STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Ira A. Fulton Schools of Engineering
Arizona State University

CSE/CEN 598
Hardware Security & Trust

Secure Computation Approaches: Trusted Execution Environment (TEE)

Prof. Michel A. Kinsy

Secure Computation Approaches

Multi-Party Computation (MPC)

Pros
- Low compute requirements
- Easy to accelerate
- Provably secure
- Supports multiple threat models
- Easy to map existing algorithms

Cons
- High communication costs
- High latency
- Information theoretic proofs are weaker than PKE ones

Fully Homomorphic Encryption (FHE)

Pros
- Very low communication costs
- Requires a single round of communications, i.e., "fire and forget"
- Useful when one side is limited in compute / memory / storage
- Provably secure – relies on strength of PKE

Cons
- Very high computational requirements
- Harder to accelerate
- Mapping existing algorithms to FHE may be difficult

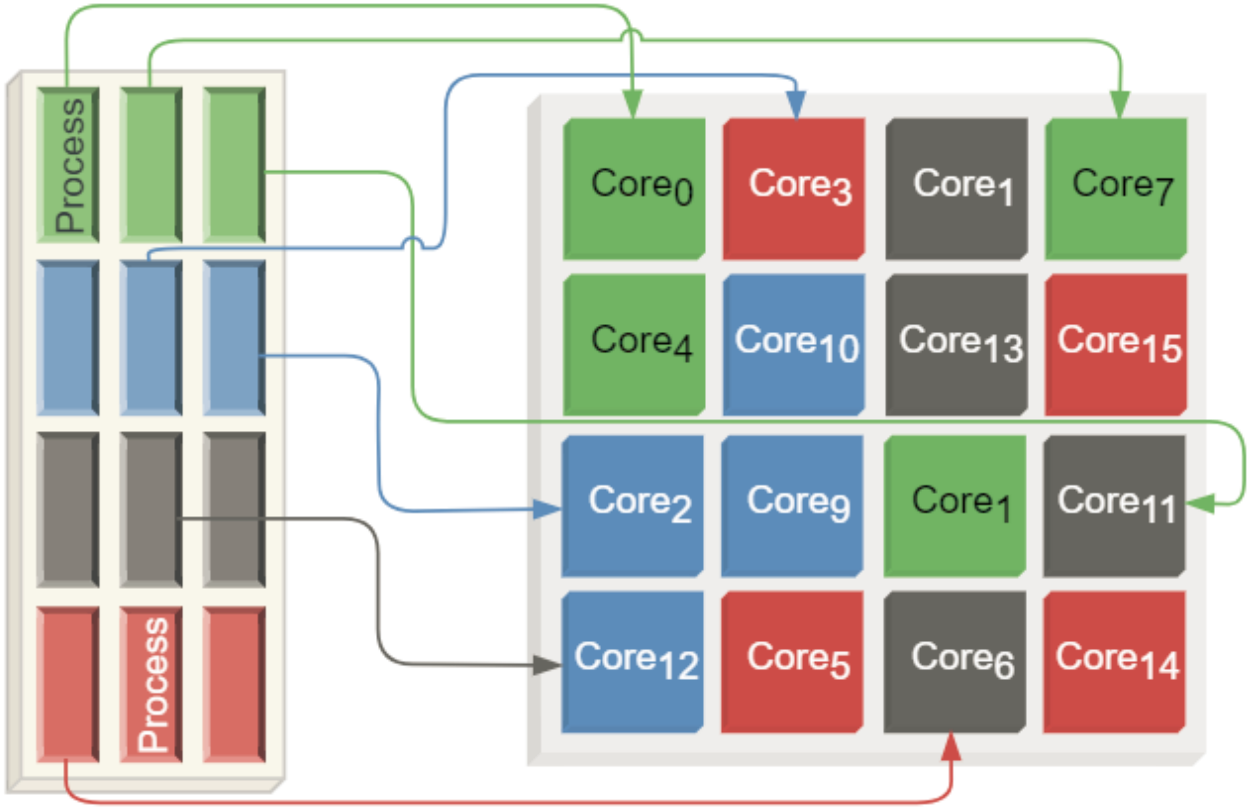
Trusted Execution Environments (TEE)

Pros
- No communication required
- Trivial to accelerate
- Great support for existing software

Cons
- Weaker security guarantees
- Cannot stop determined adversaries
- Historically plagued by vulnerabilities and breaches
- Long term deployment is difficult – TEE's can 'run out' of entropy / CRP's, etc.

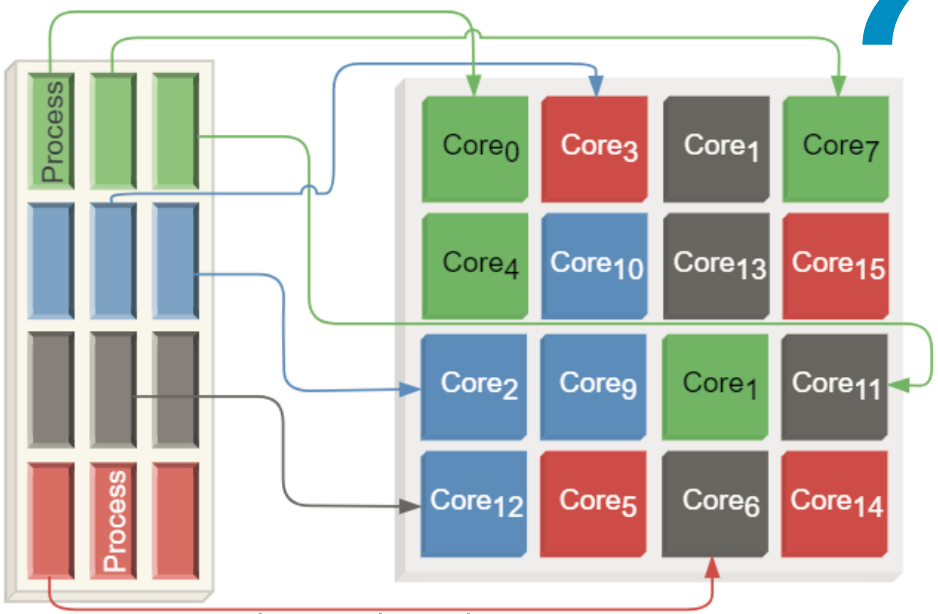
Mixed Criticality Computing Systems

- Current state of affairs:
  - Trusted/untrusted applications running on trusted/untrusted cores



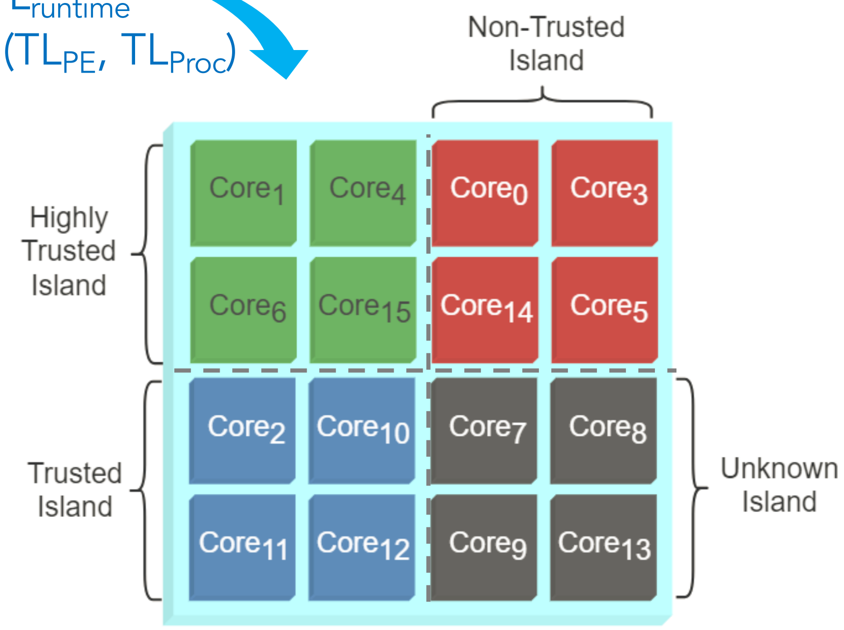
Trusted Execution Aware Design

- Develop a new trust-aware architectural framework for integrating multiple heterogeneous IPs or tenants, secure to non-secure cores, in the same chip design
  - Hardware virtualization through trusted, non-trusted and unknown island partitioning



Physical Architecture

$TL_{runtime} = \min(TL_{PE}, TL_{Proc})$



Virtual Islands

What are TEEs?

- Isolated Execution
  - Isolated data cannot be read or write by other regions
  - Dedicated memory management
- Secure Storage
  - Main memory
  - Optionally non-volatile storage

What are some major TEEs

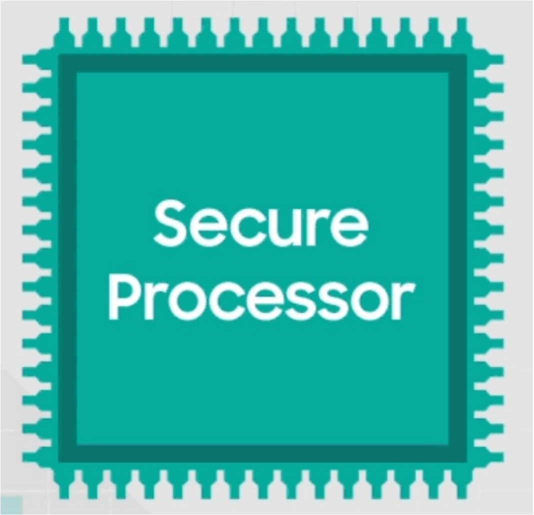
- ARM Trust Zone
  - Separates rich OS with smaller secure OS
- SGX
  - Software Guard Extension
- Sanctum
  - Builds on top of SGX
- Keystone
  - Open-source Framework, RISCV based
- AMD Platform Security Processor (PSP)
  - A trusted execution environment subsystem incorporated into AMD microprocessors

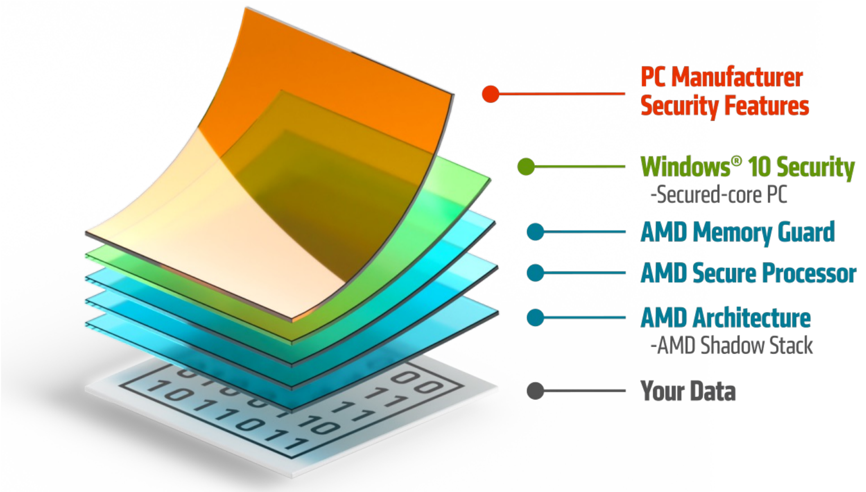
STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Ira A. Fulton Schools of Engineering
Arizona State University

