Practical Aspects of Modern Cryptography

Winter 2011

Josh Benaloh
Brian LaMacchia

Agenda

- Guest lecture: Tolga Acar, *Distributed Key Management and Cryptographic Agility*
- Hardware crypto tokens
  - Smart cards
  - TPMs (v1.2 & ".Next") – tokens for PCs
- Virtualization and virtualized crypto tokens

Agenda

- Guest lecture: Tolga Acar, *Distributed Key Management and Cryptographic Agility*
- Hardware crypto tokens
  - Smart cards
  - TPMs (v1.2 & ".Next") – tokens for PCs
- Virtualization and virtualized crypto tokens

Slide Acknowledgements

- Some of these slides are based on slides created by the following folks at MS:
  - Shon Eizenhoefer
  - Paul England
  - Himanshu Raj
  - David Wootten

Agenda

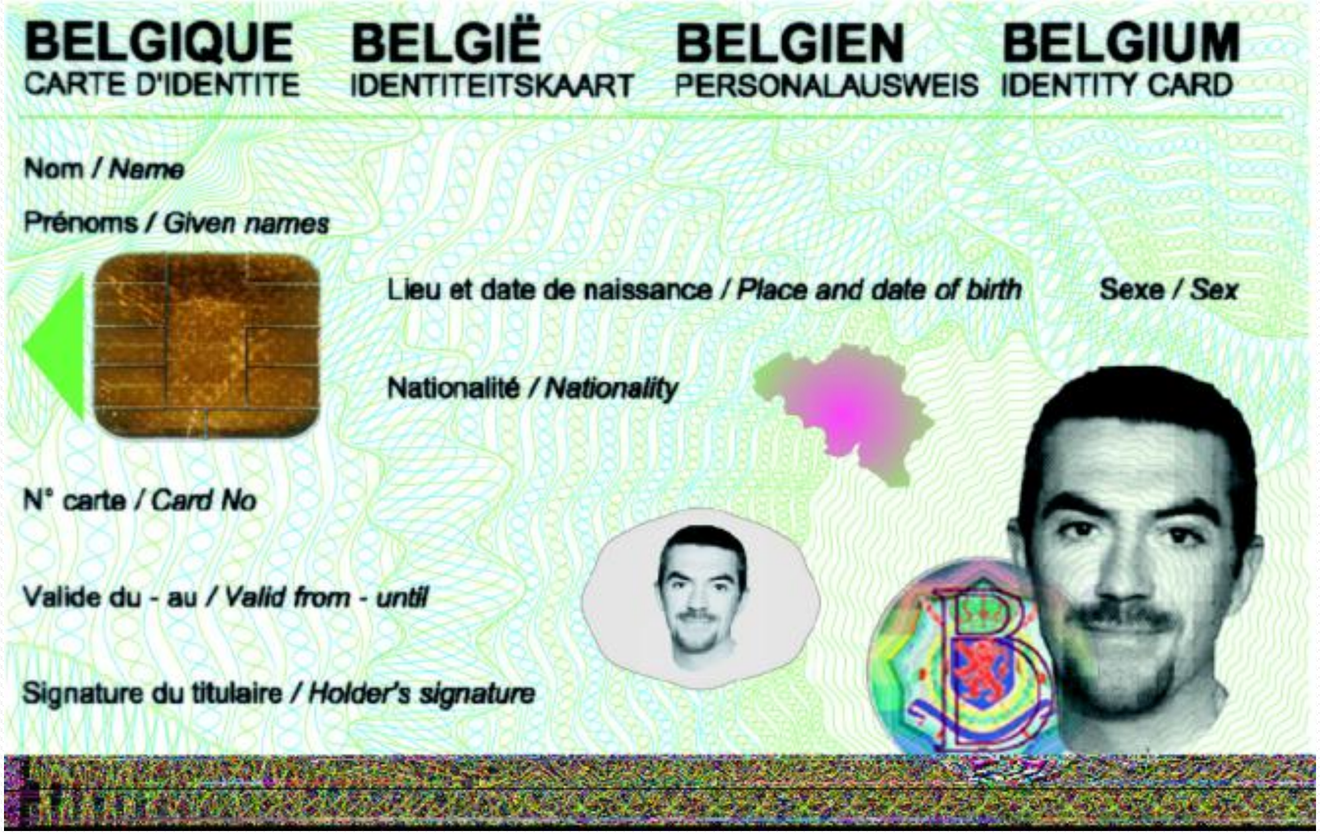
- Guest lecture: Tolga Acar, *Distributed Key Management and Cryptographic Agility*
- Hardware crypto tokens
  - Smart cards
  - TPMs (v1.2 & ".Next") – tokens for PCs
- Virtualization and virtualized crypto tokens

What is a smart card?

