Upcoming x86 Technologies for Malicious Hypervisor Protection

David Kaplan, Security Architect

11/1/19

DISCLAIMER

This presentation is a technical explanation of what the discussed technology has been designed to accomplish. The actual technology or feature(s) in the resultant products may differ or may not meet these aspirations. Each description of the technology must be interpreted as a goal that AMD strived to achieve and not interpreted to mean that any such performance is guaranteed to be fully achieved. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated.

WHY NOT TRUST THE HYPERVISOR

AMD

- Growing interest in ***Confidential Computing*** where workloads can be run in untrusted hosting environments
  - Systems managed by a 3rd party (e.g., public cloud)
  - Systems physically located in sensitive geographies
- AMD is focusing on virtualization security for
  - **Cloud Computing** – Running sensitive workloads while reducing risk of data leakage in multi-tenant systems
  - **Stronger Isolation** – Reduce risk of infrastructure bugs and software errors in shared systems
- Customers desire stronger virtualization security to reduce risk and improve scalability
- Hosters desire stronger virtualization security to attract additional customers and maintain independence

INTRODUCING SEV-SNP

AMD

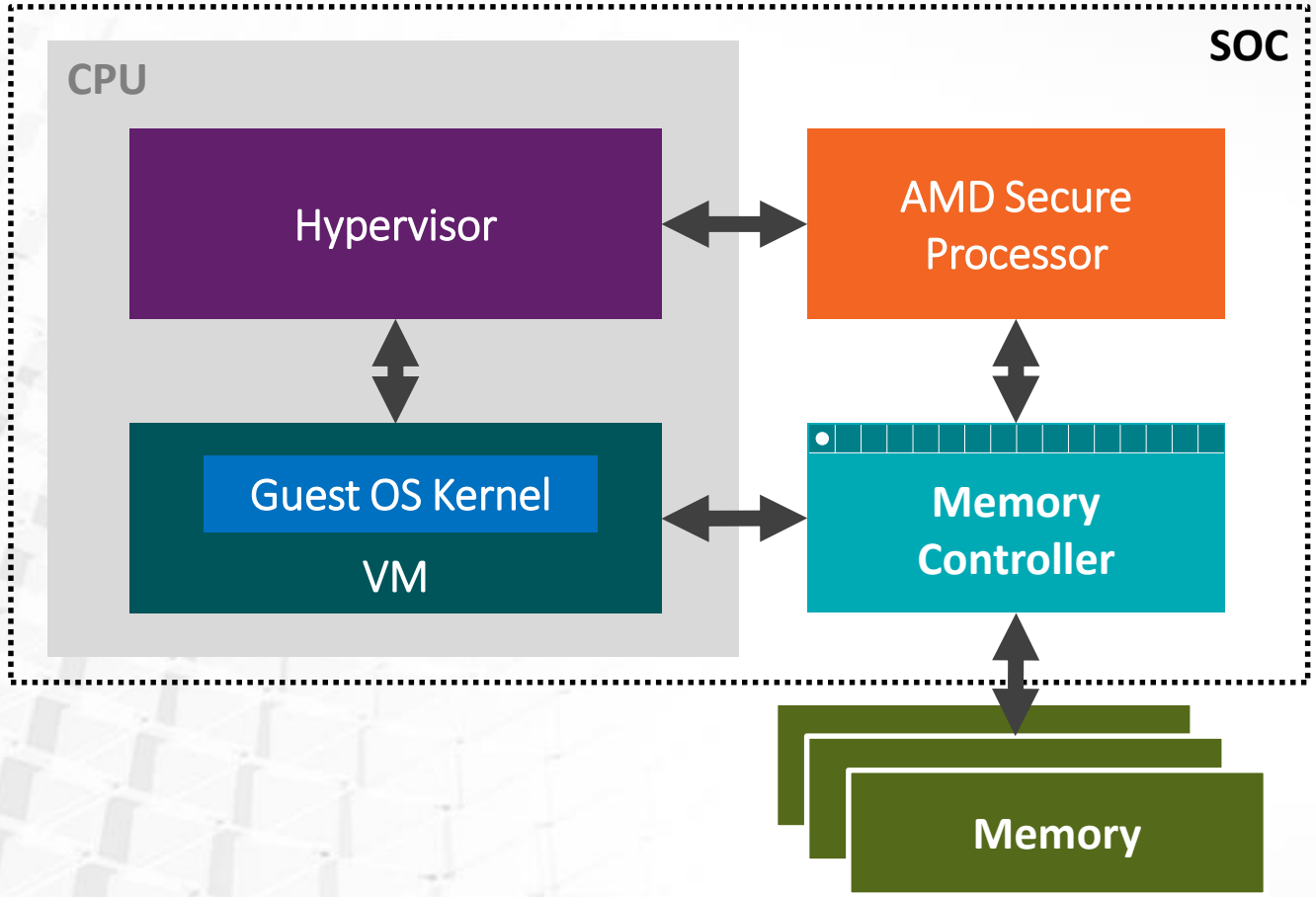
- ***Secure Nested Paging*** (SEV-SNP) is the next generation of AMD Secure Encrypted Virtualization (SEV) technology

- SEV-SNP builds on existing AMD SEV and AMD SEV-ES (Encrypted State) features to provide ***stronger security, additional use models, and more*** to protected VMs
  - SEV and SEV-ES are supported today in AMD EPYC Processors
  - SEV is supported in most recent Linux distros (kernel >=4.15), SEV-ES support is in progress

- SEV-SNP is designed to protect a VM from a malicious hypervisor in specific ways
  - Useful in public cloud and any scenario where the hosting environment cannot be trusted

