IOT DEVICE SECURITY: BUILT-IN, NOT BOLT-ON

The 10 Security Techniques Every Device Designer Should Consider

The Rising Tide of Security Threats

Limited only by designers' imaginations, the Internet of Things (IoT) is changing how people live. From medical devices and fitness trackers to tank sensors, smart thermostats, intelligent streetlights, water monitors, and more, the IoT is in more places than ever.

However, by relying on wireless networks, those hundreds of millions of IoT devices present a greater "attack surface," making them tempting frontline targets for cybercriminals, hackers and other bad actors. Unfortunately, the tools and techniques we've applied to PC/smartphone platforms often don't work as well in the IoT, for several reasons:

- **RESOURCE LIMITATIONS**
  Small-footprint IoT devices typically have far less battery power, processing speed and memory than PCs or phones. They lack the power and sophistication required to support traditional security measures.
- **DATA COMPLACENCY**
  Many companies view the data in their IoT networks as mundane and having little intrinsic value outside the organization. But many, breaches are motivated not by access to the data, but by access to the system. The data isn't the goal — the hack is.
- **AVAILABILITY OF TOOLS**
  The tools and expertise to analyze and modify embedded IoT devices are widely available — even to hobbyists.
- **NO PHYSICAL ACCESS REQUIRED**
  One of the advantages of the IoT is that devices can be remotely configured or upgraded without the need for an expensive, onsite service call to access the device. However, thanks to wireless connections, hackers also don't need physical access to devices such as USB or other I/O ports.
- **INTERFACE DIFFERENCES**
  Embedded devices often have no GUIs, and error messages can be as basic as a coded series of beeps or flashing lights. This is particularly true for security status and control functions, allowing security alarms to be overlooked.
- **HARDWIRED PORTS**
  These provide unfortunate opportunities for compromise.

IoT solutions often can't simply implement a strong password over a transport layer security (TLS) connection — the most common approach for PC/Internet applications. IoT solutions need a different approach, and the effort required to identify and mitigate the unique security risks in embedded systems is often underestimated, if not overlooked entirely.

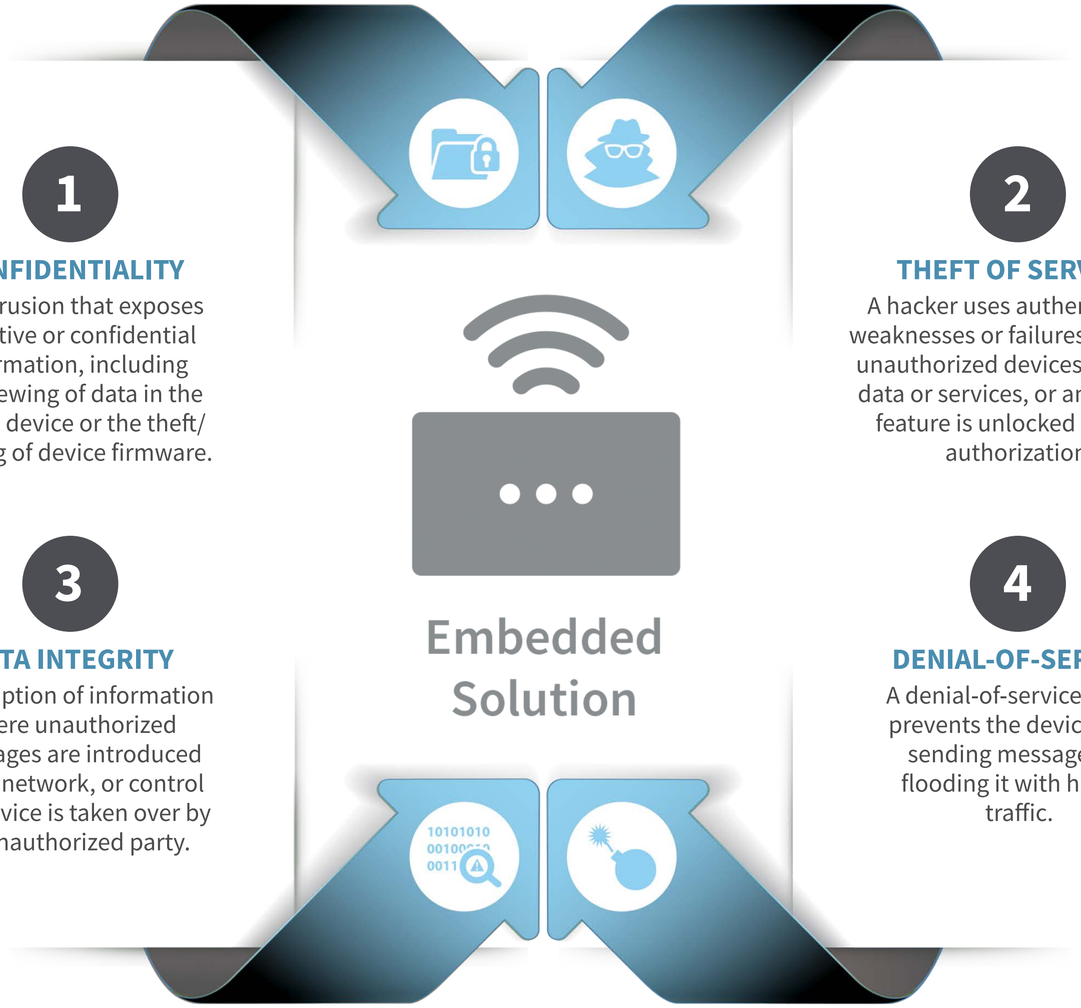
The risks of this rising tide of security threats are significant. Beyond reputational damage, competitive threats, eroding customer confidence, and safety challenges, regulators are paying increasing attention as well. For example, security breaches that violate U.S. federal HIPAA regulations can lead to fines of $50,000 per violation.[1] Credit card processors that fail to comply with the Payment Card Industry Data Security Standard (PCI DSS) may be fined up to $100,000 per violation. [2]

