Rambus

SECURITY • CRYPTOGRAPHY
White Paper

The CryptoManager Root of Trust

Implementing Security by Design

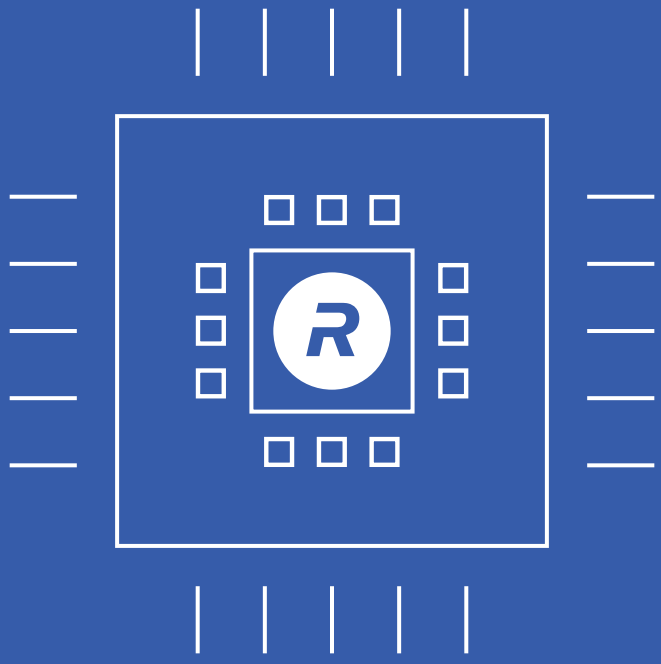


Table of Contents

Introduction

In January 2018, **Meltdown** and **Spectre** were independently disclosed by multiple security researchers, including senior Rambus technology advisor Paul Kocher and senior Rambus security engineer Mike Hamburg. The two security flaws **exploit critical vulnerabilities** across a wide range of modern processors, including Intel, ARM and AMD. Notably, however, existing RISC-V processors **remain unaffected** by both Meltdown and Spectre.

Although Meltdown and Spectre are certainly not the first high-profile semiconductor security flaws to gain widespread attention, they do represent a new class of vulnerabilities related to out-of-order and speculative execution.

"It shouldn't be surprising that microprocessor designers have been building insecure hardware for 20 years. What's surprising is that it took 20 years to discover it," **security researcher Bruce Schneier wrote in a 2018 article** about Meltdown and Spectre. "In their rush to make computers faster, [companies] weren't thinking about security. They didn't have the expertise to find these vulnerabilities. And those who did were too busy finding normal software vulnerabilities to examine microprocessors."

Paul Kocher expressed similar sentiments in an interview with **Inside HPC**.

"We need to stop trying to build one processor architecture that is great for playing video games and doing wire transfers," he stated. "We need to build architectures where there are cores and software stacks designed for security that can be slower, that can be simpler – and we need separate ones that are optimized for performance."

From our perspective, Meltdown and Spectre illustrate the critical need for a new generation of processors that execute sensitive cryptographic functions in a secure core which is physically siloed (separated) from the primary CPU.

Part 1:

The Rambus CryptoManager Root of Trust RT630

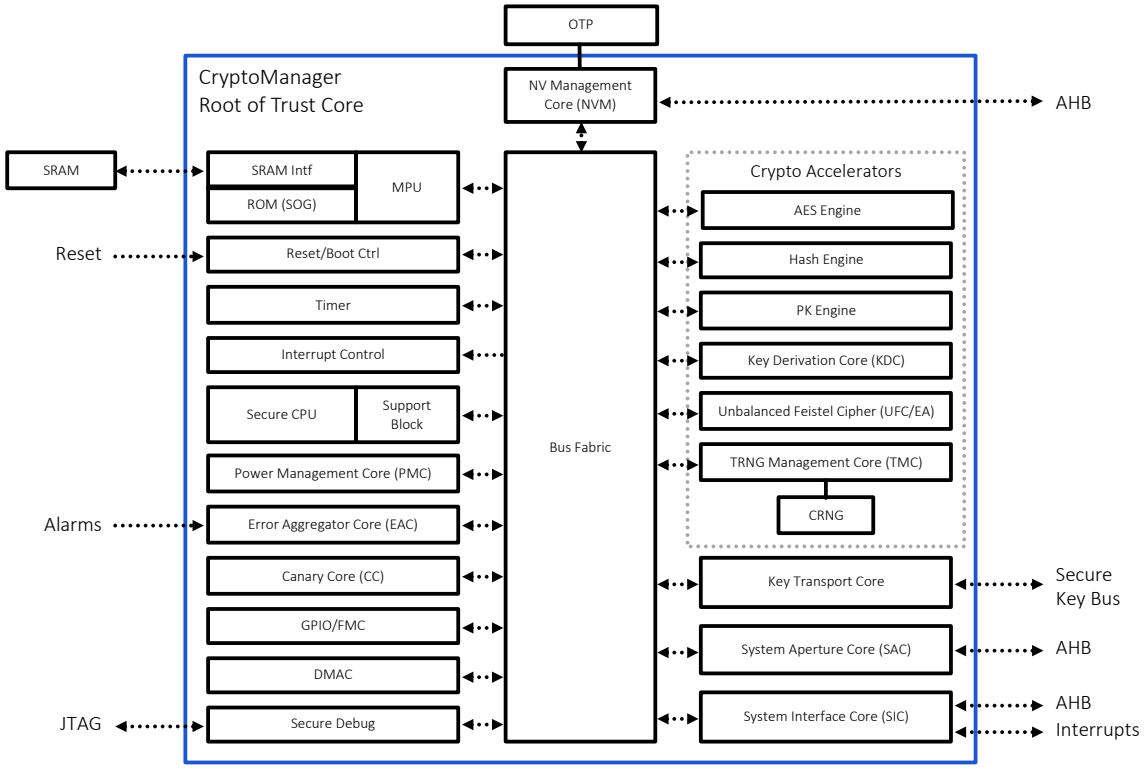
Primary Security Features

Built around a custom RISC-V CPU, the Rambus CryptoManager Root of Trust (CMRT) RT630 is at the forefront of a new category of programmable hardware-based security cores. Siloed from the primary processor, it is specifically designed to securely run sensitive code, processes and algorithms. In addition to the CPU, the CMRT contains a large set of hardware blocks arranged around an internal bus fabric.

More specifically, the CMRT comprises the following hardware components:

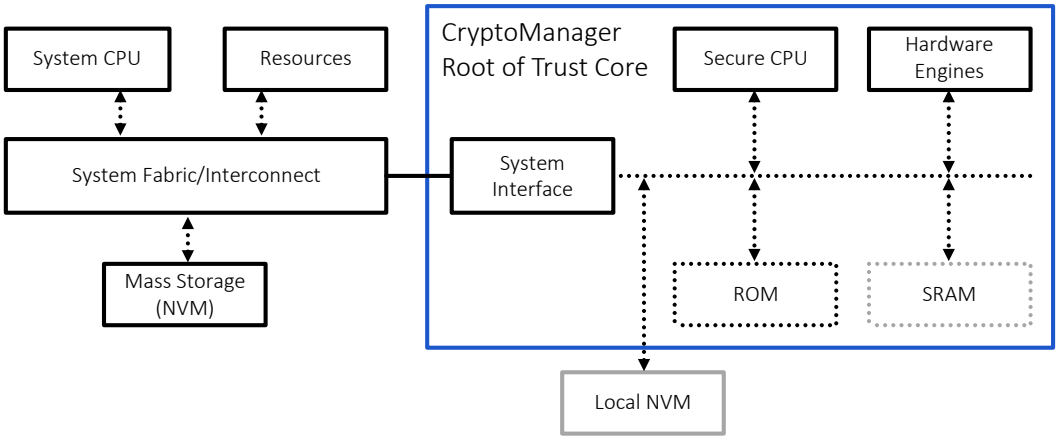
- A secure 32-bit RISC-V CPU
- AES core, secure SHA-2 hash core and an asymmetric public key engine
- Dedicated secure memories (SRAM and ROM) and non-volatile storage management core (NMC)
- Interrupt controller and timer
- Reset/boot controller
- Secure test and debug
- Both a register-mapped and memory mapped system interfaces
- DMA controller for high speed memory transfers
- Secure power management core
- True random number generator (TRNG) based on Rambus-designed chaotic entropy source
- Key derivation core (KDC) for deriving ephemeral keys form root keys
- Error aggregator core (EAC)
- Canary core for detection of glitching and over-clocking
- General purpose I/O (GPIO) and feature management (FMC) for control and monitoring of logic outside of the CMRT