Secure Processor Design

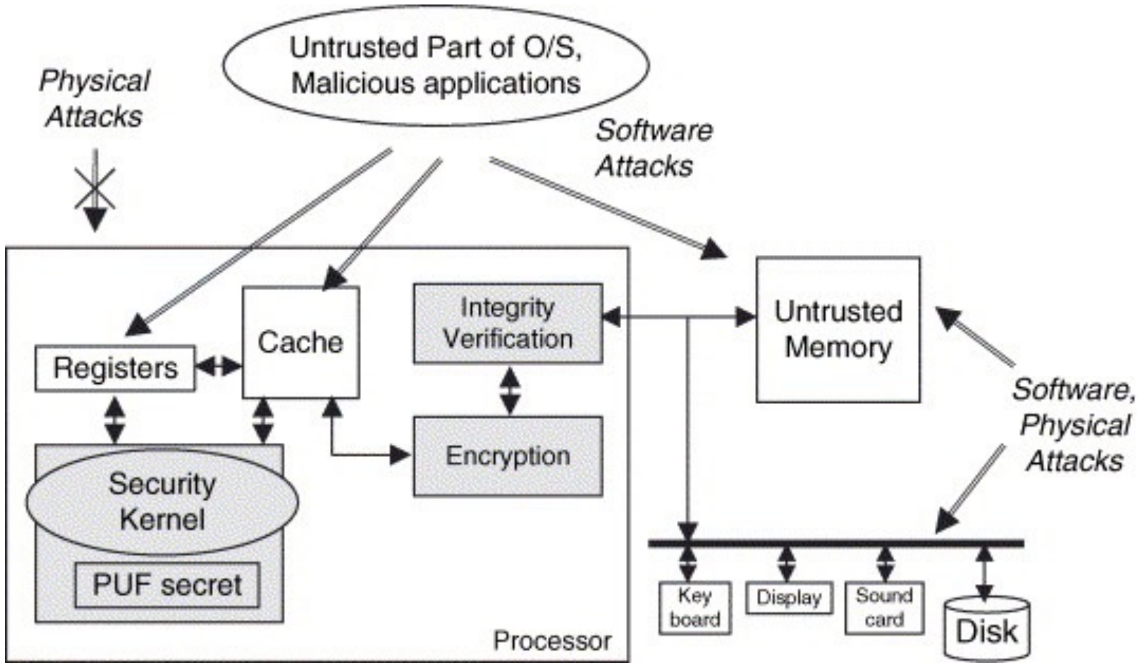
- Common approaches among the techniques:
  - A mechanism to categorize the trusted and non-trusted processes / programs / memory regions etc.
  - Separation (physically or logically) of trusted and non-trusted parties
  - Hardware-based cryptography (authentication, secret key or random number generation) to provide higher level of trust than software-based





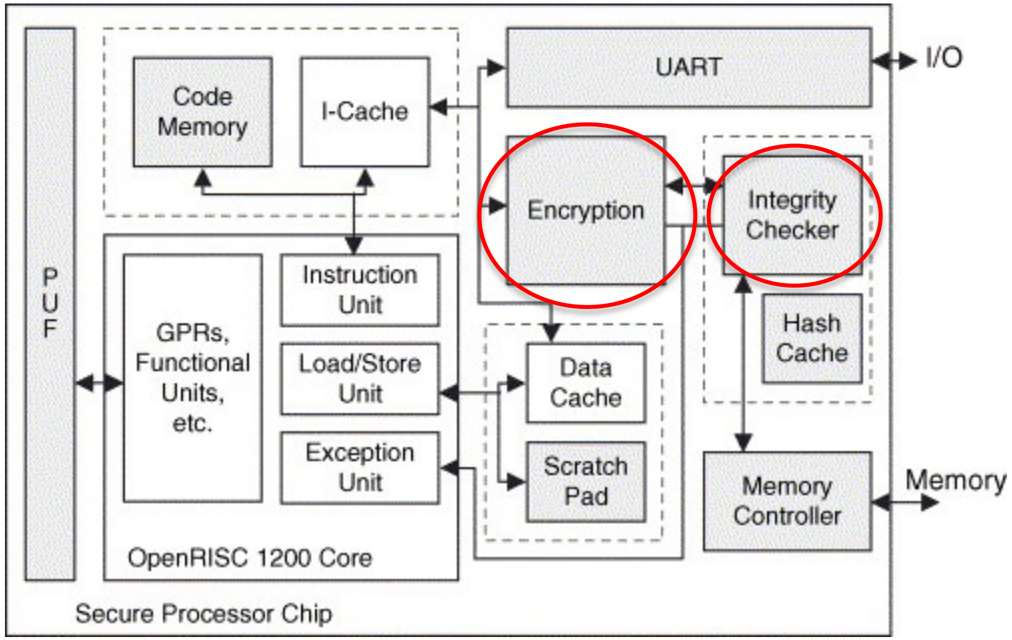
Secure Processor Design: MIT Aegis

- A single-chip secure processor that ensures the authenticate execution of programs under physical attacks
- Security foundations
  - Having all trusted components in a single tamper/probing-resistant processor
  - PUF for chip authentication and cryptographic key generation;
  - Off-chip (untrusted) memory protection



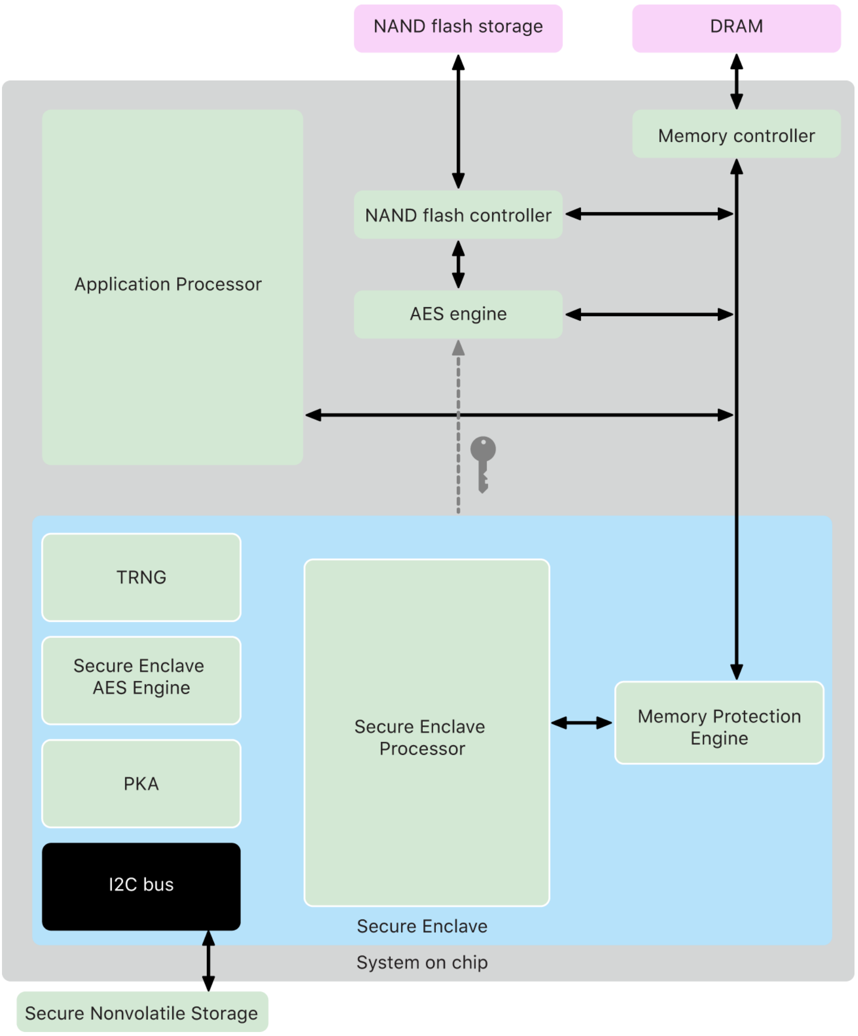
Secure Processor Design: MIT Aegis

- Pros
  - The single chip solution is more convenient to apply protection to and cheaper than multi-chip solutions
  - PUF provides unique cryptographic key that is hard to predict or model.
  - Off-chip memory is protected by integrity verification (IV) and memory encryption (ME)
- Cons
  - Latency brought by hash verification in IV and decryption in ME



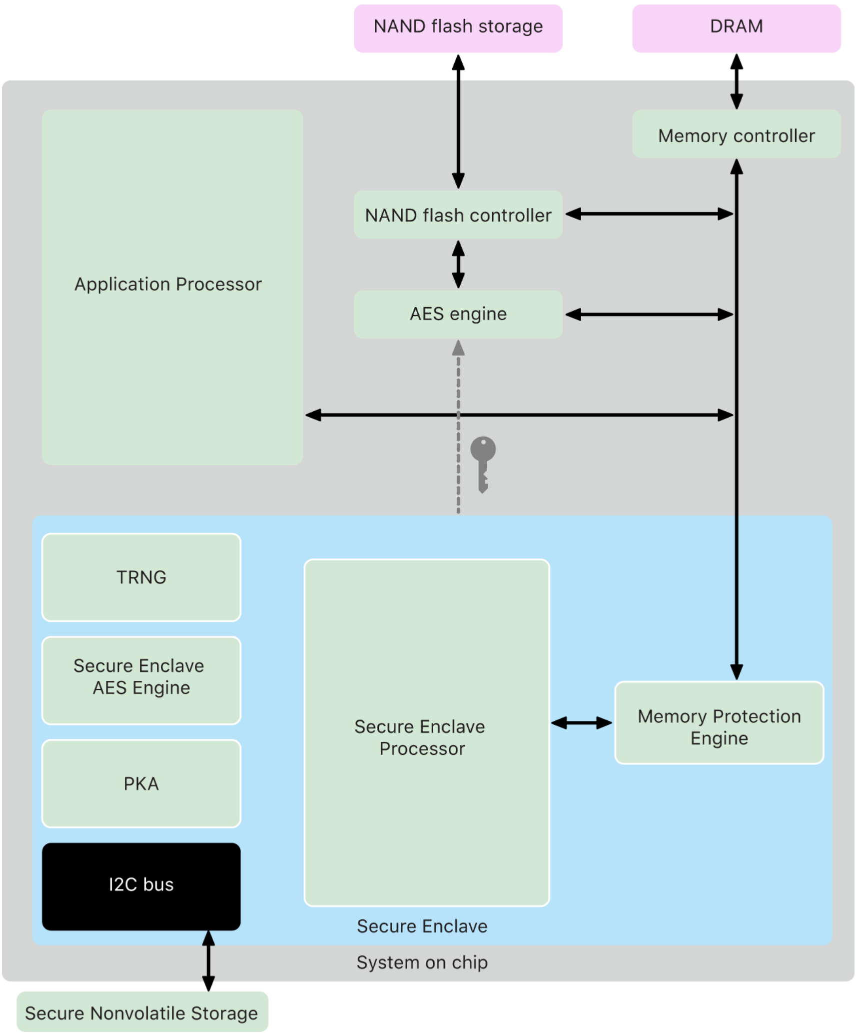
Secure Processor Design: Apple Solution

- Apples Secure Enclave Processor (SEP)
  - A processor creating a logical wall between software and sensitive security functions
- Security foundations
  - Secure Enclave Processor
    - The SEP provides the main computing power for the Secure Enclave (SE)
    - To provide the strongest isolation, the SEP is dedicated solely for SE use
      - This helps prevent side-channel attacks that depend on malicious software sharing the same execution core as the target software under attack



