Property-Based TPM Virtualization

Marcel Winandy

Joint work with: **Ahmad-Reza Sadeghi, Christian Stüble**

Horst Görtz Institute for IT Security
Chair for System Security
Ruhr-University Bochum, Germany

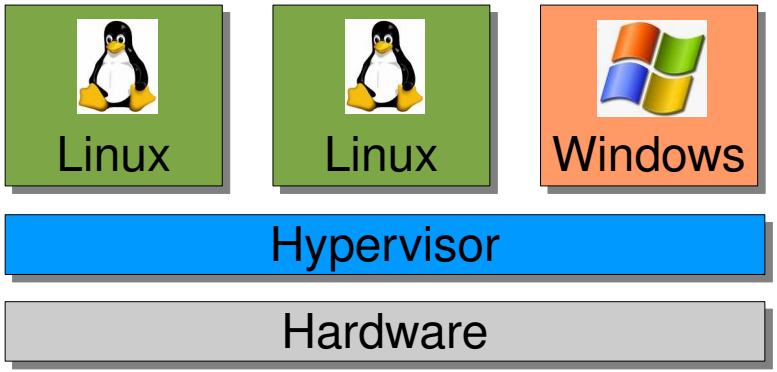
Sirrix AG
security technologies

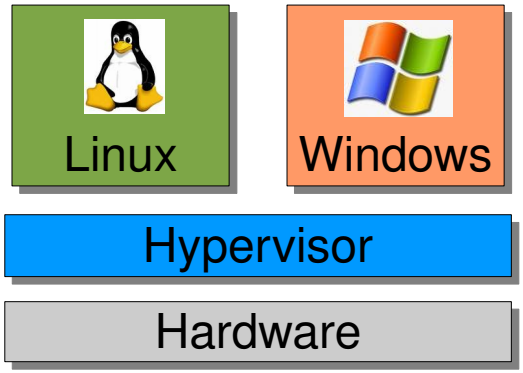
Sirrix AG security technologies
Bochum, Germany

Introduction: Virtualization

Horst Görtz Institute for IT-Security

- Features
  - Commodity operating systems on various hardware platforms
  - Virtual machines: suspend & resume, migration
  - Security: isolation of virtual machines
  - Application scenario: corporate/private computing
    - Isolated work loads for private and corporate working
    - Isolated work loads for different security levels

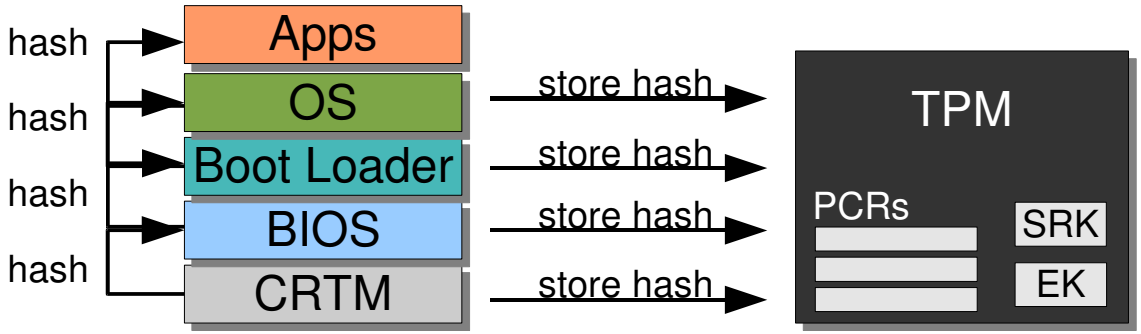




Introduction: Trusted Computing (TPM)

