Exploring the Trusted Platform Module to Establish Mutual Trust in High Performance Computing

Devon Bautista and Rebecca Whitten

August 12, 2021

Mentors: Christian Storer, J. Lowell Wofford, and Marc Santoro

LA-UR-21-28002

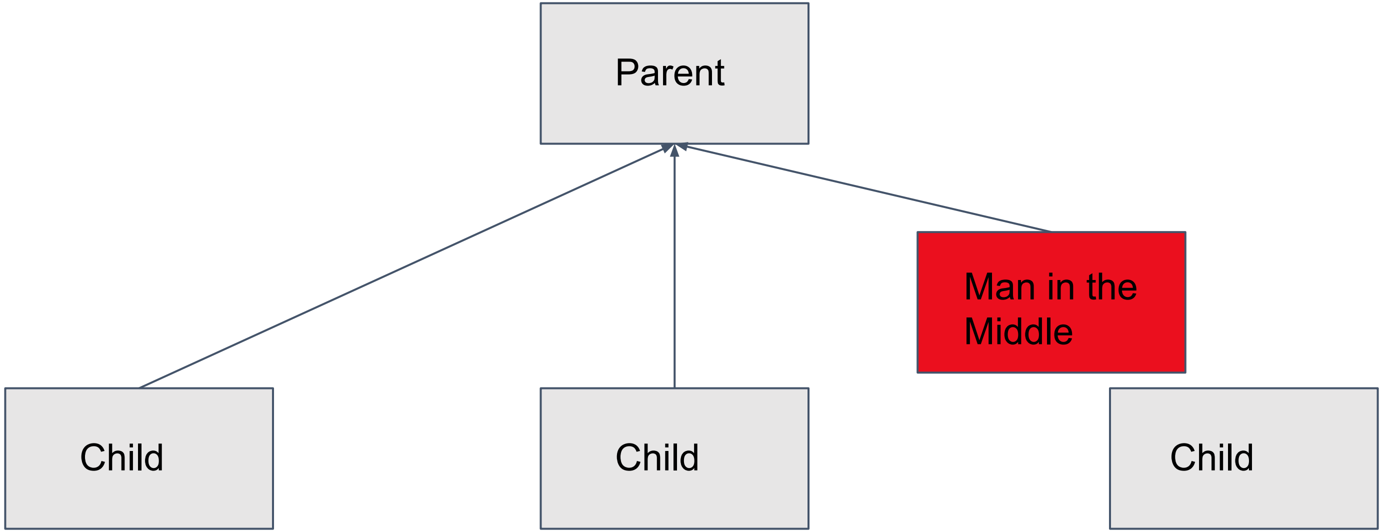
Bootstrapping of a Typical Stateless Cluster

Stateless = No secondary storage (e.g. hard disk)