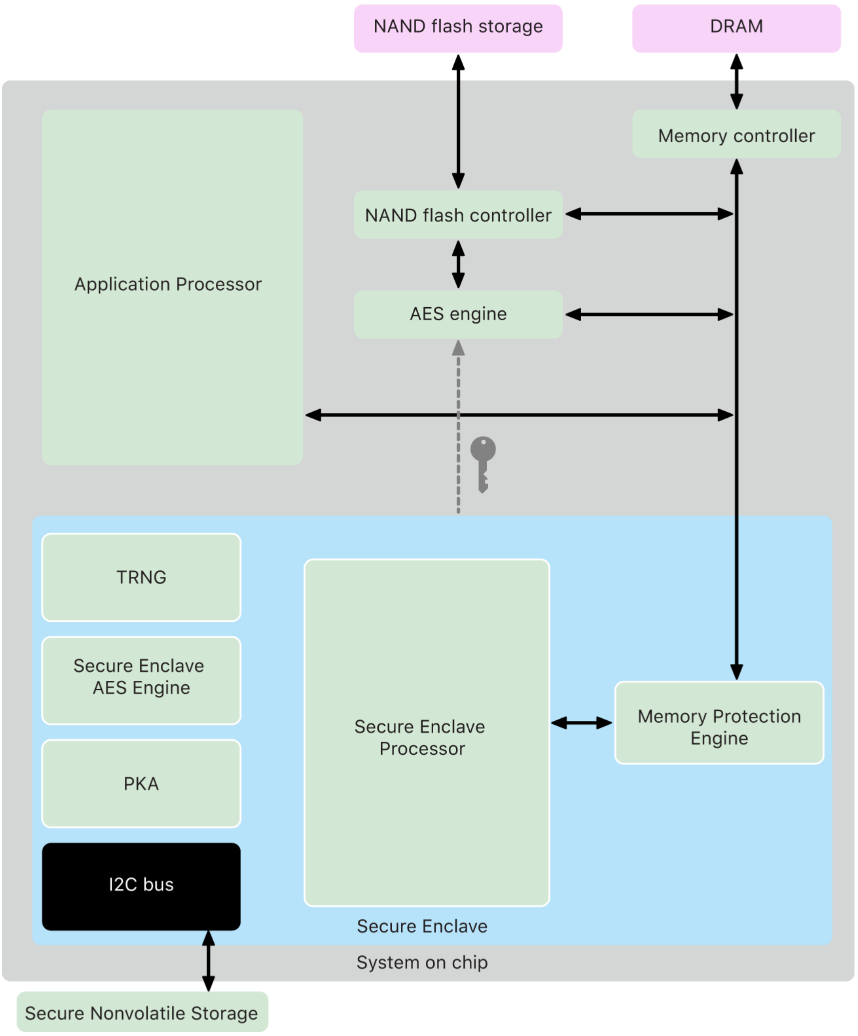
Secure Processor Design: Apple Solution

- Security foundation
  - Memory Protection Engine
    - The SE operates from a dedicated region of the device's DRAM memory
      - Whenever the Secure Enclave writes to its dedicated memory region, the Memory Protection Engine encrypts the block of memory using AES in Mac XEX mode, and calculates a Cipher-based Message Authentication Code (CMAC) authentication tag for the memory
      - The Memory Protection Engine stores the authentication tag alongside the encrypted memory
        - When the Secure Enclave reads the memory, the Memory Protection Engine verifies the authentication tag



Secure Processor Design: Apple Solution

- Security foundations
  - True Random Number Generator
    - The True Random Number Generator (TRNG) is used to generate secure random data
  - Root Cryptographic Keys
    - The Secure Enclave includes a unique ID (UID) root cryptographic key
      - The UID is unique to each individual device
      - A randomly generated UID is fused into the SoC at manufacturing time
    - The Secure Enclave also has a device group ID (GID), which is common to all devices that use a given SoC
  - Secure Enclave AES Engine
    - The Secure Enclave AES Engine is a hardware block used to perform symmetric cryptography based on the AES cipher
      - The AES Engine is designed to resist leaking information by using timing and Static Power Analysis (SPA)
      - The AES Engine supports hardware and software keys
    - Hardware keys are derived from the Secure Enclave UID or GID
  - Secure nonvolatile storage
    - The Secure Enclave is equipped with a dedicated secure nonvolatile storage device
    - The secure nonvolatile storage is connected to the Secure Enclave using a dedicated I2C bus, so that it can only be accessed by the Secure Enclave
    - The secure nonvolatile storage is used for all anti-replay services in the Secure Enclave.



Secure Processor Design: Apple Solution

- Pros
  - Restricted access and dedicated peripherals enable the isolation of SEP from possible attacks
  - Memory allocated by AP for SEP is encrypted, enforcing privilege rules upon external access requests
  - Secure mailbox to talk to the outside
- Cons
  - No validation of external memory blocks
  - SEP decrypted and secret key published [Mimoso, 2017]. Although no user info/data will leak because of the breach, it provides a way to explore the details of SEP

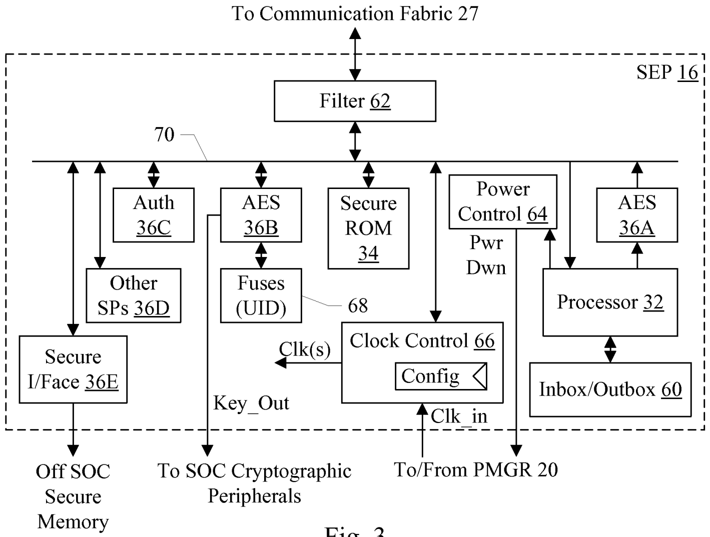


Fig. 3

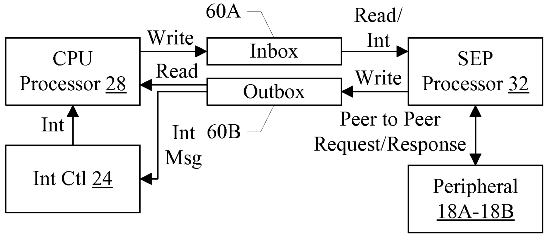
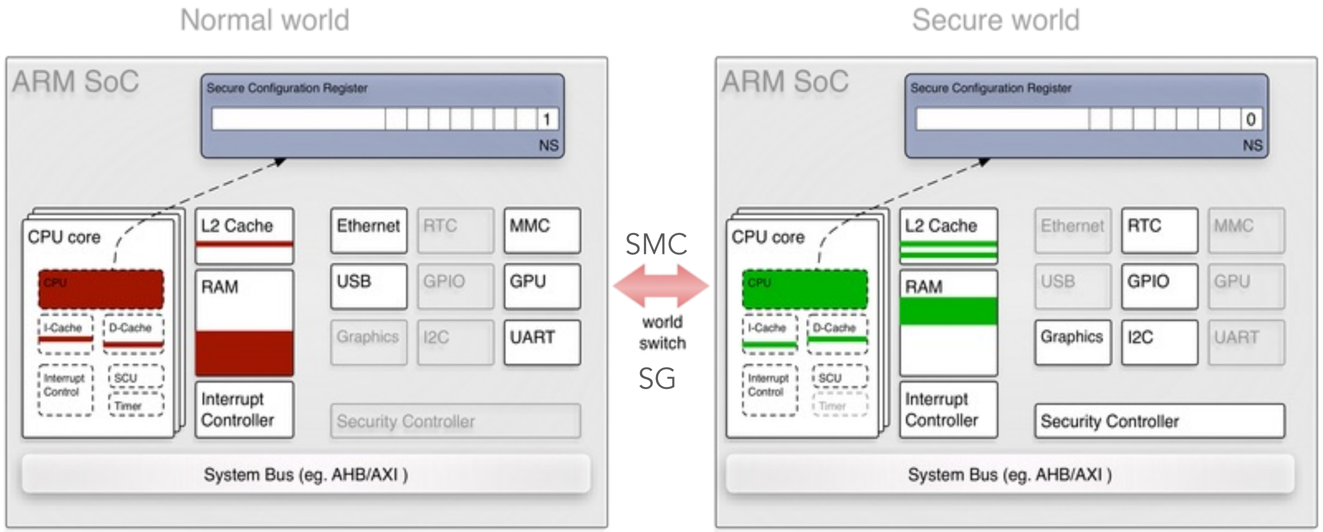