Distributed Denial of Service (DDoS) attacks are becoming more and more prevalent. These attacks may not be targeted at the average IoT edge device but the hijacking of a connected IoT edge device may be used to create a 'BotNet', a group of hijacked devices working in unison to attack a central point on the IoT network or an external server or computer outside of the local network. Even if these attacks are not targeted at the local IoT network, they still pose multiple problems by preventing the normal functioning of the IoT network or even simply draining the battery on a mobile IoT edge device, creating maintenance costs for the administrators.

[1] What are the Penalties for HIPAA Violations? HIPAA Journal, Jan 23, 2022
www.hipaajournal.com/what-are-the-penalties-for-hipaa-violations-7096/

[2] Six Serious consequences of PCI noncompliance, Security News, Dynamic Solutions Group, July 14, 2021
www.dsolutionsgroup.com/consequences-of-pci-noncompliance/

Four Types of Security Threats that Disrupt IoT Devices



1 CONFIDENTIALITY

An intrusion that exposes sensitive or confidential information, including the viewing of data in the actual device or the theft/cloning of device firmware.

2 THEFT OF SERVICE

A hacker uses authentication weaknesses or failures to enable unauthorized devices to access data or services, or an upgrade feature is unlocked without authorization.

3 DATA INTEGRITY

A corruption of information where unauthorized messages are introduced into a network, or control of a device is taken over by an unauthorized party.

4 DENIAL-OF-SERVICE

A denial-of-service attack prevents the device from sending messages by flooding it with hostile traffic.

Security is a Balance between Economic Cost and Benefit

Given enough time, money and expertise, any system can be hacked. So, it's important to design a system that will deter an attacker by making it uneconomical — the cost or effort of an attack outweighs any benefit to the attacker.