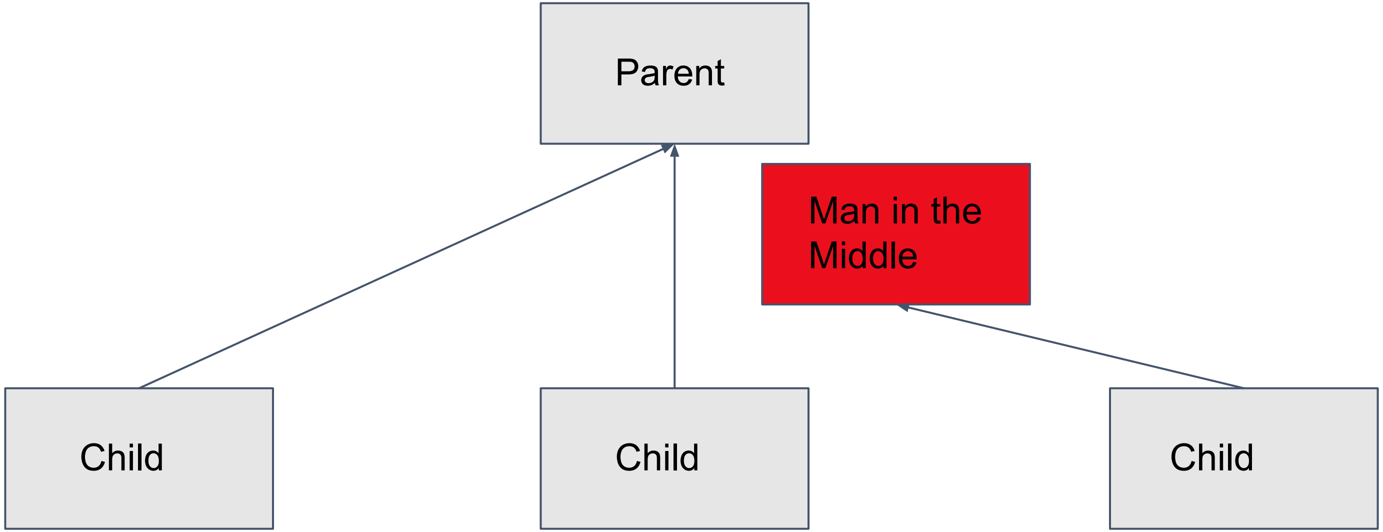
- Parent and nodes power on
- Nodes look for and connect to parent
- Parent configures nodes and provides OS image
- Nodes boot OS image

What's to stop an adversary from imitating a node? Stealing secrets (e.g. SSH keys)?

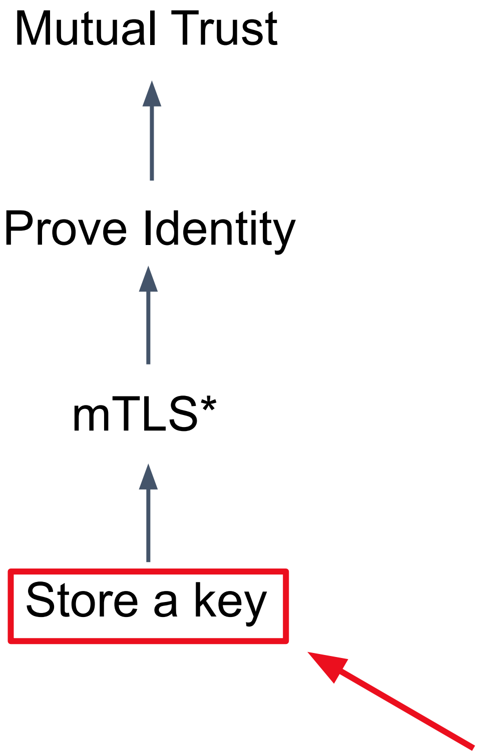
Problem in Stateless Boot



Problem in Stateless Boot



The Problem



Mutual Trust

Prove Identity

mTLS*

Store a key

*mTLS = mutual Transport Layer Security

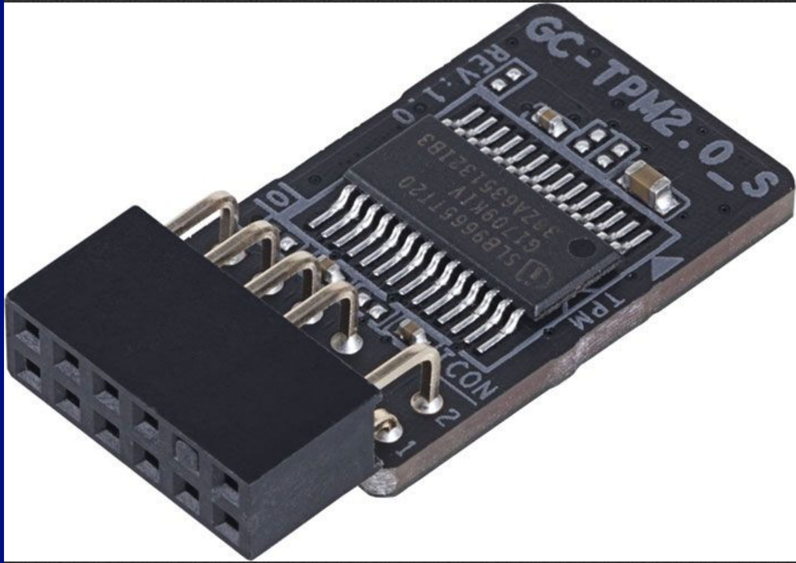
- a two-way cryptographic authentication protocol

