U-Boot: Verified RSA

Boot on ARM target

JagannadhaSutradharudu Teki
U-Boot Mini Summit- Edinburgh, 2013 Oct
Agenda

- Zynq U-Boot
- SPI Custodianship
- Verified Boot
- RSA Concept
- U-boot Verified RSA Boot
- Current u-boot state (Simon’s support)
- U-boot needs
- Demo run
- TODO
- References
Good customer support till now – feature additions SPI/QSPI, support new boards, d-caches and bug fixes

~75% of u-boot-xlnx code is in ML, rest will push soon.
SPI Custodianship

[u-boot.git] / doc / SPI / status.txt

1 Status on SPI subsystem:
2 ==========================
3
4 SPI COMMAND (common/cmd_sf, cmd_spi):
5 -
6
7 SPI FLASH (drivers/mtd/spi):
8  - sf_probe.c: SPI flash probing code.
9  - sf_ops.c: SPI flash operations code.
10  - st.c: SPI flash interface, which interacts controller driver.
11  - Bank Address Register (Accessing flashes > 16Mbytes in 3-byte addressing)
12  - Added memory_mapped support for read operations.
13  - Common probe support for all supported flash vendors except, ramtron.
14
15 SPI DRIVERS (drivers/spi):
16 -
17
18 TODO:
19  - Runtime detection of spi_flash params, SFDP(if possible)
20  - Add support for multibus build/accessing.
21  - Extended read commands support(dual read, dual IO read)
22  - Quad Page Program support.
23  - Quad Read support(quad fast read, quad IO read)
24  - Dual flash connection topology support(accessing two spi flash memories with single cs)
25  - Banking support on dual flash connection topology.
26  - Need proper cleanups on spi_flash and drivers.
Verified Boot

- Verified – Secure – Trusted boot
- Verify the loaded software to ensure that it is authorized during boot.
- Benefits:
  - Prevent from malware
  - Provide authorized read access
  - Machine safe – runs only signed software
  - Possible to filed-upgrade the software
RSA Concept

Signer

Verifier

openssl

.key .crt

Public Key

Private Key

signed_img

software image

Trusted source

Public Key

Yes/No
**U-boot Verified RSA Boot**

**Signing**
- openssl
- .key .crt
- Private Key

**Verification**
- u-boot dts
- Public Key

**mkimage**
- fit input
- software image
- signed_img

**U-boot Verification**
- OK/Bad Data Hash

Note: We can also use mkimage to write pubkey on dtb
<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Description</th>
<th>Branch</th>
<th>Commit</th>
<th>Tree</th>
<th>Snapshot</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-06-26</td>
<td>Dirk Behme</td>
<td>spil: mxc_spi: Fix pre and post divider calculation</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add verified boot information and test</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>sandbox: config: Enable FIT signatures with RSA</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>image: Add support for signing of FIT configurations</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>libbfd: Add fdt_find_regions()</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>mkimage: Add -r option to specify keys that must be</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>mkimage: Add -c option to specify a comment for key...</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>mkimage: Add -f option to modify an existing .fit file</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>mkimage: Add -k to write public keys to an FDT blob</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>mkimage: Add -k option to specify key directory</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>image: Add RSA support for image signing</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>image: Support signing of images</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>image: Add signing infrastructure</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>x86: config: Add tracing options</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>x86: Support tracing function</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>exynos: config: Add tracing options</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>exynos: Avoid function instrumentation for microsecond...</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>arm: Implement the 'take' go command</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add a 'take' go command to the boot command</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Refactor the bootm command to reduce code duplication</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Clarity bootm OS arguments</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add a simple test for sandbox trace</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>sandbox: Support trace feature</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add printf to decode profile data</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add trace support to generic board</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Support tracing in config.mk when enabled</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add a trace command</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add trace library</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Add function to print a number with grouped digits</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>bootstage: Correct print types</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Show stdout on error in fit-test</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
<tr>
<td>2013-06-26</td>
<td>Simon Glass</td>
<td>Fix missing return in do_mem_loop()</td>
<td></td>
<td>commit</td>
<td>tree</td>
<td>snapshot</td>
</tr>
</tbody>
</table>
U-boot needs

- **Enable FIT**
  - `CONFIG_FIT` - enable support for the FIT uImage format

- **Enable FDT**
  - `CONFIG_OF_CONTROL`
  - `CONFIG_OF_SEPARATE`

- **Enable verified boot**
  - `CONFIG_FIT_SIGNATURE` - enables signature verification of FIT images
  - `CONFIG_RSA` - enables the RSA algorithm used for FIT image verification
Demo run...

- Build FDT u-boot
- Build rsa_signed image
- Build FDT u-boot with public key
- Run rsa_signed image
Build FDT u-boot

- Setup the toolchain:
  http://www.wiki.xilinx.com/Zynq+Base+TRD+14.5#x-5 Building the U-boot Boot Loader

