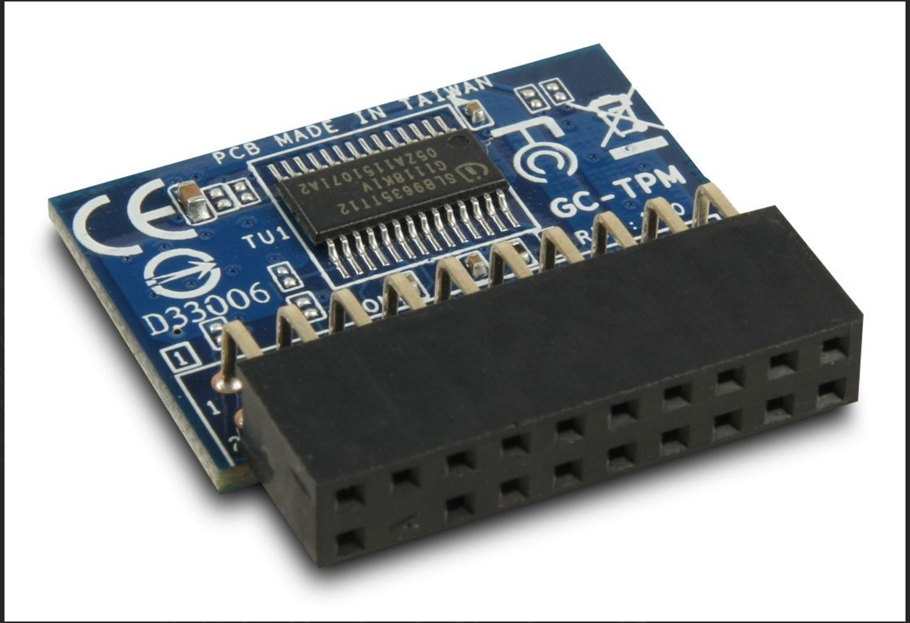
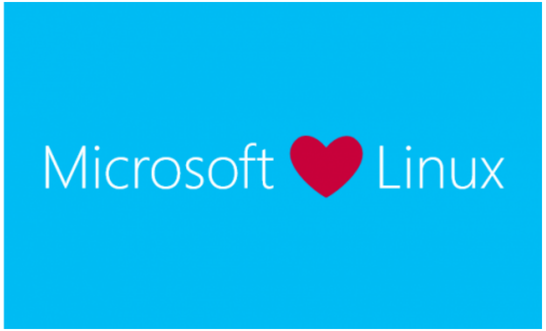
TPM update for LSS 2016