Rambus CryptoManager Root of Trust RT630



The CMRT provides the primary processor with a full suite of security services, such as secure boot and runtime integrity checking, remote attestation and hardware acceleration for symmetric and asymmetric cryptographic algorithms. Access to cryptographic accelerators, keys, memory ranges and I/O pins is restricted and enforced on a hardware level. Similarly, critical operations, such as key derivation and key unwrap, are performed by – and in – hardware.

Security Services



Comprehensive Attack Resistance

The CMRT core utilizes advanced anti-tamper techniques to provide the highest level of security and protection against a wide range of attacks, including fault injection. These include logic and crypto redundancy, secure state encoding and ephemeral keys that are generated on-the-fly from multiple splits and flushed immediately after use. In addition, the CMRT core features an optional Entropic Array with a proprietary logic structure that provides robust protection against emulation and reverse engineering. The CMRT core also helps protect against:

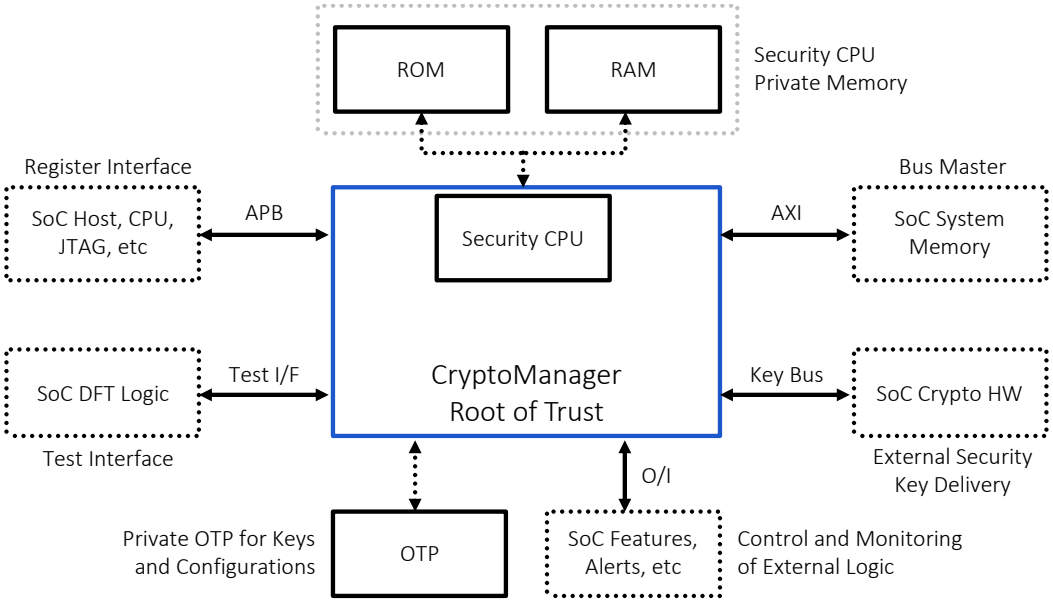
- Host processor compromise
- Non-volatile memory (NVM) key extraction, tearing and other attacks against NVM writes
- Corruption of non-volatile memory or fuses
- Test and debug interface attacks Power/EM analysis (SPA/DPA) and other **side-channel attacks, including timing attacks**
- Manufacturing/personalization facility compromise (insider attack)
- Man-in-the-middle and replay attacks
- Probing of external buses

A Fully Programmable Security SoC

The CMRT is a fully programmable, custom security SoC that offers customers a complete development environment – with full support for custom applications (called containers). SDK components include a compiler, symbolic debugger, reference HLOS components, application libraries and drivers, reference container source code, as well as FPGA models and an emulation board.

