Prerequisites:
1. Assuming Grub 2.02-2.06 is running
2. presence of a TPM 2.0 module (or intels built-in PTT) activated in Bios
3. Second workstation running ubuntu 20.04
4. Live USB containing ubuntu 20.04

(**Rob** was used as the name of the encrypted container – any name can be used)

- Needed packages on host & setup workstation

```
sudo apt update
sudo apt install cryptsetup cryptsetup-initramfs \
                 tpm2-tools tpm2-tss tpm2-abrmd \
                 clevis clevis-luks clevis-tpm2 clevis-initramfs
```

LUKS2 no support for older grub versions (below 2.06 – no auto-unlock-mount)
      → **LUKS1**

With the disk attached to another Ubuntu PC (20.04)

1.1 Preparation - Split boot directory from the root partition

- resize root-fs partition on disk containing the backup so it fits inside the container

sudo e2fsck -f /dev/sdb2 – FS check
sudo resize2fs /dev/sdb2 460G - resize FS to 460gb

- Copy root-FS(ext4), Boot-FS(ext4) & EFI-FS(fat32) from the backup
  (I used “cp”/”partclone” for this process)

```
sudo partclone.ext4 -c -s /dev/sdb2 -o ~/Desktop/Backups/Rob4_os.img
sudo partclone.fat32 -c -s /dev/sdb1 -o ~/Desktop/Backups/Rob4_efi.img
sudo cp -r /mnt/boot/ ~/Desktop/Backups/boot/
```

1.2. Create/delete partitions

- 2 partitions → 3 partitions

```
sudo fdisk /dev/sdb
      d: delete partition
      n: new partition
      t: change partition type (Linux-FS for root&boot – Efi-System for for efi)
      w: write changes
      → sdb1 → Root-FS & sdb2 → boot & sdb3 → efi

sudo mkfs.vfat /dev/sdb3
sudo mkfs.ext4 /dev/sdb1 & /dev/sdb2
```

1.3. Encrypt the root partition

- Commands for encrypting/opening/mounting

```
sudo cryptsetup luksFormat --type luks1 /dev/sdb1
sudo cryptsetup open /dev/sdb1 Rob

sudo mkfs.ext4 /dev/mapper/Rob
sudo mount /dev/sdb2 /mnt/boot
```

1.4. Transfer /, /boot and */boot/*efi FS to target Partitions

```
sudo partclone.fat32 -C -r -s ~/Desktop/Backups/Rob4_efi.img -o /dev/sdb3
sudo partclone.ext4 -C -r -s ~/Desktop/Backups/Rob4_os.img -o /dev/mapper/Rob
sudo cp -a ~/Desktop/Backups/boot/. /mnt/boot/.

sudo cryptsetup close Rob
sudo umount /dev/sdb2
```

2. Activate TPM/PTT (Intel T.T.P.)

Enter Bios of host PC
→ PCH-FW Configuration
    select PTT
→ Trustet Computing
    Enable it
→ CPU Configuration → Intel Trusted Execution Technology
    Enable it

3. Boot into live usb (Ubuntu 20.04)

Mount & Chroot into the filesystem

- conf network

```
sudo dhclient
```

```
sudo cryptsetup open /dev/sdXn Rob
sudo mount /dev/mapper/Rob /mnt/
```

- Separated /boot partition mount via:

```
sudo mount /dev/sdXn /mnt/boot
```

- mount the /boot/efi:

```
sudo mount /dev/sdXn /mnt/boot/efi
```

- bind-mount pseudo FS

```
sudo mount --rbind /proc /mnt/proc
sudo mount --rbind /sys /mnt/sys
sudo mount --rbint /dev /mnt/dev
```

- copy internet conf file

```
sudo cp /etc/resolv.conf /mnt/etc/resolv.conf
```

- chroot into the root FS - operate just like remotely logged in

```
sudo chroot /mnt /bin/bash
```

Configure Grub and Crypttab

(fstab aswell – if boot directory was seperated or partition order was altered)
- Add line for mounting the boot/root-container partition (Use UUID except for encrypted container)

```
sudo nano /etc/fstab   :
```

```
/dev/mapper/Rob   /   ext4   defaults   0  1
/dev/disk/by-uuid/...   /boot   ext4   defaults   0  1
/dev/disk/by-uuid/...   /boot/efi vfat defaults 0  2
```
“UUID=” *- arg. also works*

- If boot is still in the root – follow these steps to edit the grub file:

```
sudo nano /etc/default/grub
```
add new line: **GRUB_ENABLE_CRYPTODISK=y**

(If you enabled Ubuntu SELinux (usually disabled by default)
→ then add enforcing=0 kernel parameter as value of GRUB_CMDLINE_DEFAULT.
(Can be removed after first succesful boot)