AMD SEV COMPONENTS

- **AMD Secure Processor**
  - Manages keys during VM lifecycle
  - Runs SEV API firmware
- **Hypervisor**
  - Interfaces with SEV API
  - Allocates resources/schedules VMs
- **Guest OS**
  - Chooses which pages to encrypt via page tables
- **Guest Applications**
  - Unaffected and unaware of SEV technology



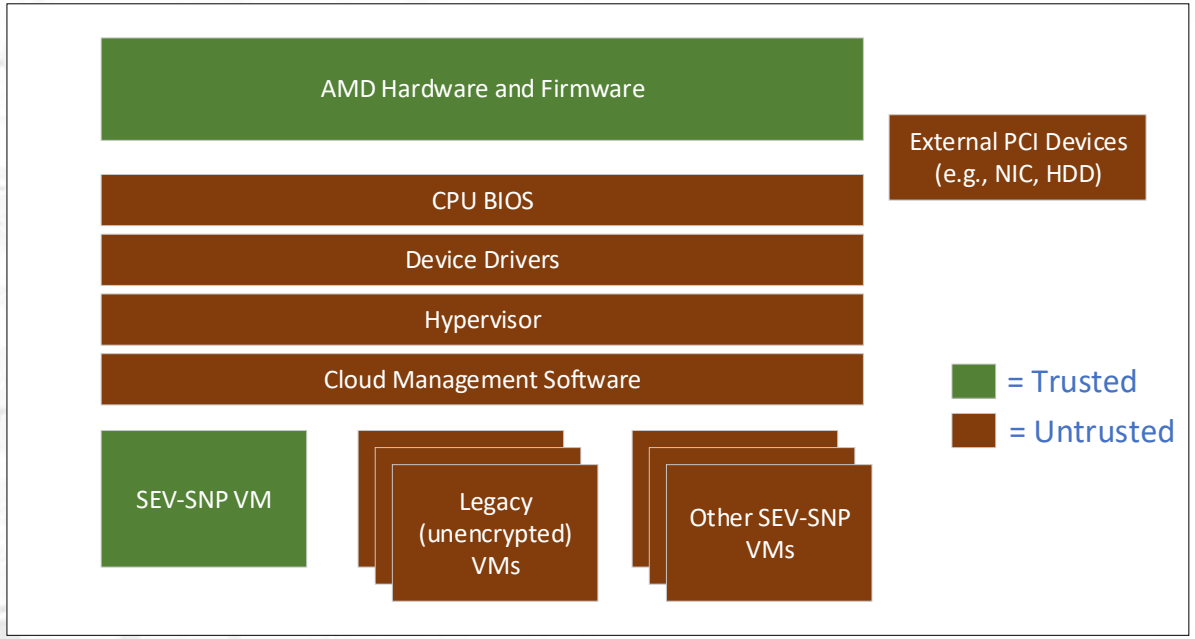
- SEV is designed to be useful across many virtualization scenarios with minimal performance overhead

THREAT MODEL

- SEV-SNP features a stronger threat model than past SEV features

- **Trusted components**
  - AMD hardware and signed firmware
  - OS software inside the target VM
- **Untrusted components**
  - Everything else (including other VMs)
  - External devices

- Untrusted components are assumed to be malicious
  - May be conspiring with each other



VM THREATS

- **Confidentiality** of VM memory is protected via AES-128 memory encryption
  - All guest register state is encrypted and swapped atomically starting with SEV-ES
- **Integrity** of VM memory is new with SEV-SNP
- **Integrity** ensures that *if an SEV-SNP VM is able to read a page of private (encrypted) memory, it must always read the value it last wrote*
- Integrity protection must hold in all cases including page swapping, VM migration, etc.

✓ = Mitigated ★ = Optionally Mitigated ⊘ = Not Mitigated	*SEV*	*SEV-ES*	*SEV-SNP*
Potential Threats			
Confidentiality			
VM Memory *Example attack: Hypervisor reads private VM memory*	✓	✓	✓
VM Register State *Example attack: Read VM register state after VMEXIT*	⊘	✓	✓
DMA Protection *Example attack: Device attempts to read VM memory*	✓	✓	✓
Integrity			
Replay Protection *Example attack: Replace VM memory with an old copy*	⊘	⊘	✓
Data Corruption *Example attack: Replace VM memory with junk data*	⊘	⊘	✓
Memory Aliasing *Example attack: Map two guest pages to same DRAM page*	⊘	⊘	✓
Memory Re-Mapping *Example attack: Switch DRAM page mapped to a guest page*	⊘	⊘	✓

VM THREATS

- **Availability** is protected for the hypervisor but not for guests
  - A malicious HV can choose never to run a guest
  - A malicious guest cannot prevent the HV from gaining control
- **Offline Physical Access Attacks** are protected, but Online attacks (e.g., modifying data on a bus during execution) are not due to their attack difficulty
  - Note that device DMA, including from malicious devices, is protected

✓ = Mitigated ★ = Optionally Mitigated ⊘ = Not Mitigated	*SEV*	*SEV-ES*	*SEV-SNP*
Potential Threats			
Availability			
Denial of Service on Hypervisor *Example attack: Malicious guest refuses to yield/exit*	✓	✓	✓
Denial of Service on Guest *Example attack: Malicious hypervisor refuses to run guest*	⊘	⊘	⊘
Physical Access Attacks			
Offline DRAM analysis *Example attack: Cold boot*	✓	✓	✓
Active DRAM corruption *Example attack: Manipulate DDR bus while VM is running*	⊘	⊘	⊘

