

Blocksign Tool

Building an Image

To build a blocksigned binary, you will need two files:

- Input Binary
- Config File (XML)

Config File

The config file is an XML file containing the parameters necessary to build the blocks. Included with the tool is a sample config file. Future releases will contain a GUI to allow easy creation of this file.

There are a lot of configurable parameters, please refer to the PFR HAS for details. The parameters you should change are highlighted below:

NOTE: All file paths within the config file should be relative to the directory in which the config file resides.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML file for Block Sign Tool -->
<blocksign>
  <!-- Version -->
  <version>1</version>
  <!-- Block 0 -->
  <block0>
    <magic>0xB6EAFD19</magic>
    <pctype>3</pctype>
  </block0>
  <!-- Block 1 -->
  <block1>
    <magic>0xF27F28D7</magic>
    <!-- Root key -->
    <rkey>
      <magic>0xA757A046</magic>
      <curvemagic>0xC7B88C74</curvemagic>
      <permissions>-1</permissions>
      <keyid>-1</keyid>
      <!-- Root Key Public Key File -->
      <pubkey>Secp256r1PublicKey</pubkey>
    </rkey>
    <!-- Code signing key -->
    <cskey>
      <magic>0x14711C2F</magic>
      <curvemagic>0xC7B88C74</curvemagic>
      <permissions>-1</permissions>
      <keyid>1</keyid>
      <pubkey>Secp256r1PublicKey</pubkey>
      <sigmagic>0xDE64437D</sigmagic>
      <hashalg>sha256</hashalg>
      <signkey>Secp256r1PrivateKey</signkey>
      <!--<script>./sign_external.sh</script>-->
```

```

</cskey>
<!-- Signature over Block 0 -->
<b0_sig>
  <magic>0x15364367</magic>
  <sigmagic>0xDE64437D</sigmagic>
  <hashalg>sha256</hashalg>
  <signkey>Secp256r1PrivateKey</signkey>
  <!-- <script>./sign_external.sh</script> -->
</b0_sig>
</block1>
<!-- CPLD Bitstream Specific -->
<padding>
  <!-- Pad block1 such that combined block length is 1024b -->
  <blockpad>1024</blockpad>
  <!-- Align total package to 128 bytes -->
  <align>128</align>
</padding>
<cpld>
  <swapbytes>true</swapbytes>
  <cpldsvn>15</cpldsvn>
</cpld>
</blocksign>

```

Key Cancellation

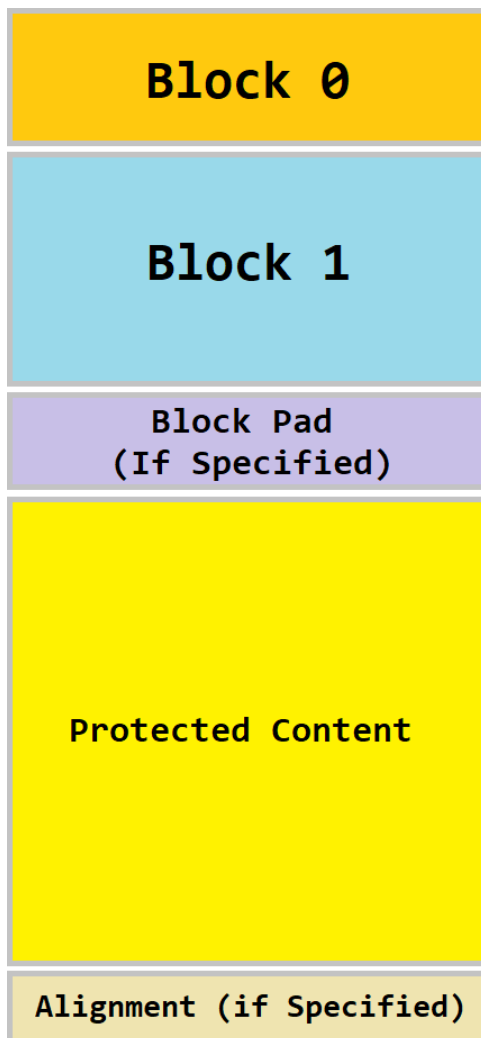
If no CSKEY is specified in Block 1, then the tool will assume that the key cancellation bit is to be set.

Definitions

All highlighted values are documented in the HAS except for the following:

- Pubkey – This is the public key file (PEM formatted) for the respective block.
- Hash Alg – The hashing algorithm used when signing
- Sign Key – If the blocksign tool is to perform signing, specify a private key here (otherwise remove)
- Script File – Use for external signing. The tool will generate two files:
 - Data.raw (raw data to be signed)
 - Data.hsh (the hash of raw data)
 - After these files are generated, the tool will execute the script. Once the script has completed, the tool will look for Data.sig. Data.sig is a DER encoded signature.
- Block Pad – This specifies the total length for Block 0 and Block 1. Example: If you specify 1024, and actual blocksize is 1000, the tool will pad 24 bytes (0x00) to the end of the blocks
- Align – This will perfectly byte align your blocksigned package. Alignment padding bytes are 0xFF
- Swap Bytes – This is a CPLD bitstream specific parameter. It will swap word endianness and reverse the bits in each byte.
- SVN – This is a SVN number to be used by the CPLD. The value is prepended to the bitstream.

Blocksigned binary:



Running the Tool

Signing a Binary

```
blocksign.exe -c config.xml -o out.bin in.bin
```

Config.xml contains all the parameters necessary to sign the input binary. Out.bin is the file to be generated. In.bin is the raw protected content. Also, you can pass a `-v` flag for verbosity.

Parsing and Verifying a Binary

```
blocksign.exe -p out.bin -c config.xml
```

Config.xml is not required to parse the output binary. However, if specified, the tool will read in the public keys, and test that the signatures verify correctly.