no matter the application — and you can't underestimate the impact of these attacks on your reputation and finances. Security is only as strong as the weakest link, and you need to be aware of the vulnerabilities and how to prevent them.

- ***Protect yourself and your customers* — make protecting sensitive data and well-being a top priority**
- ***Be compliant* — adhere to government and industry mandates**
- ***Understand cryptography* — the foundation of IoT security**
- ***Choose the right security partner* — integrate the most appropriate security solutions**

Open the book and find:

- **Why you need security for IoT**
- **The main IoT vulnerabilities shown through use cases**
- **The role of cryptography**
- **How to counter attacks with the right security solution**
- **The top ten IoT security best practices**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

FOR DUMMIES®
A Wiley Brand

Also available as an e-book

ISBN: 978-1-119-21119-8
Not for resale

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.