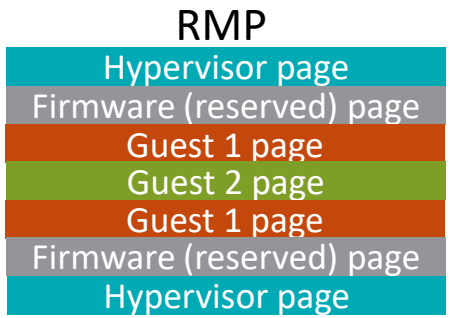
VM THREATS

- SEV-SNP adds protection against firmware rollback attacks by default
- A number of protections are optional:
  - Malicious interrupt injection (requires SW support)
  - Spectre v2 poisoning protection
  - Hardware debug registers
  - CPUID spoofing
- Certain side channels are not protected by HW
  - Architectural side channels due to SW algorithms
  - Page fault side channels
  - Performance counters
- SEV-SNP focuses on protecting VM data. VM code execution confidentiality is not guaranteed in this technology

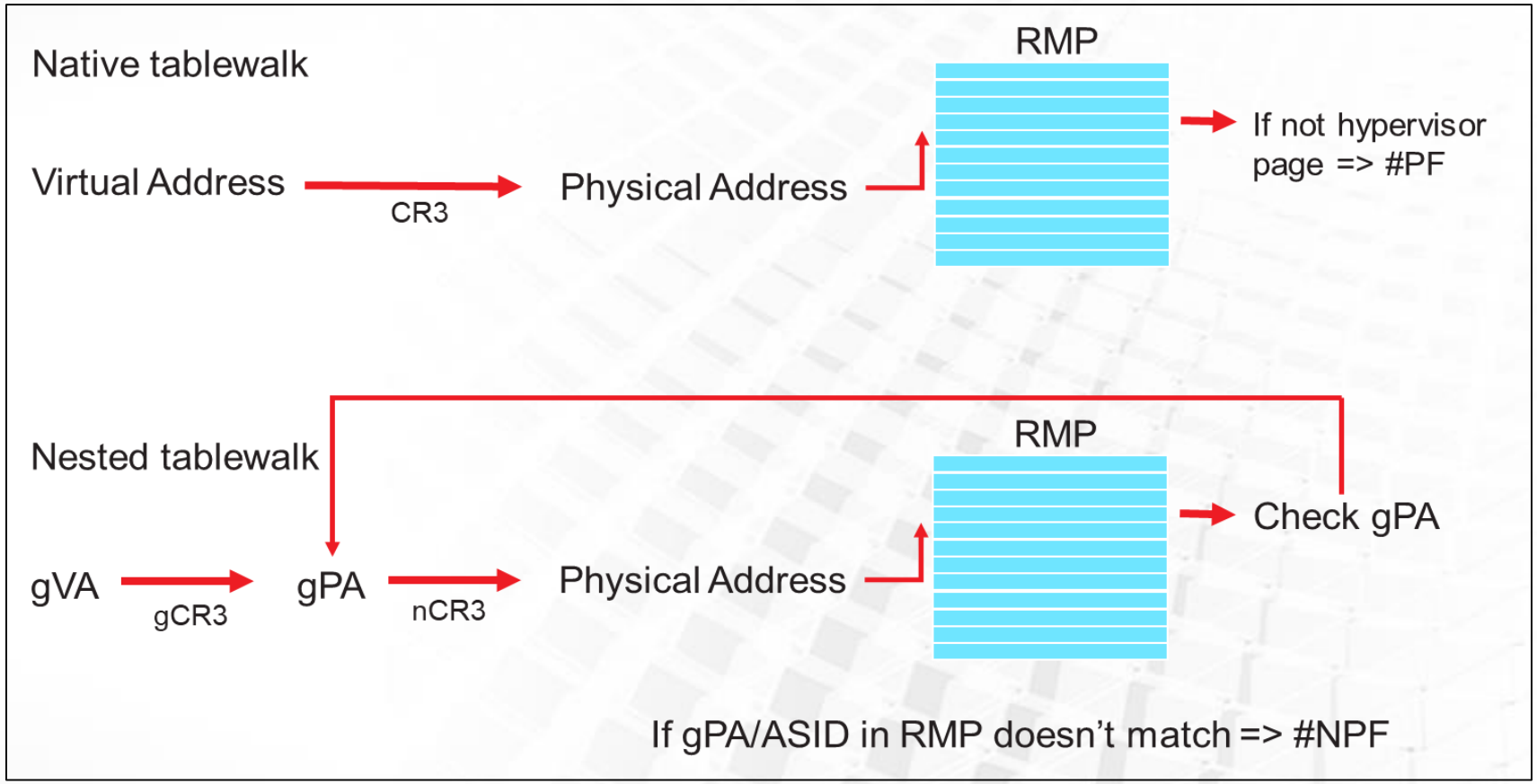
✓ = Mitigated ★ = Optionally Mitigated ⊘ = Not Mitigated	*SEV*	*SEV-ES*	*SEV-SNP*
Potential Threats			
Misc.			
TCB Rollback *Example attack: Revert AMD-SP firmware to old version*	⊘	⊘	✓
Malicious Interrupt/Exception Injection *Example attack: Inject interrupt while RFLAGS.IF=0*	⊘	⊘	★
Indirect Branch Predictor Poisoning *Example attack: Poison BTB from hypervisor*	⊘	⊘	★
Secure Hardware Debug Registers *Example attack: Change breakpoints during debug*	⊘	⊘	★
Trusted CPUID Information *Example attack: Hypervisors lies about platform capabilities*	⊘	⊘	★
Architectural Side Channels *Example attack: PRIME+PROBE to track VM accesses*	⊘	⊘	⊘
Page-level Side Channels *Example attack: Track VM access patterns through NPT*	⊘	⊘	⊘
Performance Counter Tracking *Example attack: Fingerprint VM apps by performance data*	⊘	⊘	⊘