Horst Görtz Institute for IT-Security

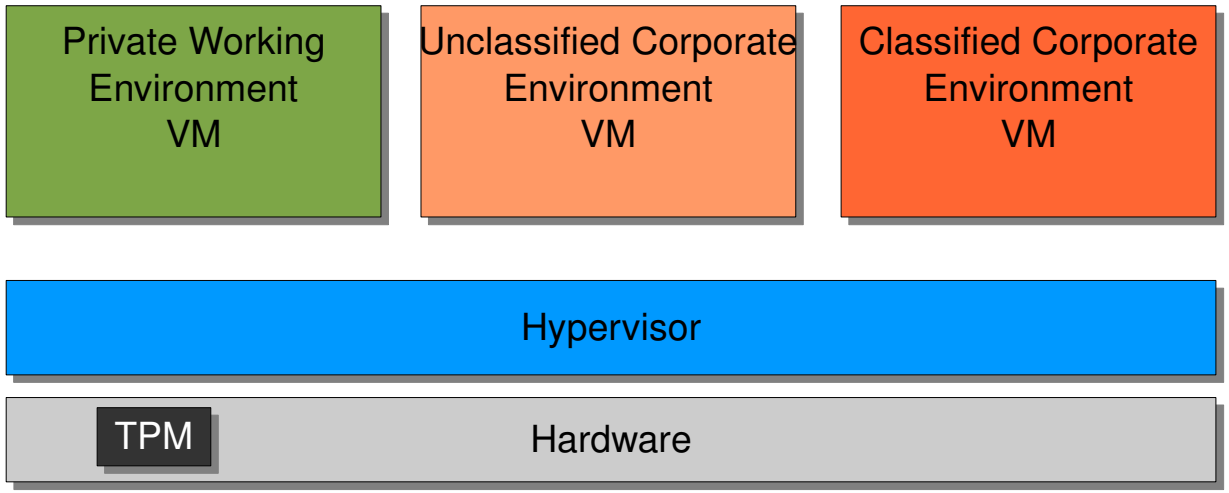
- **TPM**: cheap, tamper-evident hardware security module
  - Cryptographic functions (RSA, SHA-1, key generation, RNG)
  - Protected storage for small data (e.g. keys)
  - Special keys: Endorsement Key (EK) and Storage Root Key (SRK)
- **Authenticated Boot** (recording integrity measurements)
  - Measurements stored in Platform Configuration Registers (PCRs)
  - Each component measures next component (*chain of trust*)



- **Attestation and Sealing**
  - Attestation Identity Key (AIK) signs PCRs for (remote) attestation
  - Binding key is used to encrypt data to the current PCR values (decrypting only possible with same PCR states)

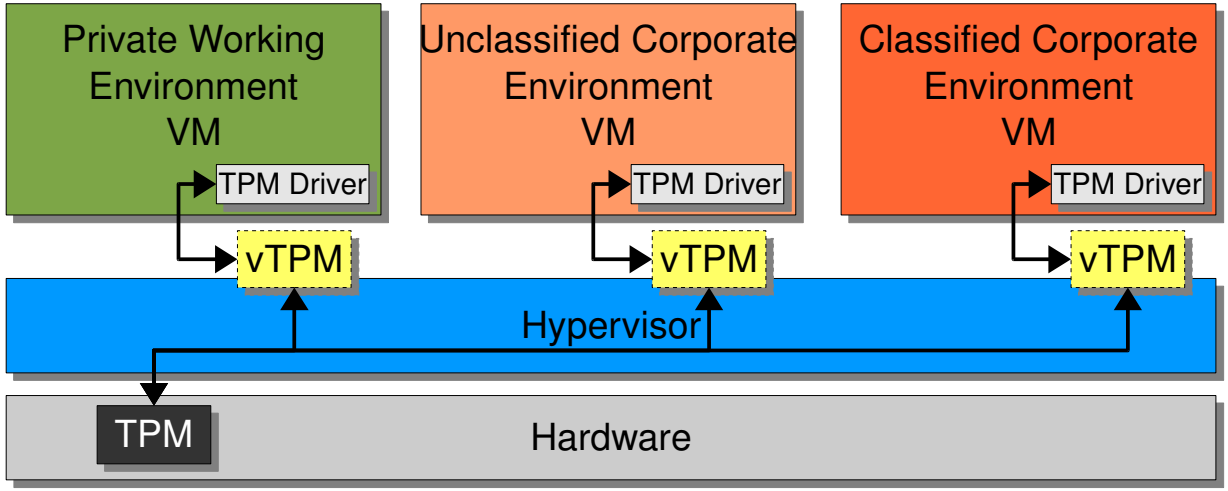
Introduction: Virtual TPM (vTPM)

- Each VM should be able to use TPM
  - Providing protected storage and crypto coprocessor
  - Assurance about the booted hypervisor and virtual machines
  - Support for migration



Introduction: Virtual TPM (vTPM)

- Each VM should be able to use TPM
  - Providing protected storage and crypto coprocessor
  - Assurance about the booted hypervisor and virtual machines
  - Support for migration
- **Virtualization of the TPM**
  - Emulation in software, but binding to VM and hardware TPM
  - Berger et al. (USENIX 2006), Scarlata et al. (2007)



