S32G Host Secure Debug

by Yuan Yuan (nxa22694)

本文说明S32G在Password 或 Challenge&Response 模式下用ADKP或者ADKPm的key时，secure debug的实现方法，基于Lauterbach tool调试器来实现。

历史	说明	作者
V1	● 创建本文	● Yuan Yuan

目录

1 参考资料 2
1.1 参考资料 2
1.2 版本匹配说明 2
2 Host Secure Debug相关概念 3
2.1 ADKP 3
2.2 ADKPm 4
2.3 如何使用HSE FW来对ADKP进行编程 5
2.4 Host Debug 5
3 Host Secure Debug的具体实现 6
3.1 直接写入ADKP 6
3.2 通过ADKPm计算ADKP并写入 7
3.3 设置授权模式 7
3.4 演进LC 8
3.5 调试器相关设置 8

1 参考资料

涉及的文档和相关软件包，S32DS 的工具，Lauterbach 的 工具等。

1.1 参考资料

以 S32G3 RDB3 为例：

序号	资料	说明	如何获取
1	HSE_DEMOAPP_S32G3XX_0_2_16_1_ReadMe.pdf	HSE DEMO 手册	HSE_DEMOAPP_S32G3XX_0_2_16_1.exe 软件包中安装获取
2	HSE_FW_S32G3_0_2_16_1 ● HSE_FW_S32G3_0_2_16_1.exe	HSE 固件安装包	NXP.COM 官网下载
3	HSE_DEMOAPP_S32G3XX_0_2_16_1 ● HSE_DEMOAPP_S32G3XX_0_2_16_1.exe	HSE DEMO	NXP.COM 官网下载
4	SW32G_RTD_4.4_3.0.0_HF02 ● SW32G_RTD_4.4_3.0.0_HF02_D2205.exe	RTD	NXP.COM 官网下载
5	Volkano ● S32DS3.4 版本中自带	Key 管理工具	S32DS 自带
6	Trace32 Powerview 工具 ● Lauterbach debug 工具 Software Version: N.2022.05.000147980 Build: 147980.	调试工具	Lauterbach.com 下载

1.2 版本匹配说明

上述软件版本只是参考，具体以各个不同版本的 HSE_DEMOAPP 中的 readme 为准。注意一点的是，S32G2 Rev2.0 和 Rev2.1 是用不同版本的 FW，同样，S32G3 Rev1.0 和 Rev1.1 也是用不同版本的 FW，用错会导致 HSE 不能正常工作。

2 Host Secure Debug 相关概念

2.1 ADKP

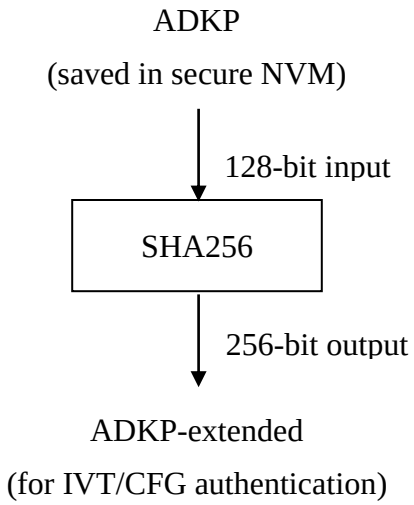
ADKP是Application Debug Key / Password的缩写，是SOC的一次性可编程参数，128bit。一旦写入不能修改，不能读取，只能通过HSE来使用。常见用法是

- 用来计算IVT/CFG的GMAC
- 用来计算AppBL的GMAC
- 演进生命周期到OEM_PROD/IN_FIELD，用来保护Debug接口
- ……

ADKP可以是明文也可以密文，用hseAttrApplDebugKey_t 或hseAttrSecureApplDebugKey_t 结合同样的设置属性服务（HSE_APP_DEBUG_KEY_ATTR_ID）来设置ADKP明文或密文。密文的密钥来自HSE的RAM/NVM key，AES-128 key。