ENFORCING INTEGRITY

- Memory integrity is enforced using a new DRAM structure called the **Reverse Map Table (RMP)**
- There is 1 RMP for the entire system, it is created by software during boot
- Basic properties:
  - RMP contains 1 entry for every 4k of assignable memory
  - RMP is indexed by System Physical Address (SPA)
  - RMP entries may only be manipulated via new x86 instructions
- The RMP indicates **page ownership** and dictates write-ability. Examples:
  - A page assigned to a guest is only writeable by that guest
  - A page assigned to the hypervisor cannot be used as a private (encrypted) guest page
  - A page used by AMD firmware cannot be written by any x86 software



RMP CHECKS



RMP is checked on:

Writes in any mode

Reads from SEV-SNP guests

The RMP is not checked on reads in certain modes (e.g., HV mode) because memory encryption ensures confidentiality

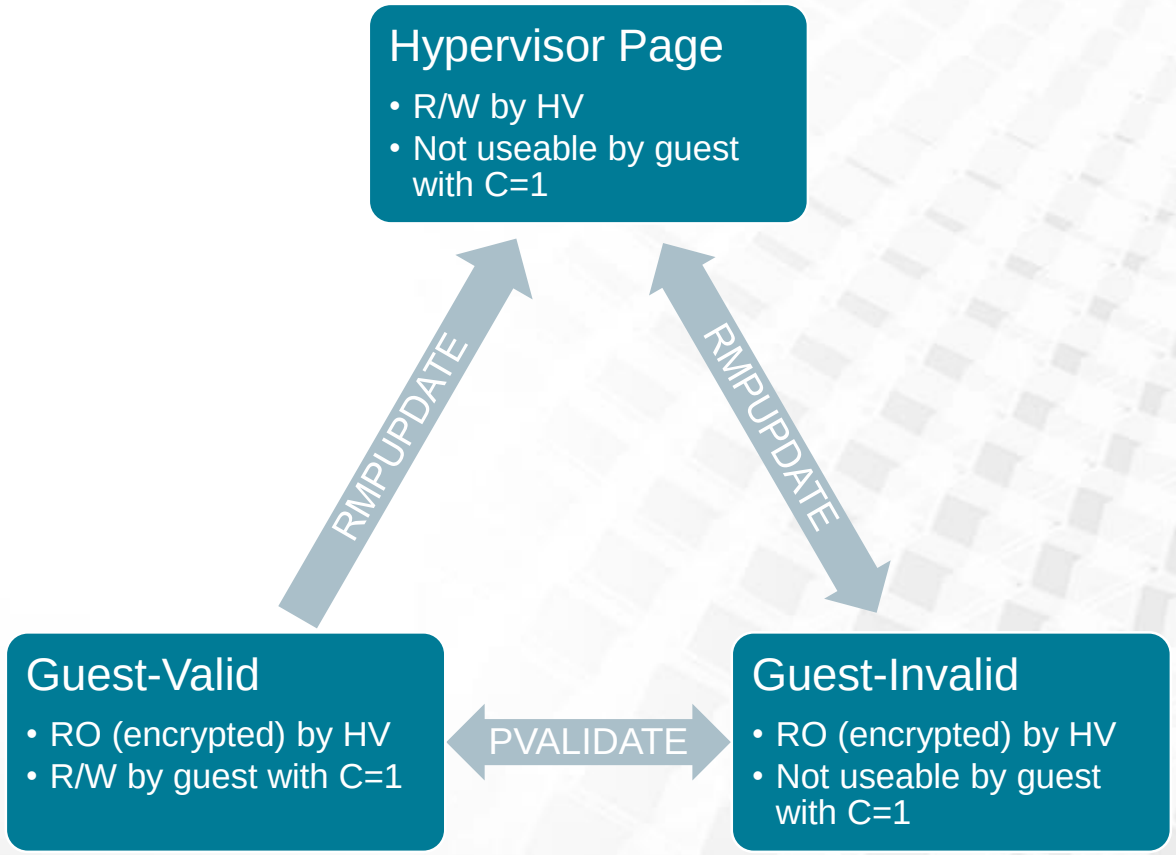
The RMP directly protects against

Data corruption/replay (only assigned guest can write to a page)

Memory aliasing (one page can only be mapped to one guest at a time)