How to store these securely?

What is a TPM?

- “Trusted Platform Module”
- A secure and separate cryptoprocessor
- Defined by TCG Specification*
  - “Trusted Computing Group”
- Separate, Non-Volatile RAM
- Access controls for certain operations

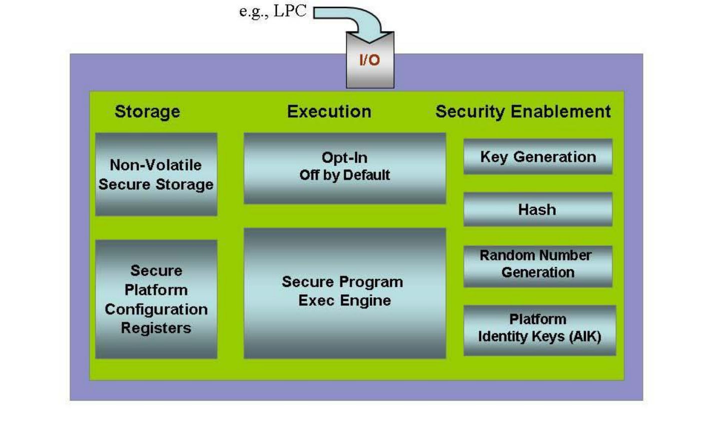
*https://trustedcomputinggroup.org/work-groups/trusted-platform-module/



A discrete TPM 2.0 chip manufactured by GIGABYTE. Source:https://hothardware.com/news/microsoft-allow-bypass-windows-11-tpm-20-requirement

What Can a TPM Do?

- Securely generate keys and store them
  - RSA and ECC
  - Private key never leaves the TPM
- Perform cryptographic operations
  - Sign/Decrypt by “asking” the TPM
  - Generate random numbers
  - Hashing (e.g. SHA-256)
- Store secrets
  - In “NV Indices”
- Measure system state
  - via Platform Configuration Registers (PCR)
- Much more

