Secure Boot a Raspberry Pi 3 B, with u-boot and slb9670 infenion TPM2

We need a Raspberry Pi 3 B with 32bit raspberry OS, a TPM2 slb9670 Infenion and U-boot as second bootloader.
U-boot on a RPi 3 B, cannot communicate with TPM slb9670, during boot. So new code added to u-boot/arch/arm/dts/bcm2837-rpi-3-b.dts.
Also code added to u-boot/cmd/tpm-v2.c, so that u-boot can read non volatile memory of slb9670, during boot.

STEP 1:

install the TPM framework, so that the OS can communicate with TPM.
First update/upgrade OS and then install the dependecies

```
sudo apt-get update
sudo apt-get upgrade
```

```
sudo apt-get –y install \
autoconf-archive \
libcmocka0 \
libcmocka-dev \
procps \
iproute2 \
build-essential \
git \
pkg-config \
gcc \
libtool \
automake \
libssl-dev \
uthash-dev \
autoconf \
doxygen \
libjson-c-dev \
libini-config-dev \
libcurl4-openssl-dev \
U-boot sudo apt-get \
uuid-dev \
pandoc \
bison \
flex \
libncurses-dev
```

for the TPM framework

```
git clone https://github.com/tpm2-software/tpm2-tss ~/tpm2-tss
cd ~/tpm2-tss
sudo ./bootstrap
sudo ./configure
sudo make -j$(nproc)
sudo make install
sudo ldconfig
```

```
git clone https://github.com/tpm2-software/tpm2-tss-engine ~/tpm2-tss-engine
cd ~/tpm2-tss-engine
sudo ./bootstrap
sudo ./configure
sudo make -j$(nproc)
sudo make install
sudo ldconfig
```

```
git clone https://github.com/tpm2-software/tpm2-tools ~/tpm2-tools
cd ~/tpm2-tools
sudo ./bootstrap
sudo ./configure
sudo make -j$(nproc)
sudo make install
sudo ldconfig
```

STEP 2:
Compile the device tree for tpm, "tpm-slb9670.dts", which is provided in the repository

```
dtc -@ -I dts -O dtb -o tpm-slb9670.dtbo tpm-slb9670.dts
```

add the compiled file "tpm-slb9670.dtbo" to boot/overlays directory

add to conig.txt

```
dtparam=spi=on
dtoverlay=tpm-slb9670.dtbo
```

Reboot the RPi and check if TPM is working using terminal and commands
sudo tpm2 get_random, it should return a random number from TPM
sudo tpm2 pcr_read , it should return the content of PCRs of TPM

in a terminal execute

```
sudo sh hash.sh
```

this script (provided in the repository) will calculate the hash of the desired files and store it to non volatile memory of TPM

STEP 3:
Install u-boot

```
git clone https://github.com/u-boot/u-boot ~/u-boot
```

replace the ~/u-boot/arch/arm/dts/bcm2837-rpi-3-b.dts, with the one provided in this repository
replace the ~/u-boot/cmd/tpm-v2.c, with the one provided in this repository

compile u-boot

```
cd ~/u-boot
sudo make distclean
sudo make rpi_3_32b_defconfig
sudo make menuconfig
```

Choose options

```
Boot options ->
        [*] Enable preboot
        (pci enum; usb start; setenv bootdelay 5) preboot default value

Library routines -> Security support ->
        [*] Trusted Platform Module (TPM) Support

Device Drivers -> [*] SPI Support ->
        [*] Enable Driver Model for SPI drivers
        [*] Soft SPI driver

Device Drivers -> TPM support ->
        [*] TPMv2.x support
        [*] Enable support for TPMv2.x SPI chips

Command line interface -> Security commands ->
        [*] Support 'hash' command
        [*] Enable the 'tpm' command

Misc commands ->
        [*] gettime command
        [*] timer command
```

```
sudo make all
```

file u-boot.bin is ceated
copy it to boot directory

```
sudo cp u-boot.bin/boot
```

Add to config.txt

```
kernel=u-boot.bin
```

Reboot RPi, break the booting sequence of u-boot and try some TPM commands to verify communication between u-boot and TPM.

```
tpm2 init
tpm2 info
```

it should return info about TPM.

STEP 4:

Final step.

In a terminal, execute the script hash.sh. This script calculates the hash of kernel7.img and bcm2710-rpi-3-b.dtb and stores it to non volatile memory of TPM

Transform the text file “boot.scr” to a format readable by u-boot. It’s a script that instructs u-boot to calulate hashes and compare with NV memory of TPM

```
~/u-boot/tools/mkimage -A arm -T script -C none -n "u-boot script" -d boot.scr boot.scr.uimg
```

The file “boot.scr.uimg” is created. Copy it to boot directory.

During boot the u-boot loads and executes the boot.scr.img which calculates the hash of kernel7.img and bcm2710-rpi-3-b.dtb and compares it with the one stored in NV memory of TPM. If they are the same then continue to boot else halt the booting.

Reboot RPi. If everything is Ok RPi will boot with information on screen about the stages of the secure boot.

IMPORTANT. Every time you update/upgrade the OS, you should execute the script hash.sn to calculate the new hashes. If you forget this RPi will not boot.

Be aware that this method is partialy secure boot. A person with physical access to the SD card of RPi can easily delete the “kernel=u-boot.bin“ from config.txt so that the RPi will not use u-boot and loose secure boot cabability.