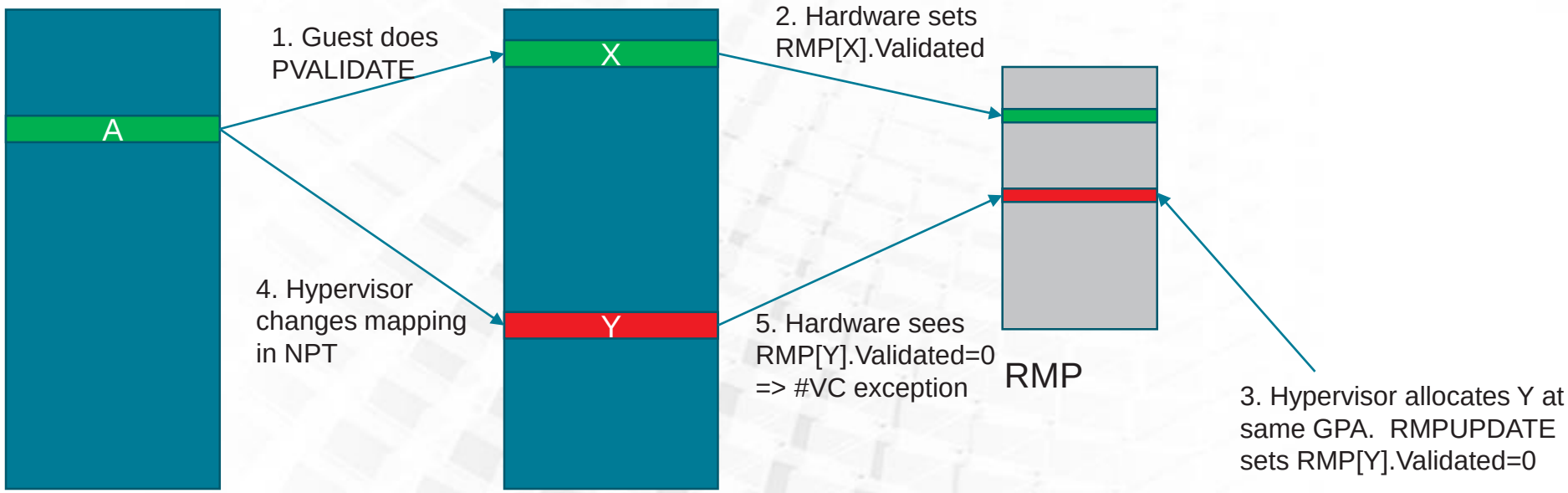
PAGE VALIDATION

PROTECTION AGAINST MEMORY RE-MAPPING



- Hypervisor assigns a page to a guest via **RMPUPDATE** instruction
  - CPU hardware sets up the new RMP entry
  - Puts page into Guest-Invalid state
    - Validated bit cleared to 0
  - Page is now not useable by guest or hypervisor
- Guest does **PVALIDATE** to accept the page
  - Transitions page into Guest-Valid state
    - Validated bit set to 1
  - Guest is expected to validate each GPA only once
- Page validation guarantees that each GPA only maps to one valid SPA at any given time