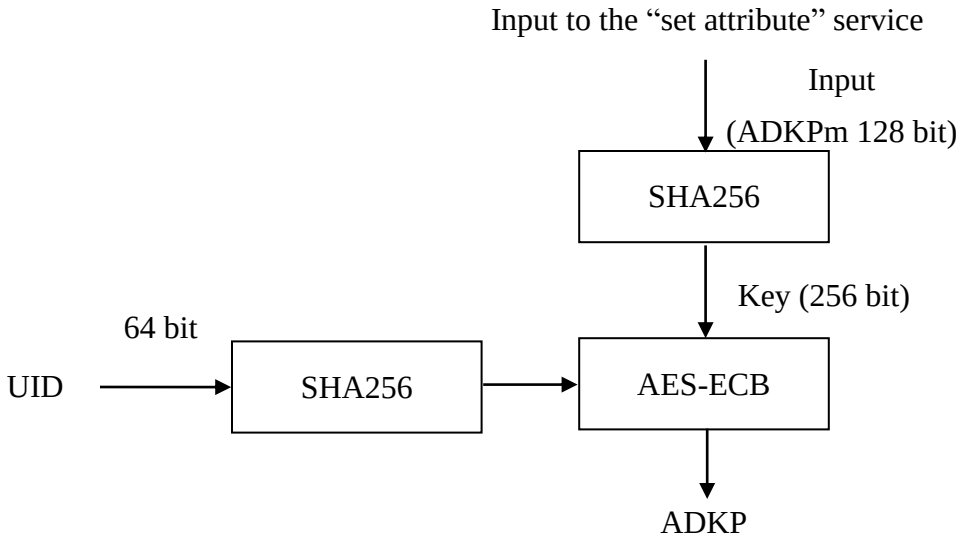
128Bit的ADKP用作GMAC，按如下方式来扩展到256Bit。GMAC Tag的计算方法如下：

GMAC TAG = GMAC(random_IV, message:IMAGE, key:SHA256(ADKP))



2.2 ADKPm

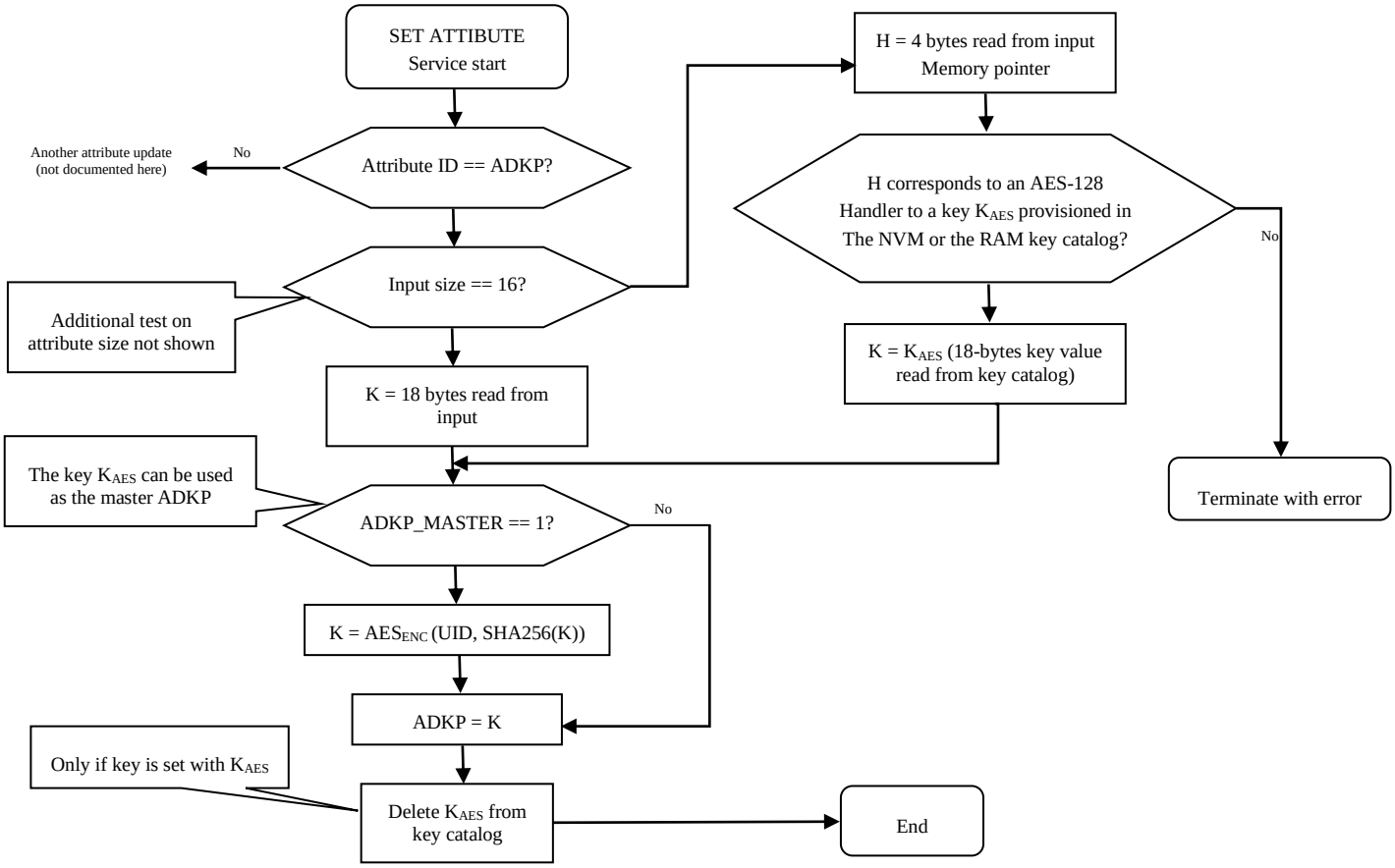
ADKPm，即 ADKP master，如下图中的 128Bit 的数，与 UID 一起通过 AES-ECB 来生成一个 ADKP。这样的好处是当 ADKPm 唯一的时候，可以通过 UID 来保证一机一密。