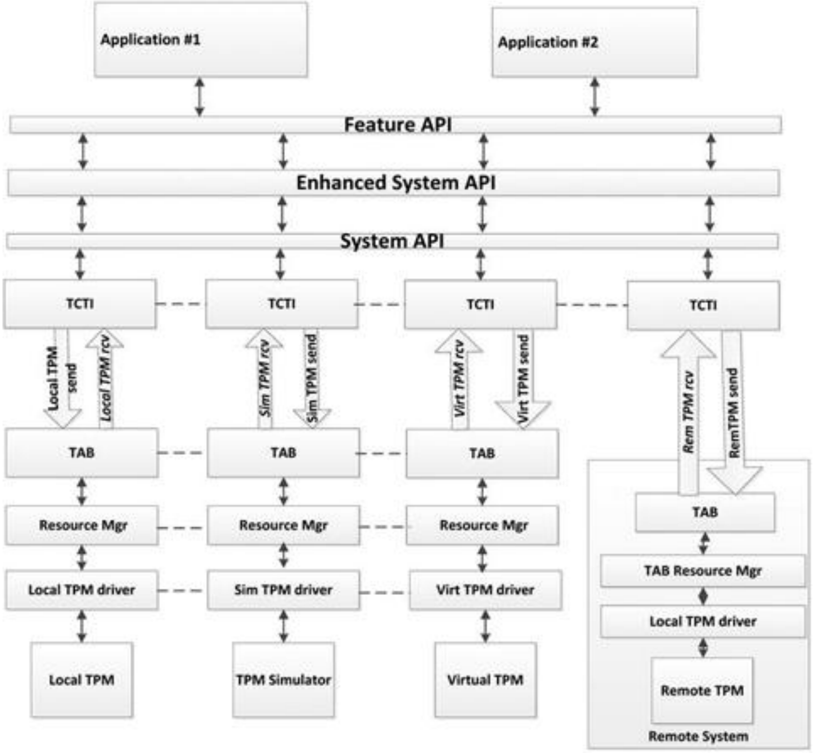

e.g., LPC
I/O
Storage
Execution
Security Enablement
Non-Volatile Secure Storage
Opt-In Off by Default
Key Generation
Hash
Secure Platform Configuration Registers
Secure Program Exec Engine
Random Number Generation
Platform Identity Keys (AIK)

Internal structure of the TPM 2.0.

Source: Trusted Computing Group, *Trusted Platform Module (TPM) Summary*

Interacting with the TPM

- Commands defined in specification
  - Byte stream
  - e.g. `TPM2_Startup`
- **Trusted Software Stack (TSS)**
  - TCG-specified API
- **Feature API (FAPI)** used for high-level communication with the TPM
- Several implementations
  - tpm2-tss (C)
  - tpm2-tools (CLI)
  - tpm2-pytss (Python)
  - go-tpm (Go)



The Trusted Software Stack, representing layers of TPM interaction with most abstract at the top to most granular at the bottom.

Source: Arthur, Challenger, Goldman. *A Practical Guide to TPM 2.0*

Interacting with the TPM

TPM2 Software Stack

github.com/tpm2-software

- Open Source
- Fully Implements TCG Software Stack Specification
- **tpm2-tss**: A C API for interacting with the TPM version 2.0
  - Provides the Feature API (FAPI), the high-level interface for interacting with the TPM
  - Also provides the System API (SAPI) and Enhanced SAPI (ESAPI), more low-level interfaces that provide 1-to-1 mappings of TPM commands specified in the TPM 2.0 specification
- **tpm2-tools**: - Command line utilities for interacting with the TPM
  - CLI wrapper for tpm2-tss, the TPM Trusted Software Stack
  - Thorough documentation; lots of examples

Interacting with the TPM

TPM2 Software Stack Continued

- **tpm2-tss-engine**: An OpenSSL engine for TPM 2.0
  - Used for doing OpenSSL-related things with the TPM
  - E.g. Creating a CSR from a private key stored in the TPM
- **tpm2-pkcs11**: A library/specification for creating/manipulating cryptographic tokens, such as those that may be stored within a TPM
  - Needed for e.g. using the TPM to store/use SSH keys
- **tpm2-pytss**: Python bindings for interacting with the TPM through the ESAPI (with FAPI in progress)
  - Code is heavily transitory
  - Documentation currently does not match API
  - Difficulty setting up in CentOS
  - Chose to skip because of the above, possibly unstable (for now) API, and significant setup overhead

Interacting with the TPM

Go-TPM

github.com/google/go-tpm

- Golang API for the TPM 2.0
- Does not yet implement entire TCG Specification
- Less thorough documentation
- Requires Go 1.16
- Easier installation:

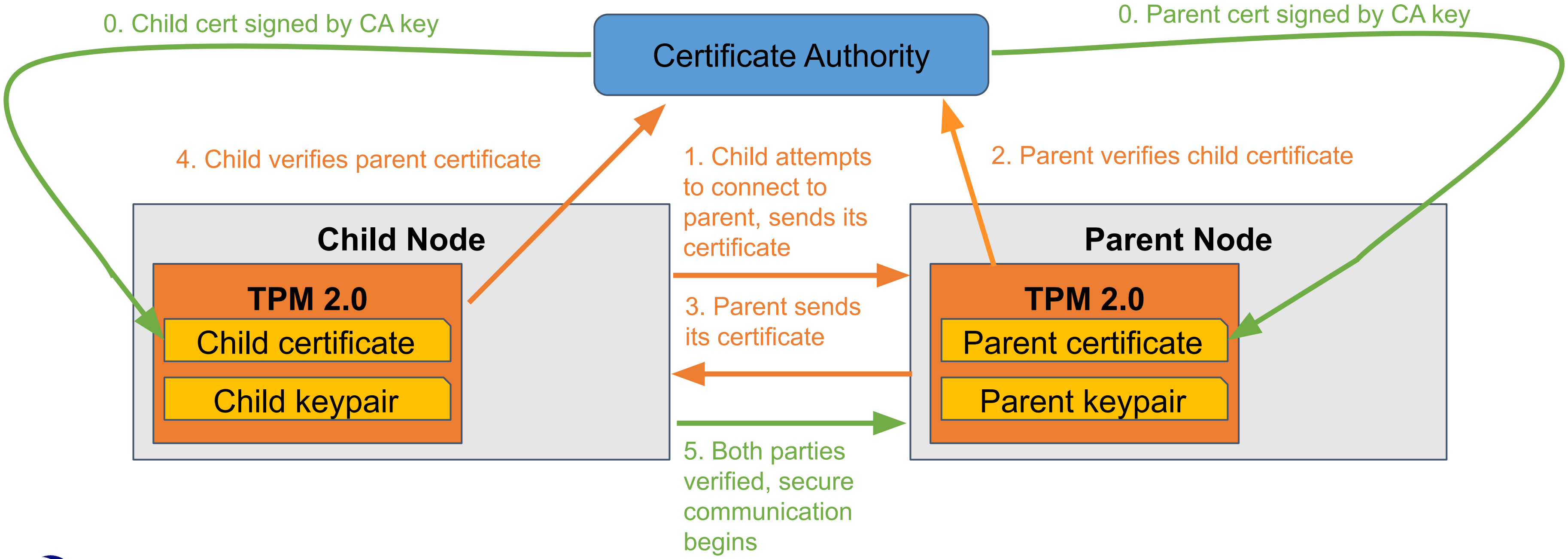
```
$ go get github.com/google/go-tpm/tpm2
```

How Does the TPM Address the Problem?

- **Secure Storage:** Able to store secrets without need for storage in disk, firmware, etc.
  - Discrete TPMs are tamper-resistant
  - A PKI for node/parent verification, independent of OS
- **Independent Access Control:** Storage/Operation access depends on authorization *independent* of the OS
  - Keys aren’t used or transferred unless authorized by the TPM
  - Just because you have root doesn’t mean you can access the TPM