PAGE REMAPPING

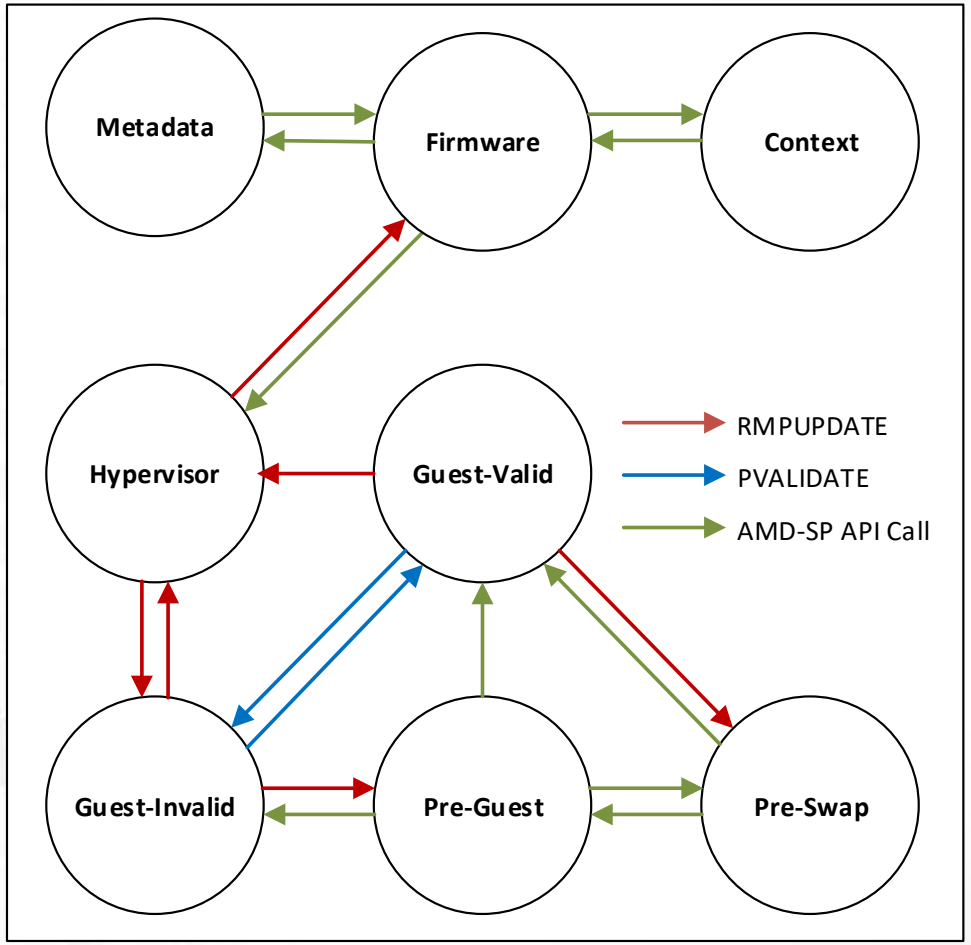


6. After seeing #VC exception, guest realizes something is wrong because it had already validated page A

PAGE STATES

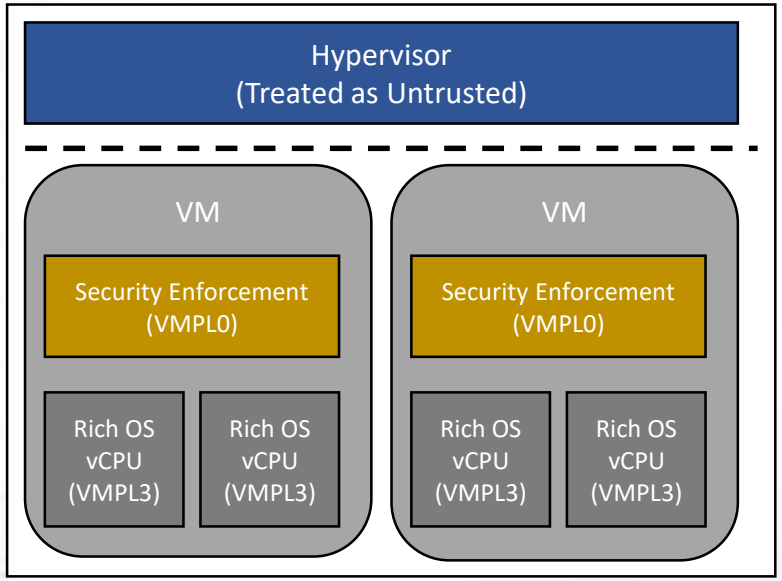
- Pages can exist in one of 8 states, used for various purposes
- Page states are transitioned via x86 instructions or AMD-SP API calls
  - The AMD-SP API is used for VM lifecycle management and page swapping

STATE	DESCRIPTION	NOTES
HYPERVISOR	Default state for otherwise unassigned memory	Used for hypervisor memory, non-SNP-VM memory, and shared (C=0) memory
GUEST-INVALID	Page is assigned to a guest but not ready to be used	Not useable by SEV-SNP VMs until validation has occurred
GUEST-VALID	Page is assigned to a guest and useable	Page may be used as private (C=1) memory by the assigned SEV-SNP VM
PRE-GUEST	Page is Immutable and not validated	Used when initially launching SEV-SNP VMs
PRE-SWAP	Page is Immutable and validated	Used when swapping guest pages to disk
FIRMWARE	Page is Immutable and reserved for AMD-SP use	Typically used as transitory state until AMD-SP has configured the page
METADATA	Page is Immutable and used for metadata	Metadata is used when swapping guest pages to disk
CONTEXT	Page is Immutable and used for context information	Context pages are used by the AMD-SP to identify individual VMs and hold per-VM data



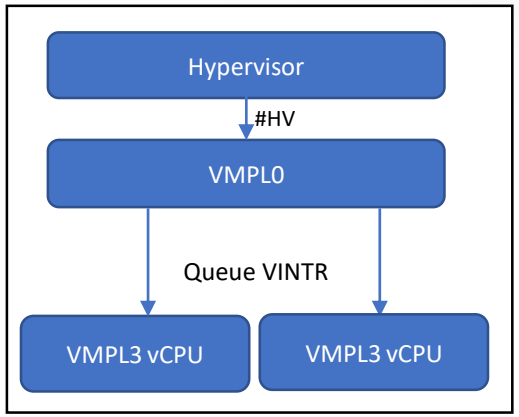
VMPLS

- **Virtual Machine Privilege Levels** (VMPLs) are a new optional concept in SEV-SNP
- VMPLs enable a guest to sub-divide its address space into 4 levels
- When enabled, RMP entries are extended with additional permission bits
  - These bits control: Read/Write/Execute-User/Execute-Supervisor at each level
  - For example: A page may be read/write at VMPL1 but read-only at VMPL2
- New x86 instruction **RMPADJUST** modifies VMPL permissions
  - Only useable inside a guest and only for their address space
- VMPLs can enable a security enforcement layer on top of a rich OS



INTERRUPT PROTECTIONS

- Optional interrupt protections can be used to help avoid a guest encountering unexpected interrupts
  - For example: Physical interrupt when EFLAGS.IF=0, or low priority interrupt when at high TPR



- Interrupt protections are designed to be used with a multi-VMPL architecture
  - **Restricted Injection** mode may be used by VMPL0
    - In this mode, no interrupt/exception injection may occur except for a single new vector (#HV)
    - A para-virtualized interface exists between VMPL0 and the HV to pass event information
  - **Alternate Injection** mode may be used by other levels
    - In this mode, all interrupt/exception injection comes from a secure area
    - VMPL0 is expected to deliver events to the vCPUs when appropriate, enforcing required security

- Combined, these modes can support full APIC emulation inside of the guest trust boundary

UN-ENLIGHTENED GUEST SUPPORT

AMD

- SEV-SNP includes new hardware that makes it easier to run un-enlightened guest OS's using VMPLs

- Specific capabilities:
  - **Virtual TOM** – New register that allows for marking memory as encrypted without modifying page tables
  - **Reflect-VC** – New capability which takes #VC exceptions (which occur when HV support is needed) and turns them into VMEXIT events

- An enlightened VMPL0 layer can therefore emulate required events on behalf of an un-enlightened lower level (e.g., VMPL3) rich OS
  - For example: VMPL3 does CPUID. This causes a VMEXIT and the HV asks VMPL0 to assist with the emulation. VMPL0 inspects the VMPL3 register state, performs the required emulation, and then asks the HV to resume VMPL3

- While not as fast as native enlightenment, these capabilities may make it easier to secure legacy workloads with VM isolation technology