在烧写ADKPm前，需要先通过属性服务（HSE_EXTEND_CUST_SECURITY_POLICY_ATTR_ID）来使能 ADKP_MASTER，然后通过上述 ADKP 编程方式来完成烧写。

ADKP_MASTER	1 bit	Selects the method to provision ADKP in secure NVM: - When 0 (default): the input value is ADKP and is written “as is” in secure NVM - When 1: the input value is considered as a master debug key and is diversified with the device’s UID before being written in secure NVM

2.3 如何使用 HSE FW 来对 ADKP 进行编程



2.4 Host Debug

Host Debug 功能要么完全开放，要么被保护，取决于 LC （Life Cycle，生命周期）的状态，具体如下表格所描述。这种保护会把仿真器的 JTAG 调试接口关闭，直到 HSE 正确授权给仿真器。本文不讨论 DEBUG DISABLE 的情况。

LC state	Host Debugging
CUST_DEL	Host debug open (unrestricted)
OEM_PROD IN_FIELD	Host debug protected (with ADKP) or permanently disabled (see DEBUG_DISABLE)
PRE_FA mode	Host debug protected (with ADKP) or permanently disabled (see DEBUG_DISABLE)
FA	Host debug open

这个授权就是基于之前提到的 128Bit ADKP/ADKPm，授权方式包括静态及动态两组方式，通过 HSE 系统属性 AUTH_METHOD 来配置。

AUTH_MODE	1 bit	Selects the method to open the host debug protection: - When 0 (default): static authentication (password) - When 1: dynamic authentication (challenge / response)

- 静态方式（password模式）：通过password来授权，这样ADKP会通过明文的方式放到仿真器中。这种方式实现简单，但安全性不够。
- 动态方式（challenge/response模式）：通过challenge/response这种方式来授权，ADKP是一个加密key，仿真器使用这个key来针对随机的challenge计算加密的response。由于这种方式不对仿真器提供任何方式的明文，安全级别会更高，推荐使用这种方式。

3 Host Secure Debug 的具体实现

一般需要如下几个步骤：

- 配置HSE相关，包括key catalog，安装key等sysimg相关配置。
- 写入ADKP/ADKPm。
- 设置授权模式，默认是password模式。
- 演进LC（life cycle）。
- 重启（destructive reset）
- 使用相应的仿真器脚本来调试板子。

下面基于HSE_DEMOAPP_S32G3XX_0_2_16_1，对上述关键步骤进行说明。

3.1 直接写入 ADKP

这部分代码在 HSE demoapp 中有，调用设置属性的 HSE_APP_DEBUG_KEY_ATTR_ID 服务即可，这里需要注意 ADKP，这 128Bit 数据的大小端关系，HSE 是大端处理的。写入到 HSE 中如果不做 swap，在 debugger 脚本中需要做 swap。

```
uint8_t applicationDebugKeyPassword[16] =
{
    0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE,
0xFF
};


srvResponse = HSE_SetAttribute(HSE_APP_DEBUG_KEY_ATTR_ID,
        sizeof(hseAttrApplDebugKey_t), &applicationDebugKeyPassword[0]);
    ASSERT(HSE_SRV_RSP_OK == srvResponse);
```

3.2 通过 ADKPm 计算 ADKP 并写入

根据 ADKPm 的使用场景，需要先读取该芯片的 UID，可以使用读 OCOTP 对应的寄存器，如下 Shadows4& Shadows5，然后 swap 后得到 UID= D8B363550B101002。当然也可以用仿真器脚本文件来获取，获取到的 UID 不需要 swap。