Shortcomings of Existing vTPM Solutions

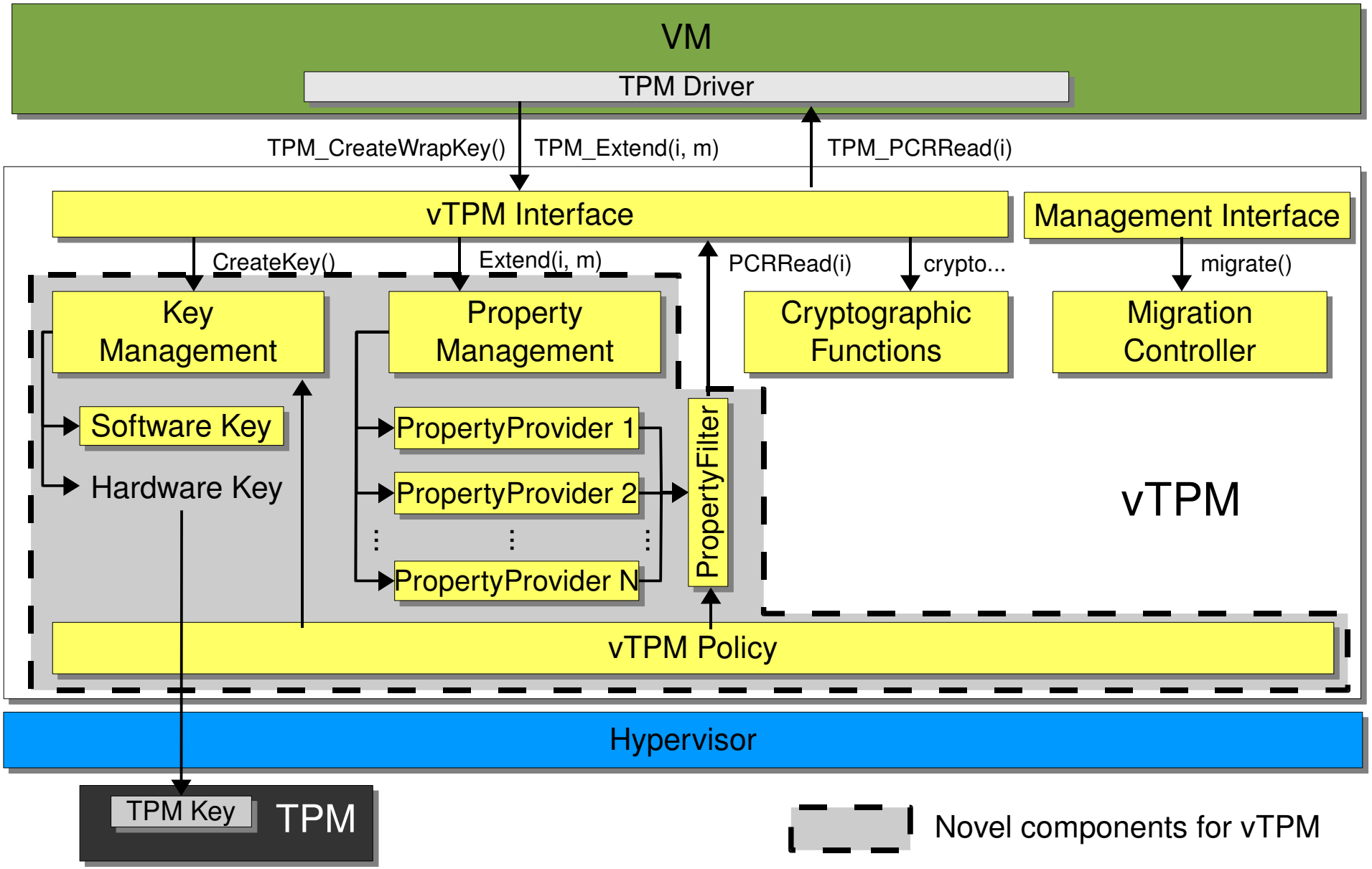
- Migration
  - Protected data bound to binary representation of hypervisor
    - VM's data may be unavailable after migration to another platform
- Keys
  - Differentiated strategies for key generation missing
    - some IT environments demand hardware-protected keys
    - wheras others would benefit from flexibility of software keys
- Privacy
  - Revealing information about system configuration
    - (v)TPM reveals information during remote attestation of PCR values
    - Profiling (security risk) and discrimination possible