CMRT containers support a wide range of security-related applications. These include secure boot; secure storage/management of keys and other sensitive assets; user data privacy; secure firmware upgrade; device authentication and attestation; secure debug and lifecycle control; key provisioning and device personalization; as well as protocol implementation and biometric algorithms.

CryptoManager Root of Trust Core – Fully Programmable



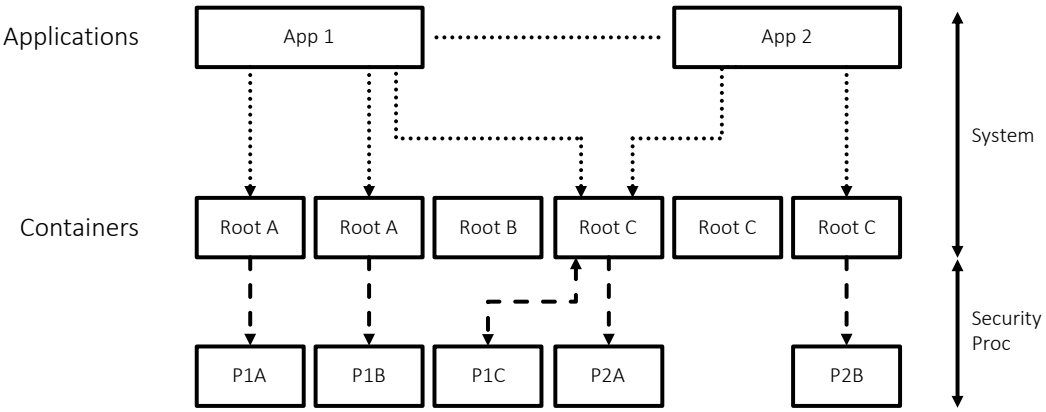
Multiple Roots of Trust

The CMRT supports multiple roots of trust, with hardware ensuring isolation of resources, keys and security assets. Each entity – such as a chip vendor, OEM or service provider – has access to its own “virtual” security core and performs secure functions without having to “trust” other entities.

This allows individual entities to possess unique root and derived keys, as well as access only to specified features and resources such as OTP, debug and control bits. Moreover, support for multiple roots of trust enables the CMRT to assign or delegate permissions to other entities at any point in the device lifecycle, while isolating (in hardware) unique signed apps that are siloed away from other programs.

These multiple roots of trust effectively create a hierarchical and secure execution environment in which mutually distrusting entities are safe to execute on the same CPU. As illustrated in the image above, applications are seen executing in the system and making calls to code fragments (containers) that must execute securely within the secure CPU domain as processes. Each of these code fragments (containers) is associated with an owner (i.e. root) that has a set of capabilities/privilege/permissions assigned to it. Moreover, each container is only permitted to utilize those capabilities directly associated with its specific signing root. To support a secure execution environment, each container is only permitted to request the privileges it actually requires.

Multiple Roots of Trust



Part 2:
Market Verticals

The versatile CMRT is targeted at multiple verticals, including the Internet of Things (IoT), the automotive space, connectivity and sensors.

IoT

According to the U.S. Department of Defense (DoD), **IoT devices pose numerous security challenges** that need to be addressed, both in specific instances and as part of a holistic approach to risk management in the information age. For its part, **the U.S. Department of Homeland Security (DHS) recommends** that IoT devices utilize chips with security integrated at the transistor level (embedded in the processor) to provide encryption, anonymity and other security functions. Meanwhile, **the European Union Agency for Network and Information Security (ENISA) recommends** that IoT devices employ a hardware-based immutable root of trust, with hardware incorporating security features to strengthen the protection and integrity of the device.