Jarkko Sakkinen <jarkko.sakkinen@linux.intel.com>

TPM in a nutshell

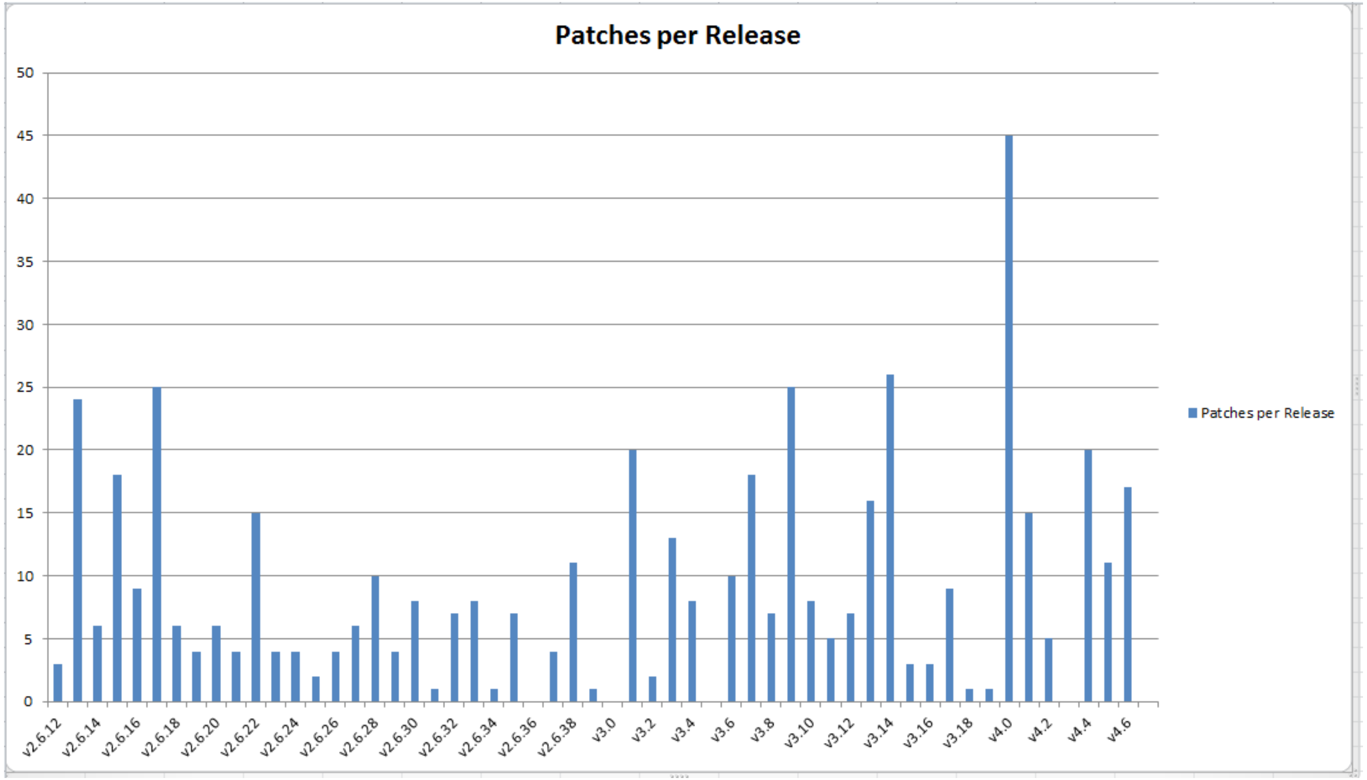
- Industry standard for cryptographic co-processors by Trusted Computing Group.
- Provides capabilities for identification, attestation, key management and storage, hashing, measurement and encryption.
- Desktop adoption increasing because of increasing security concerns.
- Cloud adaptation is taking its first steps.
- Could provide means for authentication and authorization in the IoT space?



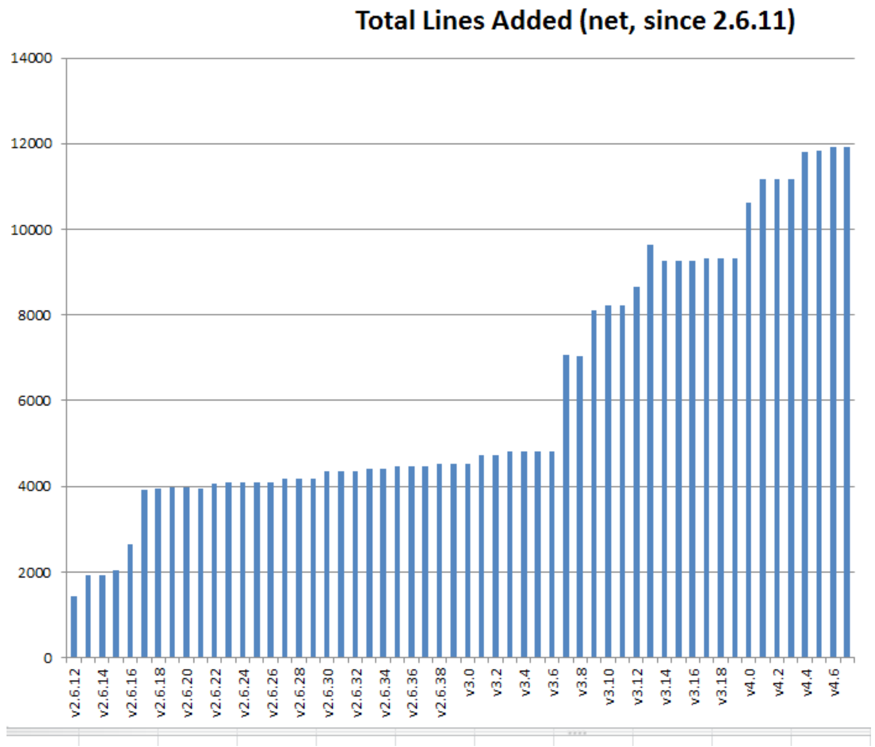
Quick recap of the TPM standard history

- The development of the TPM started in the end 90s.
- The first widely deployed version was TPM 1.1b (2003).
- TPM 1.2 (2009) brought protection from dictionary attacks and support for direct anonymous attestation.
- TPM 2.0 (2015) is the latest standard, which brings algorithmic agility, policy based authorization (logical expression of TPM conditions) and symmetric encryption.