TRUSTED PLATFORM INFORMATION

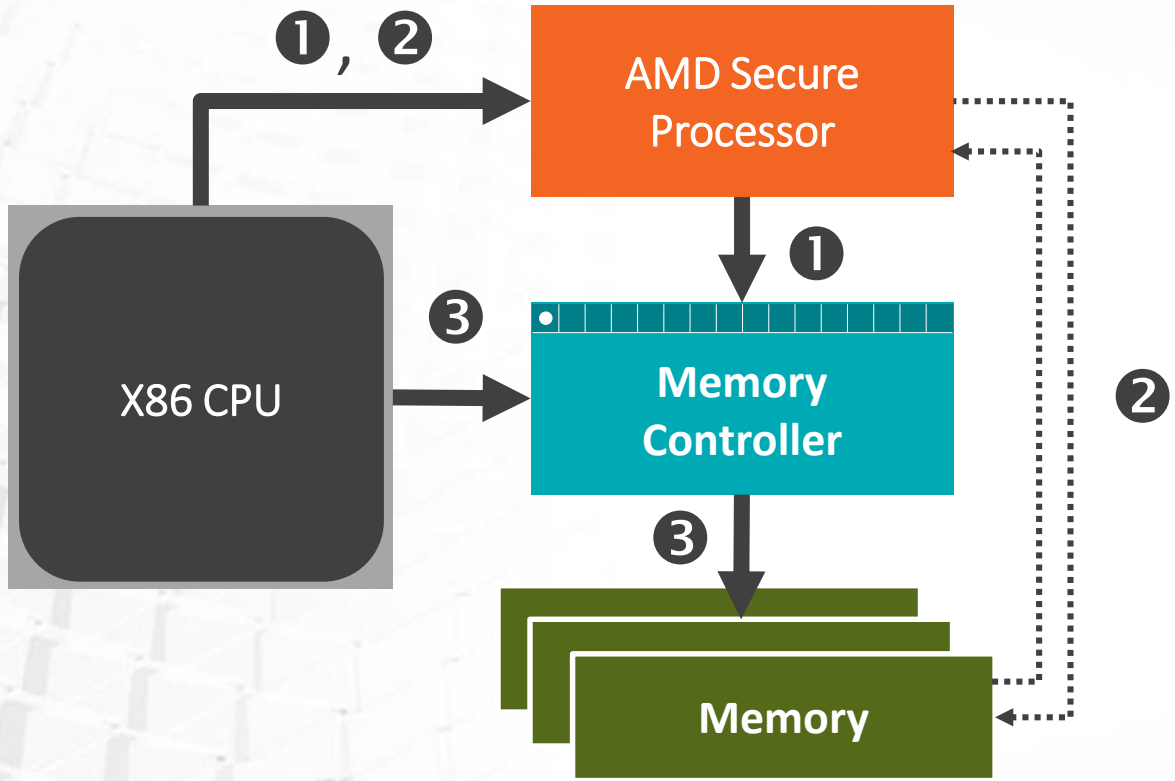
AMD

- Traditionally, platform capabilities are discovered by CPUID, which is emulated by the hypervisor
- SEV-SNP adds new capabilities for CPUID filtering so guests can ensure that security sensitive information is correct
  - For example: XSAVE state size is considered security sensitive to avoid potential buffer overflows
  - Also, any capabilities (e.g., AES-NI) the hypervisor wishes to report must exist on the platform
- Trusted platform information may be communicated in two ways:
  - During guest launch, the hypervisor may pre-fill in a page of CPUID information. The AMD-SP will filter this and correct security sensitive information as required
  - During runtime, the guest may make a secure call to the AMD-SP to ask it to verify CPUID information
- Filtering capability allows a hypervisor to restrict the set of features on a platform, but never add to them

VM Lifecycle

GUEST LAUNCH

1. **Host OS initializes AMD Secure Processor (AMD-SP)**
   - AMD-SP generates random memory key (VEK)
   - Host OS selects key slot in Memory Controller
2. **Host OS allocates & initializes image memory**
   - Host OS places initial image into DRAM
   - AMD-SP reads memory, writes back out with VEK
3. **Host OS completes guest launch and provides IDB**
   - AMD-SP verified the ID Block (IDB)
   - AMD-SP marks guest runnable if IDB checks pass
   - Guest OS manages encrypted pages



IDB is new in SEV-SNP and contains

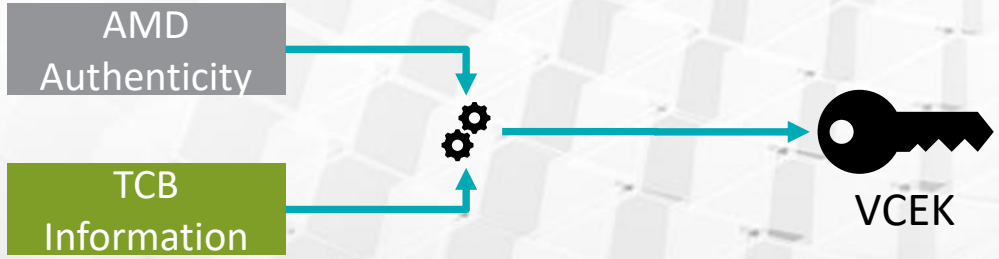
- Expected launch measurement
- Expected policy information (e.g., debug-ability)
- ID parameters
- Signature and public key

TCB VERSIONING

- The AMD-supplied TCB of SEV-SNP consists of several mutable components such as CPU microcode patch, AMD-SP firmware components, etc.
- If a TCB component has a bug, a customer should be able to verify that their VM is running on a patched system