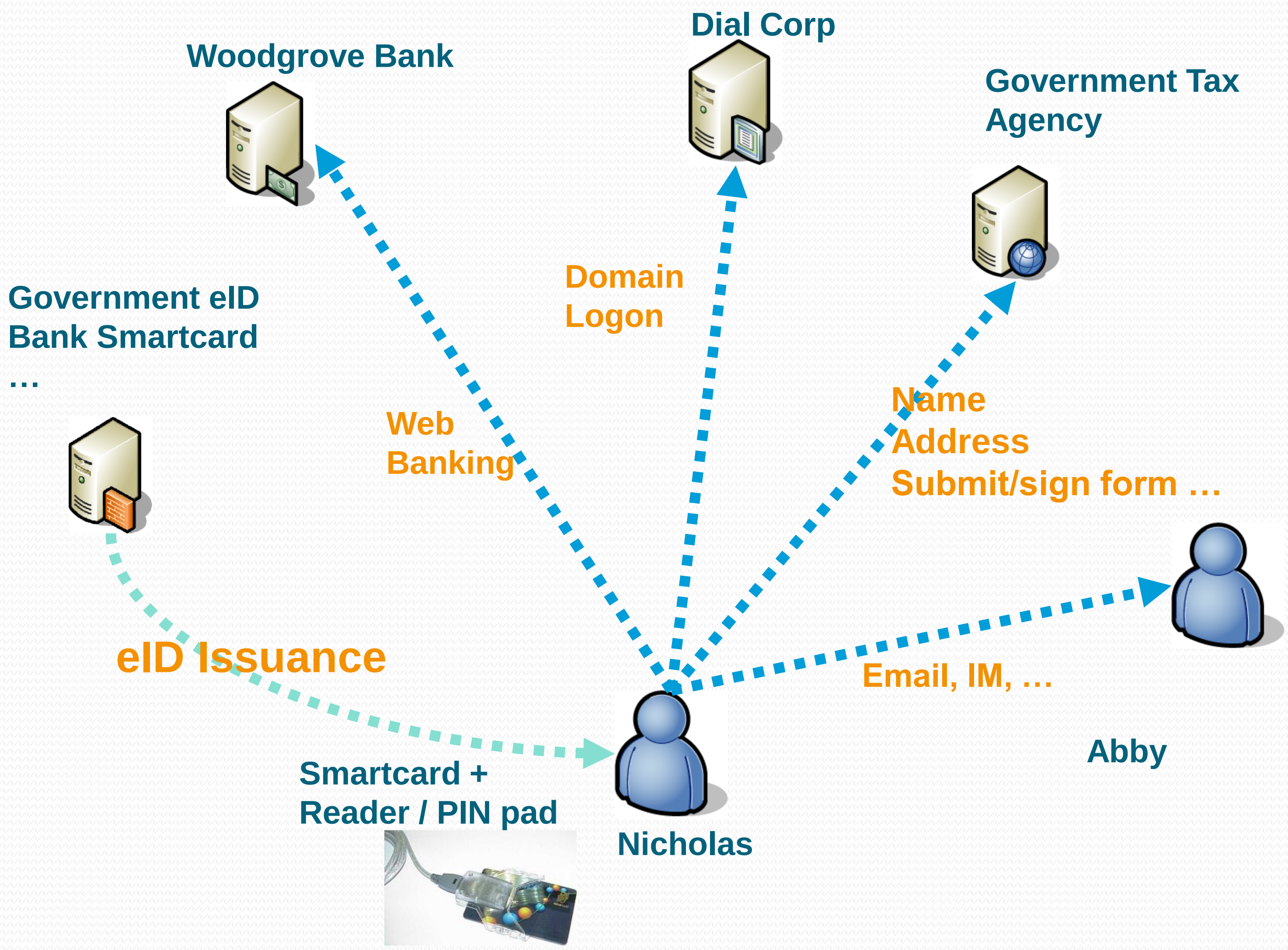
- Long history, invented in the 1970s
- Integrated Circuit Cards - ICC
- Initially used for pay phone systems in France
- Most successful deployment: GSM cell phones
- Payment: EMV – Europay, MasterCard and VISA
- Strong User Authentication. Some examples:
  - National eID programs in Asia and Europe
  - DoD CAC cards

Benefits of smart cards

- Provides secure storage for private keys & data
  - Tamper resistant
  - Cryptographically secure
- Provides two factor authentication
  - Something you have – The Card
  - Something you know – The PIN (Also referred to as Card Holder Verification- CHV)
- Programmable cards
  - Ex.: JavaCards, .NET Cards


BELGIQUE
CARTE D'IDENTITE
BELGIË
IDENTITEITSKAART
BELGIEN
PERSONALAUSWEIS
BELGIUM
IDENTITY CARD
Nom / Name
Prénoms / Given names
Lieu et date de naissance / Place and date of birth
Sexe / Sex
Nationalité / Nationality
N° carte / Card No
Valide du - au / Valid from - until
Signature du titulaire / Holder's signature

Possible eID scenarios



Stages in a Smart Card’s Life Cycle

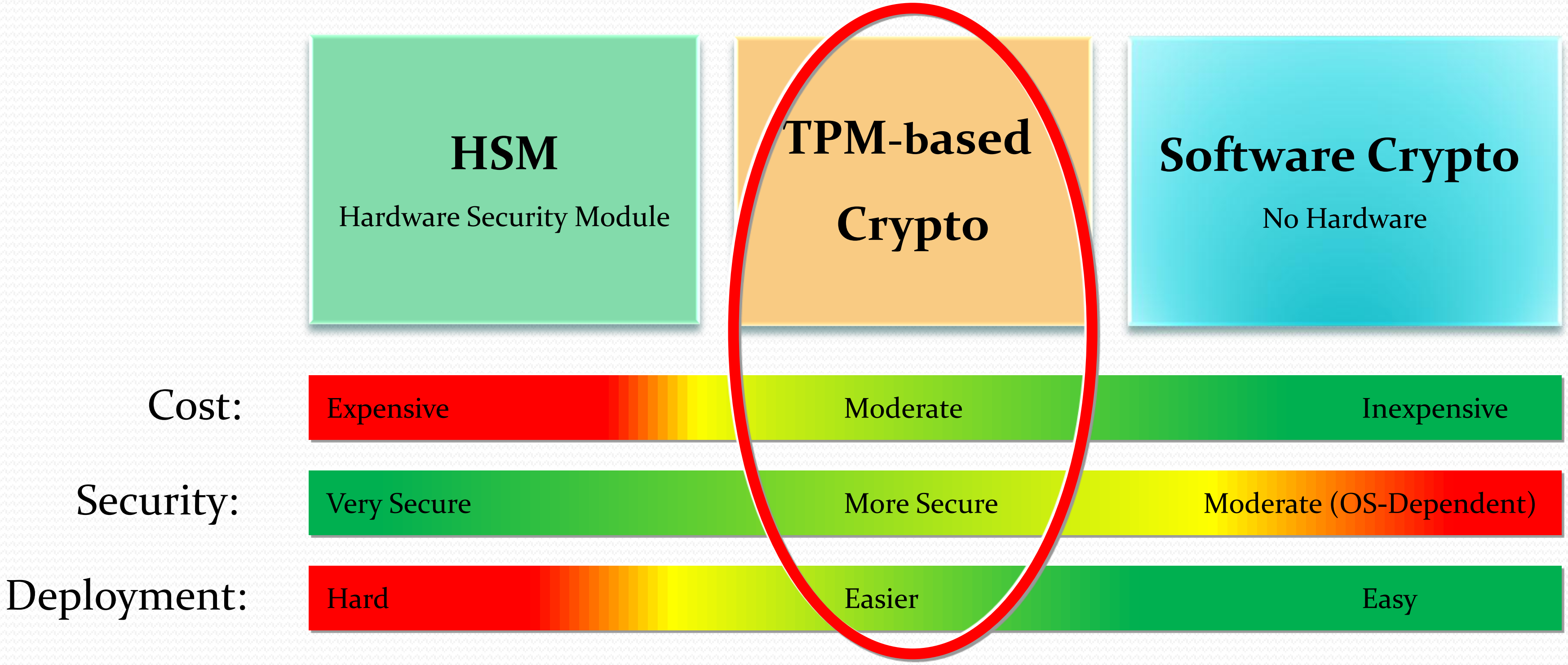
- Initial Issuance
- PIN unblock
- Renewal
- Retirement
- Revocation
- Forgotten Smart Card