New vTPM Design

Horst Görtz Institute for IT-Security

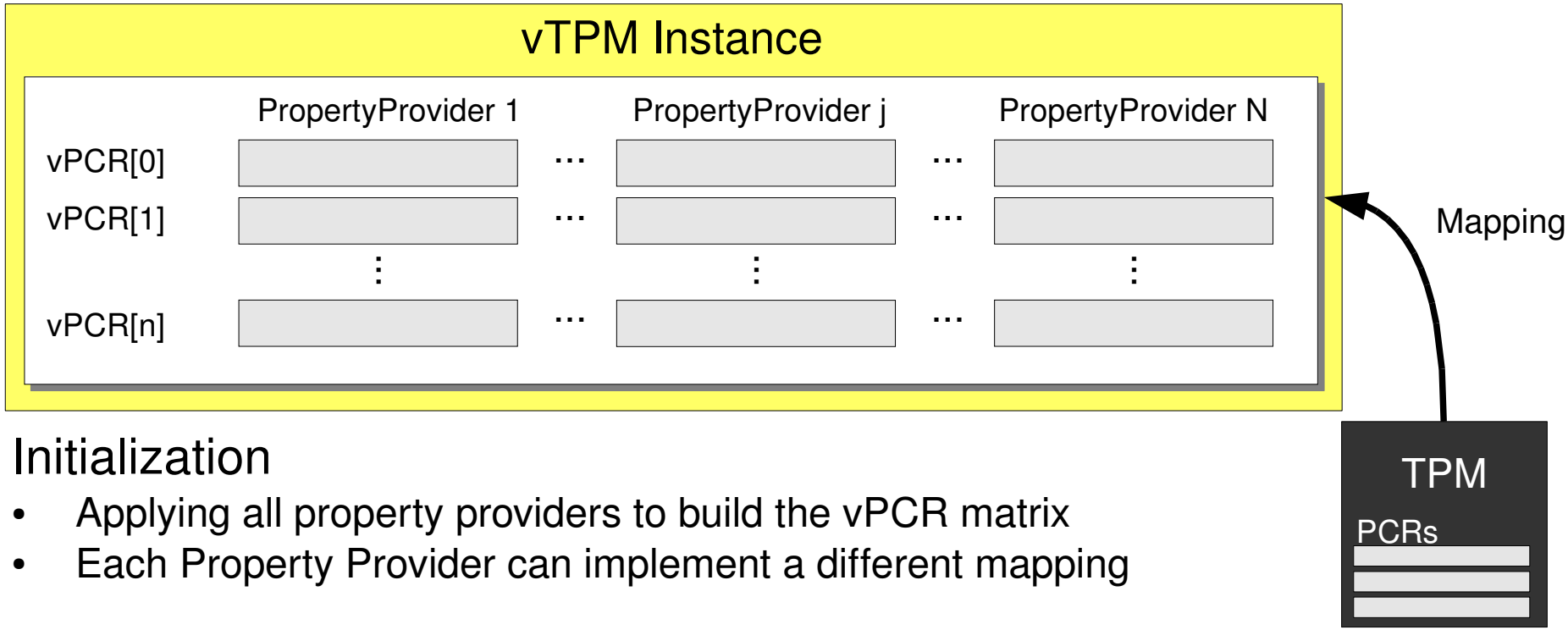
- Adding new components to internal vTPM design:
- Property Management
  - Representation of virtual PCRs
  - Different mechanisms to store and read values
  - Realizing property-based attestation and sealing
- Key Management
  - Creating and loading cryptographic keys
  - Supports software keys or keys of physical TPM
- vTPM Policy
  - User-defined policy of the vTPM instance

Flexible vTPM Architecture



Property Providers

- Each property provider has its own PCR vector
  - How to store values is up to each implementation
  - This results in a matrix of vPCRs
  - vTPM Policy decides which vector to use on which operation



  - Initialization
    - Applying all property providers to build the vPCR matrix
    - Each Property Provider can implement a different mapping

Changing the Measurement Function

- PCR extension function of the TPM:

  $Extend(i, m): PCR_i \leftarrow SHA1(PCR_i \,||\, m)$

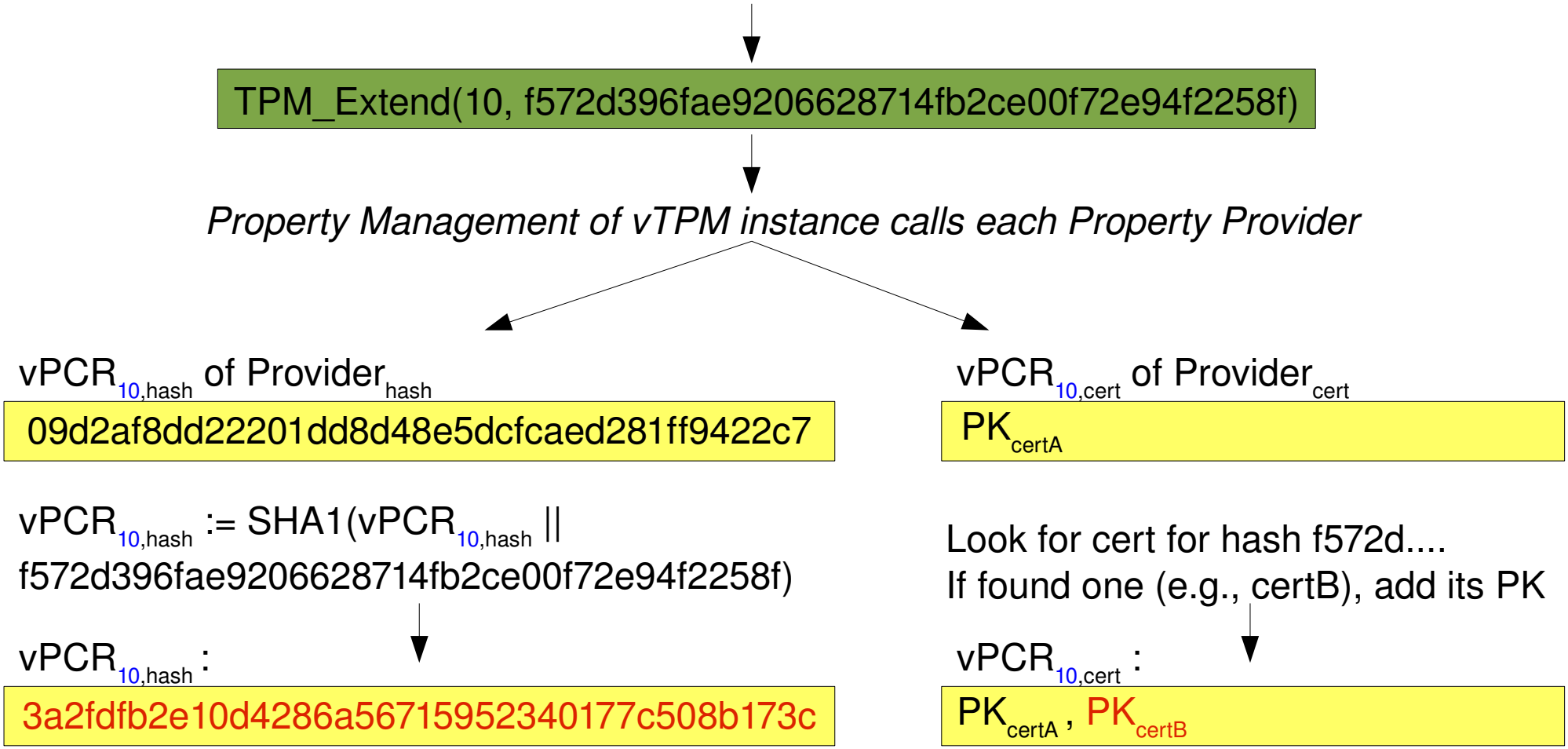
- Generalizing this for each $Provider_j$:

  $Provider_j.Extend(i,m): vPCR_{i,j} \leftarrow translate_j(vPCR_{i,j}, m)$

- Examples:
  - $translate_{hash}()$ is hashing like in hardware TPM
  - $translate_{cert}()$ looks for a certificate and stores the public key

PCR Extension: Example

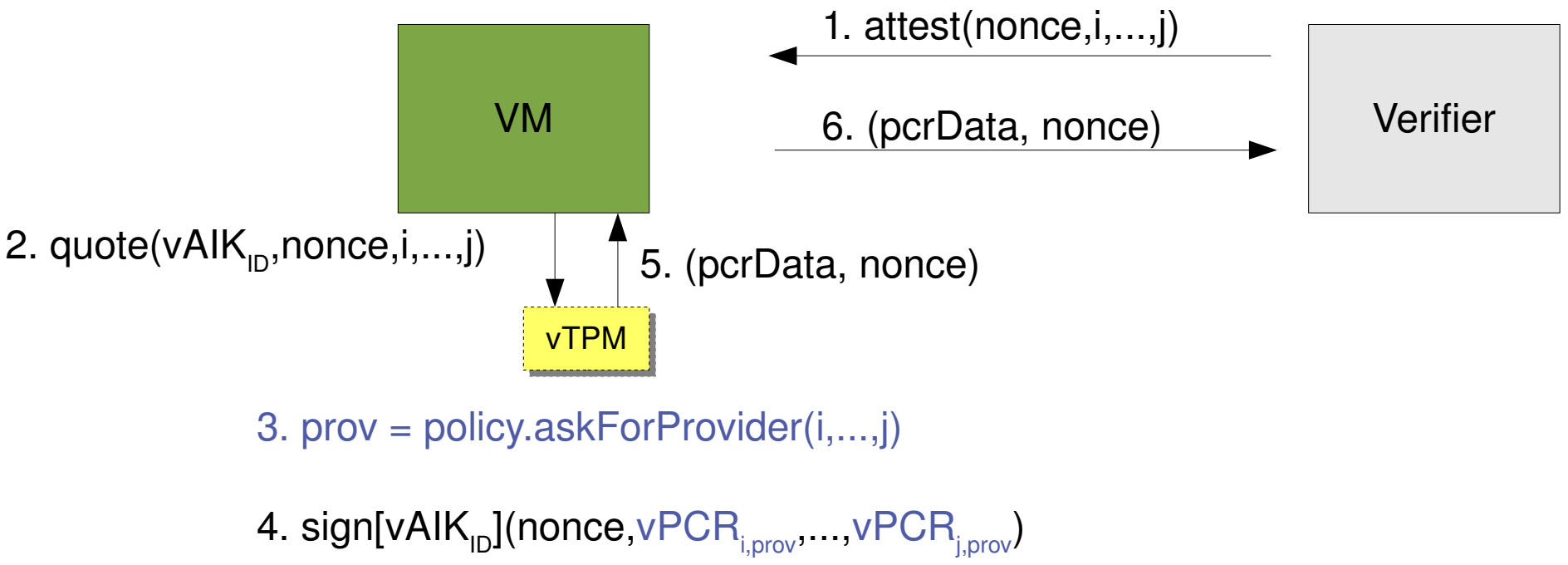
VM-OS measures a file and wants to extend the measurement in PCR 10 of the vTPM