Fig. 4

Patent #: US8832465B2

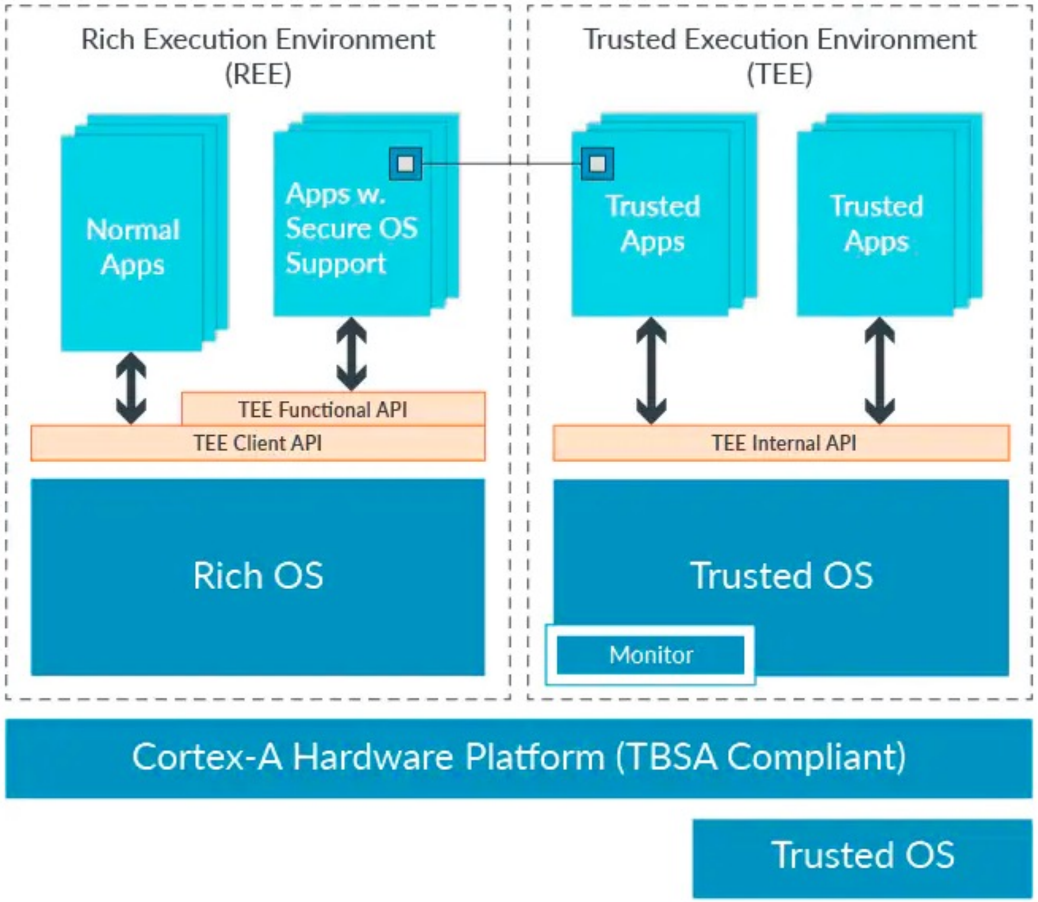
Secure Processor Design: ARM Trust Zone

- Two logic zones
  - Secure world with access to all data
  - Normal (non-secure) world with access to non-sensitive data
- Security Attribution Unit (SAU) and Implementation Defined Attribution Unit (IDAU)
  - Determine which memory region should belong to which world
- The switch of the two worlds are through a secure gateway (SG) with secure monitor calls (SMC)



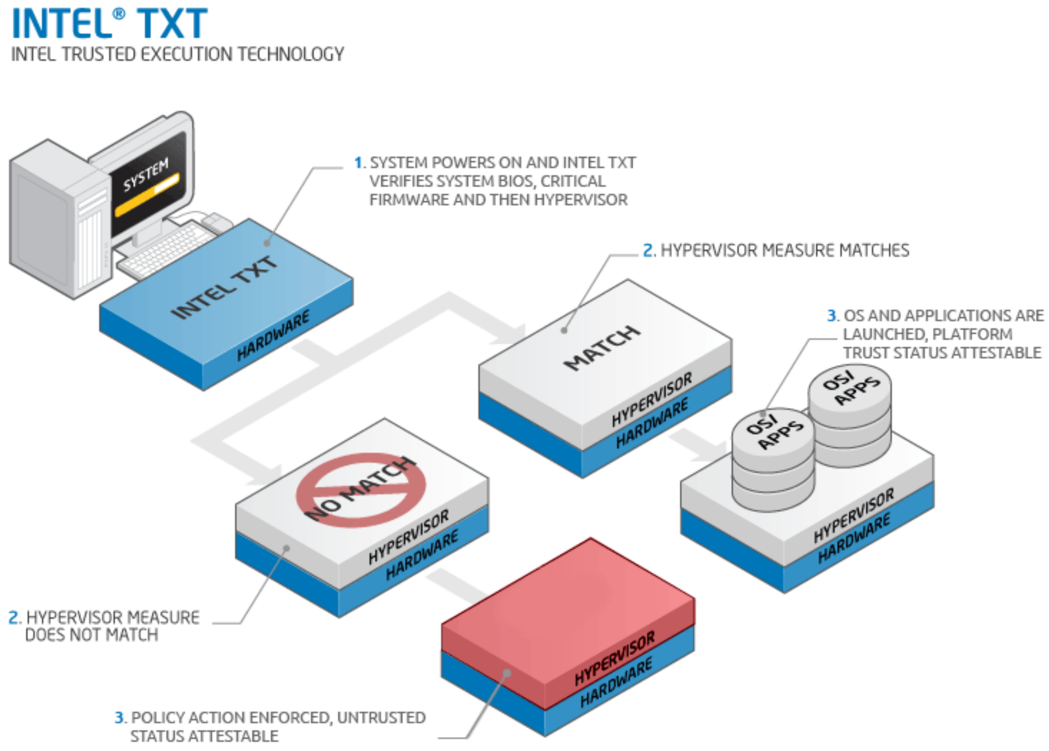
ARM Trust Zone Runtime Behavior

- ARM Cortex-A processor has 3 execution modes
  - User mode, kernel mode, and hypervisor mode
- ARM's TrustZone introduces a new mode - the Secure Monitor mode
  - In this new mode, the CPU can access all of the device's peripherals and memory
  - When not operating in this mode, the CPU can only access a subset of peripherals and specific ranges of physical memory



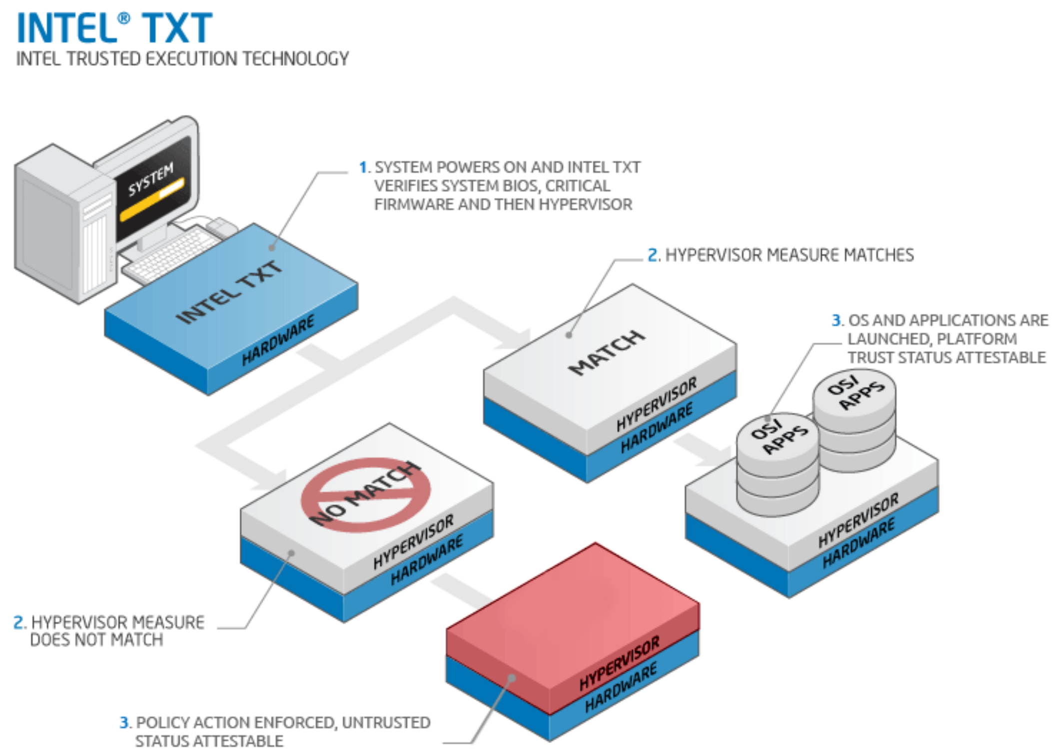
Secure Processor Design: Intel TXT

- Intel Trusted Execution Technology (TXT)
  - A hardware-based technology to examine the authenticity of the operating system and its running environment
- Security foundation
  - Trusted platform module (TPM) to provide secure storage
  - Static and dynamic chains of trust;
  - Hardware-based authenticated code module (ACM)



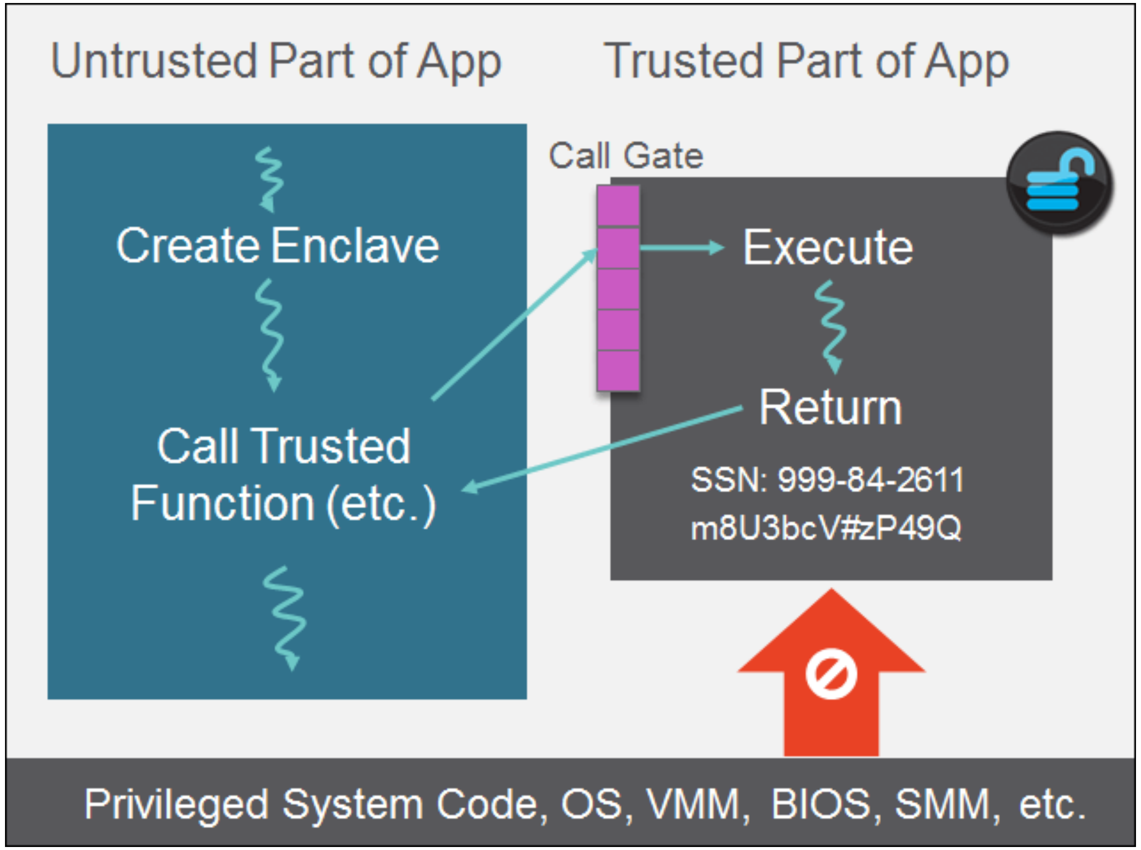
Secure Processor Design: Intel TXT

- Intel Trusted Execution Technology (TXT)
- Security foundation
  - Trusted platform module (TPM) to provide secure storage
  - Static and dynamic chains of trust;
  - Hardware-based authenticated code module (ACM)
- Known attacks
  - Butter overflow at runtime
  - System management mode (SMM) infection, which is the most privileged software loaded
  - Bootloader infection to execute the attacker's own code



