Segurança baseada em Hardware

Segurança Computacional

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

2026

Conteúdo

Mecanismos básicos

Mecanismos básicos

Mecanismos de segurança historicamente providos pelas CPUs:

- Níveis de privilégio:
  - separação entre modo núcleo e modo usuário
  - controlados por flags do processador
  - protege o núcleo do SO e os recursos
- Memória virtual:
  - cada aplicação tem suas áreas de memória RAM
  - controle pela Unidade de Gerência de Memória (MMU)
  - Acessos fora das áreas geram interrupções.

Mecanismos básicos

- Controle de acesso à memória:
  - o núcleo define permissões de acesso à memória
  - a MMU verifica essas permissões a cada acesso
  - Exemplo:
    - área `TEXT`: read + execute
    - área `STACK`: read + write
  - Política $W \wedge X$: escreve ou executa, nunca ambos

Mecanismos básicos

Outras proteções presentes em processadores modernos:

- IOMMU:
  - I/O Memory Management Unit
  - Define áreas de memória virtual para DMA
  - Áreas separadas para cada dispositivo
  - Impede a ação de dispositivos maliciosos
- Virtualização por hardware:
  - Isola ambientes virtuais por hardware
  - Cada ambiente pode executar um SO próprio
  - Os SOs não têm acesso ao hardware
  - Um hipervisor gerencia os SOs e aplicações
  - Implementações: Intel VT-x, AMD-V, ARM-VHE

Hardware criptográfico

Hardware criptográfico

Criptografia envolve operações custosas

Usar um hardware apropriado provê ganho de desempenho

Abordagens:

- Extensão criptográfica
- Cripto-coprocessador (acelerador criptográfico)
- Cripto-processador
- Cripto-array

Extensão criptográfica

Processadores mais recentes têm suporte à criptografia

Exemplos:

- Extensões Intel AES-NI e VAES
- Instruções SHA nas CPUs ARM, RISC-V e Intel
- Instrução RDRAND (*true random numbers*)

Maior desempenho ≠ maior segurança:

- Chaves e senhas ficam na memória RAM
- Podem sofrer ataques por software

Cripto-coprocessador

Coprocessador para operações criptográficas:

- Também chamado *Acelerador criptográfico*
- *Hardware* customizado
- Eficiente para funções criptográficas
- Não armazena chaves nem dados seguros
- Controlado pelo processador principal
- Indicado para alto desempenho em criptografia
- Exemplo: chip Samsung S3SSE2A (cripto pós-quântica)

Cripto array

Acelerador criptográfico vetorial

- Similar ao cripto-coprocessador
- Usa núcleos de processamento vetorial (GPUs)
- Visa alto desempenho

Cripto-processador

Processador autônomo com funções criptográficas

- Módulo de hardware (chip ou placa)
- É autônomo e programável, não depende da CPU
- Tem memória própria isolada
- Conjunto de instruções de criptografia
- As chaves são armazenadas no próprio chip
- Exemplos:
  - TPM - *Trusted Platform Module*
  - HSM - *Hardware Security Module*

Processadores confiáveis

Plataforma confiável

Para estabelecer um ambiente computacional seguro:

- a plataforma deve provar quem ela é (autenticidade)
- a plataforma não deve ter sido adulterada (integridade)

Conceitos necessários para uma plataforma confiável:

- Mensuração
- Atestação
- Raiz de confiança
- Cadeia de confiança

Um processador confiável suporta esses conceitos.

Mensuração

Mensuração (*Measurement*): relatório de segurança

- Descreve um componente de hardware ou software
- Identificação, fabricante, versão, configurações
- Inclui resumo (*hash*) de seu conteúdo
- Permite a verificação posterior do componente
- Assinada digitalmente pelo autor do mesmo

Atestação

Atestação: verificar um componente

- Permite verificar um componente
  - Autenticidade: autoria (fabricante)
  - Integridade: não foi adulterado
- Usa mensurações previamente armazenadas
- Pode ser local ou remota
  - Local: um componente atesta outro
  - Remota: um componente é atestado remotamente
- Pode exigir uma entidade confiável externa

Raiz de confiança

Raiz de confiança (RoT - *Root of Trust*)