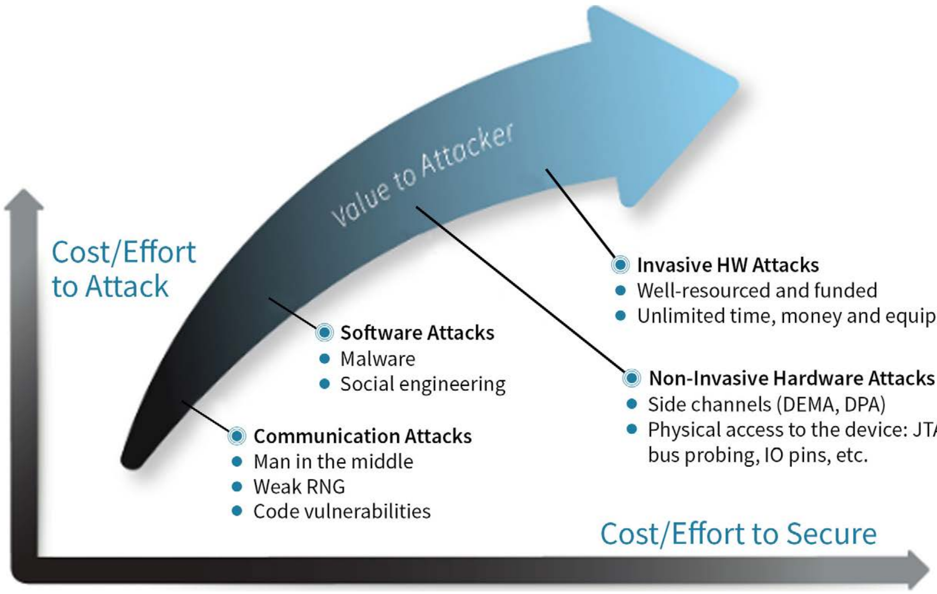
Attacks can be classified in terms of investment required by the attacker, the type of attacker and the equipment used.

These include:

- **EXPENSIVE INVASIVE ATTACKS**
  These are attacks such as reverse engineering, or sophisticated microprobing of a chip.
- **PASSIVE SOFTWARE ATTACKS**
  A key example is exploiting unintentional security vulnerabilities in the code.
- **COMMUNICATION ATTACKS**
  Communication attacks involve exploiting weaknesses in the Internet protocols, crypto or key handling.

Security is always a balance between economic cost and benefit, dependent upon the value of the assets being protected on the one hand and the cost of security features on the other.

The success of the Internet of Things depends on data and services being safe, and when the security balance is right, it can open up new opportunities and markets.



Hacks can be categorized by the cost and effort involved.

10 Security Techniques Every IoT Designer Should Consider

For design engineers who are striving to enhance the security of their IoT devices, there are numerous options at hand. Here are 10 proven strategies that engineers can use to improve device security.

Method	Complexity, Resources Needed	Notes
1. Packet Encryption	Low	Foundation for most embedded system security
2. Replay Protection	Low	Prevents resubmission of recorded messages
3. Message Authentication Code	Low	Prevents messages from being changed
4. Port Protection	Low	Secures ports that may be physically accessed by an attacker
5. Secure Bootloader	Moderate	Ensures that only authorized firmware is allowed to run
6. Pre-Shared Keys	Low	Preferred for smaller systems
7. Secure Shell (SSH)	High	Generally on OS-based systems; can prevent malicious connections
8. P ublic Key Exchange (PKE)	High	Generally on OS-based systems; can prevent malicious connections
9. Transport Layer Security (TLS)	High	Generally on OS-based systems; can prevent malicious connections
10. Wi-Fi Protected Access (WPA2/WPA3)	High	Generally on OS-based systems; can prevent malicious connections

1. Packet Encryption

This is the basic “go-to” method for protecting data exchanges in IoT solutions with smaller embedded terminal devices. Most systems have the resources to implement basic encryption, such as FIPS-197/AES, which can protect messages from unauthorized viewing or malicious changes. This method is easy to implement and use, especially in conjunction with private keys.

2. Message Replay Protection

In this approach, encrypted packets are enhanced with data fields that vary in a way known to the recipient (which could be as simple as a date stamp). The recipient enforces a rule that messages are only accepted once or in a defined sequence. This prevents recorded, but not necessarily decrypted, messages from being resubmitted at a later time to cause the original action, such as “open door.” This method is simple to implement and is often used when individual messages can cause state changes. This can also be part of an encryption mode that will use this information within a block cipher. An example of this is the AES counter mode block cipher.