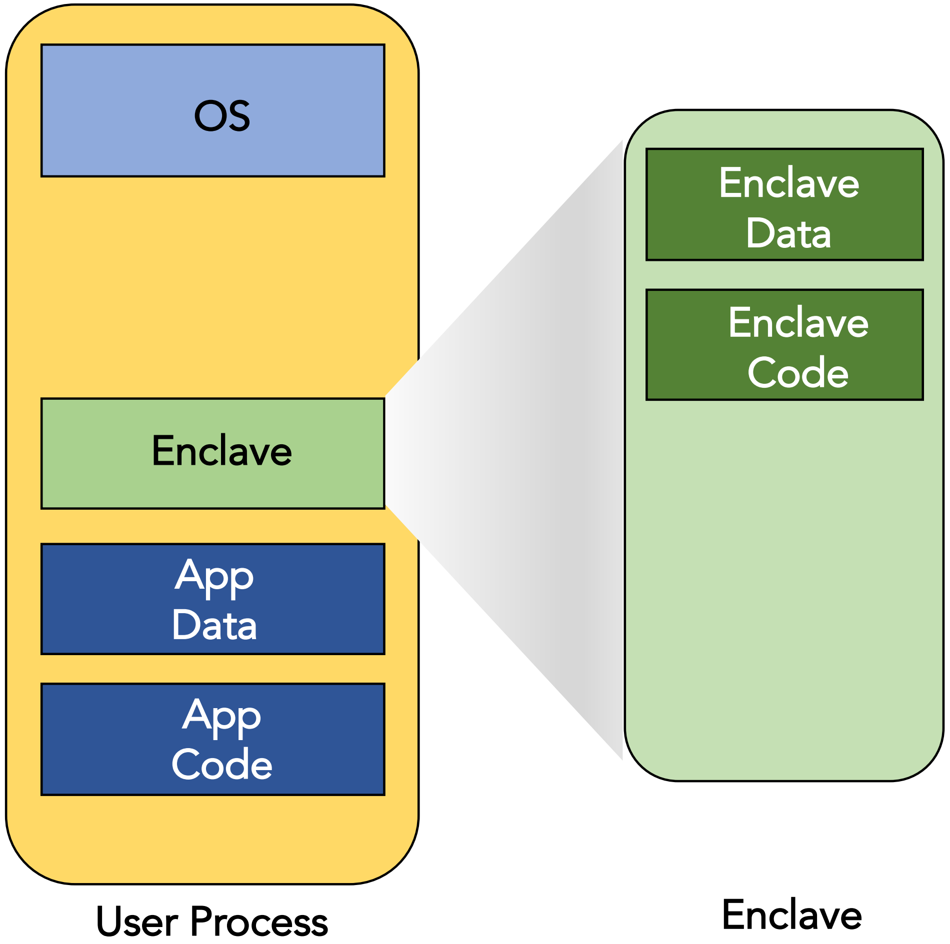
Secure Processor Design: Intel SGX

- Software guard extensions
- Allow definition of regions of memories called enclaves
  - Contents intended to be protected and unreadable by any process outside of the enclave including processes at higher privilege levels
- Even though OS is untrusted, it should still be able to manage page translation and page tables of the enclave



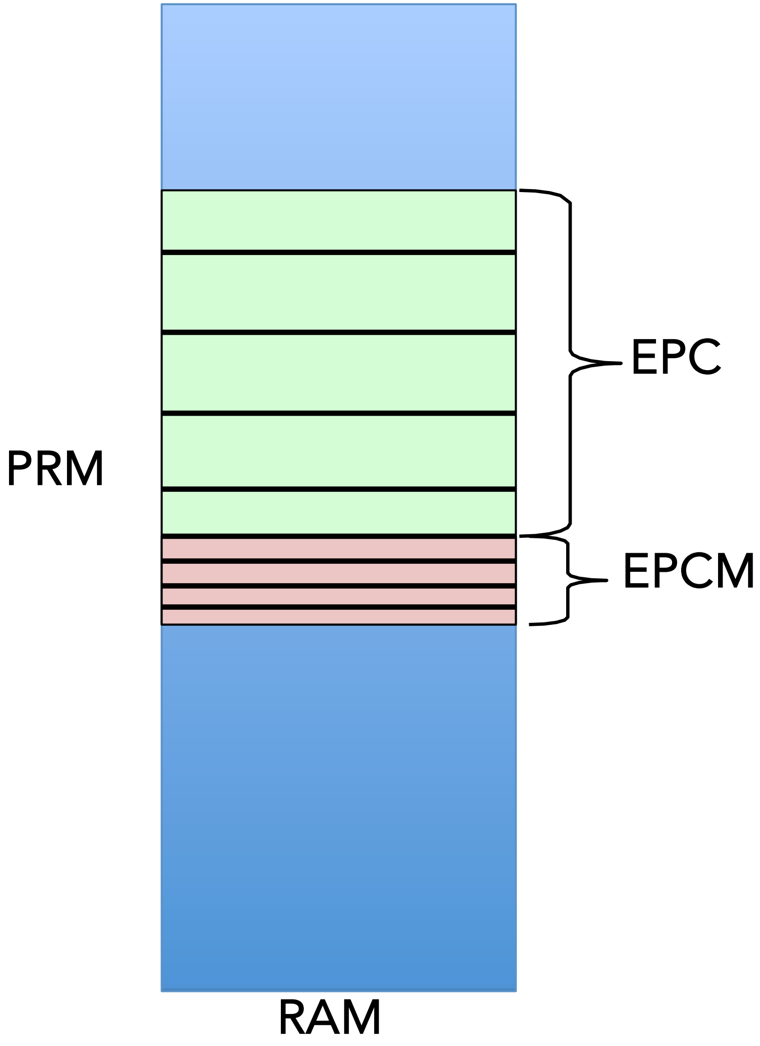
Secure Processor Design: Enclaves

- Enclave has its own code and data areas. Provides confidentiality and integrity with controlled entry points
- Enclave code and data cannot be accessed from outside the enclave, even by the OS
- TCS: Thread Control Structure
  - SGX supports multithreading; one TCS for each thread supported



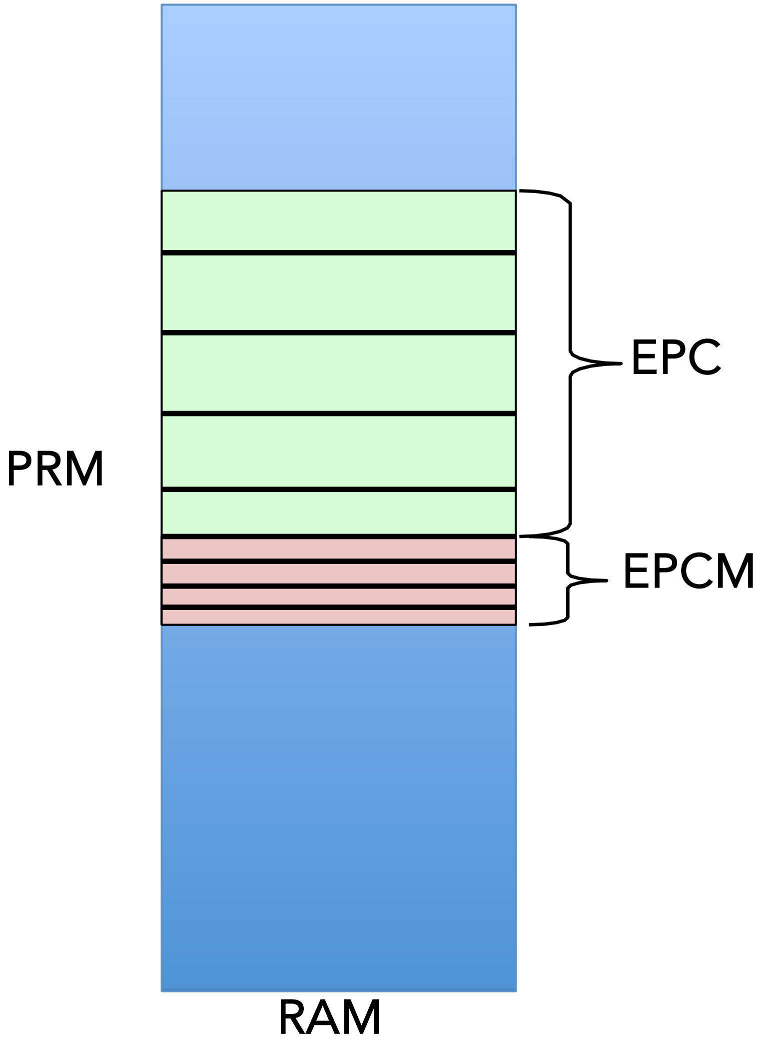
Physical Memory

- PRM – Processor Reserved Memory allocated by the BIOS. Access to PRM is blocked by external agents (DMA, graphics engine, etc.)
  - To other devices this range is treated as non-existent memory
  - All SGX enclaves mapped into the PRM
- EPC Pages: Enclave page cache holds enclaves from any application.
  - Divided into 4KB pages
  - If an EPC page is valid, it either contains an SGX enclave page or EPCM (EPC micro-architecture structure)



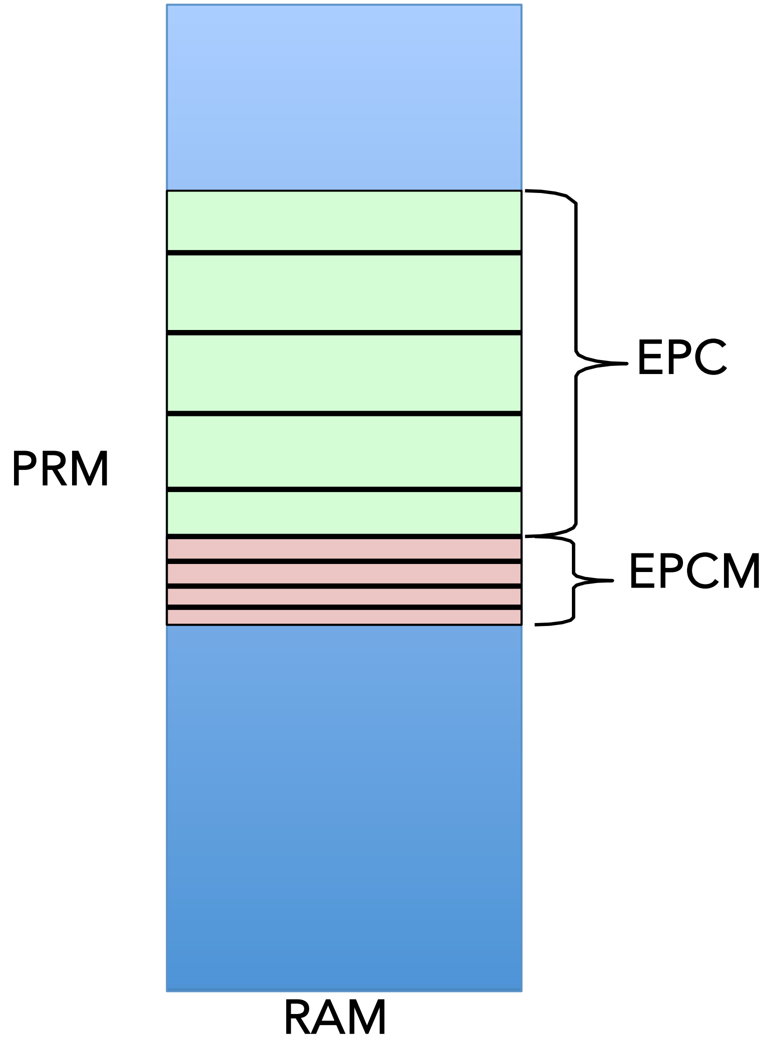
Physical Memory

- EPCM: Enclave page cache map
  - One for each EPC
  - Used by hardware for access control
  - It stores management related aspects for the corresponding EPC
    - Aspects such as valid/invalid; r/w/x permissions
    - Type of page
    - Virtual address range through which EPC can be accessed
    - It is an additional layer of security compared to legacy paging and segmentation since we do not trust the OS