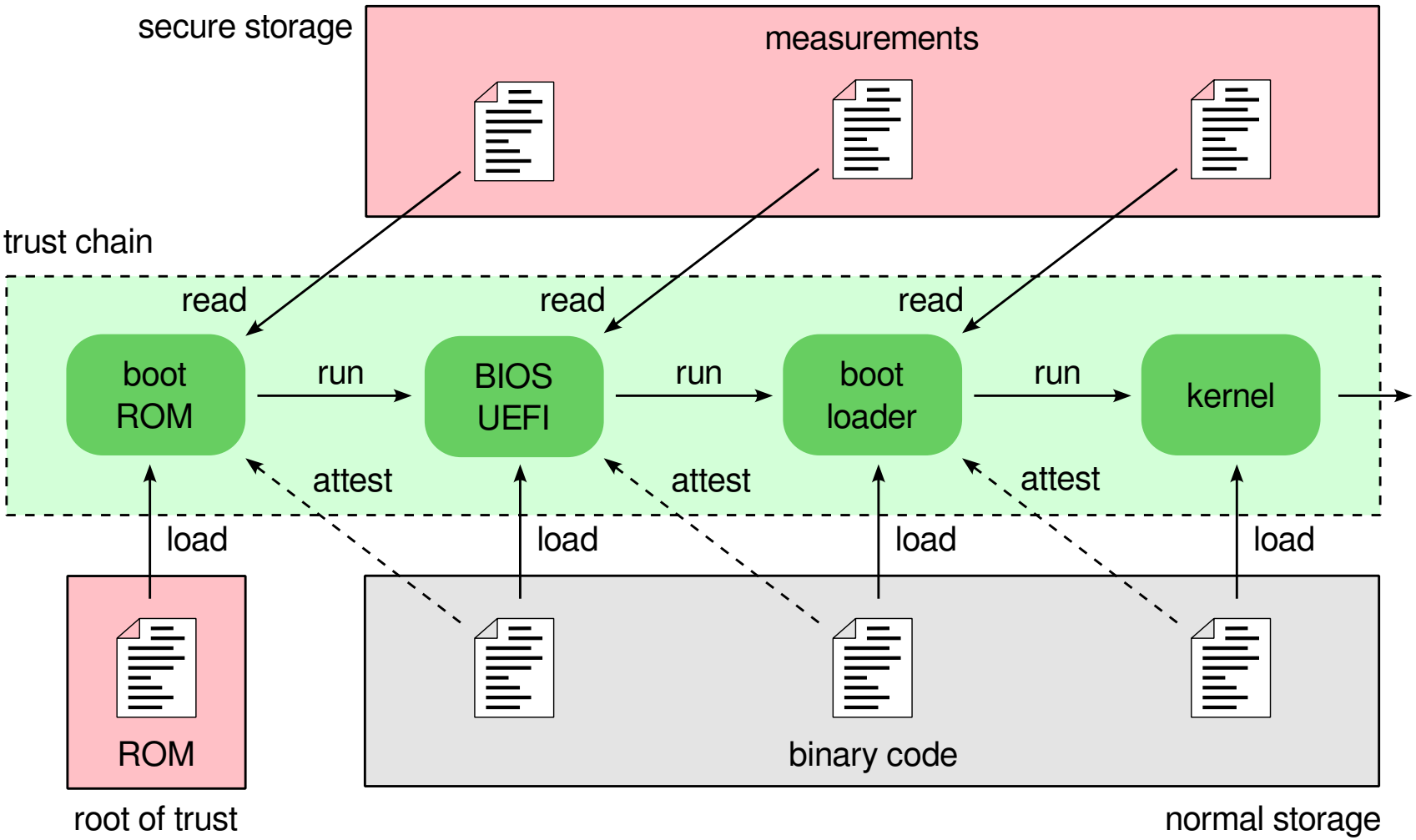
- Estabelece a confiança inicial da plataforma
- Assegura a confiança das mensurações
- Hardware imutável provido pelo fabricante
- Pode ser acessado mas não alterado

Cadeia de confiança

Cadeia de confiança (*Chain of Trust*)

- Um componente pode atestar outro, sucessivamente
- Usa mensurações prévias armazenadas e assinadas
- Construir uma "cadeia de confiança" transitiva
- Similar a uma cadeia de certificados na Web
- Usada na inicialização segura (*Secure boot*)

Cadeia de confiança



TPM - Trusted Platform Module

Chip único contendo:

- Processador
- Memória volátil (RAM) e não-volátil (ROM e *flash*)
- Interface de barramento SPI ou LPC

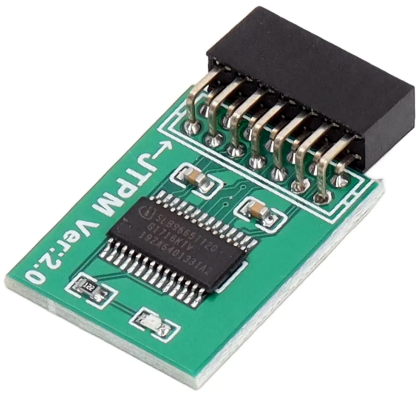
Finalidade:

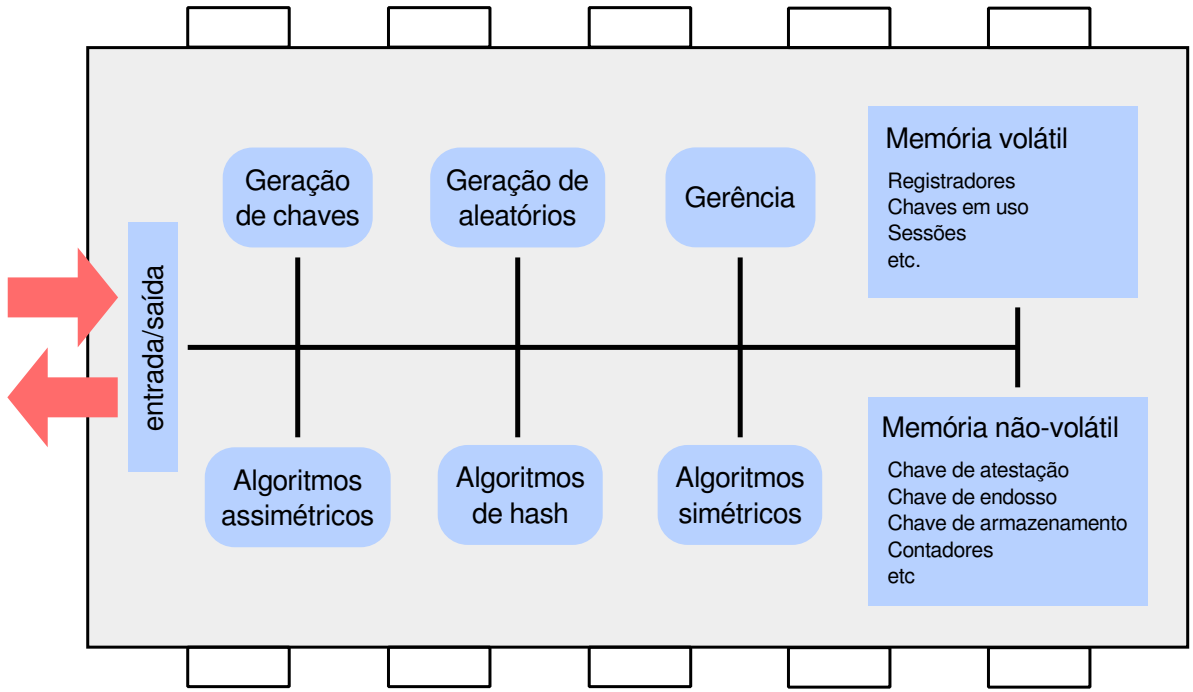
- Armazenar chaves de criptografia
- Realizar operações criptográficas

Usos:

- Servidores, computadores pessoais, smartphones
- Dispositivos de rede e centrais automotivas

TPM - Arquitetura v2.0





TPM - Funcionalidades

O TPM implementa:

- Algoritmos de criptografia simétricos e assimétricos
- Resumos criptográficos
- Assinaturas digitais
- Gerador de números aleatórios
- Geradores de chaves criptográficas
- Armazenamento seguro de chaves

TPM - Chaves

Chaves mantidas pelo TPM:

- *Chaves de Endosso* (EK - *Endorsement Keys*)
  - Par de chaves assimétricas
  - Gravadas pelo fabricante (única)
  - A chave privada **nunca** sai do chip
  - Certificado assinado pelo fabricante
  - Usadas para derivar outras chaves
- *Chave Raiz de Armazenamento* (SRK - *Storage Root Key*)
  - serve para "selar" conteúdos externos ao TPM
- Chaves gerada pelo TPM para as aplicações

TPM - *Secure Boot*

Inicialização segura (*secure boot*):

- Validar a integridade do software de sistema
  - BIOS
  - Bootloader
  - Núcleo do SO
- TPM é a raiz da cadeia de confiança
- A mensuração de cada componente é previamente armazenada no TPM
- Cada componente atesta o próximo componente
- Se atestação falhar, o *boot* é interrompido
- Garante que os componentes críticos estão íntegros

TPM - Tipos

- Discreto: chip separado
  - Totalmente isolado (mais seguro)
  - Exemplo: chip Infineon Optiga SLB 9670
- Integrado: hardware dedicado dentro do processador
  - Exemplo: Intel *Platform Trust Technology* (PTT)
- Em firmware: implementado pelo *firmware* (BIOS)
- Virtual: implementado pelo hipervisor (VMWare, KVM)
  - Usado na nuvem (vTPM)
  - Pode usar o TPM da plataforma física
- Emulado: em software
  - Inseguro!
  - Útil para desenvolvimento

AMD PSP - *Platform Security Processor*

Presente nos processadores AMD

- Processador criptográfico integrado ao chip
- Usa um microcontrolador ARM 32 bits com *TrustZone*

Provê funcionalidades de TPM:

- Algoritmos criptográficos (AES, RSA, ...)
- Gerador de números aleatórios
- Armazenamento de chaves
- Chave única da plataforma (raiz de confiança)
- Suporte à inicialização segura

Intel TXT - *Trusted Execution Technology*

Solução de plataforma confiável da Intel

- Implementa um TPM em *firmware*
- Raiz de confiança para inicialização segura
- Mensuração e atestação local dos componentes
- Depende fortemente do *Intel Management Engine*

Intel ME - *Management Engine*:

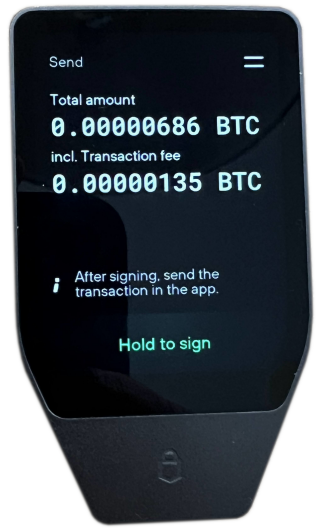
- Processador integrado à CPU principal
- Executa seu próprio SO (Minix 3)
- Funcionalidades de gerência da CPU
- Invisível ao SO e aplicações

HSM - *Hardware Security Module*

Dispositivo seguro separado:

- Armazena e manipula informações críticas
- Realiza operações criptográficas
- Pode ter acelerador criptográfico
- Resistente a violação física (arrombamento)
- Resistente a ataques eletromagnéticos
- Usados em sistemas bancários, militares e CA
- *Cripto wallets* são um exemplo simples