Agenda

- Guest lecture: Tolga Acar, *Distributed Key Management and Cryptographic Agility*
- Hardware crypto tokens
  - Smart cards
  - TPMs (v1.2 & ".Next") – tokens for PCs
- Virtualization and virtualized crypto tokens

Recall DKM-TPM Motivation from Tolga's talk:

Secret Protection Technology:

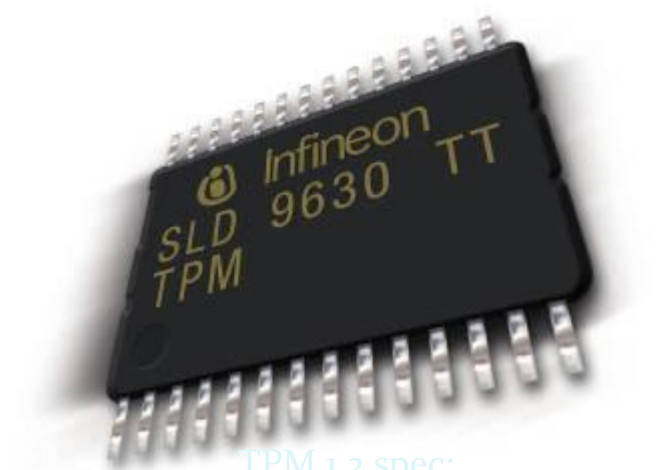


- Approach sits between a pure HSM solution and a full software solution.

What Is A Trusted Platform Module (TPM)?

Smartcard-like module on the motherboard

- Protects secrets
- Performs cryptographic functions
  - RSA, SHA-1, RNG
- Performs digital signature operations
- Anchors chain of trust for keys and credentials
- Protects itself against attacks
- Holds Platform Measurements (hashes)
- Can create, store and manage keys
  - Provides a unique Endorsement Key (EK)
  - Provides a unique Storage Root Key (SRK)



TPM 1.2 spec:
www.trustedcomputinggroup.org

TPM v1.2 Key Features

- Platform measurements
  - TPM can “measure” (hash w/ SHA-1) instruction sequences & store the results in “platform configuration registers” (PCRs)
- Encryption
  - TPM can encrypt arbitrary data using TPM keys (or keys protected by TPM keys)
- Sealed Storage
  - TPM can encrypt arbitrary data, using TPM keys (or keys protected by TPM keys) and *under a set of PCR values*
  - Data can only be decrypted later under the same PCR configuration
- Attestation (in a moment)

Sealed Storage

- Why is Sealed Storage useful?
- Provides a mechanism for defending against boot-time attacks
- Example: Full Volume Encryption (FVE)
  - BitLocker™ Drive Encryption on Windows

Information Protection Threats

Internal threats are just as prevalent as external threats

Accidental

Loss due to carelessness

- System disposal or repurposing without data wipe
- System physically lost in transit