SHADOWS4	5563B3D8	EFUSES	5563B3D8
SHADOWS5	0210100B	EFUSES	0210100B
SHADOWS6	00002100	EFUSES	00002100

接着，如果 ADKPm 是 00112233445566778899AABBCCDDEEFF，则通过 ADKP 的计算方式，利用 S32DS 中的 volkano_utils（c:/NXP/S32DS.3.4/S32DS/tools/S32Debugger/Debugger/Server/CCS/bin）可以计算出 ADKP，如下，黄色高亮部分就是最终的 ADKP。这样，在 ADKPm 相同的情况下，实现了一机一密的操作。这里的 ADKPm，在用代码写入时使用。ADKP 则是 secure debug 时使用的。

```
$ ./volkano_utils.exe -cmd derive_adkp -uid D8B363550B101002 -adkpm
00112233445566778899AABBCCDDEEFF
9734D6AF0319E4867DC5257B7AA84503
```

在代码实现时，需要先写入 ADKP_MASTER 使能位，再写入 ADKPm。

```
uint8_t applicationDebugKeyPassword[16] =
{
    0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE,
0xFF
};
```

```
    hseAttrExtendCustSecurityPolicy_t test;
    test.enableADKm = 1;
    test.startAsUser = 0;
       srvResponse = HSE_SetAttribute(HSE_EXTEND_CUST_SECURITY_POLICY_ATTR_ID,
            sizeof(hseAttrExtendCustSecurityPolicy_t), &test);
        ASSERT(HSE_SRV_RSP_OK == srvResponse);

    srvResponse = HSE_SetAttribute(HSE_APP_DEBUG_KEY_ATTR_ID,
        sizeof(hseAttrApplDebugKey_t), &applicationDebugKeyPassword[0]);
    ASSERT(HSE_SRV_RSP_OK == srvResponse);
```

3.3 设置授权模式

默认说明也不操作的情况下是 Password 模式，需要改成 CR 模式的话，设置属性 HSE_DEBUG_AUTH_MODE_ATTR_ID 为 1 即可。

```
debugAuthMode = HSE_DEBUG_AUTH_MODE_CR; //1
srvResponse = HSE_SetAttribute(HSE_DEBUG_AUTH_MODE_ATTR_ID,
    sizeof(hseAttrDebugAuthMode_t), &debugAuthMode);
ASSERT(HSE_SRV_RSP_OK == srvResponse);
```

3.4 演进 LC

LC 生命周期演进有如下三种情况，

HSE_LC_CUST_DEL -> HSE_LC_OEM_PROD
HSE_LC_CUST_DEL -> HSE_LC_IN_FIELD
HSE_LC_OEM_PROD -> HSE_LC_IN_FIELD

分别对应到属性设置上，三个周期对应的值如下，

```
#define HSE_LC_CUST_DEL                 ((hseAttrSecureLifecycle_t)0x4U)
#define HSE_LC_OEM_PROD                 ((hseAttrSecureLifecycle_t)0x8U)
#define HSE_LC_IN_FIELD                 ((hseAttrSecureLifecycle_t)0x10U)
```

在演进 LC 时，设置 HSE_SECURE_LIFECYCLE_ATTR_ID 属性即可，该属性也可以读出来作为是否设置成功的判断，具体代码如下：

```
/* Advance the Life Cycle to target value */
    lifeCycleToSet = targetLifeCycle;
    srvResponse = HSE_SetAttribute(HSE_SECURE_LIFECYCLE_ATTR_ID,
        sizeof(hseAttrSecureLifecycle_t), &lifeCycleToSet);
    ASSERT(HSE_SRV_RSP_OK == srvResponse);
    /* Read the LC issuing a get attribute request to HSE */
    srvResponse = HSE_GetAttribute(HSE_SECURE_LIFECYCLE_ATTR_ID,
        sizeof(hseAttrSecureLifecycle_t), &gHseCurrentLC);
    ASSERT( (HSE_SRV_RSP_OK == srvResponse) && (gHseCurrentLC == targetLifeCycle));
```

3.5 调试器相关设置

首先强调一下，secure debug 只能在 bootmod 是非 serial mode 的情况下才能有效，即 HSE 必须正常跑之后，secure debug 功能才能工作。

- Password 模式下：

  在 HSE_DEMOAPP_S32G3XX_0_2_16_1 工程的 config.h 中，使能如下宏。按 guide 生成带 IVT 头的 image 后，烧写到 flash 中。代码跑完后，ADKP 与 LC 就设置好了。