In fact, the ENISA specifically highlights specialized security chips and coprocessors that integrate security at the transistor level, embedded in the processor to provide:

- A trusted storage of device identity and authentication
- Protection of keys at rest and in use
- Protection against unprivileged users accessing security sensitive code
- Protection against local and physical attacks

The CMRT can help companies follow the above-mentioned recommendations. Indeed, its hardware-based root of trust supports device identification, mutual authentication, (verification), routine attestation checks, secure over-the-air (OTA) device updates, disaster recovery and key management, as well as the decommissioning and re-assignment of keys to better manage devices and mitigate various attacks, including distributed denial of service (DDoS).

Clearly, building security in at the design stage can help reduce potential IoT service disruptions such as those caused by DDoS attacks. Moreover, integrated security features allow manufacturers to avoid the difficult and expensive endeavor of adding security measures to IoT devices after they have already been deployed.

Automotive

Semi-autonomous and fully autonomous vehicles are essentially **a network of networks** equipped with a range of embedded communication methods and capabilities. Potential security exploits include intercepting unprotected vehicle-to-vehicle communication, the unauthorized collection of driver or passenger information, seizing control of critical systems such as brakes or accelerators, intercepting vehicle data and altering over-the-air (OTA) firmware updates.

Recently, the automotive research consortium known as **FASTR** published a **comprehensive document that provides a detailed framework** (guidelines) for secure OTA (SOTA) vehicle updates. Co-authored by Rambus security researchers, the guidelines identify a number of potential threats and attack vectors targeting OTA updates including spoofing, tampering, repudiation, escalation of privileges, information leakage and distributed denial of service (DDoS). The FASTR document also offers a detailed list of SOTA threat mitigation guidelines such as: encrypting software updates; using a signed certificate containing the public key of the entity requesting the update; digitally signing updates after encryption, with the private key of the entity requesting the updates; securing all network transactions with TLS public key authentication (signed by a trusted Certificate Authority); and (clients) performing hostname verification to ensure they are connecting the correct server.

The CMRT – which follows the SOTA guidelines – can help maintain the integrity of automotive OTA updates. Additional automotive security features supported and enabled by the CMRT include secure boot and authentication, multiple roots-of-trust, advanced anti-tamper resistance, anti-emulation/RE, E2E services, secure key storage and device personalization capabilities.

The CMRT can be seamlessly embedded in electronic control units (ECUs), infotainment head-end/gateway processors, as well as advanced driver assistance systems (ADAS) and autonomous car chips.

Connectivity

Cellular modems, network processors and WiFi chips all provide connectivity for a wide range of systems and devices. From a holistic perspective, ensuring the security of these types of connectivity chips is crucial to protecting the systems and devices that use them to communicate. An embedded hardware-based root of trust can help protect the data passing through the chips, prevent unauthorized access and mitigate distributed denial-of-service (DDoS) attacks.

More specifically, the programmable nature of CMRT allows it to effectively secure a variety of communication protocols by placing sensitive elements within the siloed boundary of the security core. This protects keys and certificates, while preventing (credential) tampering. In addition, the CMRT can also be used to facilitate the encryption and decryption of data sent over insecure links.

Sensors

Various types of sensors are increasingly being used for security-sensitive applications. Examples include biometric sensors such as fingerprint and iris scanners, security cameras and environmental sensors. Adding a hardware-based root of trust to sensor chips can help prevent unauthorized access to sensitive user data, including fingerprints and video streams.

Since the CMRT is programmable, sensor data can be processed inside the security boundary of the root of trust and linked to security-critical functions without exposing the sensor data itself. For example, an iris matching algorithm can be implemented within the core and linked to a payment application – without exposing user iris templates to the main processor.