Intentional

Data intentionally compromised

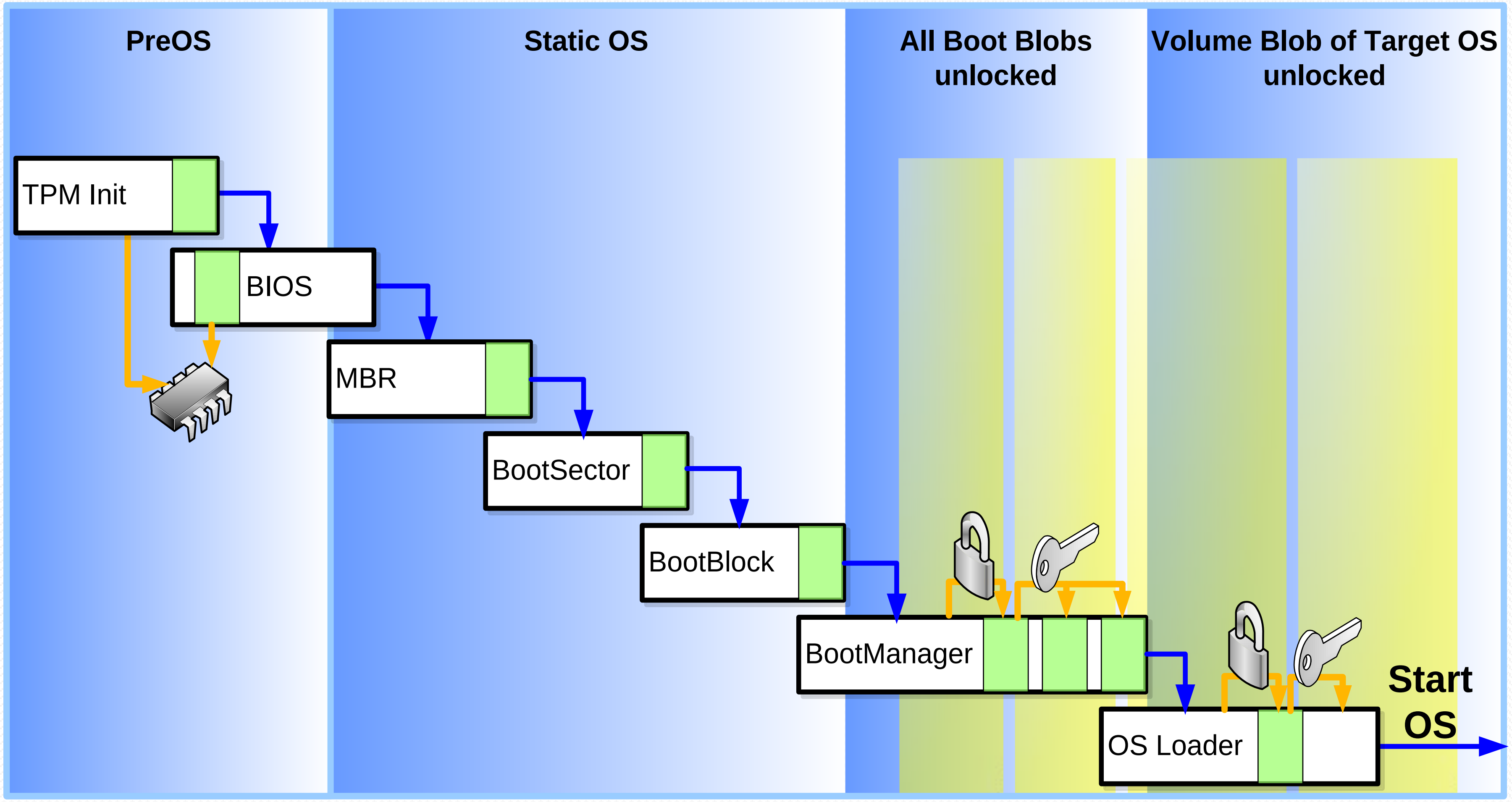
- Insider access to unauthorized data
- Offline attack on lost/stolen laptop

Targeted

Thief steals asset based on value of data

- Theft of branch office server (high value and volume of data)
- Theft of executive or government laptop
- Direct attacks with specialized hardware

Booting w/ TPM measurements



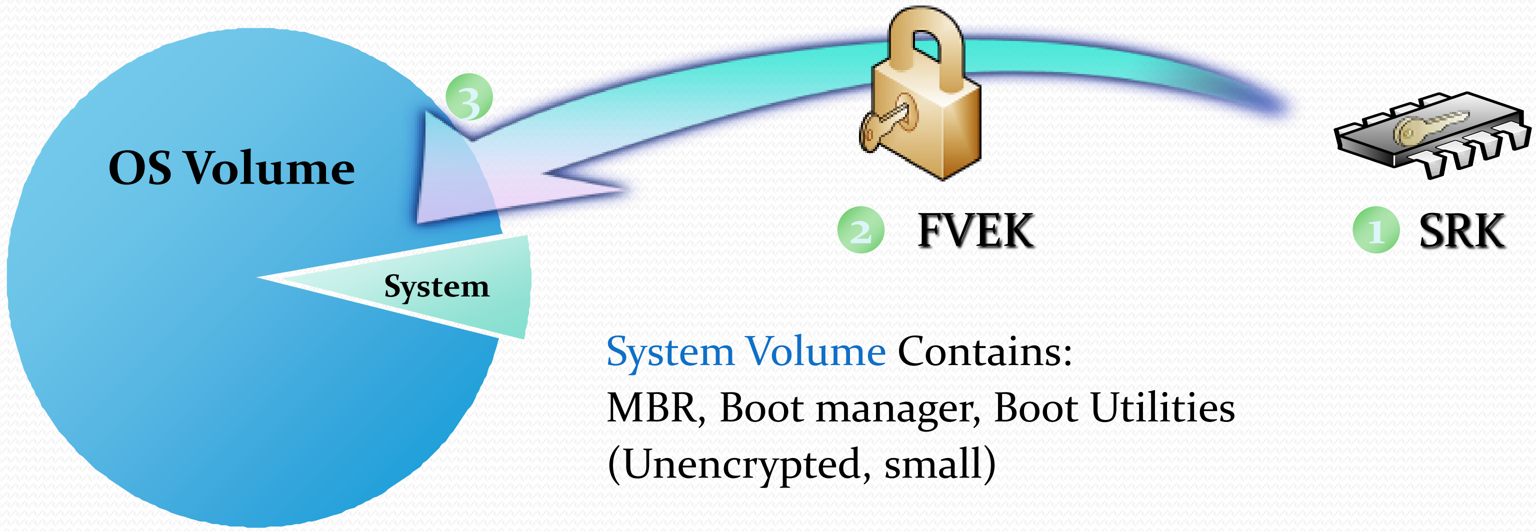
Disk Layout And Key Storage

OS Volume Contains

- Encrypted OS
- Encrypted Page File
- Encrypted Temp Files
- Encrypted Data
- Encrypted Hibernation File

Where's the Encryption Key?