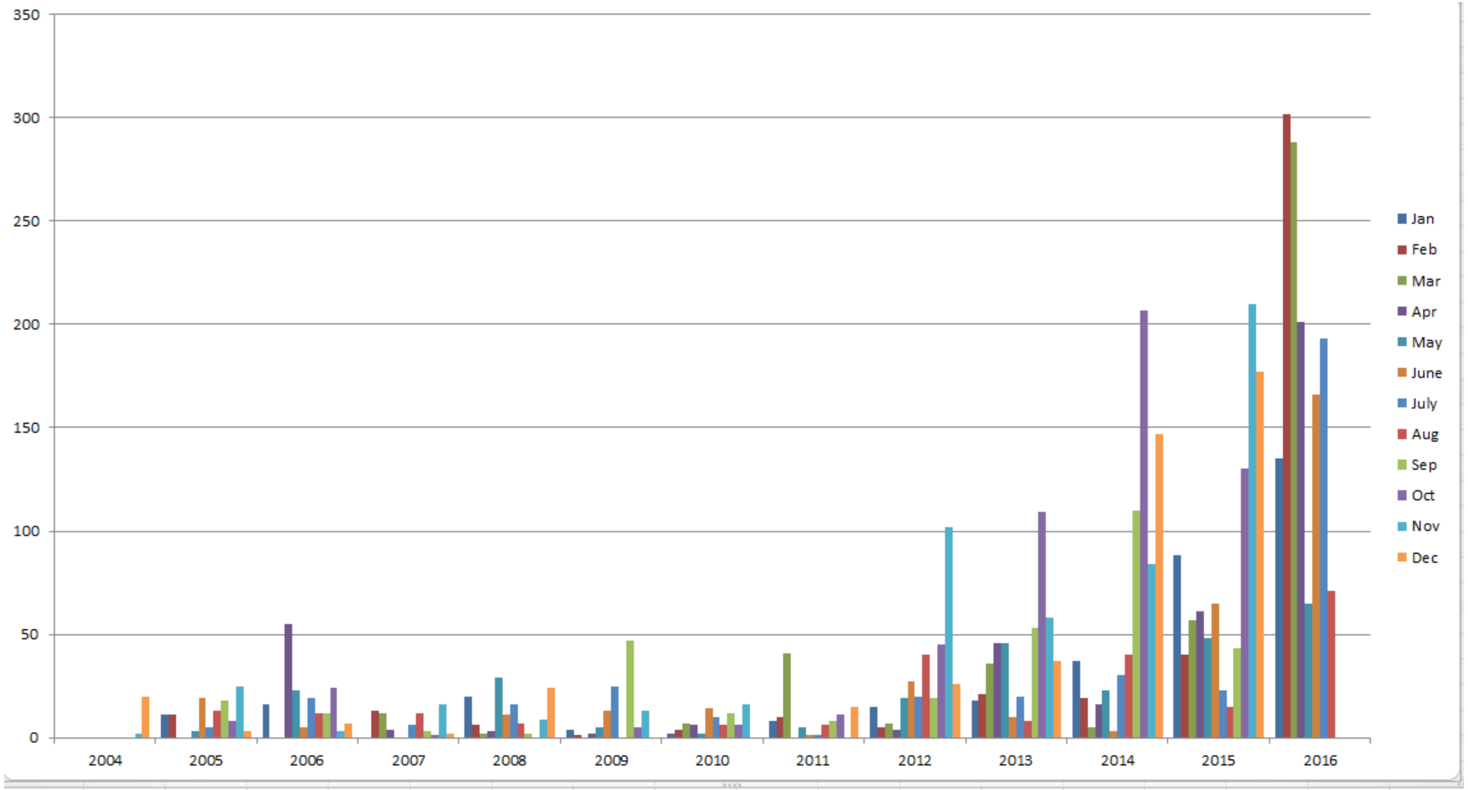
Patches per release (courtesy of Peter Hüwe)



Added lines (courtesy of Peter Hüwe)



Mailing list activity (courtesy of Peter Hüwe)



Recent (or not so recent) developments

- During last couple of years major part of time has been gone cleaning up and modernizing the subsystem.
- TPM 2.0 support including trusted keys
- Virtual TPM support (a bit like pseudo TTYs).
- Multi-backend support for tpm_tis (MMIO, SPI, I2C ready)
- New hardware support (Infineon, ST, Nuvoton etc.)