Part 3:
CryptoManager – A Complete Security Solution

The CMRT creates a secure foundation for the comprehensive Rambus CryptoManager platform, which also includes the CryptoManager Provisioning Infrastructure and CryptoManager IoT Security Service. Put simply, our CryptoManager platform spans silicon to services, beginning with the chip manufacturing process and continuing throughout the device lifecycle.

CryptoManager Provisioning Infrastructure

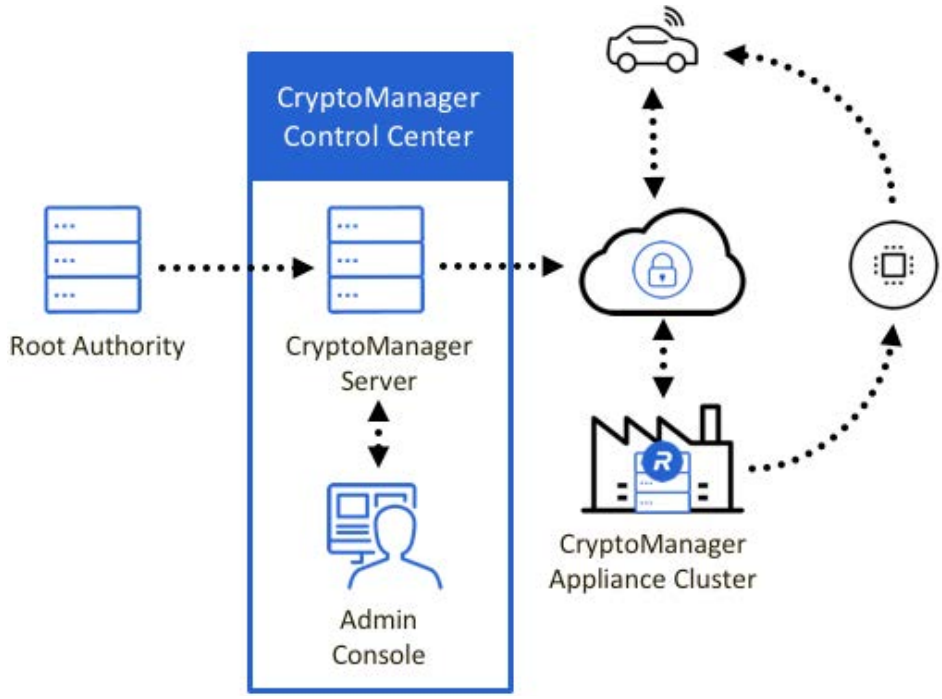
The CryptoManager Infrastructure is designed to seamlessly operate with the CryptoManager Root of Trust core, thereby enabling devices to be personalized and configured during the early stages of the semiconductor manufacturing process. The Rambus CryptoManager Infrastructure also facilitates the secure provisioning of cryptographic keys and other sensitive data in the CMRT throughout the distributed manufacturing supply chain – and even in the field.

A wide range of secure operations are supported, including key delivery and programming, protection of sensitive debug and test traces, automated configuration of chip features, as well as secure audit and logging. The CMRT and Infrastructure can also help companies create new revenue streams by facilitating in-field value add features, enabling downstream key provisioning and supporting end-to-end security services.

Additional capabilities of the CryptoManager Provisioning Infrastructure and the CMRT include:

- Protects keys in insecure manufacturing environments
- Provisions keys without TEE or OS boot as early as wafer sort or final test
- Delegates flexible security functionality throughout the supply chain without exposing manufacturing secrets
- Enables secure unlock of debug and test ports
- Grants different levels of temporary and permanent access to multiple entities at various stages of the manufacturing lifecycle

