Module 1.3
Zymbit Library Structure and Binding

CompuThings *Technology;

Zymkey Libraries

- Zymbit develop libraries and software ‘driver’ in order host system talk to zymkey.
- Script install_zk_sw.sh is the installation script that download the libraries and dependencies to operate zymkey.
- The script install required software and do the binding process between zymkey and Rpi.
- Development binding is the temporary binding between zymkey and Rpi.
- Production binding is the real binding setup for real production grade zymkey.

Inside Zymkey Install Script

1. Ensure we are running as root
2. Ensure group gpio exist in the system
3. Install python and other required Linux libraries
4. Install zymbit gpg key
5. Add zymbit repo to system
6. Install zymbit package
7. Restart zkifc
8. Reboot Rpi

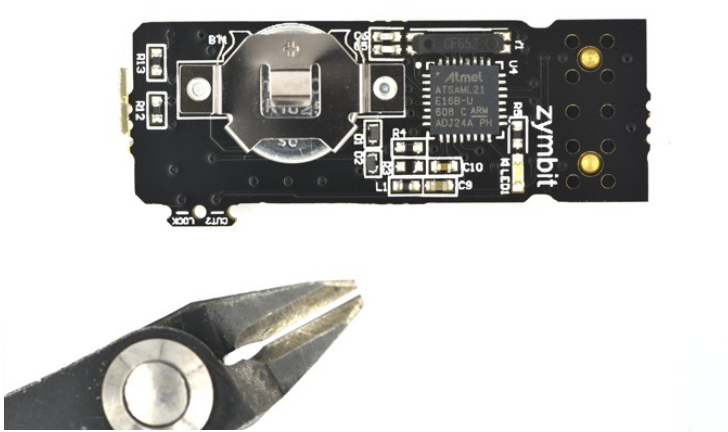
List of Zymbit Package and Libraries

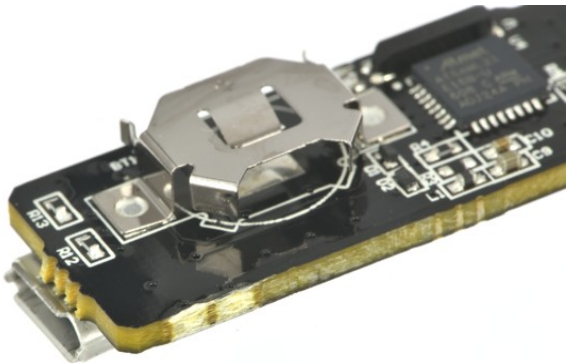
- Libzk - main library that perform the operation to zymkey HSM. Close Sources
- Libzkkeyssl
- Zkbootrtc
- Zkifc - user space C code that control the link to Zymkey. Act as ‘driver’
- Zkapputilslib
- Zksaapps
- Zkpkcs11
- Cryptsetup – user space program to handle SD card encryption
- Zku
- zk_luks
- zk_prep_encr

Binding Process

- Binding is where the Device ID of zymkey and unique system information is being bind or tie together.
- The process is done internally in the install script.*(no detail process explained)*
- Developer Mode is temporary binding. Zymkey can be move to other Rpi.
- Production Mode binding is permanent binding. Zymkey will permanently bind to one Rpi.
- For production mode, we need to configure the perimeter detect event, encrypt SD card and cut the cut-2-lock pin in zymkey.

Zymkey Cut-2-Lock





Conclusion

- Zymbit forbid us to know detail mechanism of hardware access to secure element and Atmel ARM processor inside zymkey.
- The binding process is where zymkey is being tie (permanently or temporarily) to a host (Rasberry Pi in our case).