Future developments

- Allow to conditionally compile out TPM 1.2 support
- Drop TPM 1.1b support (proposed by Peter Hüwe)
- Support for I2C in tpm_tis
- In-kernel access broker
- Algorithm agility support for IMA
- Event log

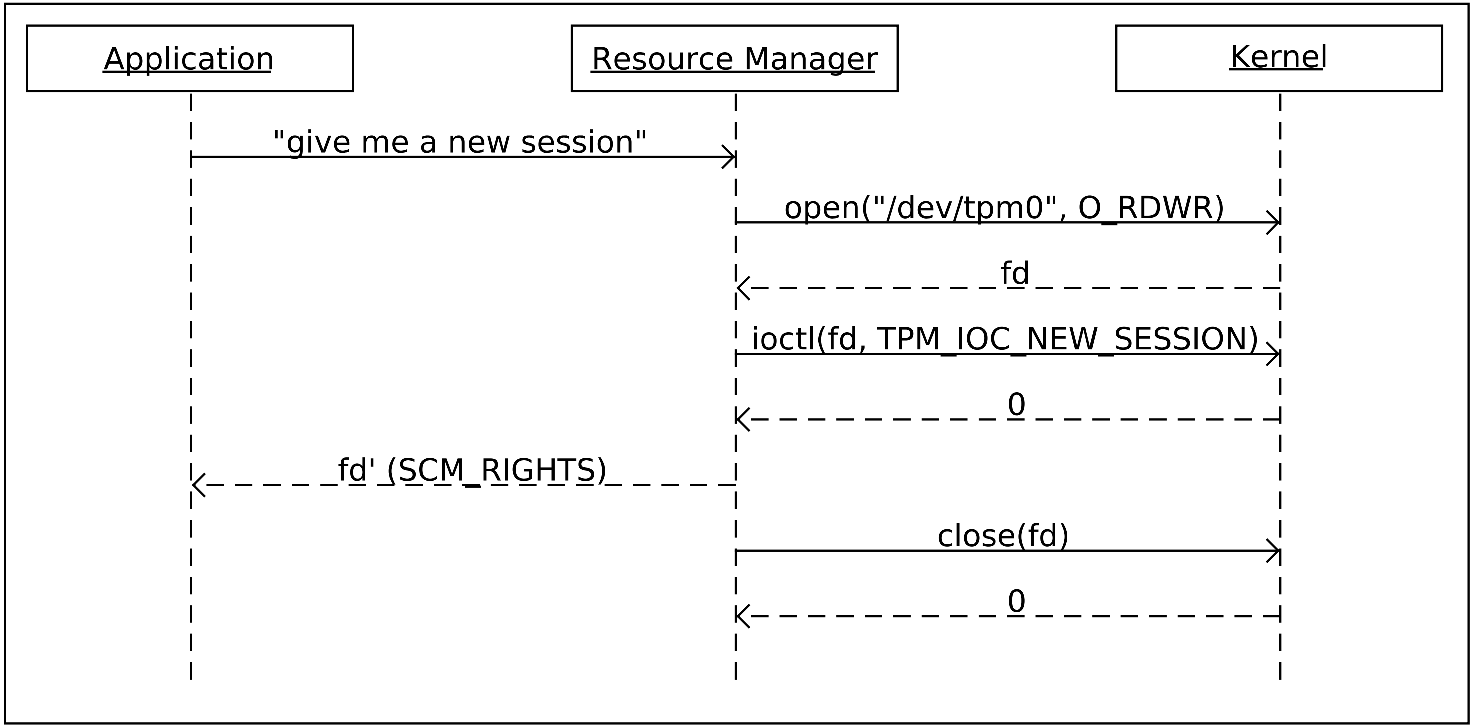
In-kernel access broker proposal draft (1/3)

- After the system is booted there’s one root session.
- The keyring always uses the root session.
- New session can be created with ioctl(fd, TPM_IOC_NEW_SESSION). It is alive until close(fd).
- Transient objects are faulted and swapped with TPM2_ContextLoad and TPM2_ContextSave.
- TPM_CAP_COMMANDS gives the meta-data for virtualizing the handle area of commands and responses.
- Each session has a shmem_file for swapping.

In-kernel access broker proposal draft (2/3)

- For each transient object of a session we need to have virtual and physical handle. When first created they are identical.
- When an object is faulted we replace the value of the physical handle.
- For commands we do virtual → physical substitution for the handle area.
- For responses we do physical → virtual substitution for the handle area.
- TPM_CAP_HANDLES requires a special case for the response. The handles in the body of the message needs to be substituted.

In-kernel access broker plan (3/3)



That was it!

TWENTY QUESTIONS
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
DICK
HERB
VAN
FLORENCE
ALDO RAY