3. Message Authentication Code

In this method, a cipher or hash algorithm is run on the content of a data packet to create a short signature that accompanies the message packet. The recipient uses the same cipher or hash to confirm that the message has not changed. Message authentication provides explicit protection from tampering and enables some systems to safely use clear-text messages. For example, this method can be used for systems that transmit non-confidential data (e.g., air temperatures) that nonetheless must not be tampered with. This is another low-complexity method that is useful for many types of embedded systems.

4. Debug Port Protection

Hardware ports used for configuration, control, and analysis (e.g., JTAG ports and serial logging ports for firmware development and debugging) are also vulnerable and tempting targets for security attacks. To start, these ports can be protected with a different factory password per unit before further actions are allowed. Of course, the better move is to internally disable these ports in field-deployed units.

5. Secure Bootloader

Even for a development team with unrestricted access to required technical information, it can be daunting to correctly build and load firmware into a resource-limited embedded device, which makes it unlikely you'll experience a successful attack based on a malicious firmware modification. But the rapidly increasing sophistication of embedded-system attackers, combined with product requirements for easier field upgrades of device firmware, have created a risk that must not be overlooked. One best practice is to configure the device to check for a hash-based message authentication code (HMAC) signature in the firmware image during startup to ensure it is authorized to run on the product. The image may also be encrypted for further protection. Secure bootloader solutions demand careful management of keys and support for debugging.

6. Pre-Shared Keys

Secure IoT communications require access to compatible keys. The use of pre-shared keys (PSKs) minimizes the demands on the resource constrained device. Keys can be transferred through an independent, secure channel and then manually entered into the terminal device. While the overall system to share the keys may have some complexity, the demands on the actual terminal device are minimal.

7. Secure Shell

The Secure Shell (SSH) protocol protects ports used for debug and configuration operations. SSH implements a standard protocol to encrypt console connections (e.g., Linux shell access) to prevent unauthorized viewing or operations. This substantially extends protection beyond a simple debug port password. This can often be too complex to implement on smaller embedded systems. But it's quite straightforward and feasible on larger OS-based systems because the necessary resources are typically present.

8. Public Key Exchange

Sometimes, pre-shared keys aren't a viable option, such as when the terminal device can't have the key configured at the factory, the necessary field-installation expertise is unavailable, or there is no key distribution system available. In these instances, public-key exchange (PKE) is an ideal solution — though it adds considerable complexity. With PKE, one of several methods is used to select and combine two large numbers, and then send one number and the resulting combination to the recipient. The recipient derives a session key that is known to the sender and this establishes a channel to encrypt/decrypt traffic.

While technically complex and potentially too resource-intensive for an embedded system, PKE can actually simplify system deployment and operation because the sender and receiver don't need prior knowledge of one another and manual configurations can be minimized. This approach is often used on Linux-based systems that communicate over IP, because the necessary resources for PKE are often already present.

9. Transport Layer Security

Transport Layer Security (TLS) is the current standard for the widely implemented Secure Sockets Layer (SSL) protocol. It provides a standard framework for PKE and encryption to secure traffic between devices. However, for resource-limited embedded systems, the memory and processing requirements for the TCP/IP stack may be impossible to support. That's why TLS is often used on larger embedded systems (e.g., those running Linux) where communication occurs in IP sessions such as TCP. Even smaller embedded systems may have the resources to support TLS, but this requires careful evaluation.