1. SRK (Storage Root Key) contained in TPM
2. SRK encrypts FVEK (Full Volume Encryption Key) protected by TPM/PIN/USB Storage Device
3. FVEK stored (encrypted by SRK) on hard drive in the OS Volume



System Volume Contains:
MBR, Boot manager, Boot Utilities
(Unencrypted, small)

Attestation

- Sealed Storage lets a TPM *encrypt* data to a specific set (or subset) of PCR values
- Attestation is an authentication technology
  - But more than “simple signing”
- Attestation allows a TPM to *sign* data and a set (or subset) of the current PCR values
  - So the TPM “attests” to a certain software configuration (whatever was measured into those PCR registers) as part of its digital signature
  - “Quoting”

Key Recovery Scenarios

- Lost/Forgotten Authentication Methods
  - Lost USB key, user forgets PIN
- Upgrade to Core Files
  - Unanticipated change to pre-OS files (BIOS upgrade, etc…)
- Broken Hardware
  - Hard drive moved to a new system
- Deliberate Attack
  - Modified or missing pre-OS files (Hacked BIOS, MBR, etc…)

TPM.Next

- The TPM architecture after TPM v1.2
- More than 3 years of specification development
- Current work on TPM.Next is happening within the Trusted Computing Group (TCG) consortium
- The actual TPM.Next specification is currently confidential
  - The only publicly available information is not very technical
- I can talk about things that Microsoft has submitted to the TCG
  - But this may or may not show up in TPM.Next

Cryptographic Algorithm Agility

- TPM 1.2 is based on RSA 2048-bit and SHA-1 with little variability possible.
- SHA-1 is being phased out.
- Support for new asymmetric algorithms (ECC) is needed in some important markets.
- Requirements to be able to support localization.
- Can’t react quickly to a broken algorithm.

Potential Solutions

- Every use of a cryptographic algorithm should allow the TPM user to specify the algorithm to be used.
  - Much wider range of algorithm options while maintaining interface compatibility
- Every data structure should be tagged to indicate the algorithms used to construct it.
  - No assumptions required or allowed.
- Define sets of algorithms for interoperability.
  - Set is a combination of asymmetric, symmetric, and hash algorithms.
- Allow multiple sets to be used simultaneously.
  - Support different security and localization needs.
  - Make it easy to replace broken algorithms without having to develop an entirely new specification or product.

TPM Management

- User has a difficult time understanding the TPM controls.
  - What is the difference between TPM enable and activate?
- Security and privacy functions use the same controls.
  - Need to take ownership of TPM to use the Storage Root Key but that also enables Endorsement Key operations which are privacy sensitive.
- PCR sealing model is brittle.
  - Makes it difficult to manage keys that rely on PCR values.
  - System updates that change a PCR measurement can be very disruptive.

PCR “Brittleness”

- Many configuration changes leading to PCR changes are benign
  - But still result in keys becoming unusable, etc.
- *Sometimes* if you plan ahead you can prevent this
  - E.g. *seal* to a future known good configuration
- *Sometimes* we can fix this with smarter external software
  - E.g. *extend* hashes of authorized signing keys and check certificates
- But it’s caused enough problems that TPM support makes sense

Potential Solutions

- Change to simpler model for control – on/off
- Should split controls.
  - Security functions based on Storage Root Key – default on
  - Identity/privacy functions based on Endorsement Key – default off
  - Provisioning functions based on BIOS controls – always on
- Allow a recognized authority to approve different PCR settings.
  - An authority over the PCR environment in which the key may be used much like migration authority controls the hierarchy in which a key may be used.

Ecosystem Issues

- TPM/TCM are not interchangeable.
  - No BIOS level abstraction for a security token (TPM/TCM) as there is for a disk (read/write logical blocks).
  - Makes it hard to adopt boot code for alternative algorithms.
- Trusted computing crosses national boundaries.
  - Neither the TPM nor the TCM has the ability to meet both local and international cryptographic requirements at the same time.
- The sunset of SHA1 has demonstrated the importance of not being tied to a fixed set of algorithms.
  - It will be a major upset to the ecosystem (chip, system, software) to switch to a new TPM with a new software interface.
- Changing the TPM algorithms is going to cause a major upheaval in the ecosystem.

Potential Solutions

- TPM.next should have an interface that is not tied to a specific set of algorithms.
  - Boot code can use the BIOS interface without being aware of the underlying cryptographic algorithms.
  - Makes for a better abstraction.
- TPM.next should allow multiple sets of algorithms to co-exist at the same time on the same TPM.
  - Give the ability simultaneously to support both local and international standards.
- TPM.next should allow new algorithms to be introduced as needed without having to re-specify the interface.
  - Avoid future upset of the ecosystem when an algorithm is broken or better algorithms are needed.

Summary

- TPM.next tries to keep the best ideas of the TPM and incorporate the best ideas from the TCM.
- TPM.next tries to improve the sub-optimal parts of the TPM and TCM especially with respect to algorithm flexibility.
- TPM.next is intended to be an international standard that can address local requirements while maintaining software compatibility over a broad range of applications.
- Please join with TCG to create a TPM.next design which will satisfy both China-market and international requirements through a single unified world-wide standard.

Agenda

- Guest lecture: Tolga Acar, *Distributed Key Management and Cryptographic Agility*
- Hardware crypto tokens
  - Smart cards
  - TPMs (v1.2 & ".Next") – tokens for PCs
- Virtualization and virtualized crypto tokens

Virtualization

- Sharing a single physical platform among multiple virtual machines (VMs) with *complete isolation* among VMs
- Benefits
  - Consolidation of workloads, Fault tolerance, Extensibility, Ease of Management, Better security