Hardware Security Module



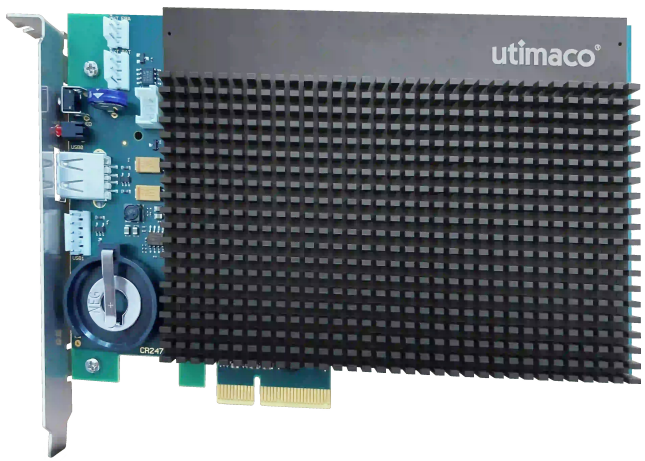


Figura: HSM autônomo, placa HSM e carteira de criptomoedas

Ambientes de Execução Confiável

Ambientes de Execução Confiável

TEE - *Trusted Execution Environment*:

- Espaço de RAM isolado dentro do sistema
- Abriga código e/ou dados críticos
- A área isolada é denominada *Enclave*
- Alguns TEEs suportam múltiplos enclaves

Algumas implementações:

- ARM *TrustZone* e *Confidential Compute Architecture*
- Intel SGX e TDX
- *Apple Secure Enclave*
- *AMD Secure Encrypted Virtualization*
- *IBM Secure Execution* e Protected Execution Facility

Ambientes de Execução Confiável

Objetivos:

- Prover confidencialidade, integridade e autenticidade
- Bloquear acessos vindos de fora, mesmo do núcleo
- Proteger a aplicação de uma plataforma maliciosa

Como usar o enclave:

- Separar as partes de uma aplicação
  - Parte confiável/crítica dentro do enclave
  - Parte normal fora do enclave
- Ambas comunicam por mecanismos do hardware

ARM TrustZone

Características:

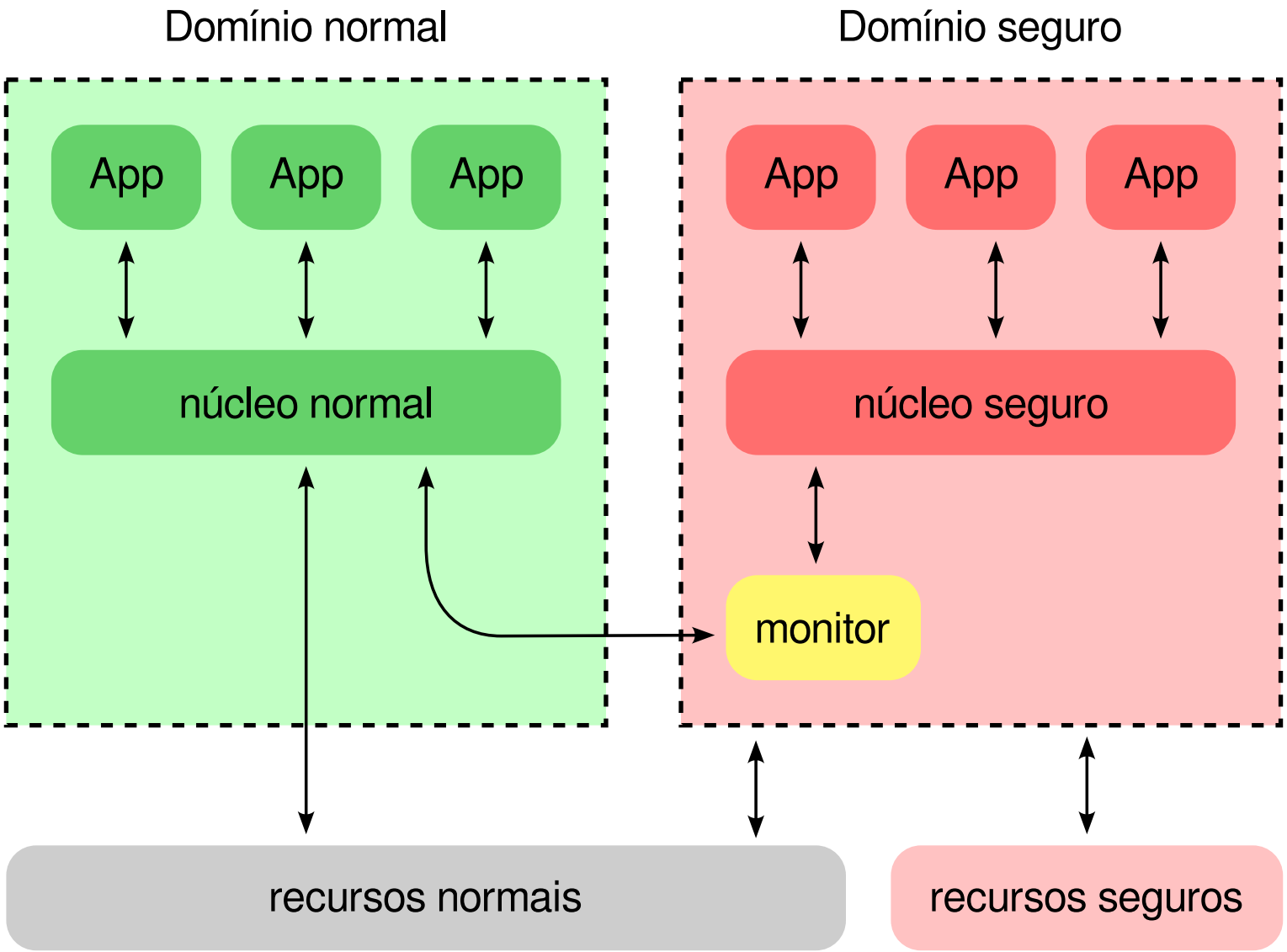
- Processadores ARM Cortex-A, Cortex-M23 e Cortex-M33
- Muito usado em *Smartphones* Android e Apple
- Define dois domínios de execução:
  - *Normal World* : domínio normal
  - *Secure World* : domínio seguro (enclave)
- Cada domínio tem seus próprios recursos:
  - Áreas de memória
  - Dispositivos de entrada/saída
  - Registradores