10. Wi-Fi Protected Access (WPA2/WPA3)

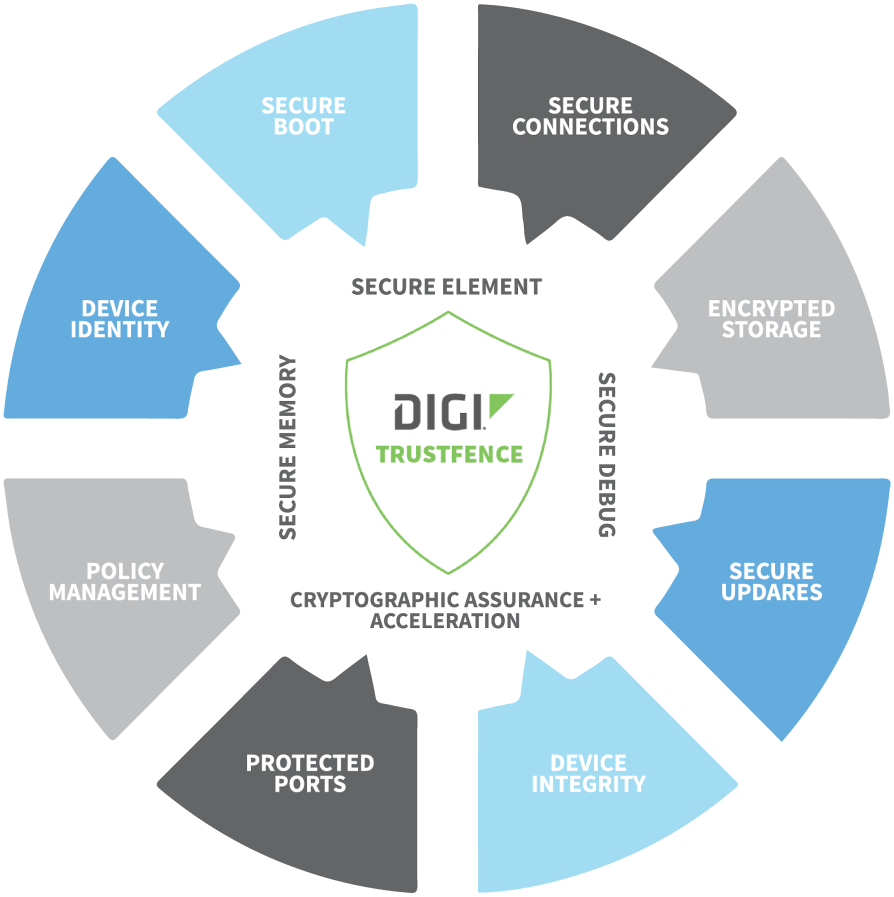
When a wireless device uses Wi-Fi (802.11) for communication, the WPA2 suite of standards can secure the communication channel with a secure key exchange and encryption. This widely deployed protocol allows interoperability of systems from different manufacturers. The latest generation of Wi-Fi Protected Accesss is WPA3, which brings the next level of security to devices, while allowing backwards-compatibility with older WPA2 systems.

Digi TrustFence for Digi ConnectCore SOMs

To help designers and builders effectively respond to the IoT security mandate, Digi offers Digi TrustFence®, an industry-leading security framework that leverages the built-in security features of application processors with state-of-the-art software to simplify the process of securing connected devices. Digi TrustFence is integrated into all Digi ConnectCore® System-on-Modules (SOMs).

Digi TrustFence security features include:

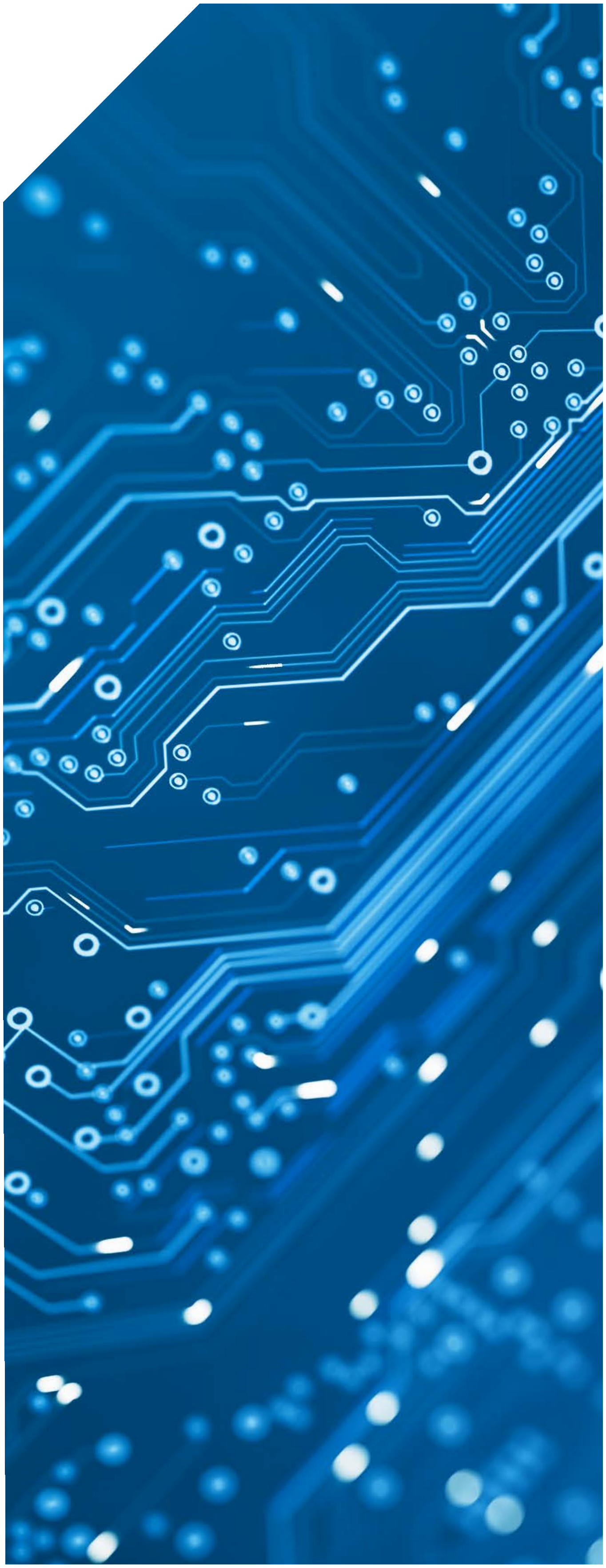
- **SECURE BOOT**
  TrustFence ensures that programs and code running on the device are from an approved source or manufacturer and have not been modified.
- **SECURE CONNECTIONS**
  Enterprise-level data encryption protocols for data in motion and over-the-air (OTA) transmissions ensure the integrity of data flowing across the network.
- **ENCRYPTED STORAGE**
  Local file system encryption keeps your internal data safe.
- **PROTECTED PORTS**
  Hardened, access-controlled internal and external ports prevent unwanted local intrusion.
- **DEVICE IDENTITY**
  Root of trust, certificate management, and secure key storage protect the identity of your device.
- **DEVICE INTEGRITY**
  Tamper-proofing and device-integrity monitoring with low-power support protect against physical intrusion.
- **ONGOING MONITORING AND SUPPORT**
  Digi provides ongoing threat measurement and monitoring services, performs external security audits and proactively communicates with customers regarding possible upcoming threats.



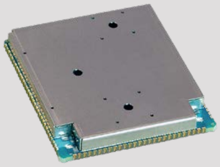
Digi System-on-Modules (SOMs) incorporate the comprehensive security features of Digi TrustFence, enabling OEMs to build secure, reliable and pre-certified connected products faster, and with less cost and less risk. OEMs can build connected, embedded products on Digi ConnectCore SOMs and capitalize on the out-of-the-box, integrated security of Digi TrustFence.

Digi ConnectCore 8

The Digi ConnectCore® 8 scalable family of SOMs is built on the NXP i.MX 8 processor series. It provides a comprehensive development platform with developer tools and software libraries, as well as Android and Yocto-based Linux. Digi ConnectCore 8 modules enable sophisticated capabilities in Human-Machine Interface (HMI), audio/voice, edge compute, machine learning, Artificial Intelligence (AI), cybersecurity and more.



- **Digi ConnectCore 8X:** The Digi ConnectCore® 8X SOM is based on the NXP i.MX 8X processor. It offers OEMs a cost-effective SOM platform that measures just 40 mm x 45 mm with the Digi SMTplus® surface mount form factor, and easy-to-use edge castellated SMT technology or a versatile LGA option for access to virtually all interfaces, including Wi-Fi and Bluetooth.



Digi ConnectCore 8X

- **Digi ConnectCore 8M Nano:** Digi ConnectCore® 8M Nano is based on the NXP i.MX 8M Nano processor. It helps OEMs get to market faster, with a lower total cost of ownership. The 8M Nano offers advanced connectivity and multimedia capabilities and streamlines development for advanced industrial, medical, agricultural and transportation applications. It offers pre-certified dual-band 802.11a/b/g/n/ac 1x1 and Bluetooth® 5 connectivity.