Our Solution

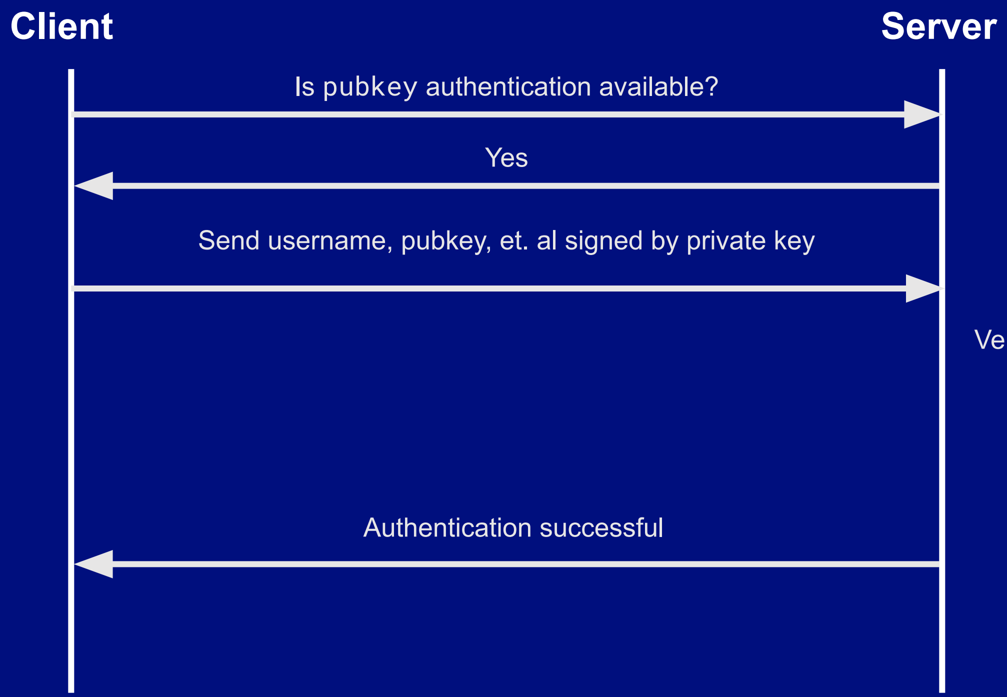
Implement a mutual authentication protocol using keys/certificates stored in the TPM to bilaterally authenticate compute nodes and their parent(s).



0. Child cert signed by CA key
Certificate Authority
0. Parent cert signed by CA key
4. Child verifies parent certificate
1. Child attempts to connect to parent, sends its certificate
2. Parent verifies child certificate
Child Node
TPM 2.0
Child certificate
Child keypair
3. Parent sends its certificate
Parent Node
TPM 2.0
Parent certificate
Parent keypair
5. Both parties verified, secure communication begins

How SSH Works Using a Keypair

Possesses
private key



Client

Server

Is pubkey authentication available?

Yes

Send username, pubkey, et. al signed by private key

Authentication successful

Possesses
public key

Verify that:

- Verify the signature with the provided public key
- Supplied pubkey is in user’s `authorized_keys`

Using the TPM for SSH Authentication

1. Set up PKCS#11 key database

```
$ tpm2_ptool init
```

2. Create a cryptographic token in the PKCS#11 storage

```
$ tpm2_ptool addtoken --pid 1 --label sshtok \
    --sopin <supervisor_pin> --userpin <user_pin>
```

3. Generate key pair associated with the above token

```
$ tpm2_ptool addkey --algorithm <rsa2048_or_ecc256> \
    --label sshtok --key-label <key_label> --userpin <key_pin>
```

4. Place public component of key into remote host’s authorized_keys file

```
$ ssh-keygen -D /path/to/libtpm2_pkcs11.so | ssh <host> \
    ‘cat >> ~/.ssh/authorized_keys’
```

5. SSH into the machine using the TPM key

```
$ ssh -I /path/to/libtpm2_pkcs11.so <host>
```

Using the TPM for mTLS

- Generate CA Key Pair and Certificate

```
$ openssl x509 ...
```

- Create an authorization policy

```
    $ tpm2_startauthsession ...
    $ tpm2_policypassword ...
    $ tpm2_flushcontext ...
```

- Define an NV Index with authorization policy

```
$ tpm2_nvdefine -L policy -C o -s 2048 -p samplepassword 1
```

- Write certificate to NV Index

```
$ tpm2_nvwrite -Q 1 -C o -i client.crt -P samplepassword
```

- Lock Index from Further Writes [Optional]

```
$ tpm2_nvwritelock -C o 1
```

Future Work

- Finish mTLS implementation using the TPM
  - PoC for authenticating nodes with certificate
  - Integrate into Kraken/Layercake?
- More research/testing into NV Index policies
  - NVName policy to prevent attacker deleting and recreating index
- Using the PCR functionality to verify and attest the entire boot process

References

[1] C. M. Lonvick and T. Ylonen, The Secure Shell (SSH) Authentication Protocol. RFC Editor, 2006. doi: 10.17487/RFC4252.

