Example on Secure boot implementation for WP76xx (version 7)

# Signing server environment setup

1. The test below was conducted on WP76xx R16 yocto source code.
2. Environment setup

export ANDROID\_SIGNING\_DIR=/home/owner/Yocto/tools/signing\_dir\_R16

cd /home/owner/Yocto/tools/signing\_dir\_R16

1. Extract the following zip file and copy the android\_signature\_add\_R16.sh to your signing directory (i.e /home/owner/Yocto/tools/signing\_dir\_R16)



1. copy hdrcnv from yocto build directory to signing directory

mkdir common/

mkdir common/sierra/

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/cwetool-native/usr/bin/hdrcnv ./common/sierra/

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

cp -rf /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/android-signing-native/usr/share/android-signing/\* ./

cp -rf /home/owner/Yocto/WP76/R16/meta-swi/meta-swi-mdm9xxx/recipes-bsp/android-signing/files/swi-make-cert-chain.sh ./

1. Set up your own 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. Open the file swi-key-cwe.sh, line 46, change as the following highlighted part:

if [[ -n "$3" && "$3" =~ ^[A-Z]{3}[0-9]$ ]]

then

 KEY\_ID=$3

elif [ -z "$2" ]; then

 echo "Generate swi-key.cwe for 9x28 series."

else

 echo "Wrong keyid format. Exiting."

 exit

fi

1. Extract the following zip file and copy the android\_sign.sh to your signing directory



1. Open android\_sign.sh and modify line 2 and line 49 according to your environment:





1. Extract the following zip file and copy the rootfs-sign.sh to your signing directory



1. Open rootfs-sign.sh and modify line 2 and line 13 according to your environment:



# Bootloader and kernel authentication

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

chmod 777 swi-key-cwe.sh

chmod 777 ./common/sierra/hdrcnv

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

1. Copy the unsigned images (i.e. aboot, kernel) and default RootFS image to signing directory

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/appsboot.mbn.unsigned ./

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/boot-yocto-mdm9x28.4k.unsigned.img ./

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/mdm9x28-image-minimal-swi-mdm9x28-wp.ubi ./

1. Sign the aboot and kernel

chmod 777 ./android\_signature\_add\_R16.sh

./android\_signature\_add\_R16.sh /boot boot-yocto-mdm9x28.4k.unsigned.img boot-yocto-mdm9x28.4k.img verity

(need to enter your password set)

./android\_signature\_add\_R16.sh /aboot appsboot.mbn.unsigned appsboot.mbn verity

(need to enter your password set)

1. Combine the signed files as yocto.cwe
* cp /home/owner/Yocto/WP76/R16/meta-swi/common/recipes-core/cwetool/cwetool/yoctocwetool.sh ./
* cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/cwetool-native/usr/bin/hdrcnv ./hdrcnv\_cwetool
* open yoctocwetool.sh in local directory, in line 54, change it to

HDRCNV=./hdrcnv\_cwetool

* chmod 777 ./hdrcnv\_cwetool
* echo "SWI9X07Y\_02.37.10.00" > version\_file.txt
* ./yoctocwetool.sh -pid '9X28' -platform '9X28' -o yocto.cwe -fbt appsboot.mbn -vfbt version\_file.txt -kernel boot-yocto-mdm9x28.4k.img -vkernel version\_file.txt -rfs mdm9x28-image-minimal-swi-mdm9x28-wp.ubi -vrfs version\_file.txt

(Please note that only bootloader and kernel part are signed with your certificate in this yocto.cwe. The rootFS image is not signed with your certificate in this 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 combined image yocto.cwe to module by FDT which can be downloaded here in <https://downloads.sierrawireless.com/tools/swiflash/swiflash.zip>

fdt2.exe -f C:\Users\jyi\Desktop\signing\_dir\_R16\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\_R16\swi-keys.cwe

1. Now secure boot is enabled and module should be able to boot into console. You can confirm LK/Kernel authentication is enabled by the following:

at!oemauth?

!OEMAUTH: Enabled

OEM cert hash: A63F021F999537C9A7A33296A75A6741A679F53CB7860D4947779D04E84AC7AE

OK

1. For a new built yocto image next time, start with step 3 above to sign the image.

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

1. In case downloading a not-signed yocto image on this secure boot enabled module with LK/kernel authentication, we will see the following in UART console and command prompt respectively:

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 yocto image (yocto.cwe) again in step 9 above by fdt tool.

# LegatoFS authentication

1. Please note that you must finish the above bootloader and kernel authentication before starting to sign the legato image as the RootFS/LegatoFS authentication requires bootloader/Kernel authentication.
2. Copy your new key pair to correct location

cp mykey.pk8 security/verity.pk8

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

1. Generate LegatoFS keystore CWE (LGT0-keys.cwe):

$ANDROID\_SIGNING\_DIR/swi-key-cwe.sh security/verity.x509.pem 9X28 LGT0

1. Copy files to current signing folder

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/legato-af/git-r0/recipe-sysroot-native/usr/bin/hdrcnv ./

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/legato-af/git-r0/legato-af/build/wp76xx/legatoimg/ubi/ubinize.cfg ./

(if your yocto build\_bin folder does not have legato-af folder, you can use this command “cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/cwetool-native/usr/bin/hdrcnv ./” and use the following ubinize.cfg file )



1. Copy unsigned root hash from legato build folder to current signing folder

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/legato-image/1.0-r0/images-wp76xx/rhash-unsigned.bin ./

(if you are building the legato image from LEAF, you can use this command “cp /home/owner/LEAF/WP76/leaf-data/WP76\_stable/wp76-legato/build/wp76xx/rhash.bin ./rhash-unsigned.bin” )

1. Open the ubinize.cfg which is in current signing folder, in line 3 and line 9, change the directory to be the same as the directory where you copy the unsigned root hash in previous step.