Digi ConnectCore 8M Nano

- **Digi ConnectCore 8M Mini:** Digi ConnectCore i.MX 8M Mini is a quad-core SOM that supports Digi Embedded Yocto and Digi Embedded Android for application flexibility. The available development kit helps you build products faster, on a productization-ready platform.

Digi ConnectCore 8M Mini

Digi ConnectCore 6 and 6UL

Digi ConnectCore 6 and 6UL SOMs are built on the NXP i.MX6 and i.MX6 UL application processors.

- **Digi ConnectCore 6:** Based on the NXP i.MX6 processor, this surface-mount module offers an ultra-compact, highly integrated embeddable computing solution. With speeds up to 1.2 GHz and fully pin-compatible, single-/dual-/quad-core variants, the ConnectCore 6 offers a future-proof solution with scalable performance and pre-certified wireless 802.11a/b/g/n and Bluetooth 4.0/Bluetooth LE connectivity.

Digi ConnectCore 6

- **Digi ConnectCore 6UL:** The Digi ConnectCore 6UL SOM is based on the NXP i.MX6UL-2 528 MHz processor. This surface-mount module offers scalable performance and pre-certified wireless 802.11a/b/g/n/ac and Bluetooth 5 connectivity. ConnectCore 6UL is industrial temperature rated, making it an ideal SOM for a wide variety of industrial and enterprise applications.

Digi ConnectCore 6UL

About NXP

NXP Semiconductors N.V. enables secure connections and infrastructure for a smarter world, advancing solutions that make lives easier, better, and safer. As the world leader in secure connectivity solutions for embedded applications, NXP is driving innovation in the secure connected vehicle, end-to-end security and privacy, and smart connected-solutions markets. Built on more than 60 years of combined experience and expertise, the company has 44,000 employees in more than 35 countries.

Summary

Security threats to embedded devices in IoT solutions are increasingly common, as attacks have become easier to carry out. These can include confidentiality breaches, service theft, and attacks on data integrity, and service availability. The Digi TrustFence device-security framework complements the security design techniques described above and simplifies the process of securing your connected devices.

Visit the Digi website to learn more about Digi ConnectCore products or contact a Digi expert and get started today.

Why Digi?

Digi is a complete IoT solutions provider, supporting every aspect of your project, from mission-critical communications equipment to design and deployment services to get your application designed, installed, tested, and functioning securely, reliably and at peak performance.

Digi builds its products for high reliability, high performance, security, scalability, and versatility so customers can expect extended service life, quickly adapt to evolving system requirements, and adopt future technologies as they emerge. Digi embedded modules, routers, gateways, and infrastructure management solutions support the latest connected applications across verticals, from the enterprise to transportation, energy, industrial and smart cities use cases.

Our solutions enable connectivity to standards-based and proprietary equipment, devices, and sensors, and ensure reliable communications over virtually every form of wireless or wired systems. Our integrated remote management platform helps accelerate deployment and provide optimal security using highly efficient network operations for mission-critical functions such as mass configuration and firmware updates, as well as system-wide monitoring with dashboards, alarms, and performance metrics.

Company Background

- Digi has been connecting the "Internet of Things" — devices, vehicles, equipment and assets – since 1985
- Digi is publicly traded on the NASDAQ stock exchange: DGII
- Headquartered in the Twin Cities of Minnesota, Digi employs over 700 people globally, and has connected over 100 million devices worldwide

As an IoT solutions provider, Digi puts proven technology to work for our customers so they can light up networks and launch new products. Machine connectivity that's relentlessly reliable, secure, scalable and managed — and always comes through when you need it most. That's Digi.

Learn more on our About Digi page.

Contact a Digi expert and get started today

PH: 877-912-3444
www.digi.com

Digi International Worldwide Headquarters
9350 Excelsior Blvd. Suite 700
Hopkins, MN 55343

/digi.international　@DigiDotCom　/digi-international

While every reasonable effort has been made to ensure that this information is accurate, complete, and up-to-date, all information is provided "AS IS" without warranty of any kind. We disclaim liability for any reliance on this information. All registered trademarks or trademarks are property of their respective owners.