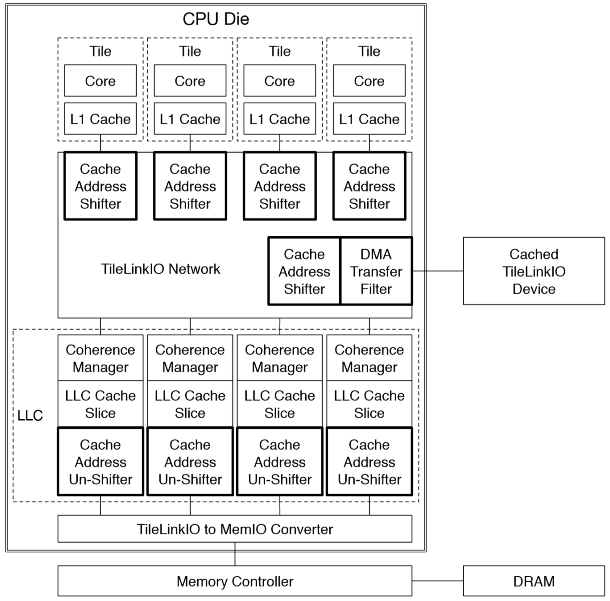
Physical memory

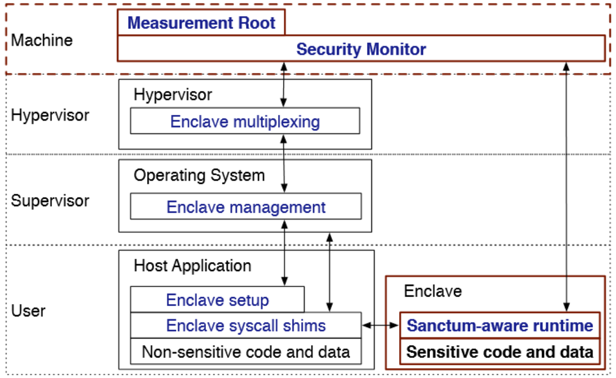
- SECS: SGX Enclave Control Store
  - One for each enclave
  - 4KB (present in an EPC)
  - Contains global metadata about the enclave
    - EPC pages that are used
      - Mapping information
      - Crypto log of each used EPC page
    - Range of protected addresses used by the enclave
    - 32/64 bit operating mode
    - Debug access



Sanctum

- Based on the analysis of SGX, offers additional protection against memory access pattern side-channeling
- HW/SW Co-design implementation; minimal and minimally invasive hardware modifications with a trusted software security monitor
- Hardware - Cache Address Shifter, shift PPN right by certain bits for obfuscation
- Software - Security Monitor, replacing SGX microcode, high privilege level; controls page walker FSM





Sanctum Memory

- Hardware extension for dual page table lookup
  - Ensure enclave page table only map to enclave memory and OS page tables only map to non-enclave memory
- Per enclave metadata used by SM Stored in DRAM regions managed by the OS
  - Page map similar to EPCM in SGX to verify actions of the OS

Keystone

- Open-source framework for customized TEEs
- Can be implemented on unmodified RISC-V hardware
  - No changes to cores, memory controllers
- Required hardware platform features
  - Trusted boot process
  - Device specific secret key (visible only to the trusted boot process)
  - Hardware source of randomness
- Support multiple enclaves
- Allow multiple stakeholders to customize a TEE

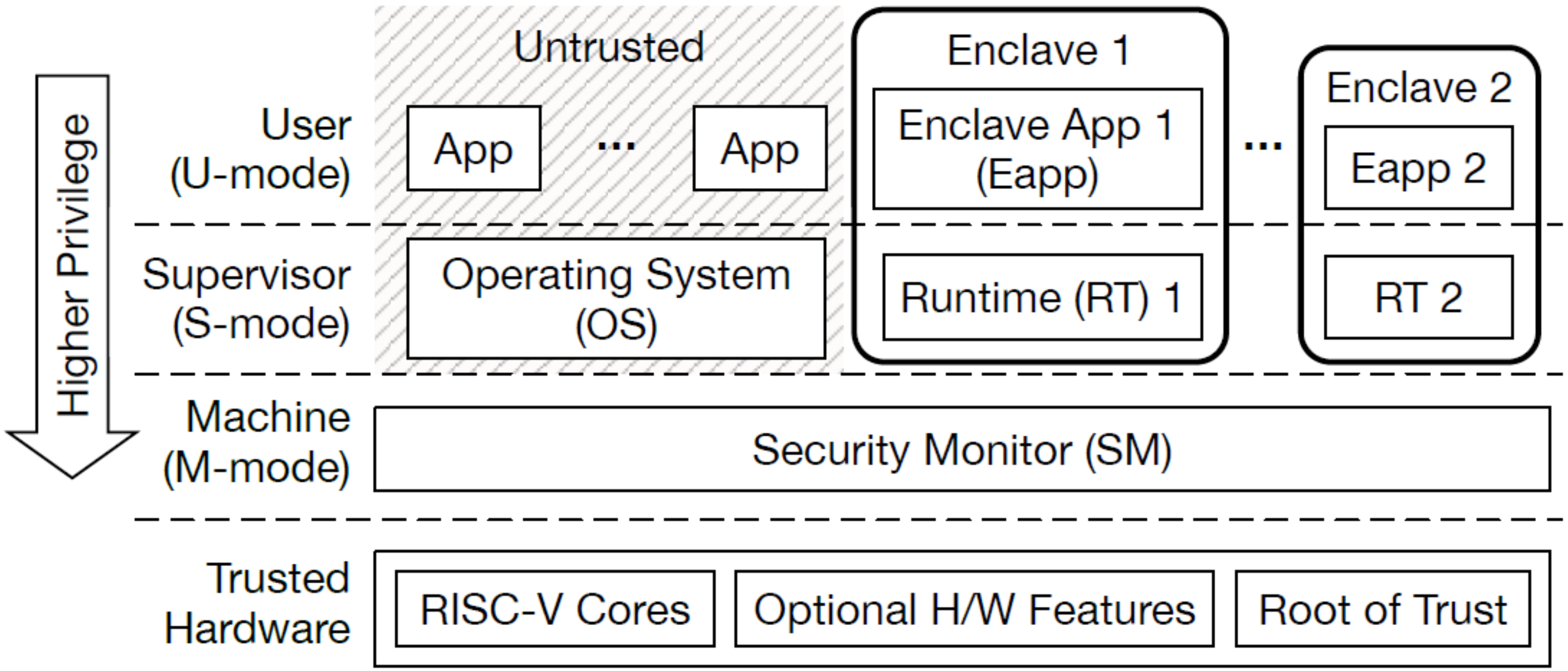
Keystone: Security Monitor (SM)

- Executed in machine mode
- Physical Memory Protection (PMP) allows enforcing access policies to physical memory
- Use hardware primitives to provide TEE guarantees
  - Secure boot
  - Memory isolation
  - Attestation
- No resource management

Keystone: Enclaves

- Two components
  - User mode: Enclave application (eapp)
  - Supervisor mode: Runtime (RT)
- Own isolated physical memory region
  - RT manages virtual memory for the enclave
- Enclave measurement after creation
  - SM performs measurement and attestation
- Page tables always inside enclave memory
- Dynamic resizing
  - Extended SBI call to OS
  - If OS succeeds, SM increases enclave size