[2] D. Goutte-Gattat, “Using a TPM for SSH authentication,” Incenp.org, 03-Jan-2020. [Online]. Available: https://incenp.org/notes/2020/tpm-based-ssh-key.html. [Accessed: 22-Jul-2021].

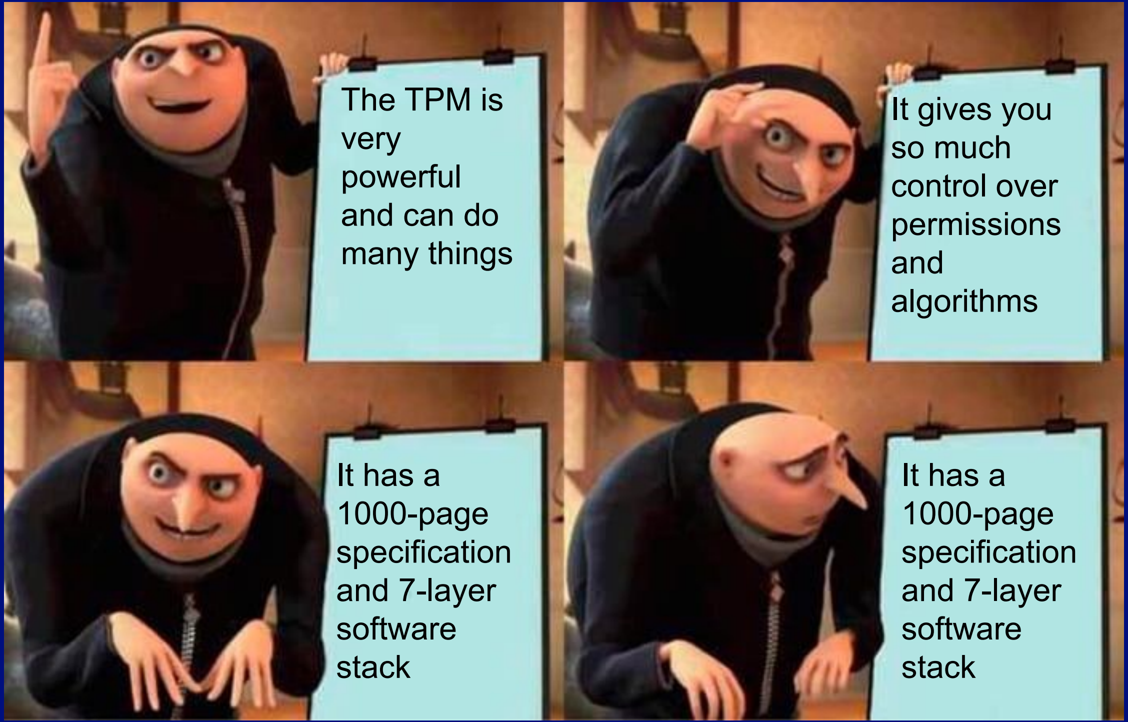
[3] Go-TPM (2021) [Source Code] https://github.com/google/go-tpm.

[4] Linux TPM2 & TSS2 Software (2021) [Source Code] https://github.com/tpm2-software.

[5] *Trusted Platform Module Library Specification, Family “2.0”, Level 00, Revision 01.59*, Nov. 2019. [Online]. Available: https://trustedcomputinggroup.org/work-groups/trusted-platform-module/

[6] W. Arthur, D. Challenger, and K. Goldman, *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*. Apress Media, 2015.

Bonus


The TPM is very powerful and can do many things
It gives you so much control over permissions and algorithms
It has a 1000-page specification and 7-layer software stack
It has a 1000-page specification and 7-layer software stack