ARM TrustZone

Funcionamento:

- *TrustZone Address Space Controller* define as áreas normais e seguras da RAM
- A memória do domínio seguro não é criptografada
- *TrustZone Protection Controller* regula o acesso aos dispositivos periféricos e interrupções
- Código no domínio seguro pode acessar o domínio normal
- Código no domínio normal deve usar uma API para interagir com o domínio seguro

ARM TrustZone



ARM TrustZone

Sistema operacional:

- Cada domínio tem seu próprio *bootloader* e SO
  - Domínio normal: Android ou iOS
  - Domínio seguro: SO proprietário

Exemplos:

- OP-TEE (vários fabricantes)
- QSEE (Qualcomm)
- Knox (Samsung)
- L4 (Apple)

SGX - *Software Guard Extensions*

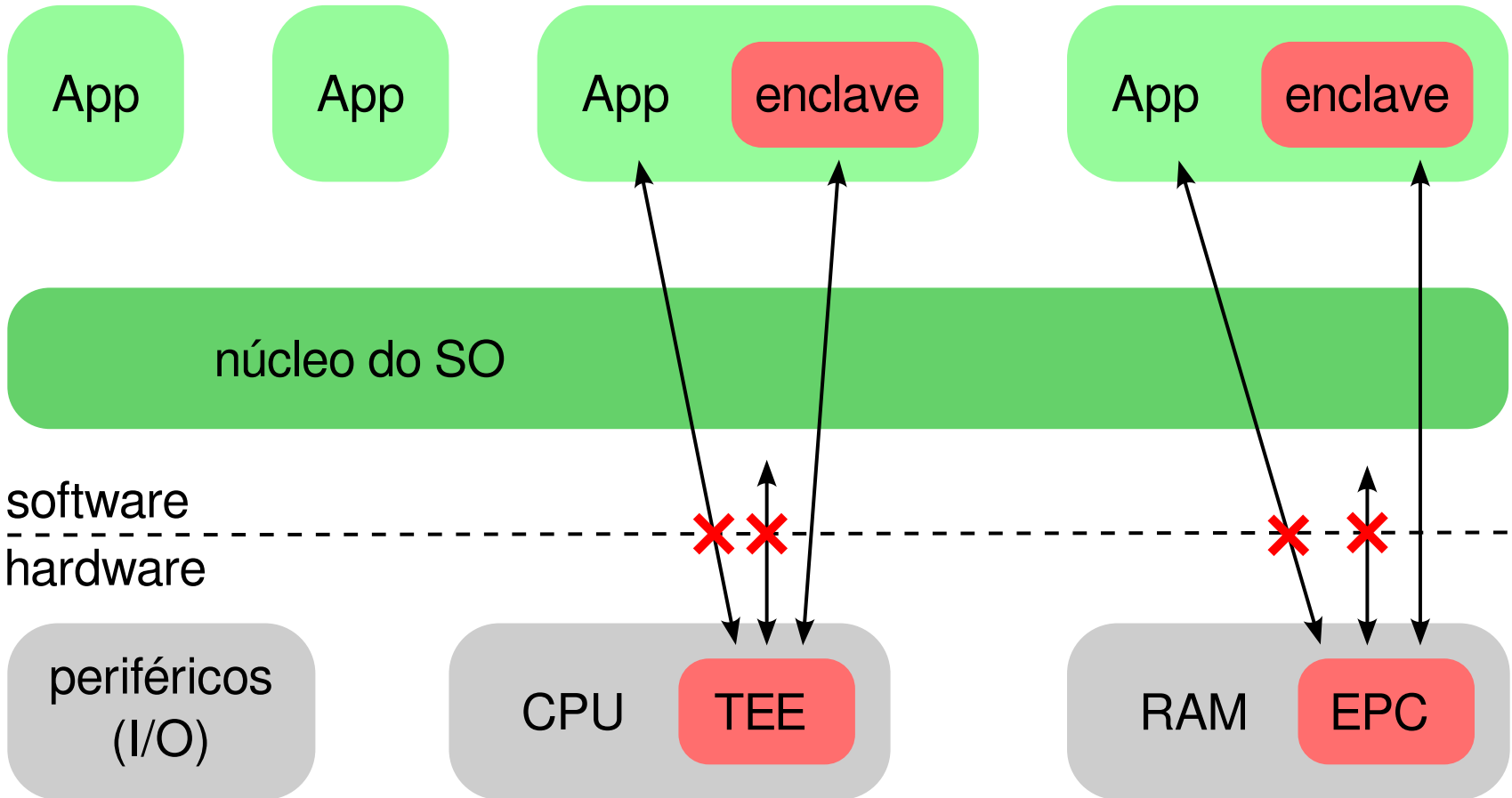
Características:

- Extensão em alguns processadores Intel (2015)
- Cria enclaves criptografados na RAM
- Cada aplicação pode ter vários enclaves

Enclave Page Cache (EPC):

- Área de RAM criptografada para os enclaves
- Protegida de acessos privilegiados
- Hashes de verificação protegem integridade
- Páginas só são decifradas dentro da CPU
- Chaves de criptografia são inacessíveis fora do enclave

SGX - Visão geral



SGX - Interação com os enclaves

Modelo de uso:

- Enclaves são parte das aplicações
- Uma aplicação pode ter vários enclaves
- Interação por *E-Calls* e *O-Calls*

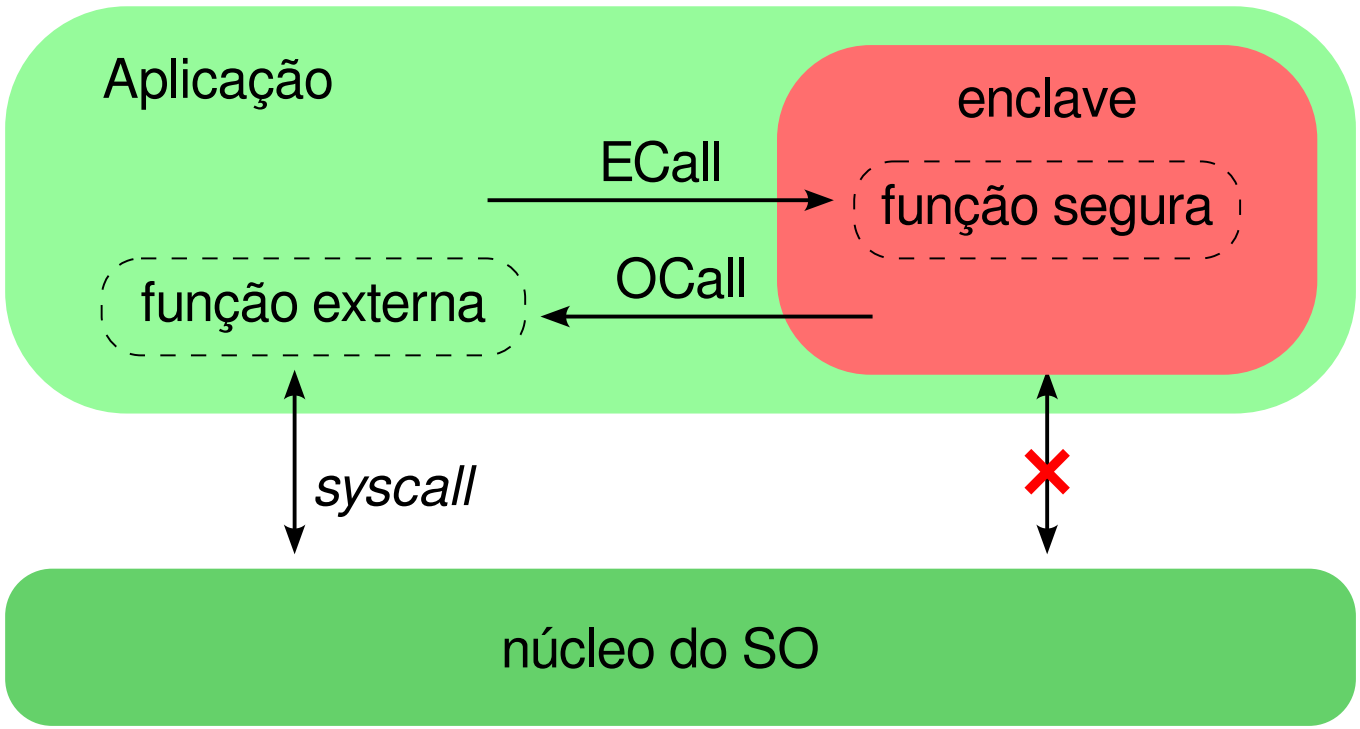
E-Call (Enclave Call):

- Aplicação solicita serviço seguro ao enclave

O-Call (Outside Call):

- Enclave solicita serviço à aplicação
- Útil para fazer chamadas de sistema

Software Guard Extensions



Software Guard Extensions

SGX provê 18 novas instruções para enclaves:

- `ECREATE`: cria estrutura do enclave
- `EINIT`: inicializa um enclave
- `EENTER` ou `ERESUME`: entra ou retorna ao enclave
- `EEXIT`: sai do enclave
- `AEX`: saída assíncrona do enclave (interrupção, exceção)
- `EREPORT`: gera uma mensuração do enclave para atestação
- `EREMOVE`: destrói o enclave
- ...