Property-Based Attestation with vTPM

Horst Görtz Institute for IT-Security

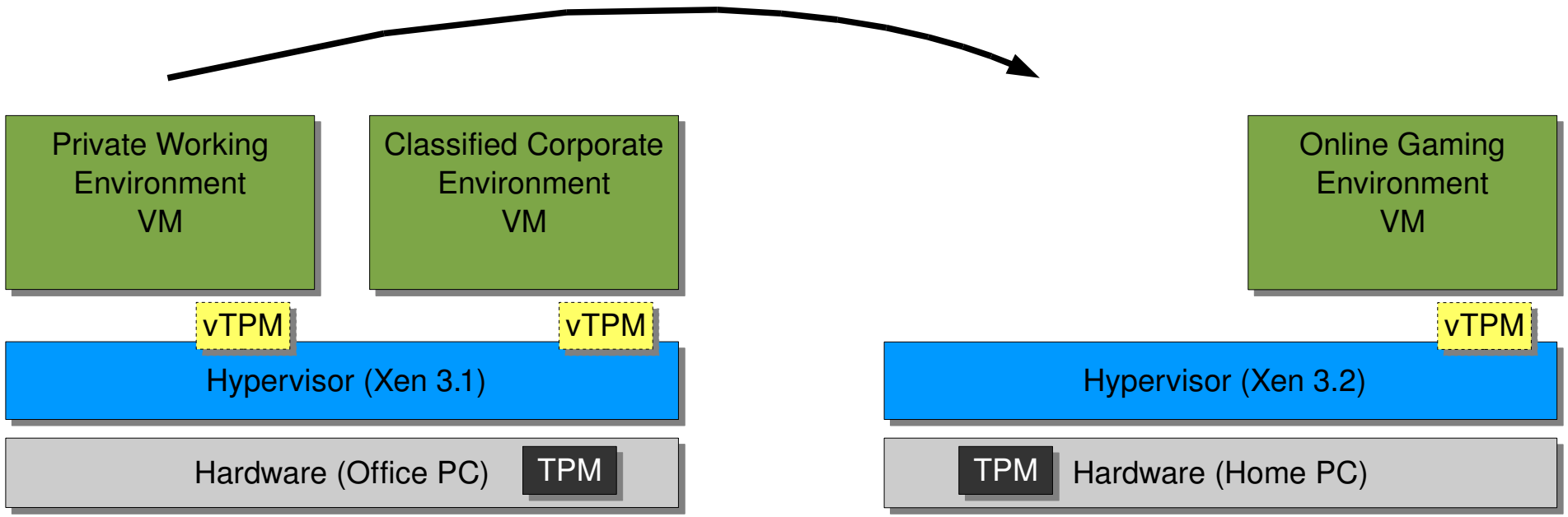
- Provider$_{cert}$ is one example to use property certificates
  - Certificates describe the properties for a particular measurement
  - Issued by a Trusted Third Party