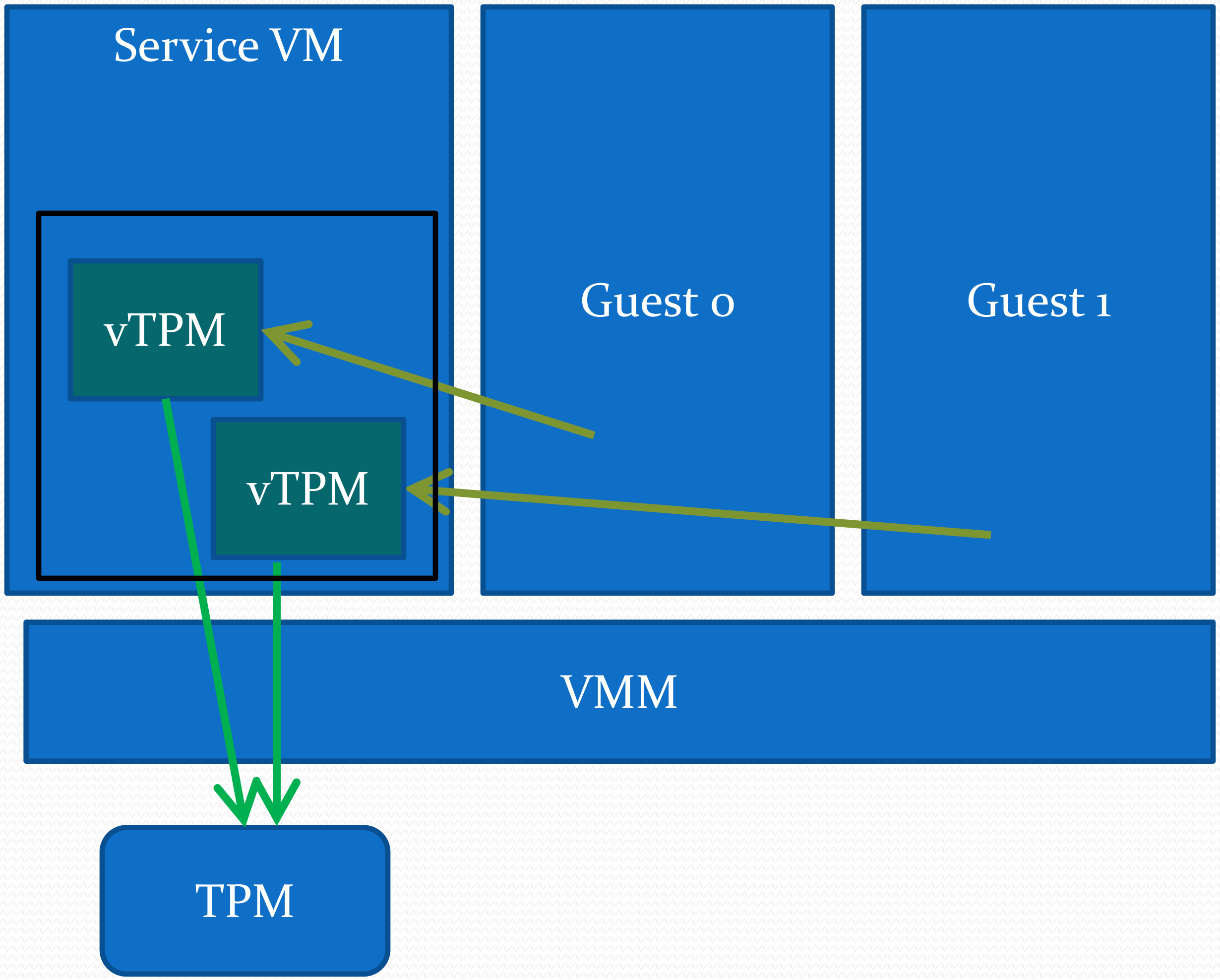
Virtualization

- With increasing h/w support, performance degradation is becoming minimal
- With multi-core, we can envision pervasive adoption
  - Solutions available for server, client, and mobile platforms
  - E.g., virtualized data centers (EC2, Azure)
  - And, Dilbert running his office VM on home computer

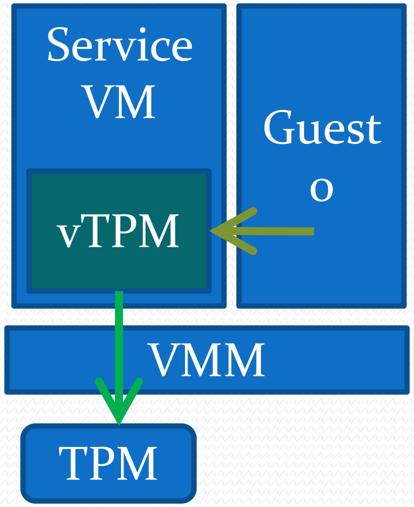
Virtual TPM

- Challenge: physical TPM itself is hard to virtualize
  - By design, TPM resists virtualization
- TPM emulation
  - Complete s/w emulation, TCG interface: vTPM [Berger06]
- Para-virtualized TPM sharing [England08]
  - Hypercall interface with Hv as mediator