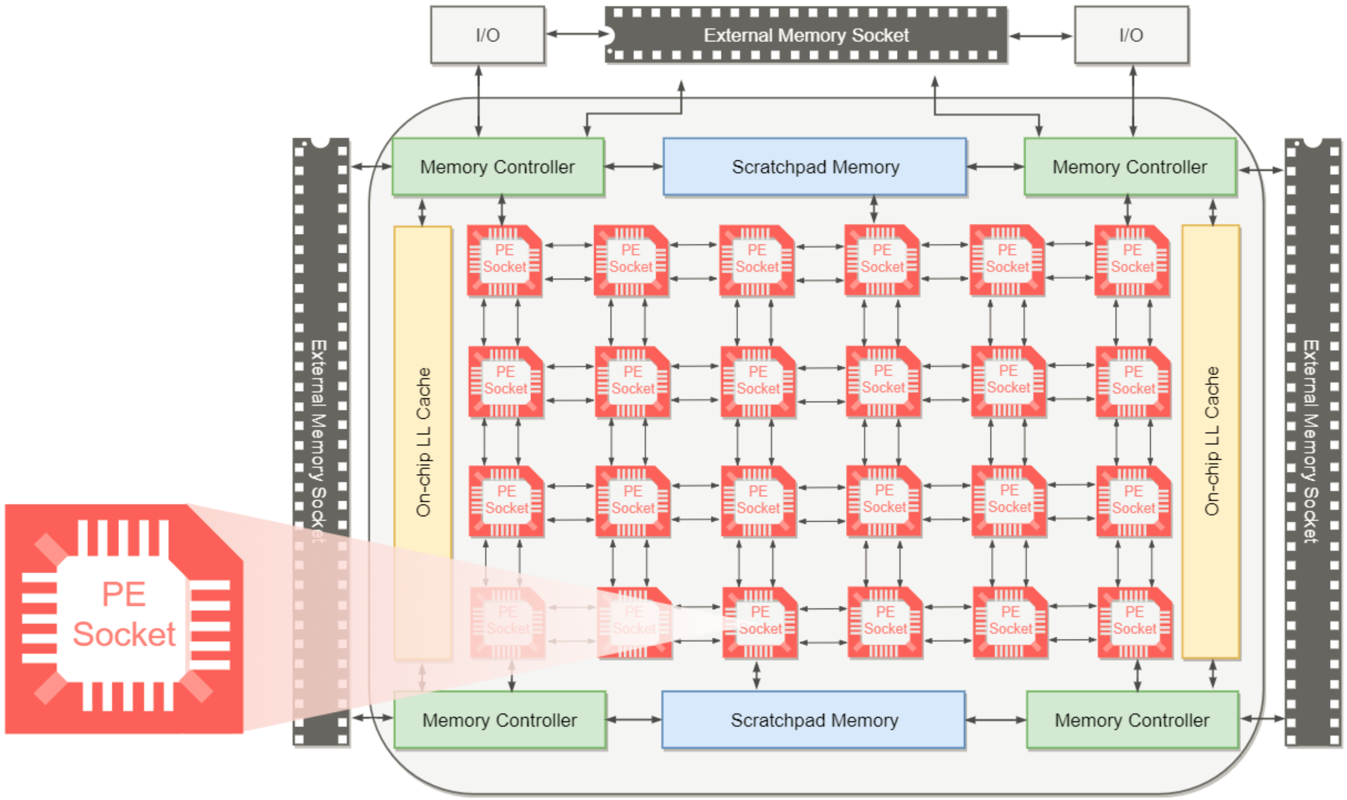
Keystone Components



D. Lee, D. Kohlbrenner, S. Shinde, D. Song, and K. Asanović, “Keystone: An Open Framework for Architecting TEEs,” 2019.

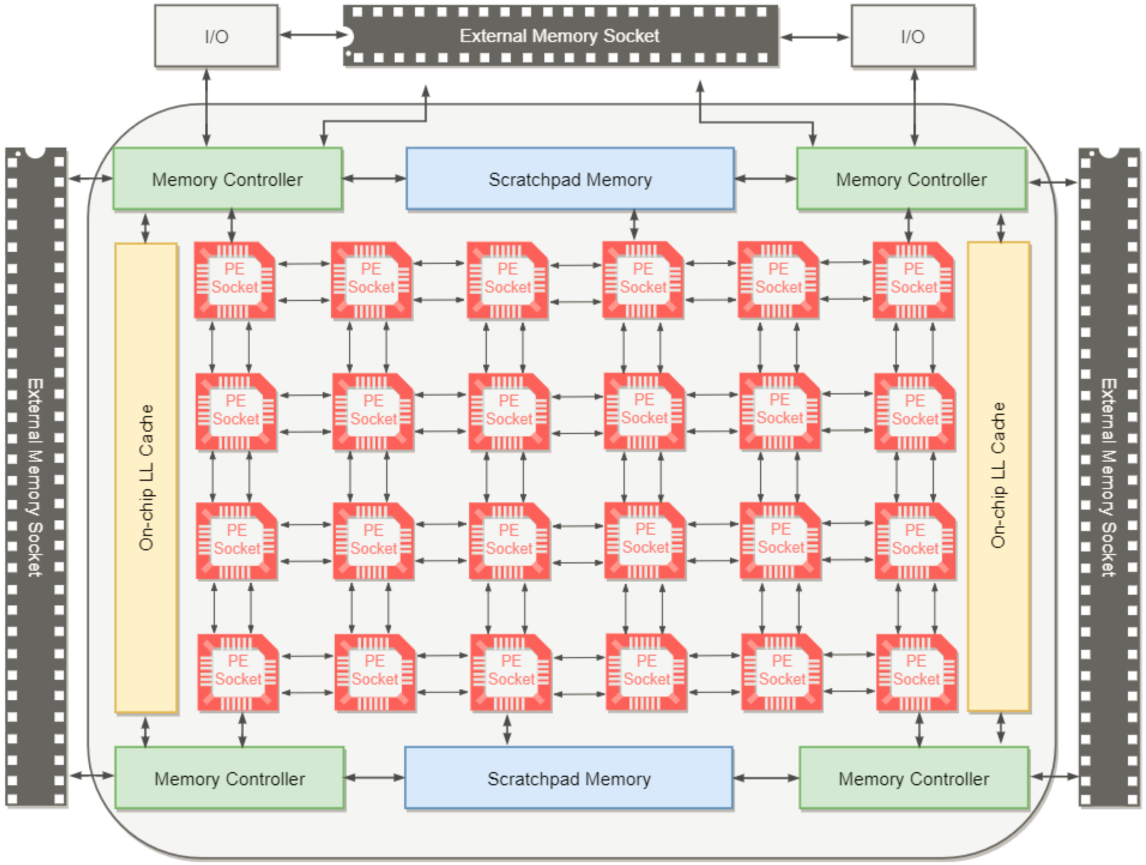
The Hermes Architecture

- An integration template to enable a secure SoC built from untrusted processing elements (PE)
  - With user-defined security policy



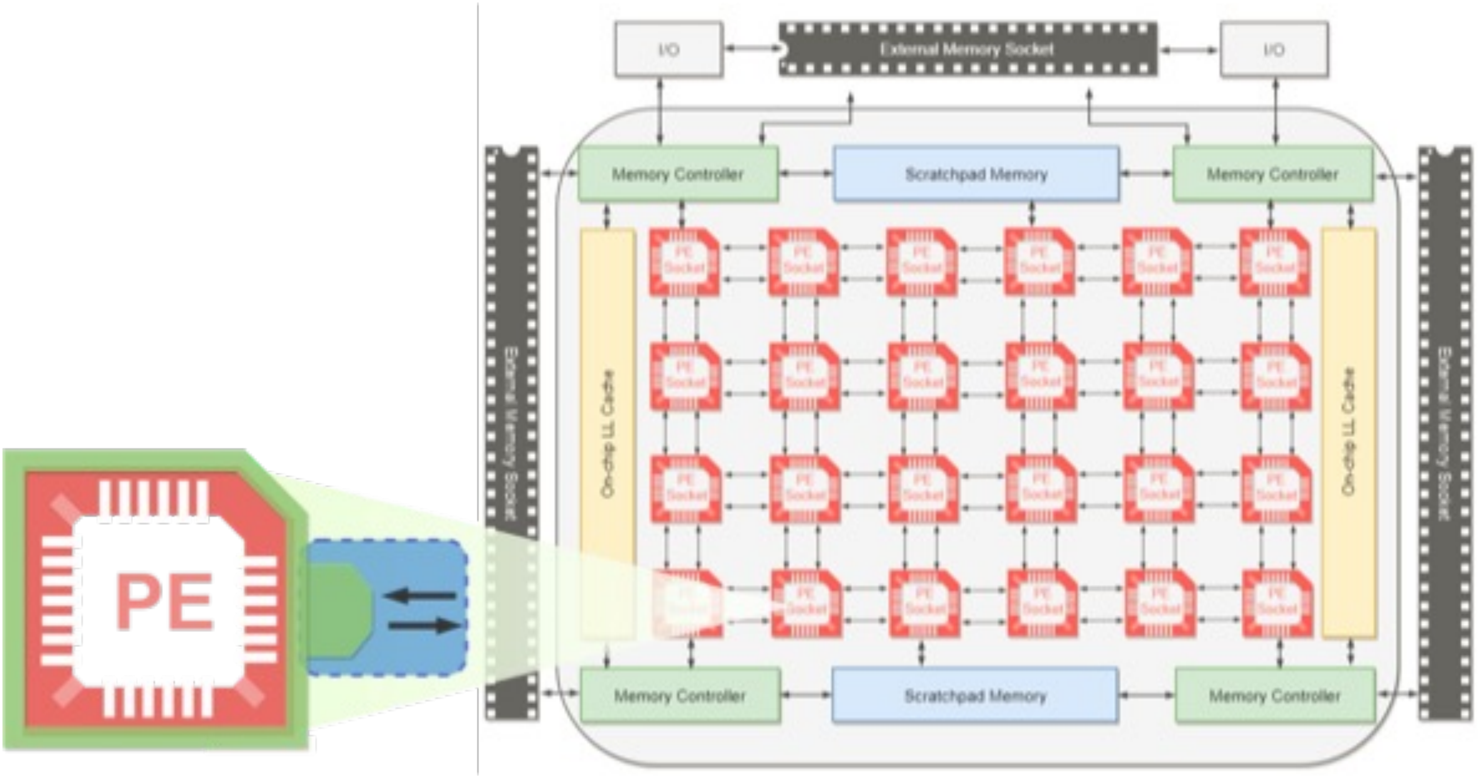
Hermes Architecture Features

- A template to integrate processing elements with different security levels
- A process isolation design to create virtual logic zones according to PEs' runtime HW & SW security
- Hardware root-of-trust and a set of formally verified secure protocols and to resist malicious behaviors of PEs
- A set of quantum-proof hardware cryptographic primitives to guarantee the model's post-quantum security



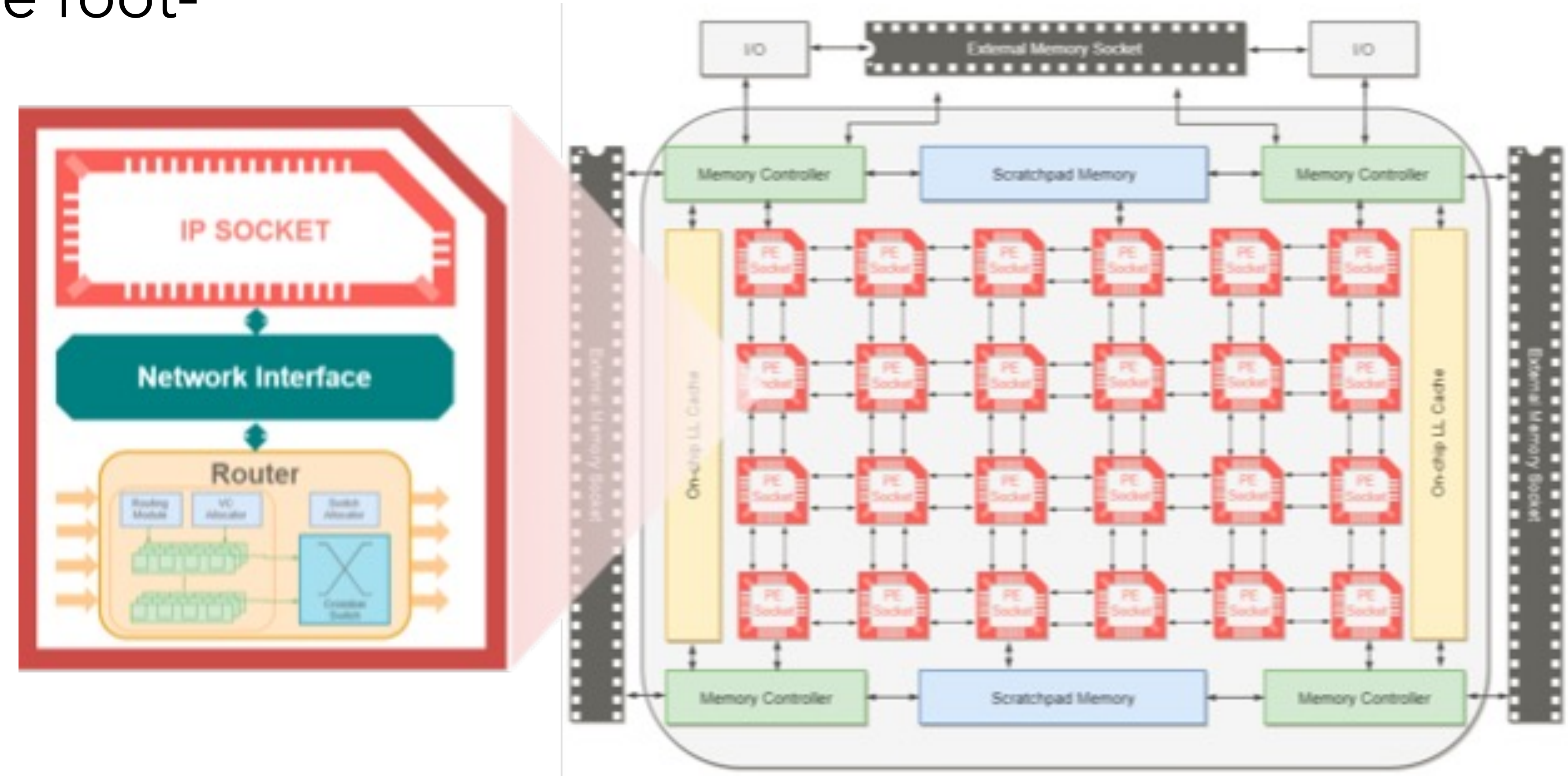
Hermes Design Principles

- Integrating processing elements with different security levels
  - In this design, no restrictions are made on the type, trust level or provenance of the cores
  - A user-programmable security wrapper built around the processing elements
  - Although we cannot control what a PE does, its interactions with the rest of the system is fully specified and verified!