(if you are building the legato image from LEAF, you need to use this directory “ /home/owner/LEAF/WP76/leaf-data/WP76\_stable/wp76-legato/build/wp76xx/” )



1. Open the ubinize.cfg which is in current signing folder, in line 15, change to current directory where we will generate the new rhash.bin in current directory in next step



1. Sign the LegatoFS

chmod 777 ./hdrcnv

chmod 777 ./android\_sign.sh

./android\_sign.sh

(need to enter your password set)

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



1. Load the signed legato image (legato-image-usrsigned.wp76xx.cwe) to module by FDT

fdt2.exe -f C:\Users\jyi\Desktop\signing\_dir\_R16\legato-image-usrsigned.wp76xx.cwe

1. Download the LegatoFS keystore CWE image (LGT0-keys.cwe) 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\_R16\LGT0-keys.cwe

1. In case downloading a not-signed legato image on this secure boot enabled module with legatoFS authentication, we will see the following in command prompt:



You need to re-download the signed legato image by fdt tool.

# RootFS authentication

1. Please note that you must finish the above bootloader and kernel authentication before starting to sign the RootFS image as the RootFS/LegatoFS authentication requires bootloader/Kernel authentication.
2. Copy your new key pair to correct location

cp mykey.pk8 security/verity.pk8

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

1. Generate RootFS keystore CWE (RFS0-keys.cwe):

$ANDROID\_SIGNING\_DIR/swi-key-cwe.sh security/verity.x509.pem 9X28 RFS0

1. Copy files to current signing folder

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/legato-af/git-r0/recipe-sysroot-native/usr/bin/hdrcnv ./

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/mdm9x28-image-minimal-swi-mdm9x28-wp-\*.rootfs.squashfs.ubinize.cfg ./ubinize\_rootfs.cfg

(if your yocto build\_bin folder does not have legato-af folder, you can use this command instead “cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/cwetool-native/usr/bin/hdrcnv ./” )

1. Copy required images from build folder to output folder in current signing folder

mkdir output

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/mdm9x28-image-minimal-swi-mdm9x28-wp.squashfs ./output

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/rootfs.hash ./output

cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/deploy/images/swi-mdm9x28-wp/rootfs.rhash.unsigned ./output

1. Open the ubinize\_rootfs.cfg which is in current signing folder, in line 3 and 11, change them to the output directory:

[sysfs\_volume]

mode=ubi

image=output/mdm9x28-image-minimal-swi-mdm9x28-wp.squashfs

vol\_id=0

vol\_type=static

vol\_name=rootfs

vol\_alignment=1

[hash\_volume]

mode=ubi

image=output/rootfs.hash

vol\_id=1

vol\_type=static

vol\_name=rootfs\_hs

vol\_alignment=1

1. Open the ubinize\_rootfs.cfg which is in current signing folder, in line 19, change it to the output directory where we will generate the new rootfs.rhash in next step

[rh\_volume]

mode=ubi

image=output/rootfs.rhash

vol\_id=2

vol\_type=static

vol\_name=rootfs\_rhs

vol\_alignment=1

1. Sign the RootFS

chmod 777 ./hdrcnv

chmod 777 ./rootfs-sign.sh

./rootfs-sign.sh

(need to enter your password set)

1. Combine the signed bootloader, kernel and rootFS images as yocto\_signed.cwe
* cp /home/owner/Yocto/WP76/R16/meta-swi/common/recipes-core/cwetool/cwetool/yoctocwetool.sh ./
* cp /home/owner/Yocto/WP76/R16/build\_bin/tmp/sysroots-components/x86\_64/cwetool-native/usr/bin/hdrcnv ./hdrcnv\_cwetool
* open yoctocwetool.sh in local directory, in line 54, change it to

HDRCNV=./hdrcnv\_cwetool

* chmod 777 ./hdrcnv\_cwetool
* echo "SWI9X07Y\_02.37.10.00" > version\_file.txt
* ./yoctocwetool.sh -pid '9X28' -platform '9X28' -o yocto\_signed.cwe -fbt appsboot.mbn -vfbt version\_file.txt -kernel boot-yocto-mdm9x28.4k.img -vkernel version\_file.txt -rfs output/fs.ubi -vrfs version\_file.txt

(Please note that in this yocto\_signed.cwe, bootloader, kernel and rootFS part are all signed with your certificate)

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



1. Load the signed yocto\_signed.cwe to module by FDT

fdt2.exe -f C:\Users\jyi\Desktop\signing\_dir\_R16\yocto\_signed.cwe

1. Download the RootFS keystore CWE image (RFS0-keys.cwe) 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\_R16\RFS0-keys.cwe

1. In case downloading the yocto.cwe which only have bootloader and kernel signed but without RootFS signed with your certificate, we will see the following in command prompt:



You need to re-download the signed yocto\_signed.cwe image by fdt tool.