SB File Preparation and Usage on LPC556xx 1B Revision

I already had several customers met SB file loading error **Inject command 'receive-sb-file'** while working on LPC55S6xx 1B revision. There is significant change of Secure Boot in LPC55S6xx 1B version and 0A version. To solve this problem, we need to understand it first.

1. SB2.1 vs. SB2.0

SB2 container is described in elftosb User's Guide. SB file config file contains configuration commands that will be processed after SB2 file is loaded in the device. The image location is stated in the "sources" .bd file section. SB key in text file is used for encryption with elftosb command line tool.

The 0A version of the LPC55S6xx silicon supports version 2.0 of the SB image format.
The 1B version of the LPC55S6xx silicon supports version 2.1 of the SB image format.
The main difference between version 2.0 and version 2.1 is in the usage of the digital signature.
SB 2.0 is e**ncrypted** and SB2.1 is **encrypted + signed.**

2. SB file Preparation and Usage

Example of use (Encrypted SB):

```
elftosb -f lpc55xx -k "sbkek.txt" -c "commandFile.bd" -o "output.sb2" "input.bin"
```

where
-f = family lpc55xx
-k = path to KEK file (SBKEK)
-c = path to command file to be processed:

```
options {
flags = 0x4; // 0x8 encrypted + signed, 0x4 encrypted
buildNumber = 0x1;
productVersion = "1.00.00";
componentVersion = "1.00.00";
}
```

```
sources {
inputFile = extern(0);
}
section (0) {
      erase 0x0..0x40000;
load inputFile > 0x0;
}
```

-o = path to output file

files… = path to files (usually image files), which will be replacing placeholders defined in command file, paths can be hardcoded in command file and then not inserted as input

Example of use (Encrypted + Signed SB):

1 root key

```
elftosb.exe -f lpc55xx -k "sbkek.txt" -c "commandFile.bd" -o "output.sb2" -s
"selfsign_privatekey_rsa2048.pem" -S
"selfsign_v3.der.crt" -R "selfsign_v3.der.crt" -h "RKTH.bin" "input.bin"
```

4 root keys

```
elftosb.exe    -f lpc55xx -k "sbkek.txt" -c "commandFile.bd" -o "output.sb2" -s
private_key_1_2048.pem -S certificate_1_2048.der.crt -R certificate_1_2048.der.crt -
R certificate_2_2048.der.crt -R certificate_3_2048.der.crt -R certificate_4_2048.der.crt
-h "RHKT.bin" "input.bin"
```

where
-f = family lpc55xx
-k = path to KEK file (SBKEK)
c = path to command file to be processed

```
options {
flags = 0x8; // 0x8 encrypted + signed, 0x4 encrypted
buildNumber = 0x1;
productVersion = "1.00.00";
componentVersion = "1.00.00";
}
```

```
sources {
inputFile = extern(0);
}
section (0) {
      erase 0x0..0x40000;
load inputFile > 0x0;
}
```

-o = path to output file
-s = path to private key of certificate used for signing
-S = path(s) to certificates in certificate chain, each certificate in chain must be specified with new -S switch in order of how was chain created (root certificate first)
-R = path(s) to root certificate(s), 1-4 root certificates can be specified, each root certificate must be specified with new -R switch, one of the root certificates must be first certificate specified by -S switch
-h = path and name of output binary file generated by elftosb, which contain hash of hashes of all root certificates (RKTH), which must be uploaded to the device register
files… = path to files (usually image files), which will be replacing placeholders defined in command file, paths can be hardcoded in command file and then not inserted as input
The SB2.0 file created with the updated binary image can be loaded into the device through ISP command handler with command “receive-sb-file“

```
blhost -p COMxx receive-sb-file <path to the secured binary(.sb2)>
```

The SB2.1 file created with the updated binary image can be loaded into the device through ISP command handler with command “receive-sb-file“ but keep in mind that before sending SB2.1 file into device has to be there already RKTH in CMPA (see AN12283 chapter 5.5 CMPA preparation) and enabled RoT keys in ROTKH_REVOKE field at CFPA page address 0x9DE18 (see chapter AN12283 5.4 CFPA preparation).

```
blhost -p COMxx receive-sb-file <path to the secured binary(.sb2)>
```

After successfully loading the SB2 file it is executed as configured in SB configuration file (.bd file). The above figure shows an example of SB configuration file. When the file is executed, the internal flash address from 0x0 to 0x40000 is erased. After flash erase operation, the image mentioned in the

sources parameter is loaded to address 0x0.
Reset the device after these operations. The updated image loaded into internal flash starts to execute.