vTPM



vTPM

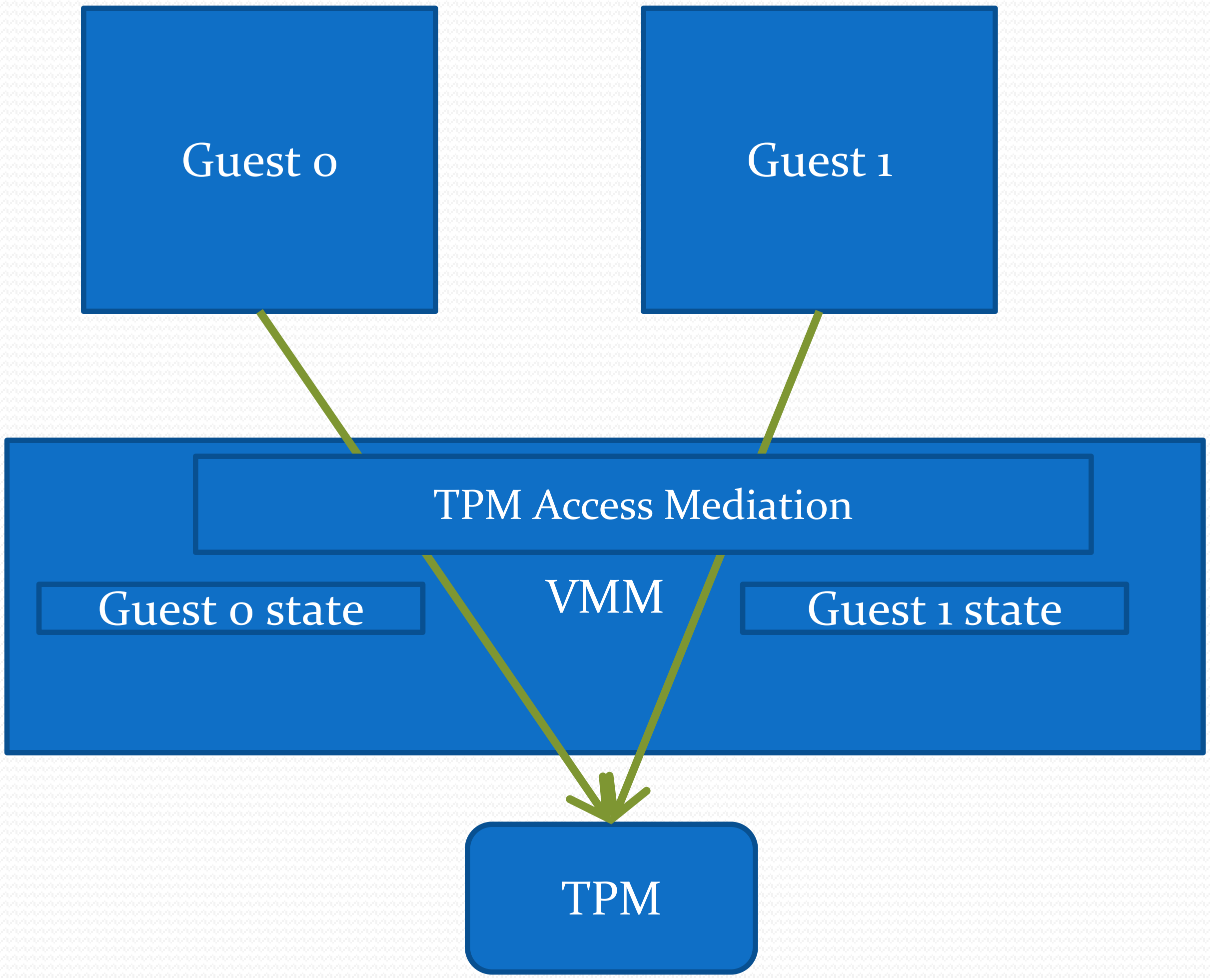


- Pros
  - Standard TCG interface
  - High fidelity: full legacy support
  - Vendors can add VM use-cases
    - Migration, suspend/resume, rollback
- Cons
  - Low resistance to physical attack
  - Reduced resistance to software attack
    - Hypervisor is more complex and exposed than TPM embedded OS
  - Trust model for TPM is complex
    - Hypervisor security model influences vTPM security

vTPM

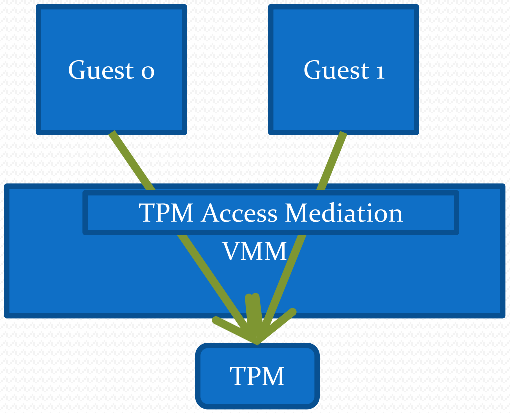
- Each vTPM has its independent key hierarchy
  - EK, SRK, AIKs …
  - May take extra precaution while storing these in memory
    - Wrapped with physical TPM's SRK?
  - Attestation using vTPM
    - In a manner similar to physical TPM
    - E.g., a signed statement using an AIK that is linked to vTPM's EK

Para-Virtualized TPM Sharing

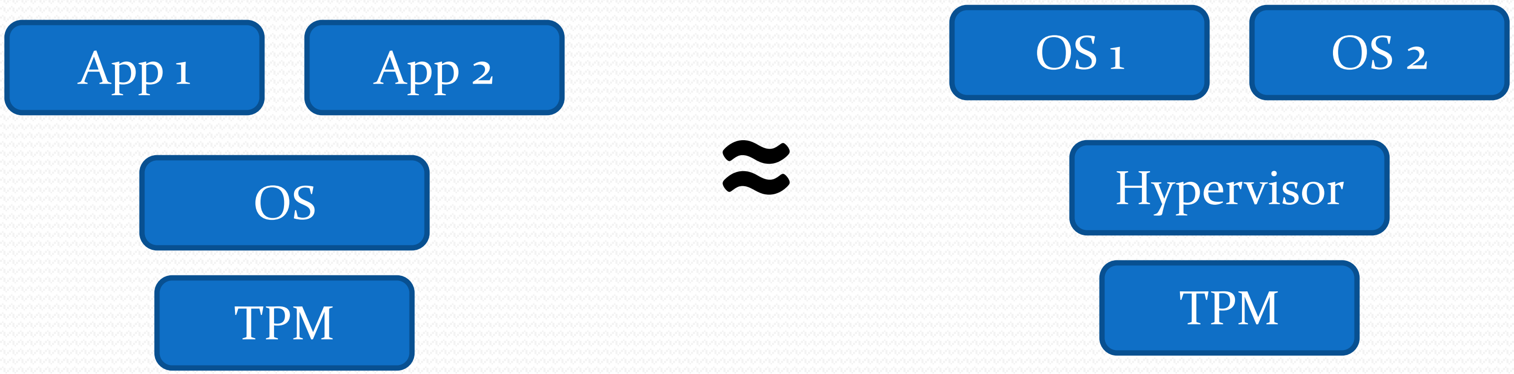


Para-Virtualized TPM Sharing

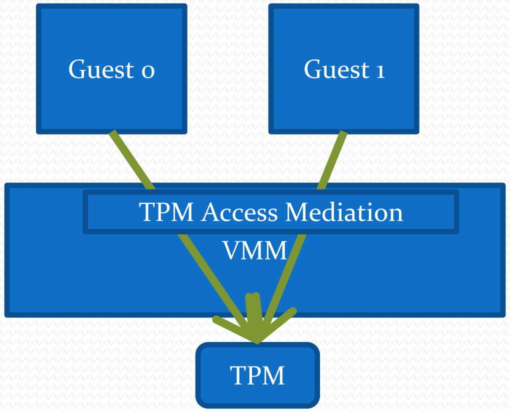
- Roll of Access Mediation Layer
  - Schedule access to TPM
  - Authenticate guests to TPM
    - Store guest measurement in resettable PCR
  - Protect Hv from guests and guests from each other
- Designed as minimal SW-stack for TPM sharing
- Minimal or no *application* changes