Software Guard Extensions

Outras funcionalidades importantes:

- Mensuração: registro assinado de cada enclave
- Atestação: verificar integridade e autenticidade do enclave sem revelar seu conteúdo
  - Local: feita por outro enclave ou código externo
  - Remota: feita via rede, exige uma autoridade certificadora externa (*Intel Trust Authority*).
- Selagem: proteger dados sensíveis armazenados fora do enclave

SGX é inovadora mas difícil de usar, pois exige a reescrita das aplicações para separar as áreas críticas em enclaves.

TDX - *Trusted Domain Extensions*

Motivação:

- SGX é difícil de usar
- Crescente adoção de serviços na nuvem
- Necessidade de máquinas virtuais seguras

Características:

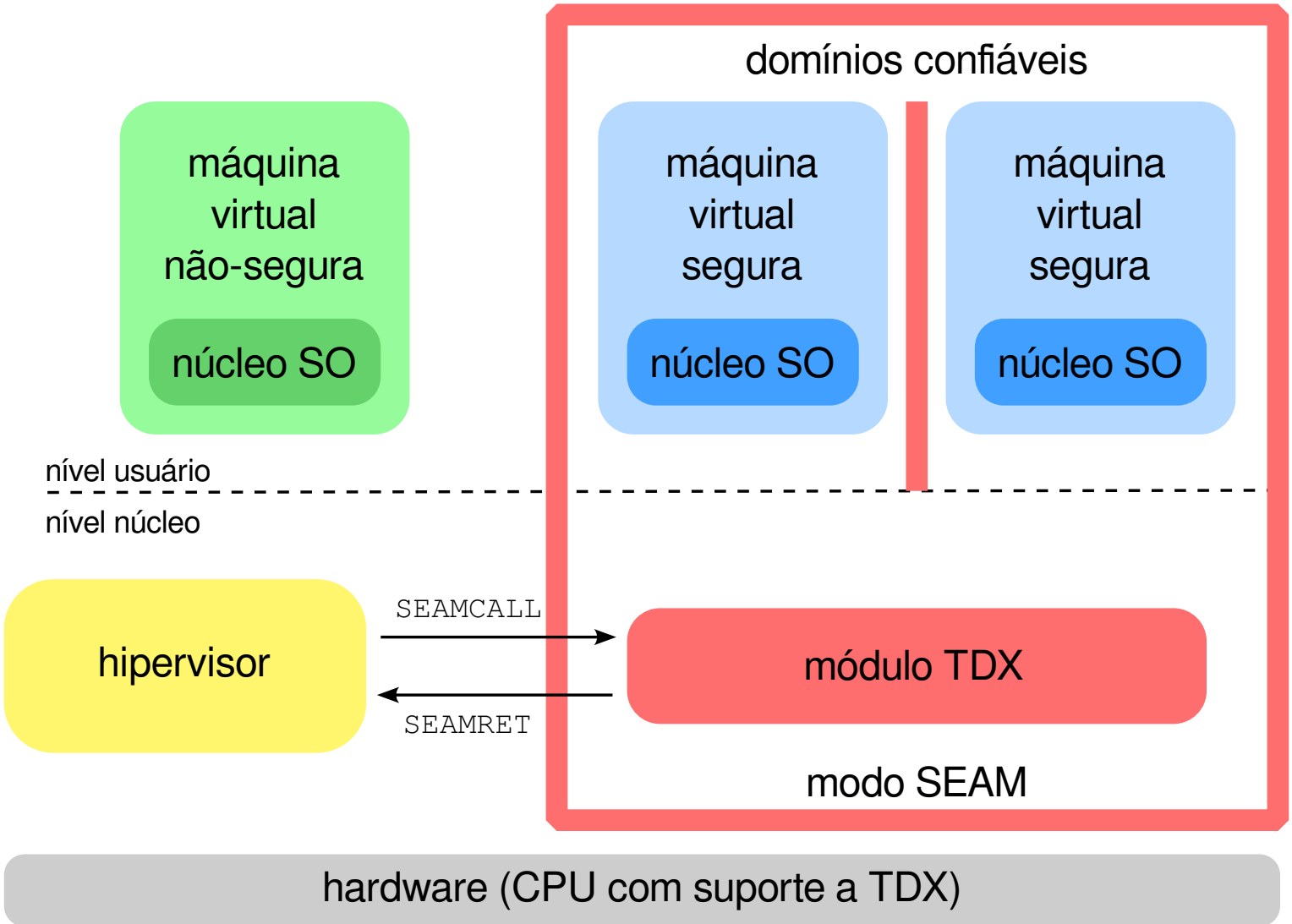
- Extensão para CPUs Intel Xeon (2023)
- Cada VM ou contêiner executa em um domínio seguro
- RAM criptografada e protegida de outras VMs
- Sem acesso do hipervisor, SO e *firmware* do servidor

TDX - Funcionamento

Funcionamento:

- Usa tecnologias anteriores (SGX, VMX, ...)
- *SEecure-Arbitration Mode* (SEAM):
  - Novo modo da CPU, mais privilegiado que o núcleo
  - Entrada e saída do modo SEAM é controlada
  - Novas instruções: `SEAMCALL`/`SEAMRET`
- *Intel TDX module*
  - Software provido pela Intel
  - Executa no modo SEAM
  - Faz a gerência dos domínios seguros

TDX - Arquitetura



TDX - Competidores

Benefícios:

- Provê cadeia de confiança, selagem de dados e atestação remota
- Suporta aplicações sem reescrita de código
- Somente o hipervisor e os SOs precisam ser adequados

Soluções similares de outros fabricantes:

- *AMD Secure Encrypted Virtualization* (SEV)
- *ARM Confidential Compute Architecture* (CCA)
- *IBM Secure Execution*
- *IBM Protected Execution Facility* (PEF)
- *RISC-V Confidential VM Extension* (CoVE).