4. For auto unlock mechanism

(while still in chroot)

sudo apt install clevis clevis-luks clevis-tpm2 clevis-initramfs

Create a random key (clevis)
→ TPM seals (encrypt) random key under policy tied to PCR 7 (changeable)

sudo clevis luks bind -d /dev/sdX tpm2 '{"pcr_ids":"7"}'

- In the following file → entries for what and how to unlock are placed
nano /etc/crypttab

Rob UUID={...} none luks

– (UUID of the partition the encrypted container is on – replace {...})

Reinstall grub

grub-install --target=x86_64-efi --efi-directory=/boot/efi --boot-directory=/boot/efi/EFI/ubuntu --recheck

“--boot-directory=/boot” (if boot is its own partition – i.e. unencrypted)

grub-mkconfig -o /boot/efi/EFI/ubuntu/grub/grub.cfg

update-initramfs -c -k all

Exit chroot
exit

(you then have to unmount everything)
or for lazy man :D just:
shutdown -P now

Creating/Applying a backup: (for application only skip to 3.2)

1. creating backup of the luks1 container headers
sudo cryptsetup luksHeaderBackup /dev/sdb1 --header-backup-file ~/luks-header-sdb1.backup
creating backup of the partitioning table
sudo sgdisk --backup /home/atlasatcal/sdb-partition-table.bak /dev/sdb

2. Creating an .img of each partition → Open Luks container
sudo cryptsetup open /*dev*/sdb1 Rob
2.1 Use Clonezilla in expert mode
sudo clonezilla
Create backup of all 3 partitions (Root-FS, Boot, EFI)
→ select /*dev*/mapper/Rob
→ select /*dev*/sdb2
→ select /*dev*/sdb3
.OR.
2.2 sudo mkdir ~/Desktop/Backups_ENC
sudo partclone.ext4 -c -s /dev/mapper/Rob -o ~/Desktop/Backups_ENC/Rob4_sdb1.img
sudo partclone.ext4 -c -s /dev/sdb2 -o ~/Desktop/Backups_ENC/Rob4_sdb2.img
sudo partclone.fat32 -c -s /dev/sdb3 -o ~/Desktop/Backups_ENC/Rob4_sdb3.img

3. Preparing the disk
(not necessary when partition table is restored onto disk → skip to **3.2**)

3.1 calculate bit map → 2gb each for efi and boot partition
sudo fdisk /*dev*/sdb
65535 → 968460171 ; 968476230 → 972624682 ; 972670470 → 976773134
.OR.
3.2 Applying disk partition backup
sudo sgdisk --load-backup=/home/atlasatcal/sdb-partition-table.bak /dev/sdb

Create LUKS by using Backup of the LUKS headers
sudo cryptsetup luksHeaderRestore /dev/sdb1 --header-backup-file luks-header-sdb1.backup
sudo cryptsetup open *dev/sdb1 Rob*

4. **Loading backup to disk**

4.1 **sudo clonezilla**
→ apply backups to /*dev*/mapper/Rob, /*dev*/sdb2 & /*dev*/sdb3
.OR.
4.2 **–** Loading backups onto disk using partclone
sudo partclone.ext4 -C -r -s ~/Desktop/Backups_ENC/Rob4_sdb1.img -o /dev/mapper/Rob
sudo partclone.ext4 -C -r -s ~/Desktop/Backups_ENC/Rob4_sdb2.img -o /dev/sdb2
sudo partclone.fat32 -C -r -s ~/Desktop/Backups_ENC/Rob4_sdb3.img -o /dev/sdb3

5. Clevis/Cryptsetup Key setup
sudo cryptsetup luksDump /dev/sdb1

How to delete keys:
(sudo cryptsetup luksRemoveKey --key-slot N /dev/sda1)
sudo cryptsetup luksKillSlot /*dev*/sda1 N
Adding Passphrase Key:
sudo cryptsetup luksAddKey –key-slot N /*dev*/sda1
– N for keyslot
Adding TPM bound key:
clevis luks bind -d /dev/sdX tpm2 '{}'
clevis luks bind -d /dev/sdX tpm2 '{"pcr_ids":"0,2,4"}'
→ Lock to firmware + bootloader + kernel

6. Activation of Secure Boot – Assuming backup images were used

Navigate to Security Panel → Secure boot

switch to setup mode → then enable secure boot

Boot → select advanced options → boot into generic Image

cd Kernel → *home/*conbotics/Kernel
    - contains all the keys for secure boot

6.1 MOK enrollment
sudo mokutil --import MOK.der

→ this command will ask for a new password which you have to enter upon startup in the MOK enrollmen screeb
→ reboot and enter the password

uname -r
    - to check whether the running kernel is the custom one – if so everything worked