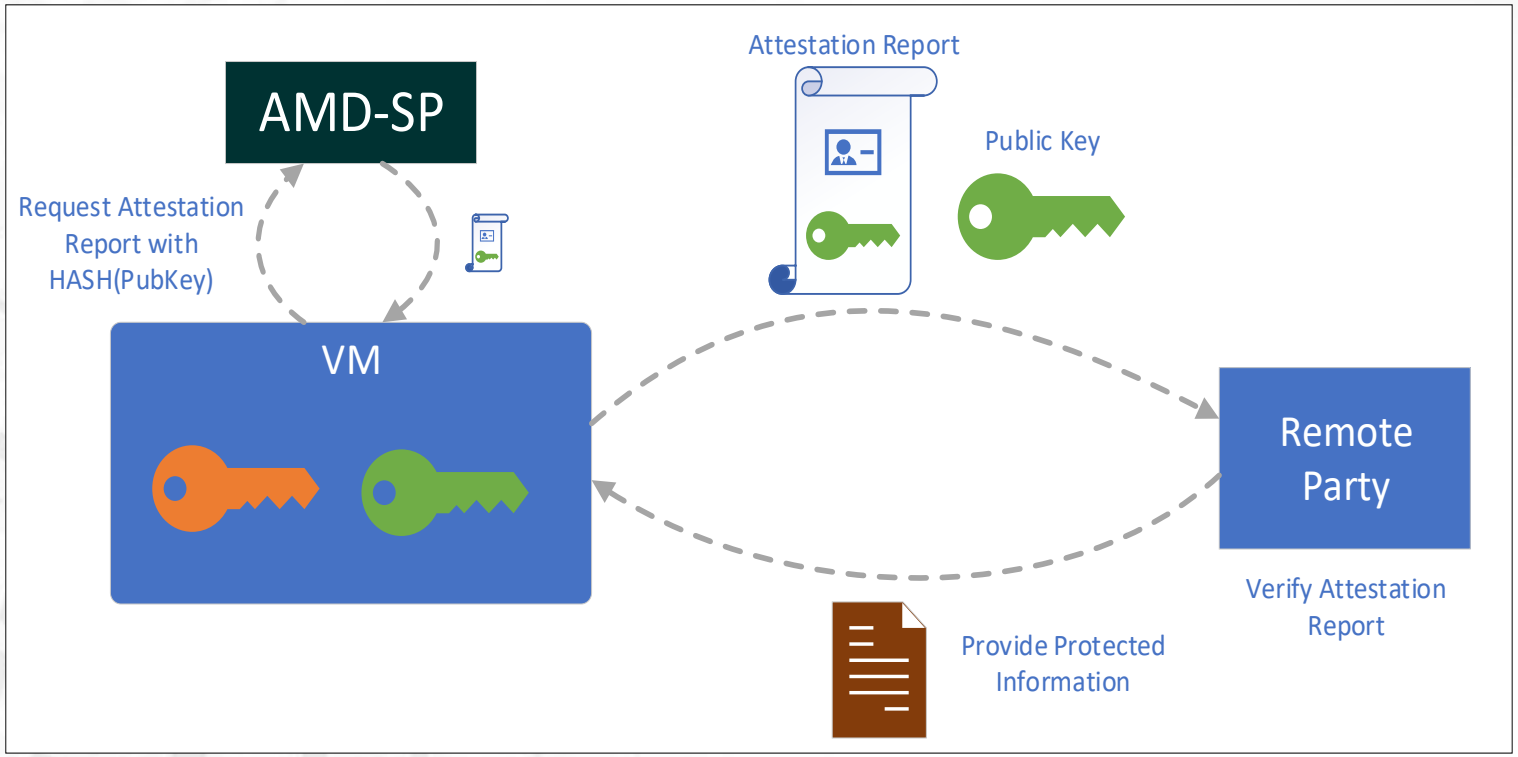
- SEV-SNP adds a new concept called a **Versioned Chip Endorsement Key** (VCEK)
- The VCEK is based on a fused chip-unique value and is derived from TCB component versions

- The mathematical construction of the VCEK makes it impossible for version N of a component to predict that the VCEK will be for version N+1
- The VCEK is used to sign attestation reports, and for other purposes



VM ATTESTATION

- During runtime, SEV-SNP VMs may ask the AMD-SP for an attestation report
- This report contains
  - VM identity information
  - TCB information
  - VM-supplied data (e.g., public key hash)
- Attestation report is signed by VCEK
- Remote part may use report to verify that supplied data came from the desired VM



VM MIGRATION

- There are 2 aspects of VM migration
  - **Authentication** – Verifying the new machine is capable as defined by the guest owner
  - **Data Movement** – Securely moving the VM memory to the new machine

- SEV-SNP adds the concept of a **Migration Agent** (MA)
  - The MA is responsible for machine **authentication and policy enforcement**
  - The MA is bound to the VM at creation time and MA information is included in the attestation report
  - The MA enables arbitrarily flexible migration policies and can facilitate offline save/restore

- Data movement is handled either through AMD-SP or VM enlightenment code
  - See Tom Lendacky's KVM forum talk for more details on proposed VM enlightenment

SIDE CHANNELS

AMD

- SEV-SNP offers a number of optional protections for side channel attack mitigation

Spectre v2 (within guest)	Virtualized SPEC_CTRL MSR
	Guest may used IBRS/IBPB/etc. modes as desired
Spectre v2 (host->guest)	Guest may opt into protection from hypervisor based BTB poisoning
	Hardware ensures that BTB entries are flushed if required on VM entry
SMT	Guests may opt into a policy that requires them to be run on an SMT disabled system
	Can help mitigate threats from shared resources in SMT designs

SUMMARY

AMD

- Protecting VMs from malicious hypervisors enables computing in untrusted environments
- SEV-SNP builds upon previous AMD technologies to offer **stronger** protection with more **flexibility**
- SEV-SNP enables
  - Integrity protection for memory in addition to confidentiality
  - New security use cases and un-enlightened guest support
  - Flexible attestation and migration architectures
  - Additional security protection around interrupts, side channels, etc.

- More details about SEV-SNP can be found in our new whitepaper
  - **AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More**

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION