Example on Secure boot implementation for WP76xx

1. The test below has been tested on R13.3 yocto source code.
2. Download the toolchain according to your software release.

E.g. R13.3 components

<https://source.sierrawireless.com/resources/airprime/software/wp76xx/wp76xx-firmware-release-13,-d-,1-components/#sthash.13BE7Elj.dpbs>

1. Install the toolchain to /opt/swi on your linux PC

sh poky-swi-ext-glibc-x86\_64-meta-toolchain-swi-armv7a-neon-toolchain-swi-SWI9X07Y\_02.28.03.05.sh

1. copy the android\_signature\_add.sh to your signing directory ,e.g. /home/owner/Yocto/tools/signing\_dir



1. Environment setup

export ANDROID\_SIGNING\_DIR=/home/owner/Yocto/tools/signing\_dir

cd /home/owner/Yocto/tools/signing\_dir

1. copy hdrcnv from toolchain to signing directory

cp /opt/swi/SWI9X07Y\_02.28.03.05/sysroots/x86\_64-pokysdk-linux/usr/bin/hdrcnv /home/owner/Yocto/tools/signing\_dir

1. Copy the following two files from the build system to the signing directory after successfully compile the Linux yocto source code

cp -rf /home/owner/Yocto/WP76/R13.3/yocto/build\_bin/tmp/sysroots-components/x86\_64/android-signing-native/usr/share/android-signing/\* /home/owner/Yocto/tools/signing\_dir

cp -rf /home/owner/Yocto/WP76/R13.3/yocto/meta-swi/meta-swi-mdm9xxx/recipes-bsp/android-signing/files/swi-make-cert-chain.sh /home/owner/Yocto/tools/signing\_dir

1. Set up the certificate

chmod 777 make\_key

./make\_key mykey '/C=US/ST=California/L=Mountain View/O=Android/OU=Android/ CN=Android/emailAddress=android@android.com'

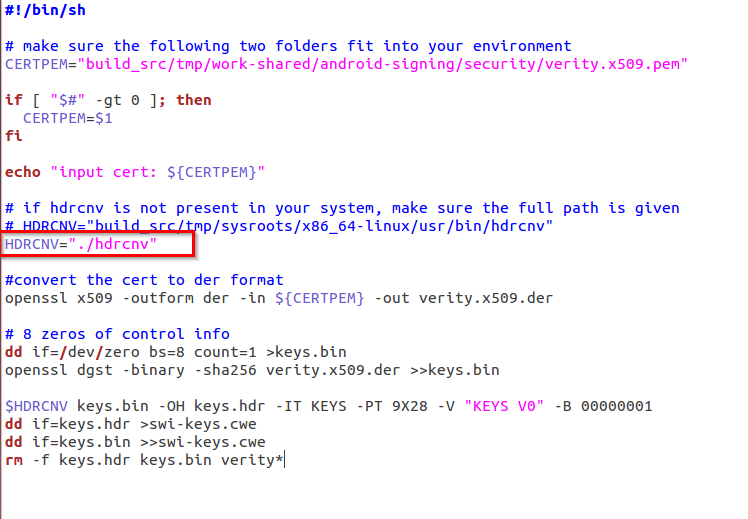
(When prompted, choose a password to protect the private key file that is being generated. The private/public key pair (mykey.pk8, mykey.x509.pem) are created in the current signing folder.)

1. Replace the default (dummy) key pair with your new key pair

cp mykey.pk8 security/verity.pk8

cp mykey.x509.pem security/verity.x509.pem

1. Open swi-key-cwe.sh, in line 14, change the path of HDRCNV according to build environment



1. Generate the keystore CWE image (swi-keys.cwe) corresponding to your new key

chmod 777 swi-key-cwe.sh

chmod 777 hdrcnv

$ANDROID\_SIGNING\_DIR/swi-key-cwe.sh security/verity.x509.pem

1. Copy the unsigned images (i.e. aboot, kernel, rootfs) to signing directory

cp /home/owner/Yocto/WP76/R13.3/yocto/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/appsboot.mbn.unsigned /home/owner/Yocto/tools/signing\_dir

cp /home/owner/Yocto/WP76/R13.3/yocto/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/boot-yocto-mdm9x28.4k.unsigned.img /home/owner/Yocto/tools/signing\_dir

cp /home/owner/Yocto/WP76/R13.3/yocto/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/mdm9x28-image-minimal-swi-mdm9x28-wp.ubi /home/owner/Yocto/tools/signing\_dir

1. Sign the aboot and kernel

chmod 777 ./android\_signature\_add.sh

./android\_signature\_add.sh /boot boot-yocto-mdm9x28.4k.unsigned.img boot-yocto-mdm9x28.4k.img

(need to your password set in step 8 above)

./android\_signature\_add.sh /aboot appsboot.mbn.unsigned appsboot.mbn

(need to your password set in step 8 above)

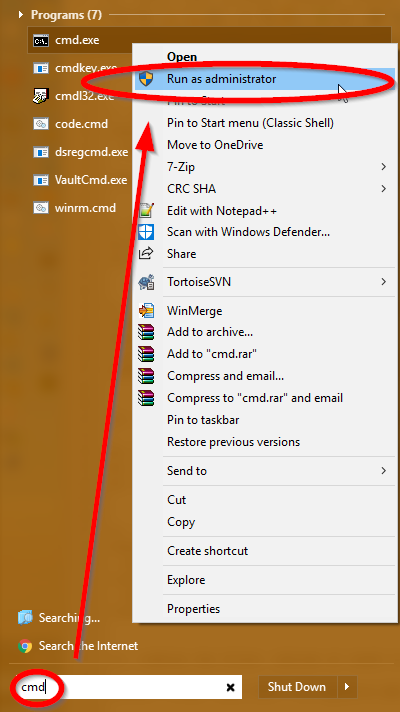
1. Install swicwe tool

<https://source.sierrawireless.com/resources/airprime/software/swicwe/#sthash.sno65XlI.dpbs>

1. Combine the signed files as yocto.cwe (swicwe version at least to be v1.2)

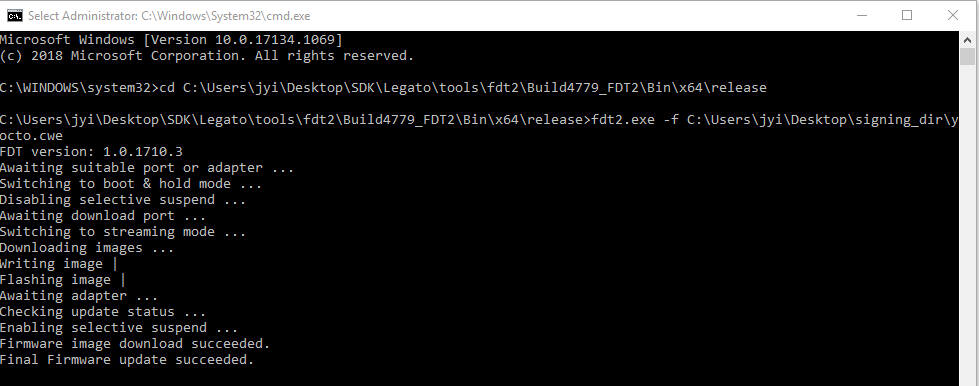
swicwe --YCWE --pid 9X28 --lk appsboot.mbn --kernel boot-yocto-mdm9x28.4k.img --rootfs mdm9x28-image-minimal-swi-mdm9x28-wp.ubi -v SWI9X07Y\_02.28.03.05 -o yocto.cwe

1. Copy the whole signing directory from linux PC to Windows 10 platform.
2. Open command prompt in Window as administrator



1. Load the final image yocto.cwe to module by FDT

fdt2.exe -f C:\Users\jyi\Desktop\signing\_dir\yocto.cwe



1. Enable Customer Secure Boot in AT command port

AT!ENTERCND="A710"

AT!OEMAUTH=1

1. Download the keystore CWE image to module by FDT (Please note that the command prompt has to be "run as administrator")

fdt2.exe -f C:\Users\jyi\Desktop\signing\_dir\swi-keys.cwe

1. Now secure boot is enabled and module should be able to boot in to console.
2. For a new built yocto image next time, start with step 12 above to sign the image.

DO NOT regenerate the key and swi-keys.cwe !!!!!

1. In case downloading a not-signed image on this secure boot enabled module, we will see the following in UART console:

B - 25372980 - cert signature failure, b

B - 25373346 - BOOT\_HOLD=3,DOWNLOADER=boot,VERSION=Thu Oct 31 04:32:40 UTC 2019,TYPE=APPL,SUBIMAGE=APPS,ERROR=Flash write failed: Secboot invalid cert chain (0X92),STATUS=FAIL

If you see this, after reboot, the device cannot run properly, you need to download the signed image again in step 18 above by fdt tool.