  - APP_CONFIG_LC_DEBUG_ACCESS
  - DEBUG_CONFIG_OPTION

- PROGRAM_AD_PASSWORD_ADVANCE_LC_TO_OEM_PROD
- PROGRAM_ADKP_M （ADKPm 下可选）

修改 T32\demo\arm\hardware\s32g2\s32g-vnp-evb\s32g-vnp-evb-m7\s32g-vnp-evb_sieve_sram_password.cmm 中的 keycode，注意需要 swap，如果 ADKP 是 00112233445566778899AABBCCDDEEFF 的话。

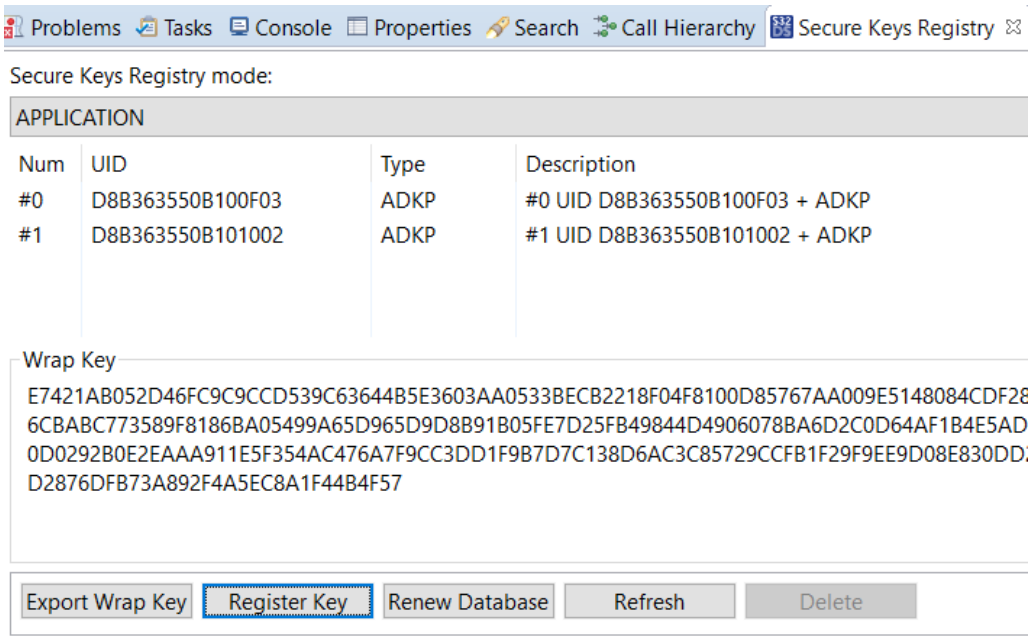
```
SYStem.Option KEYCODE 0x7766554433221100 0xFFEEDDCCBBAA9988
```

RDB 板上 bootmod 跳线到 0b10，否则会报"Failed connecting to the target using the password!"错误。

- CR 模式下：

用 S32DS 中的 Secure Keys Registey 注册 UID+ADKP，如下，#0 是 ADKP 方式下的，#1 是 ADKPm 方式下的。这里可以看到 ADKP 是不可见的，而 UID 是可见的，因为 UID 在任何时候都可以读到。

- #0：UID=D8B363550B100F03；ADKP=00112233445566778899AABBCCDDEEFF
- #1：UID=D8B363550B101002；ADKP= 9734D6AF0319E4867DC5257B7AA84503（来自于 ADKPm 的计算，ADKPm=00112233445566778899AABBCCDDEEFF）



在 HSE_DEMOAPP_S32G3XX_0_2_16_1 工程的 config.h 中，使能如下宏。按 guide 生成带 IVT 头的 image 后，烧写到 flash 中。代码跑完后，ADKP 与 LC 就设置好了。

- APP_CONFIG_LC_DEBUG_ACCESS
- DEBUG_CONFIG_OPTION
- PROGRAM_AD_KEY_SET_CHALLANGE_RESPONSE_ADVANCE_LC_TO_OEM_PROD
- PROGRAM_ADKP_M （ADKPm 下可选）

直接使用 s32g-vnp-evb_sieve_sram_challenge_response.cmm 即可，需要修改 volkano 在 cmm 中的路径（c:/NXP/S32DS.3.4/S32DS/tools/S32Debugger/Debugger/Server/CCS/bin），其他不需做任何改动，直接使用就可以了。

RDB 板上 bootmod 跳线到 0b10，否则会报"Failed connecting to the target using the password