Electronic Design

By BRAD REX, Senior Manager, Microcontroller Business Development, and KAUSHAL VORA, Director, Strategic Partnerships & Global Ecosystem, Renesas Electronics, www.renesas.com

Conquer the Common Security Challenges Plaguing Embedded IoT Designs

Multiple standards and new threats further complicate the already complex fabric of embedded IoT security. However, embedded developers can explore a number of approaches to build a stronger-than-ever root-of-trust.

Security is a fact of life for embedded IoT development—but that doesn't mean that it's simple or straightforward. In fact, even veteran developers can be puzzled by the multiple standards, evolving threats, and contrasting approaches to IoT security. Before starting design and development, embedded developers can explore a number of approaches to help ensure the security of their designs.

IoT Security Is a Moving Target

By 2020, the world will be home to an estimated 31 billion IoT (Internet of Things) devices—almost four times the number of humans on earth. However, many of these devices will have limited or flawed security controls that will make them vulnerable to hacking.

Why are so many IoT devices designed with weak security? The primary reason is that developers confront a phalanx of challenges and complexities as they begin securing their embedded applications and devices. The threat landscape continues to evolve while security standards multiply and grow more complex. Increasingly, applications are expected to meet multiple standards, limiting device compatibility and flexibility.

Memory Protection Units
- Limit CPU access to certain memory areas

Hardware Encryption
- RSA, AES, SHA
- Trusted Secure IP and the Secure Crypto Engine

Special Features
- TRNG
- Chip Unique ID

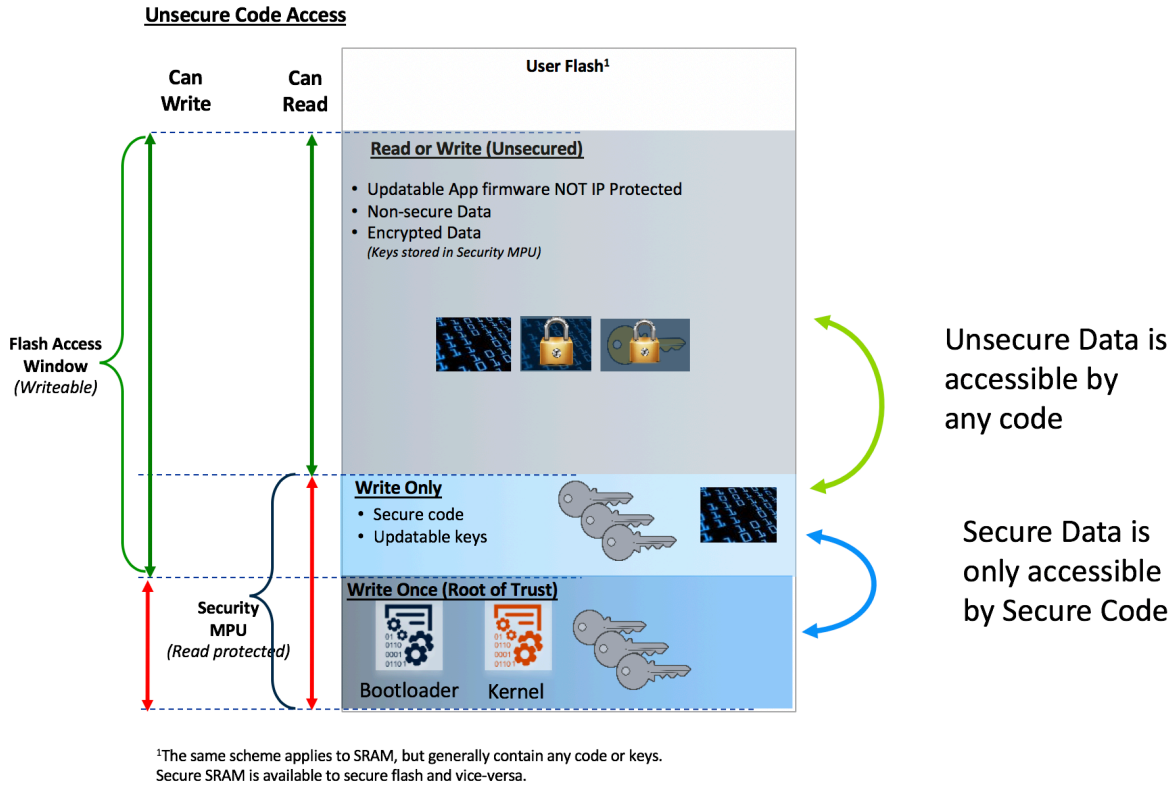
Advanced Memory Protection
- Protect customer SW IP from data access

Flash Area Protection
- Protect against unexpected erasing/programming

ID Code Protection
- Flash memory protection from programmer and debugger

Certifications / Endorsements!

1. What do you look for in a holistic security solution?



2. Memory protection units and flash area protection enable isolation and ensure that Secure Data is accessible only by Secure Code.

The Foundation for Securing Your Embedded Device

Not so long ago, securing applications wasn't such an overriding concern as it is today, because most devices and applications weren't connected like they are now. Even the most basic items—from toasters to bathroom mirrors—can now be connected through the IoT to the internet or the cloud. In the rush to get these products to market, security is often overlooked or only addressed when it's too late.

Building security into IoT devices from the start to protect data and functionality from cyber threats is now a critical concern for developers. Implementing multiple layers of defense that take advantage of the latest security advances in both hardware and software to provide in-depth, comprehensive protections should be the first step of a strategic approach to device security *(Fig. 1)*.

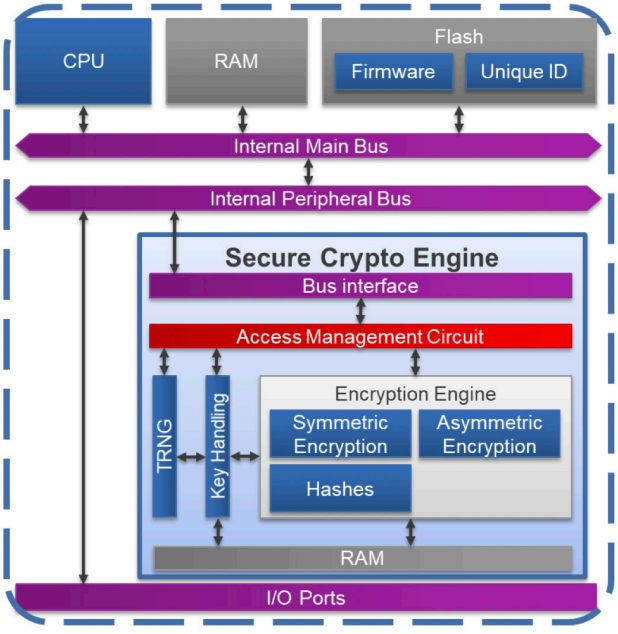
In terms of hardware, effective security should include secure key management to ensure that keys aren't accessible in an unencrypted state. For truly secure device-unique identity and provisioning, the device should be able to securely generate and store keys, including private keys. The device also should offer hardware-accelerated encryption, hashing, and true random number generation, which accelerates cryptographic operations. Secure memory access is another important hardware feature, as it enables protection of specific regions of RAM and flash memory from unauthorized access *(Fig. 2)*.

To support comprehensive security, select software that includes driver-level APIs to provide an easy interface to hardware security features. Your software should also offer cryptographic libraries with a wide range of security features available via APIs, including macro-level security functions, root-of-trust, and the ability to recognize trusted sources and code *(Fig. 3)*.

Because IoT devices require connectivity, your software should support common communication protocols and transports, such as Hypertext Transfer Protocol Secure (HTTPS), Transport Layer Security (TLS), and other cloud-specific protocols. In addition, to complete your development environment, your software should include compatible and integrated stacks, libraries, HAL drivers, and potentially a real-time operating system (RTOS).

The one-size-fits-all approach to security doesn't address the real-world requirements of device developers. Instead, there are multiple approaches to embedded security, providing a multi-tiered development infrastructure that provides in-depth security protection for a wide variety of embedded products.

For developers who prefer a platform-based approach, a comprehensive, qualified development environment such as the Renesas Synergy Platform includes production-grade soft-

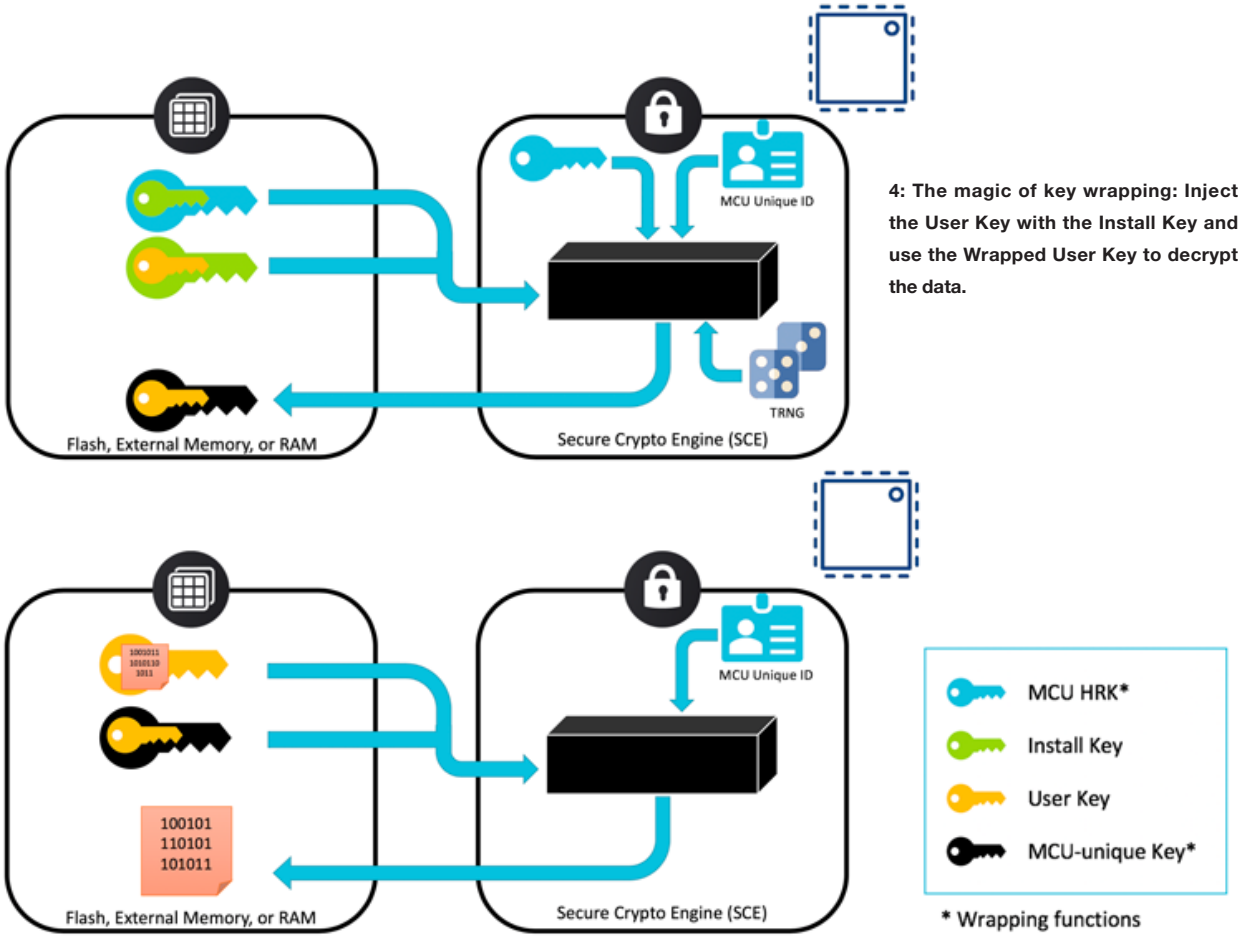


3. Example of a Secure Crypto Engine—a subsystem managed and protected by dedicated control logic.

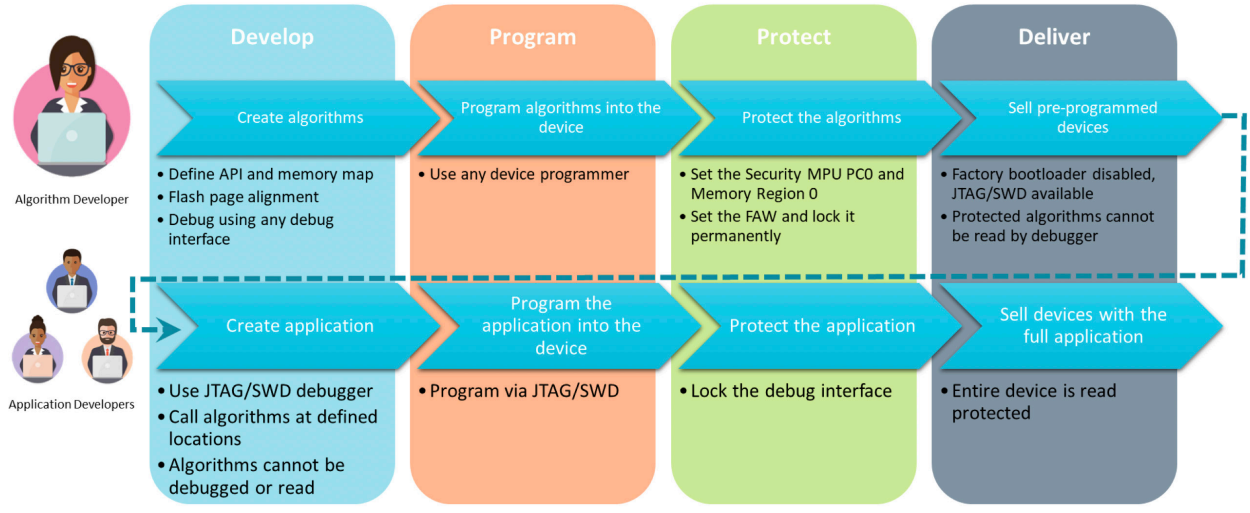
ware and a scalable family of pin-compatible MCUs, pre-integrated and pre-tested to provide security at multiple levels.

Developers who prefer greater platform flexibility can explore MCUs like the Renesas RA Family, which delivers an option that combines Arm Cortex-M cores and embedded-system peripheral IP from Renesas. The RA's Flexible Software Package (FSP) provides optimized HAL drivers as well as a baseline software platform leveraging Amazon FreeRTOS and associated middleware. Designed for flexibility, it facilitates incorporation of a developer's middleware and libraries of choice.

Whichever approach you choose, look for hardware acceleration for the most prevalently used cryptographic algorithms (RSA/ECC/DSA/AES/SHA), as well as key generation and a true random number generator (TRNG). In addition, companies like Renesas offer MCU-unique key wrapping *(Fig. 4)* that performs key binding by encrypting keys specifically for each MCU, so that keys are accessible only within the SCE module on the individual MCU that performed the



4: The magic of key wrapping: Inject the User Key with the Install Key and use the Wrapped User Key to decrypt the data.



5. This use-case example demonstrates pre-programmed algorithms—specifics for devices with a security MPU.

wrapping. The wrapped keys can be stored in non-secure memory; therefore, even if the entire MCU contents are copied onto another device, the keys can't be utilized or exposed.

Securing Your IP from Unauthorized Production

Nobody wants their products replaced by imitations or clones. Protecting your intellectual property from unauthorized production requires secure manufacturing systems to mitigate risk and maintain the integrity of your production process.

Secure firmware flash programming solutions, including secure boot manager solutions, enable developers to dependably and securely program authorized firmware into approved flash-memory devices in remote manufacturing facilities. This protects the firmware from being pirated, modified, or installed on cloned hardware.

The boot manager also delivers a strong root-of-trust that provides unique identities, hardware protected keys, secure boot loader, secure flash update module, and cryptographic APIs to interface with the MCU hardware. The boot manager pre-loads the root-of-trust through a secure connection to a high-volume programmer system designed for manufacturing and provisioning of processing units. The provisioned chip stores the data securely and maintains tight control on how it's used.

The secure boot manager is also able to securely update authorized firmware to the MCUs' flash memory even after products are in the field. The on-chip root-of-trust validates and decrypts the firmware before flash programming—all securely provisioned via secure cloud infrastructure made more reliable and trustworthy with cloud connectivity solutions.

Managing the Complexity of Security

If you're starting from scratch, designing in-depth, layered security for embedded designs can be challenging and time-consuming. The platform-based approach has all new and relevant protocols and other security safeguards built in, simplifying complex functions encountered while developing secure connected embedded systems.