- Clone u-boot-spi.git
  
  $ git clone git://git.denx.de/u-boot-spi.git
  
  $ cd u-boot-spi
  
  $ git checkout -b master-xlnx origin/master-xlnx

- U-boot build
  
  $ make zynq_zed_config
  
  $ make DEVICE_TREE=zynq-zed -j4
/dts-v1/
/
{ description = "Simple image with single Linux kernel, FDT blob and ramdisk";
 #address-cells = <8x1>;
 images {
   kernel@0 {
     description = "Zynq Linux kernel";
     data = /incbin("./vmlinux.bin.gz");
     type = "kernel";
     arch = "arm";
     os = "linux";
     compression = "gzip";
     load = <0x800000>;
     entry = <0x800000>;
     hash01 {
       algo = "sha1";
     };
     signature01 {
       algo = "sha1,rsa2048";
       key-name-hint = "dev";
     };
   };
   fdt@0 {
     description = "ZED board Flattened Device Tree blob";
     data = /incbin("./devicetree.dtb");
     type = "flat dt";
     arch = "arm";
     compression = "none";
     hash01 {
       algo = "sha1";
     };
     signature01 {
       algo = "sha1,rsa2048";
       key-name-hint = "dev";
     };
   };
   ramdisk@0 {
     description = "Ramdisk Image";
     data = /incbin("./ramdisk.image.gz");
     type = "ramdisk";
     arch = "arm";
     os = "linux";
     compression = "gzip";
     load = <0x80000000>;
     entry = <0x80000000>;
     hash01 {
       algo = "sha1";
     };
     signature01 {
       algo = "sha1,rsa2048";
       key-name-hint = "dev";
     };
   };
   configurations {
     default = "conf@0":
     conf01 {
       description = "Boot Linux kernel, FDT blob and ramdisk";
       kernel = "kernel@0";
       fdt = "fdt@0";
       ramdisk = "ramdisk@0";
     };
   };
};
RSA key generation:
- Create RSA key pair
  $ openssl genrsa -F4 -out mykeys/dev.key 2048
- Create a certificate contains public key
  $ openssl req -batch -new -x509 -key mykeys/dev.key -out mykeys/dev.crt

Create dtb for existing u-boot dts
$ dtc -p 0x1000 board/xilinx/dts/zynq-zed.dts -O dtb -o zynq-zed.dtb
$ cp zynq-zed.dtb zynq-zed-pubkey.dtb

Sign the images with mykeys
$ DTC_OPS="-I dts -O dtb -p 2000"
$ mkimage -D "${DTC_OPS}" -f rsa.its -K zynq-zed-pubkey.dtb -k mykeys -r rsa_signed.img
Build FDT u-boot with public key

- For building FDT u-boot with public key- externally
  $ make DEV_TREE_BIN=./zynq-zed-pubkey.dtb

u-boot-dtb.bin -> Is final FDT u-boot image with public key on it, hence the pubkey will used in verification process.
zyq-uboot> boot 0x2000000
## Loading kernel from FIT Image at 02000000 ... 
Using 'conf@1' configuration
Verifying Hash Integrity ... OK
Trying 'kernel@1' kernel subimage
  Description: Zynq Linux kernel
  Type: Kernel Image
  Compression: gzip compressed
  Data Start: 0x8200000
  Data Size: 2972178 Bytes = 2.8 MiB
  Architecture: ARM
  OS: Linux
  Load Address: 0x00008000
  Entry Point: 0x8000000
  Hash algo: md5
  Hash value: 3601ae0d79bd62a71a43e72886a41d24
  Hash algo: sha1
  Hash value: 5c18a3632e8399349d9ea6d42e3e9fa861d5193
  Sign algo: sha1,rsa2048:dev
  Sign value: 1b6d3d3e6c027783626779f8fa4bebaed46d97d4d3ce4ce43f10aff4e79da2a796c84619806e6a8d7ae17
  65d67d9a34f21378a84a6a1c1f7cc74d6ee9c7a619b7a636508b8c8cffe73dcb155dcd5b262c1cb9582e4d2cfd85315c701dc53
  5ae0c93e56d9e6e5c7b334aedc73e13e75d45c5d9bc92064b8d342b37a0f9c34bbbab724256e9fac5b9b3375e0c8cd9334a6
  4d3ef521b51306ae66e73802961c8e150d2a8a86c9b58d8f7447e1f1083d2d2542231579f40aae89456d9b0d9a8b50bed8e8f6
  43369425c60ab41be2aad89df5918a5a8882daca5a21313f4b2f54376d1ff4229c9587bedd95c7cc2bf448237b72372c5c5795194c56c372d
Verifying Hash Integrity ... sha1,rsa2048:dev+ md5+ sha1+ OK
## Loading ramdisk from FIT Image at 02000000 ... 
Using 'conf@1' configuration
Trying 'ramdisk@1' ramdisk subimage
  Description: RAMDisk Image
  Type: RAMDisk Image
  Compression: gzip compressed
  Data Start: 0x822d7cf8
  Data Size: 3688961 Bytes = 3.5 MiB
  Architecture: ARM
  OS: Linux
  Load Address: 0x00008000
Todo

- Possible TODO's @ doc/uImage.FIT/signature.txt
- Signed_image creations support for bootable images (SPL) or FIT support in SPL ??
References

- Zynq u-boot-xlnx.git repo
  https://github.com/Xilinx/u-boot-xlnx
- For verified boot: doc/uImage.FIT/verified-boot.txt
- For signature: doc/uImage.FIT/signature.txt
- Sample sign its: doc/uImage.FIT/sign-configs.its
- Code for this demo run
  http://git.denx.de/?p=u-boot/u-boot-spi.git;a=shortlog;h=refs/heads/master-xlnx
- Possible TODO’s on doc/uImage.FIT/signature.txt
- Any questions - mail to sjg@chromium.org CC u-boot@lists.denx.de, jagannadh.teki@gmail.com