Important Observation



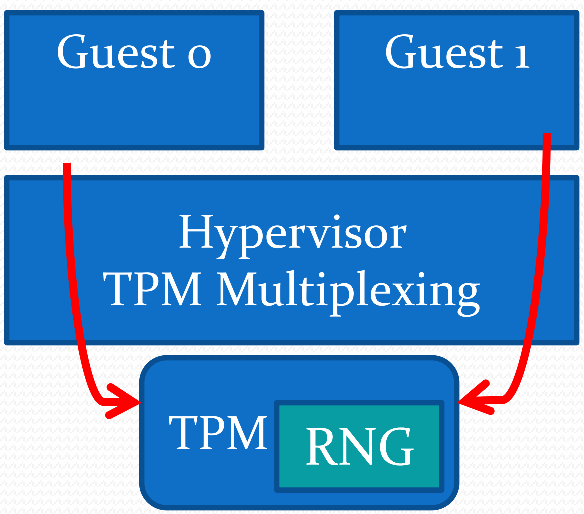
Para-Virtualized TPM



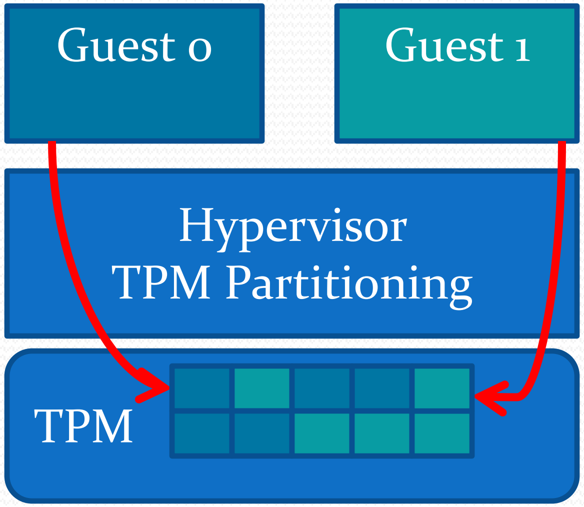
- Pros
  - Simple
  - Hardware protection for asymetric keys
- Cons
  - Requires software changes, at least at the library level
    - Hypercall based interface
    - Meaning of seal/unseal/quote
      - Which physical PCRs are mixed?
      - Ordering of vPCRs
      - Actual operation against PCR 15
  - We can only provide a "virtualization-friendly" subset of the TPM
    - similar to OS-friendly subset

Para-Virtualized TPM - Examples

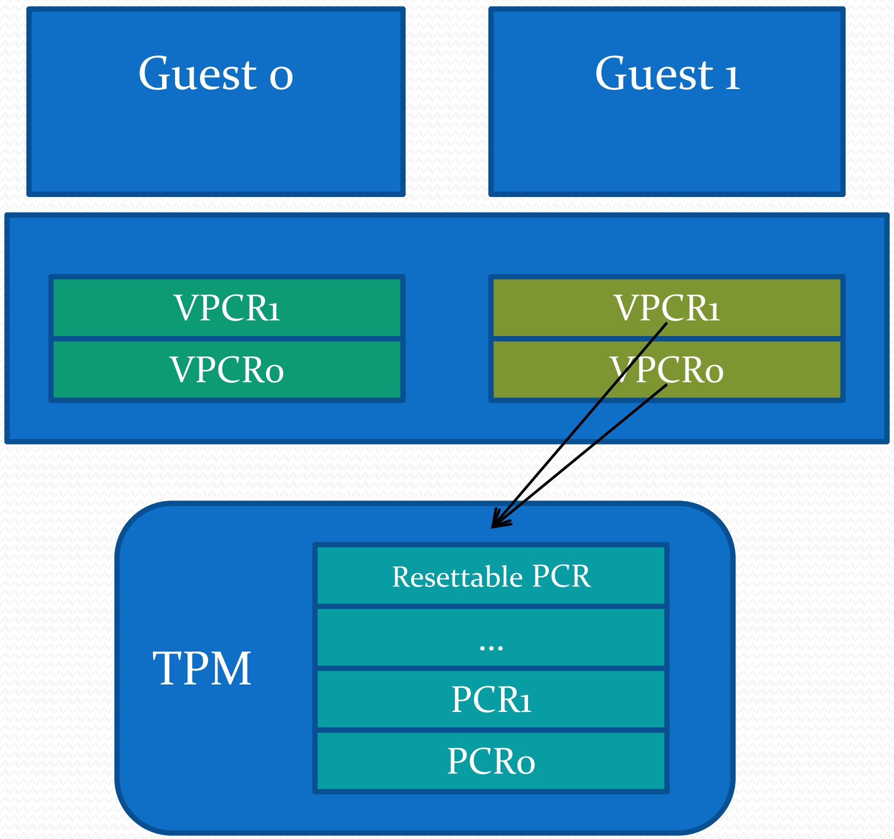
RNG



NV Storage



Para-Virtualized TPM - Attestation



HvQuote(TCB, nonce)

PcrReset(15)
PcrExtend(15,VPCR1)
Quote((0,15), nonce)

Para-Virtualized TPM

- TVP binds a VM to a physical platform
- Must re-establish the key hierarchy after migration
  - Need to signal VM about migration
  - Is this a good thing?

Backup