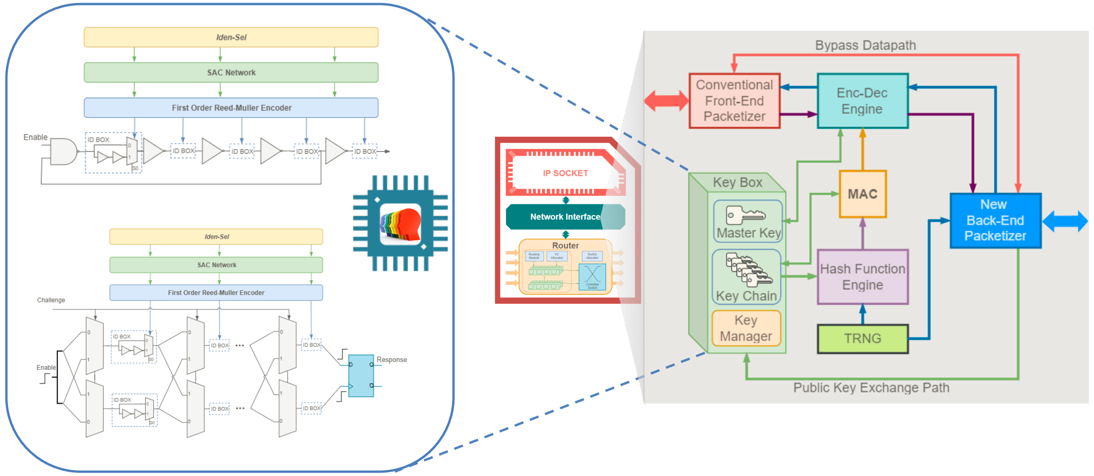
Hermes Design Principles

- Interface-based hardware as the root-of-trust design

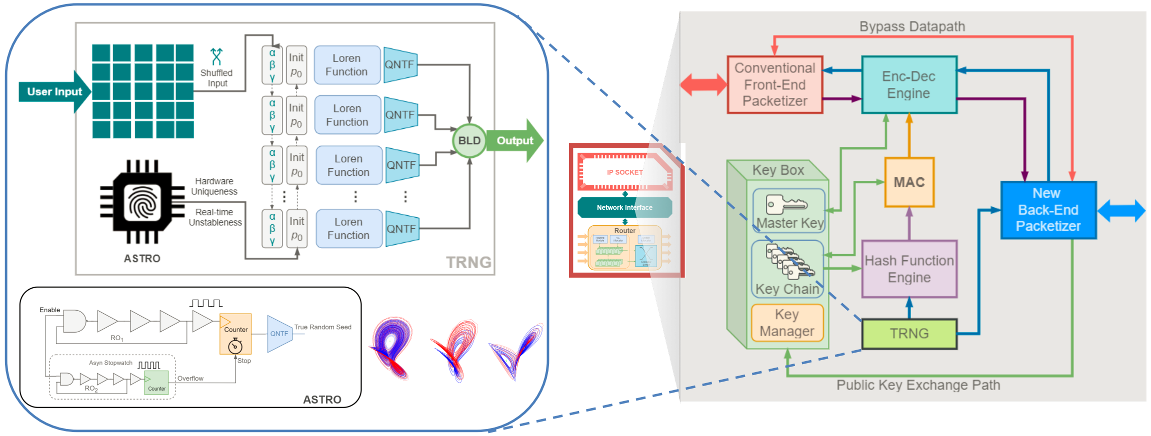


Hermes Hardware as Root-of-Trust Design

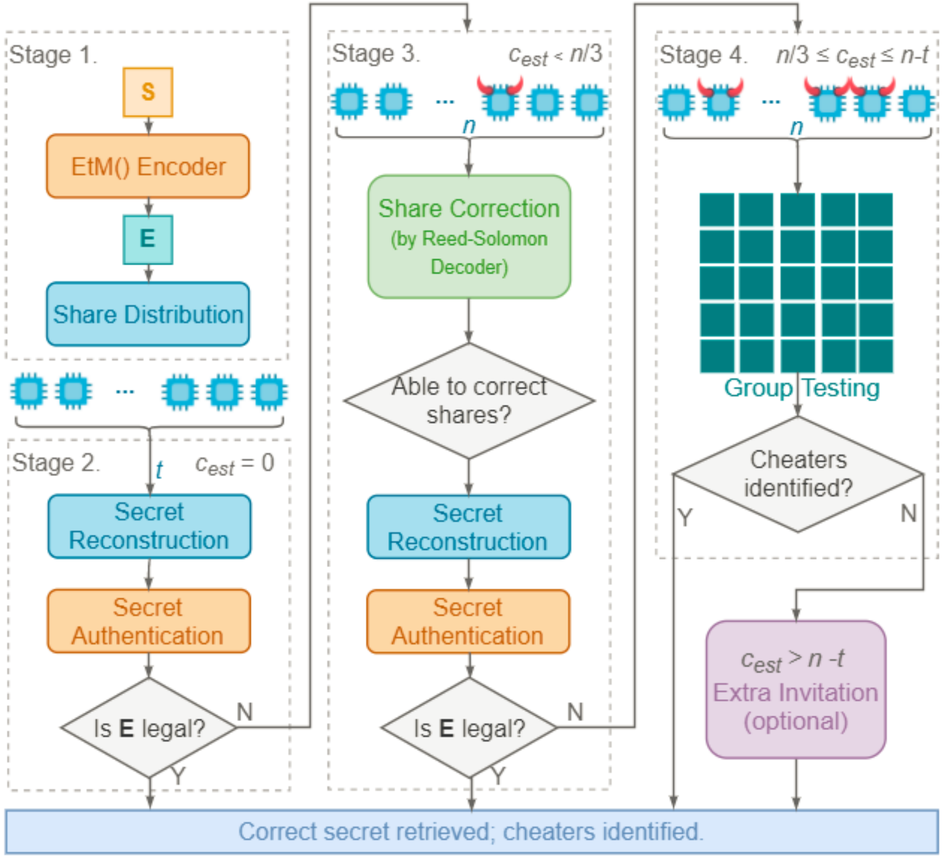
- Multi-Identity Physical Unclonable Functions (Mi-PUF)



- Programmable TRNG using Lorenz Chaotic Systems

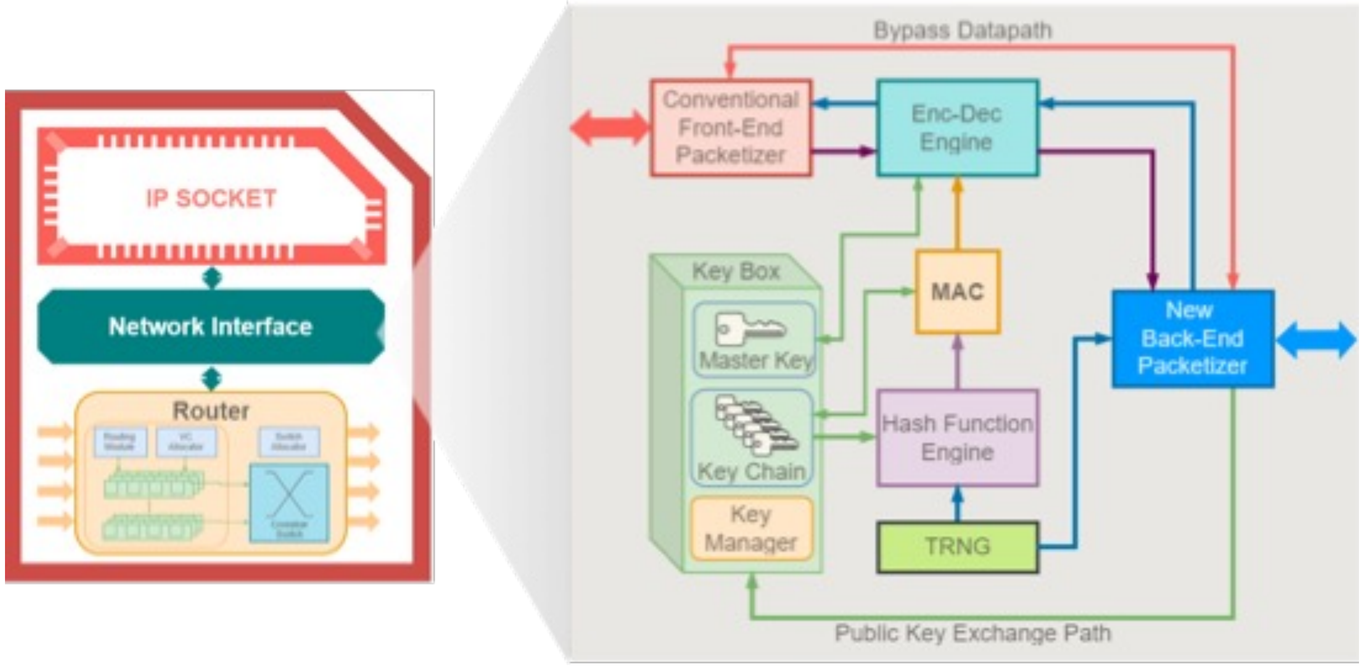


- Threshold-based authorization of services



Hermes Hardware as Root-of-Trust Design

- Support for multi-level user-defined security protocols
  - Front-end and back-end packetization
  - Processing and verifying incoming and outgoing requests
  - Generation of new session keys upon island membership change
  - Public-key & symmetric encryptions of packets
  - Access privilege identification



Secure Computation Approaches

Multi-Party Computation (MPC)

Pros
- Low compute requirements
- Easy to accelerate
- Provably secure
- Supports multiple threat models
- Easy to map existing algorithms

Cons
- High communication costs
- High latency
- Information theoretic proofs are weaker than PKE ones

Fully Homomorphic Encryption (FHE)

Pros
- Very low communication costs
- Requires a single round of communications, i.e., "fire and forget"
- Useful when one side is limited in compute / memory / storage
- Provably secure – relies on strength of PKE

Cons
- Very high computational requirements
- Harder to accelerate
- Mapping existing algorithms to FHE may be difficult

Trusted Execution Environments (TEE)

Pros
- No communication required
- Trivial to accelerate
- Great support for existing software

Cons
- Weaker security guarantees
- Cannot stop determined adversaries
- Historically plagued by vulnerabilities and breaches
- Long term deployment is difficult – TEE's can 'run out' of entropy / CRP's, etc.

Upcoming Lectures

- Secure Computation Approaches
  - Homomorphic Encryption