Rambus

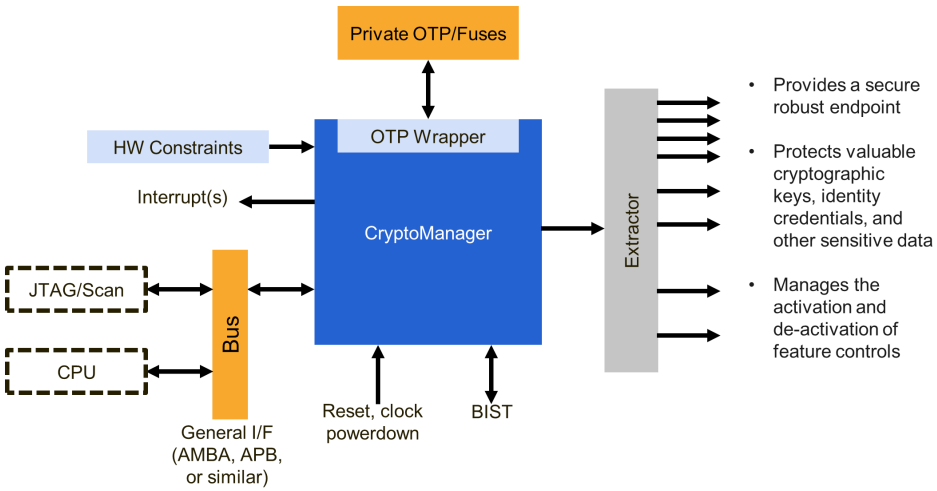
CRYPTOGRAPHY RESEARCH
CryptoManager Platform
Product Brief

CryptoManager Security Engine

A high-security core with a silicon root-of-trust for key and feature provisioning throughout a device's lifecycle.

Overview

Robust security starts with the design of the SoC and continues with the manufacturing supply chain. Our CryptoManager™ Security Engine is a high-security silicon IP core that is integrated into the SoC of an intelligent device, such as the application processor of a smartphone or a tablet. It includes a hardware root-of-trust, providing the device with a secure endpoint. The Security Engine addresses critical device security needs, including the provisioning and management of cryptographic keys, authorization of debug modes, and programming across manufacturing stages, including wafer test, package test, device assembly, and return authorization.



Highlights

Superior Security

- Provide a robust, secure endpoint
- Protect valuable cryptographic keys, identity credentials, and other sensitive data
- Protect against reverse engineering and counterfeiting
- Enhance brand protection

Improved Profitability

- Reduce revenue loss by preventing unauthorized access
- Enable new revenue with feature activation
- Reduce NRE, engineering risk, and operating costs
- Simplify meeting complex security requirements

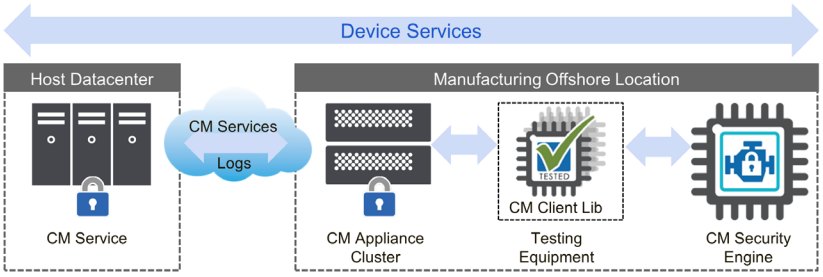
How It Works

The core stores and protects sensitive key and configuration information in One Time Programmable (OTP) memory in the SoC. For feature management, it acts as a secure endpoint for the activation and de-activation of specified feature controls. Depending on the SoC designer's requirements, this capability may be used to configure chip features during different stages of manufacturing.

The Extractor block, generated by our Configurator software tool, is a hardware and software wrapper. It allows the SoC design team to assign key and feature outputs of the Security Engine to customer-defined names and usage.

The Security Engine enforces security policies based on digitally-signed authorizations. To securely manage SoC features (such as a PLL configuration or debug mode enable), the SoC design team can simply connect the output bit(s) to the logic that controls those features. This power and flexibility enables chip companies to address complex use cases, including many unanticipated needs, while freeing SoC design teams from having to anticipate and implement complex policies.

End-to-End Security



Our Security Engine acts as a secure endpoint for the device. This allows our CryptoManager Infrastructure to communicate with the device and provision device services (operations such as feature configuration, key provisioning, and rights delegations). These device services may be authorized for provisioning at any stage of manufacturing.

Features

- Secure feature and configuration management
- Secure key management
- Trust delegation
- Entropic Array (EA) – countermeasures to protect against silicon de-processing
- Canary logic – countermeasure for glitching attacks
- Secure private memory management of OTP (or other NVM) memory
- Secure API support for the provisioning of cryptographic data and feature activation controls
- Asymmetric crypto capabilities: RSA 2048, PKCS #1, PSS, and Ferguson-Schneier key exchange
- Symmetric crypto capabilities: AES128, AES256, and SHA256
- Private bus for direct key delivery

Deliverables

Netlist or encrypted RTL

Full Documentation

- User guide
- External reference specification
- Interfaces

Tools and Scripts

- Configurator tool
- Synthesis constraints and timing check scripts

Integration deliverables

- Testbench development
- Use case vectors

Example: Debug Access Use Case

To prevent misuse of debug modes (e.g. BIST, scan, JTAG), the Security Engine can be connected to the debug mode enable, which defaults to an off (safe) setting. It can selectively enable debug features as needed, for example:

- At specified manufacturing stages (wafer test, package test), necessary debug capabilities can be temporarily enabled
- In the case of a defective chip or device, debug capability can be re-enabled for Return Merchandise Authorization (RMA) and Failure Analysis (FA)

Once the debug is completed, the Security Engine will disable the debug mode.

rambus.com/cryptomanager