Migration of VM and vTPM

Horst Görtz Institute for IT-Security

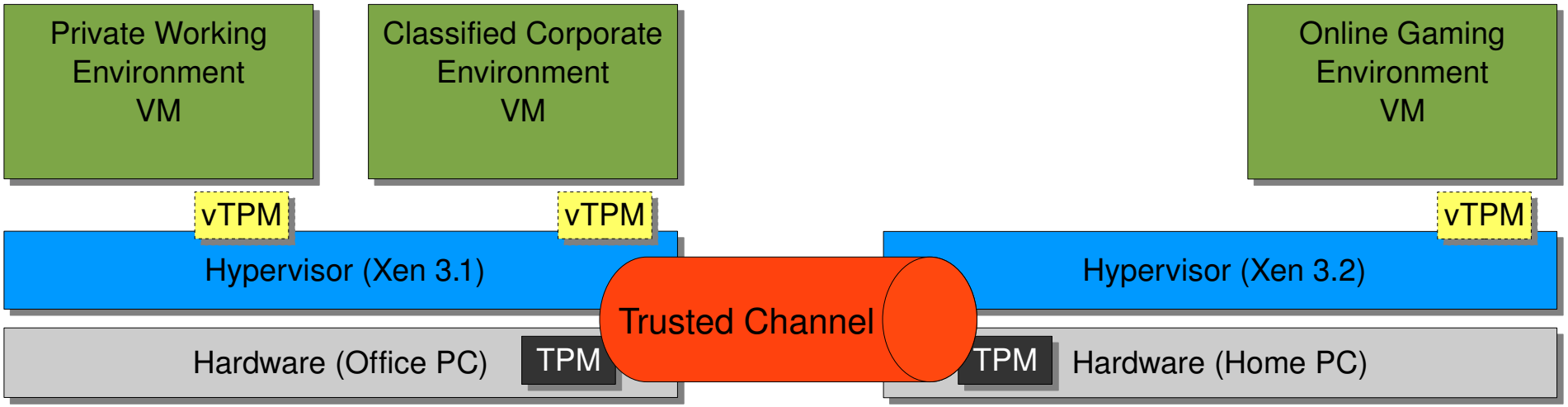
- Secure migration needed (confidentiality, integrity, authenticity)
- Example: move private working environment to home PC



Trusted Channel based Migration

Horst Görtz Institute for IT-Security

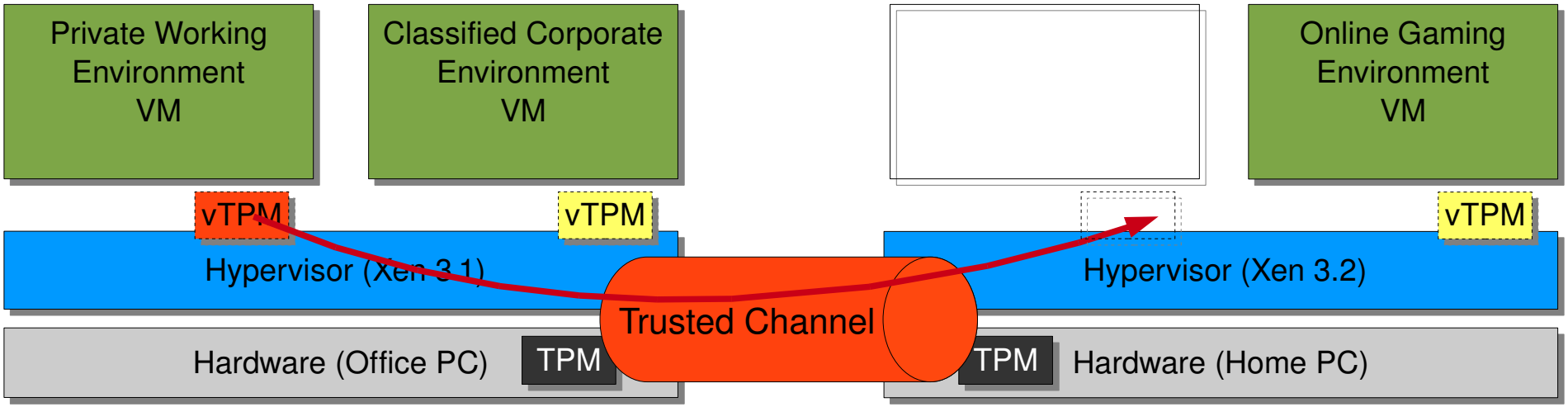
- Source platform requests trusted channel to destination
  - Creates secret encryption key bound to TPM and configuration of destination platform (assurance about integrity of end points)
  - Configuration can also be property-based
  - Re-usable for several migrations



Trusted Channel based Migration

Horst Görtz Institute for IT-Security

- Source platform requests trusted channel to destination
  - Creates secret encryption key bound to TPM and configuration of destination platform (assurance about integrity of end points)
  - Configuration can also be property-based
  - Re-usable for several migrations