Any approach, whether it's a closed platform or an open one with greater platform flexibility, support public key infrastructure (PKI) and pre-shared key (PSK) support, increasing development options. PKI is a cryptology methodology that offers authentication via digital certificates. PSK security mechanisms are an encryption model in which authentication is authorized when both peers in a digital connection specify the same key.

In addition, MCUs with integrated security offer the flexibility to reuse and expand upon existing infrastructure, as well as the ability to enhance it efficiently and precisely as required for each application *(Fig. 5)*.

Defend Against Multiple Security Threats

It's scary out there: Today's cyber-threat landscape is filled with multiple bad actors and risks, and exploits and attack vectors await the unprepared and unprotected. Protecting a device against multiple security threats requires securing the device's identity through hardware-based key generation.

Establishing a strong device identity with layered IoT security protections enables devices to be individually secured and to engage in encrypted communication with other secured devices and services.

- *Trust:* The device must authenticate its identity as soon as it connects to a network to create trust between other devices, services, and users.
- *Privacy:* Certain types of data captured and shared within IoT networks must be kept private and secure to meet regula-

tory compliance.

• *Integrity:* Data integrity is an often-overlooked requirement of layered security, involving the assurance that data shared within networks hasn't been altered.

Digital data security for stored data is also a top priority for safeguarding against multiple security threats. Data at rest refers to data not actively in motion between devices or networks, usually parked in SRAM or non-volatile storage. Controlling access to stored data reduces the attack surface and increases system security.

Offering data access controls, including read, write, read-write and write-once protections, helps to safeguard data at rest. Remote updates in the field ensure that security software and firmware are up-to-date and provide protection against the latest cyber threats.

Ensure You Deliver a Secure Device, Even If You're Not a Security Expert

Not everyone has the training or experience to understand all of the ins and outs of embedded security, but steps can be taken to ensure that you put in place the basics for delivering a secure IoT application.

First, delivering comprehensive, in-depth security protection for products based on embedded devices requires multiple protocols and safeguards that work together to provide security at many levels.

The platform-based approach can give you a head start by delivering a complete development environment complete with a unique, built-in set of hardware and software security capabilities.

Design and development resources, such as an online library of application projects with step-by-step instructions, provide guidance on building end-to-end security solutions. And a large, robust ecosystem of partners can help speed development and extend deep expertise into your security solution development.

With Security Support, Focus on Design Features that Differentiate

When it comes to streamlining the process of securing new IoT applications, choosing the right MCU is the first step. This will streamline your security workflows, allowing developers to focus on designing the features and capabilities that will make your product stand out.

Platform-based approaches provide functionalities that work together to deliver security at multiple levels. This is important because malicious agents can take advantage of vulnerabilities in embedded designs when variations in design and security protocols create weak points that are hackable. This is particularly a risk when MCU hardware, software, communication stacks, and drivers haven't been standardized into a fully integrated framework.

A platform ensures that applications are built on a secure, robust technology foundation. It also allows designers to focus their time and skills on innovations that address fast-moving IoT market opportunities and consumer demands.

MCUs outside of a platform can offer flexibility and deliver best-in-class security IP and peripherals that provide a highly optimized feature set for holistic security protections. In addition, an active ecosystem of partners and other resources, such as the Arm ecosystem, provides the flexibility and expertise to deliver innovative designs with the multiple layers of defense now required by the market.

The option of outsourcing development of specific security features or functionalities to trusted partners can save time and strengthen the final product.

Conclusion

There are multiple ways to take advantage of the latest breakthroughs in hardware and software security to deliver in-depth, comprehensive protections with layered security. Whether choosing a platform-based approach or a more flexible MCU-based approach, building on a strong root-of-trust enables developers to secure IoT devices, services, and networks at a deep level, and extend protections to secure and scalable manufacturing and defense of intellectual property across the product lifecycle.

BRAD REX is Senior Manager, Microcontroller Business Development at Renesas Electronics, where he is responsible for marketing Renesas' microcontroller hardware and software solutions.

KAUSHAL VORA is Director of Strategic Partnerships & Global Ecosystem at Renesas Electronics, where he is responsible for defining, establishing, and managing the company's microcontroller ecosystem. He leads a global marketing and application engineering team to develop software building blocks for IoT design and collaborates closely with key technology partners to complement and expand Renesas' embedded ecosystem.