Integridade do fluxo de controle

Integridade do fluxo de controle

Objetivo:

- Garantir a integridade da execução
- O fluxo de execução segue o previsto no código
- Acessos à memória são feitos como previsto
- Prevenir ataques de desvio da execução
  - *Buffer overflow*, *stack overflow*
  - *Return-oriented Programming*

Importante:

- Os mecanismos definem novas instruções e registradores
- Compiladores, ligadores e bibliotecas precisam ter suporte
- Aplicações precisam ser recompiladas

Intel CET

Intel Control-flow Enforcement Technology (CET)

Provê dois mecanismos para proteger o fluxo de execução:

- *Shadow Stack* (SS)
- *Indirect-Branch Tracking* (IBT)

Intel CET

Shadow Stack (SS):

- Cria uma pilha secundária (“sombra”) em cada aplicação
- Ao chamar uma função copia o endereço de retorno na pilha sombra
- Ao retornar da função, compara os endereços de retorno nas duas pilhas
- Permite bloquear ataques de *stack overflow*

Intel CET

Indirect-Branch Tracking (IBT):

- Nova instrução `ENDBRANCH` marca destinos válidos
- Usada em saltos de execução (*jmp*, *call*, *ret*, etc)
- Interrompe a execução se o destino do salto não for uma instrução `ENDBRANCH`
- Permite detectar ataques de *Return-Oriented Programming*
- *ARM Branch Target Identification* (BTI) é similar

Intel CET

Exemplo de uso de *Indirect-Branch Tracking*:

```c
void func() {}

int main()
{
    void (*f)();

    f = func;
    f(); // chamada indireta
}
```

```asm
main:
    endbr64
    movl $func, %edx
    call *%edx
    ret

func:
    endbr64
    ret
```

Integridade do fluxo de controle

Mecanismos de proteção de memória:

- *ARM Pointer Authentication* (PA)
- *ARM Memory Tagging Extension* (MTE)
- *Intel Memory Protection Keys* (MPK)

Protegem áreas de memória e os ponteiros que apontam para elas

Integridade do fluxo de controle

ARM Pointer Authentication (PA):

- Associa a cada ponteiro uma assinatura
- Permite detectar se o ponteiro foi alterado
- Assinatura calculada com o valor do ponteiro, uma chave secreta e um contexto (exemplo: valor do *stack pointer*
- Assinatura fica no próprio ponteiro (bits MSB sem uso)
- Instruções adicionais para gerar e testar as assinaturas

Integridade do fluxo de controle

ARM Memory Tagging Extension (MTE):

- Uma etiqueta de 4 bits é associada a cada área de memória
- Uma etiqueta de 4 bits é associada a cada ponteiro (MSB)
- Acesso do ponteiro à área de memória só é permitido se ambas as etiquetas forem iguais

Integridade do fluxo de controle

Intel Memory Protection Keys (MPK)

- Associa a cada página da memória um ID (chave) de 4 bits
- Podem existir até 16 chaves distintas
- Registrador PKRU define permissões p/ cada chave:
  - 2 bits por chave: escrita (1 bit), leitura/execução (1 bit)
- Permissões são verificadas ao acessar a página
- Registrador PKRU específico para cada *thread*
  - Pode ser alterado no espaço de usuário
  - Cada *thread* pode definir suas próprias permissões
  - Muito mais rápido que mexer na tabela de páginas

Monitoração

Monitoração

Existem mecanismos de hardware para monitorar a execução:

- Extrair dados sobre o comportamento da aplicação
- Anomalias podem indicar *malware* ou ataques
- Impacto no desempenho menor que monitorar por software

Alguns mecanismos relevantes:

- *Hardware Performance Counters* (HPC)
- *Intel Processor Trace* (PT)
- *Intel Threat Detection Technology* (TDT)

Monitoração

Hardware Performance Counters (HPC):

- Registradores especiais para a contagem de eventos
- Contadores para instruções executadas, acessos à memória, acertos/erros de cache, predição de desvios, ...
- Presentes na maioria dos processadores modernos
- Permitem observar o comportamento de cada aplicação

Monitoração

Intel Processor Trace (PT):

- Captura log de eventos do fluxo de execução (saltos, sequências de instruções, interrupções, etc)
- O registro é feito em uma área comprimida da RAM
- Baixo impacto no desempenho
- Evolução das tecnologias anteriores *Last Branch Record* (LBR) e *Branch Trace Store* (BTS)

Monitoração

Intel Threat Detection Technology (TDT):

- Usa a GPU para varrer a memória RAM
- Busca indicativos de código malicioso
- Incorpora algoritmos de aprendizado de máquina
- Detecta anomalias (*ransomware*, *spyware*, ...)
- Usado em alguns antivírus comerciais