

Anrroid R Hailstorm 4.1.1 SDK Mandatory Patch List

avb2_testkey_replace P1 readme

- Note
 - Download patch
 - For customer
 - `git clone ssh://git@source.amlogic.com/faq-mandatory-patch-b r-amlogic-hailstorm4.1.1-patch`
 - For AML FAE:
 - `git clone ssh://source.amlogic.com:29418/faq-mandatory-patch-b r-amlogic-hailstorm4.1.1-patch`
 - `git pull`
 - How to apply patches
 - Example code, need manually merge the required commits according chip used on project.

• Patch List

	Patch Number	Patch Level	Patch Content	
1	avb2_testkey_replace	P1	Jira	0TT-60127
			commit	avb2_testkey_replace/patches/
			chip type	All Chips
			root cause	Some project need replace AVB2 testkey by OTA
			how to fixed	Add example patch for replace AVB2 testkey without reset device.
			Side effect-range	No.

Test
recommendat
ion

Example include example patch for both v2015 and v2019, need choose patch base on bootloader version on your project.

Apply patch `avb2_testkey_replace/patches/bootloader/uboot-repo/bl33/v201X/0001-PATCH-debug-boot_param.patch`, check avb status by debugging boot params. (On existing(old) firmware).

```
4096 bytes read in 1 ms (3.9 MiB/s)
system_mode in storeboot: 0
active_slot in storeboot: _a
[imgread]read from part: boot_a
[imgread]read from part: boot_a
[imgread]partname_r = vendor_boot_a
[imgread]read from part: vendor_boot_a
[imgread]read from part: vendor_boot_a
avb2: 1
active_slot is _a
ab_suffix is _a
AVB2 verify with default kpub:520, vbmeta kpub:520
avb verification: locked = 1, result = 0
device_locked = 1.
verified_boot_state = 0.
verified_boot_key
7728e30f50bfa5cea165f473175a08803f6a8346642b5aa10913e9d9e6defef6
verified_boot_hash
8bb439842a228eb12d3ea345db35933ccb7ca925523bed00a7c61a60081349ca
avb2: 1
```

Apply `avb2_testkey_replace/patches/bootloader/uboot-repo/bl33/v201X/0002-avb-force-test-key-1-1.patch` and change `test_key_hash` with `verified_boot_key`

```
uboot / lib/libavb/avb_slot_verify.c Patch Set Base 1 16 Patch Set 1 16
... skipped 20 common lines ...
21 #define MAX_NUMBER_OF_LOADED_PARTITIONS 32
22
23 /* Maximum number of vbmeta images that can be loaded with avb_slot_verify(). */
24 #define MAX_NUMBER_OF_VBMETA_IMAGES 32
25
26 /* Maximum size of a vbmeta image - 64 KiB. */
27 #define VBMETA_MAX_SIZE (64 * 1024)
28
29 uint8_t boot_key_hash[AVB_SHA256_DIGEST_SIZE];
30
31 static AvbSlotVerifyResult initialize_persistent_digest(
32     AvbOps* ops,
33     const char* part_name,
34     const char* persistent_value_name,
35     size_t digest_size,
36     const uint8_t* initial_digest,
37     uint8_t* out_digest);
38
39 #define MAX_NUMBER_OF_LOADED_PARTITIONS 32
40
41 /* Maximum number of vbmeta images that can be loaded with avb_slot_verify(). */
42 #define MAX_NUMBER_OF_VBMETA_IMAGES 32
43
44 /* Maximum size of a vbmeta image - 64 KiB. */
45 #define VBMETA_MAX_SIZE (64 * 1024)
46
47 uint8_t boot_key_hash[AVB_SHA256_DIGEST_SIZE];
48
49 #define FORCE_TEST_KEY (1)
50 uint8_t test_key_hash[AVB_SHA256_DIGEST_SIZE] = {
51     0x22, 0xde, 0x39, 0x94, 0x53, 0x21, 0x96, 0xf6,
52     0x1c, 0x03, 0x9e, 0x90, 0x26, 0x0d, 0x78, 0xa9,
53     0x3a, 0x4c, 0x57, 0x36, 0x2c, 0x7e, 0x78, 0x9b,
54     0xe9, 0x28, 0x0f, 0x6e, 0x80, 0xb7, 0x7c, 0x8c,
55 };
56
57 static AvbSlotVerifyResult initialize_persistent_digest(
58     AvbOps* ops,
59     const char* part_name,
60     const char* persistent_value_name,
61     size_t digest_size,
62     const uint8_t* initial_digest,
63     uint8_t* out_digest);
```