CryptoManager Control Center



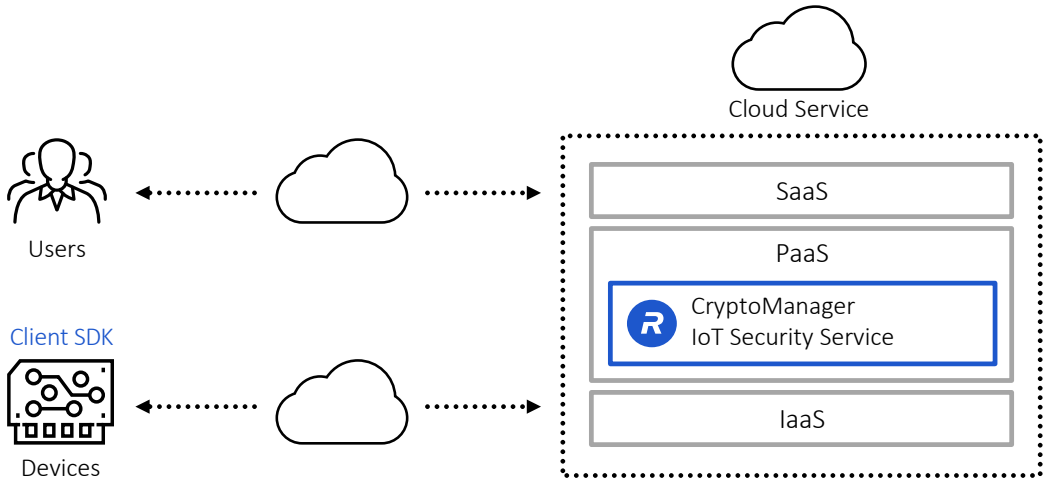
By establishing the trust chain early in the silicon manufacturing process, the CryptoManager Infrastructure enables secure provisioning and robust audit of security-related activity throughout all phases of the chip life cycle. The Rambus CryptoManager Infrastructure has been deployed and proven in the most demanding of microelectronics manufacturing environments – with billions of chips configured to date.

CryptoManager IoT Security Service

The turnkey Rambus CryptoManager IoT Security Service is the Software as a Service (SaaS) component of the comprehensive Rambus CryptoManager security platform. In its optimal configuration, hardened security is provided by the CMRT, although the service also offers support for multiple third-party chipsets and device vendors.

The CryptoManager IoT Security Service features a client software development kit (SDK) that is pre-integrated with the chipset SDK and IoT platform as a service (PaaS) provider. When a supported device first powers up and connects to the internet, it is automatically identified and authenticated. The IoT Security Service utilizes the (device) root of trust to authenticate and provision the target device, thereby creating a secure connection between the device and its service.

CryptoManager IoT Security Service Components



The Rambus CryptoManager IoT Security Service also protects IoT devices from a range of cloud-based attacks including DDoS, cross-site request forgeries and digital offensives against unprotected REST APIs. Moreover, the IoT Security Service reduces the risk of devices being hijacked, copied, re-purposed, or even disabled by leveraging strong authentication between the device and service. In addition, the service quarantines and recovers compromised or cloned devices.

Key CryptoManager IoT Security Service features include:

- Secure device lifecycle management
- Secure OTA credential provisioning
- Device attestation
- Advanced device monitoring capabilities
- Device decommissioning and re-assignment
- Disaster mitigation and recovery

Conclusion

Meltdown and Spectre illustrate the critical need for a new generation of devices that execute sensitive security functions in a secure core which is physically separated from the primary CPU. Built around a custom RISC-V CPU, the CryptoManager Root of Trust RT630 is at the forefront of a new category of programmable hardware-based security cores. Siloed from the primary processor, it is specifically designed to securely run sensitive code, processes and algorithms. Indeed, the CMRT provides the primary processor with a full suite of security services, such as secure boot and runtime integrity, remote attestation and broad crypto acceleration for symmetric and asymmetric algorithms.

Targeted at multiple verticals, the versatile CMRT creates a secure foundation for our comprehensive CryptoManager suite, which also includes the CryptoManager Provisioning Infrastructure and CryptoManager IoT Security Service. Put simply, Rambus' CryptoManager solution spans silicon to services, beginning with the chip manufacturing process and continuing throughout the device lifecycle.

For more information, visit
rambus.com/cryptomanager

1050 Enterprise Way, Suite 700
Sunnyvale, CA 94089