

Summary Of Features Under Consideration For The Next Generation Of TPM

Copyright© 2009 Trusted Computing Group

Feature Area Summary

Cryptographic Algorithm Variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic Algorithm Variability

Virtualized Systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Cryptographic Algorithm Variability – Issues

- 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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Cryptographic Algorithm Variability – 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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic Algorithm Variability

Virtualized Systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Virtualized Systems – Issues

- TPM 1.2 is designed before the identification of the requirements for virtualization were identified, and, as a consequence, the TPM 1.2 difficult to virtualize.
  - Primary use model was authentication and identity of clients
  - Virtualization was seen as an attack vector rather than a product solution.
- Market evolution over the past 7+ years.
  - TPM has found its way onto servers.
  - Virtualization is now more common on all platforms.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Virtualized Systems – Solutions

- Should enhance ability of TPM to support virtualized systems.
  - Not the same as a virtualized TPM which is just software.
- Change key hierarchy to allow a pseudo Storage Root Key (SRK) for a virtual machine.
  - Bind keys to a pseudo SRK in the same way that we bind keys to the SRK.
  - Pseudo SRK would be able to migrate under control off Virtual Machine Manager (VMM).
  - Keys would still be in hardware but they would be able to migrate under control of trusted authority.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic algorithm variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Regularity – Issues

- TPM has many special purpose keys with some limitations on many of them, such as:
  - A bound blob can't be attached to the same parent as a sealed blob.
  - A key used for attestation can't sign a certificate request.
  - A key that can sign certificate requests can't be trusted as an attestation key.
- Privacy model is not consistent.
  - In some cases we try to make the TPM responsible for privacy and in other cases we assume that the system software must be involved.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Regularity – Solutions

- Have the properties define the key type rather than having the key type define the properties.
  - Example: TPM.next should not define an AIK. Instead, it should defines a signing key that can sign anything that the TPM hashes.
- Let new operations be used uniformly on any key, when possible/sensible.
  - Example: ActivateIdentity.next() should work on any key.
- Use simple primitives instead of complex commands.
  - Example: Unseal.next() should operates on a loaded blob instead of in an atomic operation to give better consistency in blob handling.
- Address privacy in context of OS.
  - Assume that OS and TPM both want to protect the user's interests

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic algorithm variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Performance – Issue

- Loading a key hierarchy can be slow because of the asymmetric cryptography used to load keys.
  - This is aggravated because the key format was chosen for small size (*n*, *d*) rather than performance (*p*, *q*, *dP*, *dQ*, *qInv*).

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Performance – Solution

- Use symmetric encryption for the hierarchy.
  - Should follow the example of TCM and encrypt the asymmetric key data with a symmetric key.
  - Would preserve the ability to identify a node based on its public key while greatly speeding the load time.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic algorithm variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Robustness – Issues

- There are known issues when users make poor choices for authorization values.
  - An authorization value with low entropy can be guessed in a man-in-the-middle (MIM) attack
  - In some cases, the authorization value is well known and has no entropy.
- Key substitution is possible by swapping handles.
  - Key handles are not included in the authorization-keyed, Hash Message Authentication Code (HMAC). Lets someone misappropriate an authorization for use on a different key.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Robustness – Solutions

- Can protect weak authorization data by adding secret salt to session authorization HMAC.
- Can prevent key substitution by including key "Name" in the authorization HMAC.
  - Handles can still be switched but not without detection.
- Only use approved symmetric algorithms and modes.
  - All cryptographic methods will be according to recognized cryptography standards – TCG defines cryptographic applications not cryptographic algorithms.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic algorithm variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Management – Issues

- 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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Management – 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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Feature Area Summary

Cryptographic algorithm variability

Virtualized systems

Regularity

Performance

Robustness

Management

Ecosystem

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

Ecosystem Issues – 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. THIS IS VERY IMPORTANT!

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.

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.

Copyright© 2009 Trusted Computing Group – Other names and brands are properties of their respective owners.