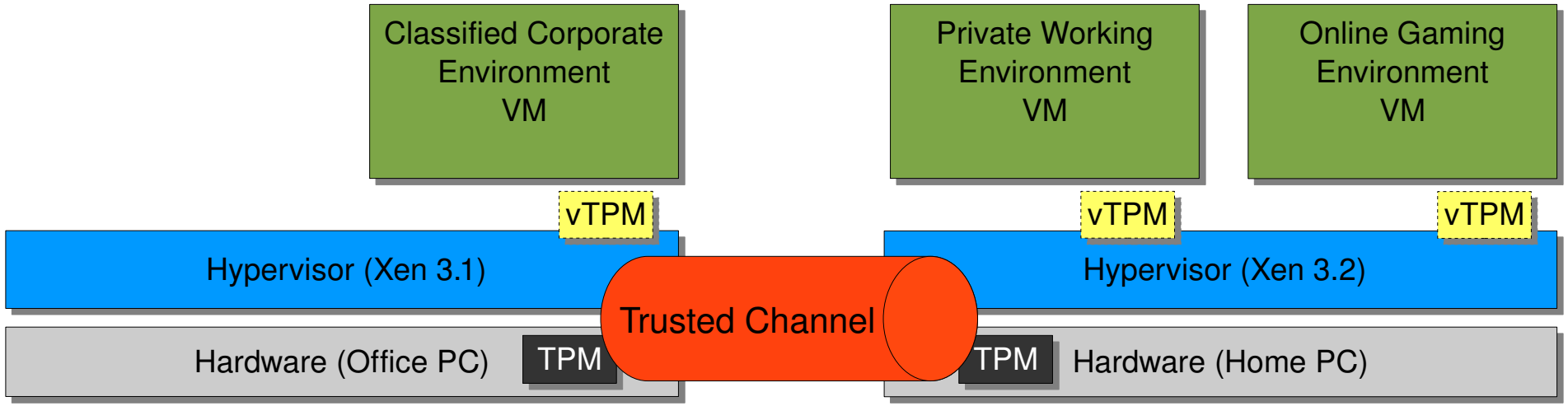


Transfer encrypted TPM state via Trusted Channel
No re-mapping of PCRs necessary (because of property providers)

Trusted Channel based Migration

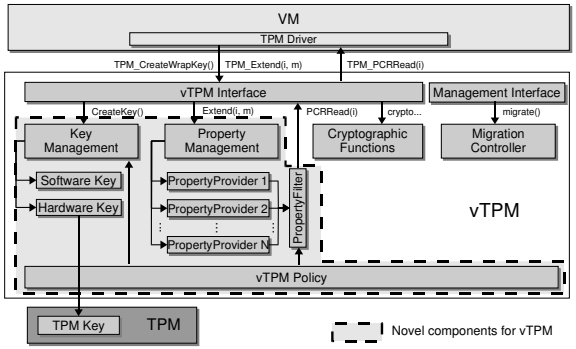
Horst Görtz Institute for IT-Security

- Source platform requests trusted channel to destination
  - Creates secret encryption key bound to TPM and configuration of destination platform (assurance about integrity of end points)
  - Configuration can also be property-based
  - Re-usable for several migrations



Transfer encrypted TPM state via Trusted Channel
No re-mapping of PCRs necessary (because of property providers)

Summary



New vTPM Design

- Property Providers
- Key Management
- vTPM Policy

- Allows to link hypervisor to vTPM based on properties
  - Availability of sealed data after migration or software updates
  - Trusted Migration protocol ensures binding to trustworthy platform
- More flexibility in key usage
  - Key Management can delegate key requests to hardware TPM
- User-defined policy decides which information to reveal
  - Policy defines which Property Provider to use on attestation

Thank you for your attention!

Questions?

Contact:

Marcel Winandy
Horst Görtz Institute for IT Security
Ruhr-University Bochum, Germany
marcel.winandy@trust.rub.de

BACKUP

Property-Based Sealing

$vTPM_k$.Seal($vBindkeyID$, [i,...,j], $data$):
provider := vTPMPolicy.askForProvider([i,...,j]);
FOR $l := i$ TO j DO $prop_l$:= provider.PCRRead(l);
pk := KeyManagement.getPublicKey($vBindkeyID$);
ed := encrypt[pk]($i||prop_i||...||j||prop_j||data$);
return ed.

$vTPM_k$.UnSeal($vBindkeyID, ed$):
(sk, pk) := KeyManagement.getKeyPair($vBindkeyID$);
$(i||prop_i||...||j||prop_j||data)$:= decrypt[sk](ed);
provider := vTPMPolicy.askForProvider([i,...,j]);
FOR $l := i$ TO j DO BEGIN
  $prop'_l$:= provider.PCRRead(l);
  if ($prop'_l \neq prop_l$) return $\emptyset$;
END
return $data$.

Migration Protocol