Q/HMAC

一汽海马汽车有限公司企业标准

Q/HMAC 103.266-244-2013

UDS 诊断安全访问算法规范

2013-07-10 发布　　2013-07-15 实施

一汽海马汽车有限公司　发布

目　　次

前　　言

标准按照GB/T 1.1-2009给出的规则，并结合本企业实际情况而起草。

本标准由一汽海马汽车有限公司研发本部提出并归口。

本标准主要起草单位：研发本部电气车身开发部

本标准主要起草人：何烈炎、庄丽兴、陈启达

本标准主要校对人：梁杰、游立伟、梁友琼、王岩、庄丽兴、周建国、陈日高、陈元清、李智、程翔、符传兴、、张国颖、张运成、王文、王崇弟、韩伟

本标准审核人：蔡刚强

本标准审批人：李文、蔡锋

本标准所代替标准的历次版本发布情况为：

—首次发布。

UDS 诊断安全访问算法规范

1 范围

本标准规定了UDS诊断的安全访问算法规范。

本标准适用于UDS诊断要用到安全访问的所有诊断服务。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO 14229 统一诊断服务（UDS）

ISO 15031-5 排放相关的诊断服务

Q/HMAC 103.266-243-2013 UDS诊断DID规范

3 术语及定义

ISO 14229 、ISO 15031-5 界定的术语和定义适用于本文件。

3.1

安全访问

服务端对某些可能会影响到自身安全性的诊断操作设置密码保护，客户端只有计算出满足服务端要求的密钥并得到服务端的认可后，才可请求服务端执行这些受保护的诊断操作。客户端计算出密钥发送给服务端，并得到服务端响应的过程即是安全访问。

3.2

符号缩写的含义

CAN 控制器局域网

UDS 统一诊断服务

DID 数据标识符

4 安全访问算法

安全访问服务须应用于控制器的诊断服务，以确保控制器数据流及诊断控制的安全。

4.1 对象

使用安全访问服务的对象定义如下：

客户端：下线电控设备，售后诊断测试设备和其它诊断测试工具；

服务端：控制器。

4.2 参数

安全访问服务的参数定义如下：
种子：用于计算密钥的数值；
密钥：用于通过安全访问的数值。
种子和密钥的字节长度都是 4 个字节。

4.3 控制器安全状态

控制器安全状态定义如下：
锁定状态：控制器不支持需要通过安全访问后才能执行的诊断服务；
解锁状态：控制器支持需要通过安全访问后才能执行的诊断服务。

4.4 安全访问流程图

安全访问流程如下：
客户端请求“种子”;
服务端回复“种子”;
客户端发送“密钥”;
服务端回复密钥确认的结果（如果密钥正确，服务端将处于解锁状态）。
根据服务端不同的安全状态，安全访问流程会有所不同，下图表示的是服务端在锁定状态和解锁状态下的安全访问流程：

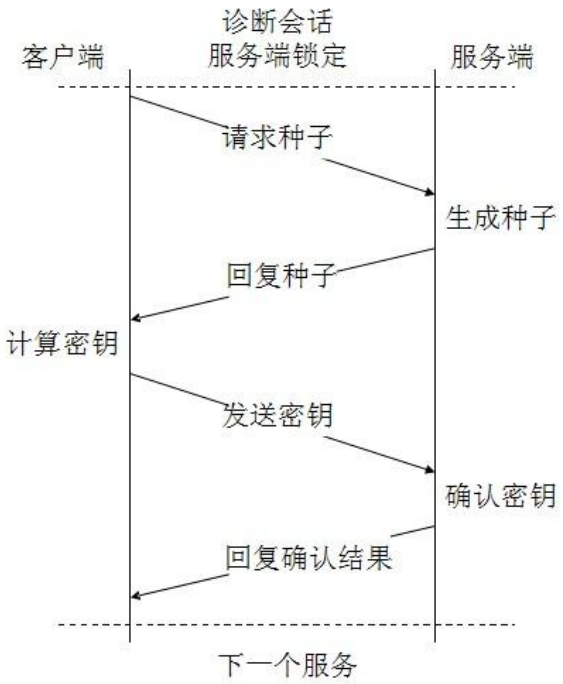


图 1 服务端处于锁定状态时的安全访问流程图

注：如果服务端回复的密钥确认结果为密钥无效，客户端若要通过安全访问，需要重新走一遍上图的流程。

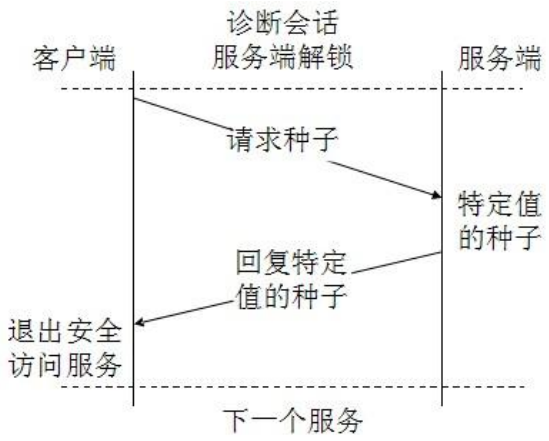


图 2 服务端处于解锁状态时的安全访问流程图

注：在解锁状态，服务端回复的种子是“00 00 00 00”。

4.5 种子的生成

种子的生成函数由供应商自定义。

建议按以下方式生成种子：

利用控制器的运行时间作为参数输入，通过随机函数生成种子。

注：种子“00 00 00 00”用于指示控制器处于解锁状态。

4.6 控制器密钥的解决办法

对于控制器的密钥，有两种解决办法：针对不同的种子，控制器可预先在内部存储对应的密钥，或通过密钥算法获取密钥。

建议控制器在生成种子后，通过密钥算法获取密钥。

4.7 安全等级

不同的诊断服务可能会对安全访问有不同的等级要求，以下定义安全访问不同安全等级的应用策略：

- 在不同的安全等级中，使用同样的运算法则；
- 在不同的安全等级中，使用不同的常数数组；
- 常数数组在下一节定义。

注：在密钥算法函数中，会用到常数数组，它被定义为密钥算法函数的输入参数“SA_constant_n”（见 5.2 节）。

4.8 常数数组定义规则

常数数组是计算安全访问密钥的基本参数，其用法在 5.2 节中具体定义。

（1）对于只有一个等级的安全访问，或多等级安全访问的第一等级，其常数数组的定义规则如下：

常数数组的数值取自于控制器序列号的后四个字节(这要求控制器必须支持读取序列号的诊断服务指令“22 F1 8C”）（DID F18C的定义见《Q/HMAC XXX.XXX-XXX-XXXX UDS诊断DID规范》）；

- 当控制器序列号的后四个字节被用作常数数组时，该四个字节的取值会被当作为十六进制格式的；
- 常数数组的取值是从后到前，即数组的最后一个数值与控制器序列号的最后一个字节对应；
- 如果控制器序列号的长度小于四个字节，在常数数组的前边补充 0x00 以确保常数数组的长度为四个字节。

示例：

如果控制器的序列号是 11 22 ... 33 44 55 66，那么常数数组是［0x33 0x44 0x55 0x66］。

如果控制器的序列号是 11 22 33，那么常数数组是［0x00 0x11 0x22 0x33］。

（2）对于多安全访问等级的控制器，常数数组的定义规则如下：

Constant[0]n = Constant[0]n-1 ROR 3

Constant[1]n = Constant[1]n-1 ROR 3

Constant[2]n = Constant[2]n-1 ROR 3

Constant[3]n = Constant[3]n-1 ROR 3

注：n 是安全等级编号；

ROR 3 指的是循环右移 3 个比特位，如下图：

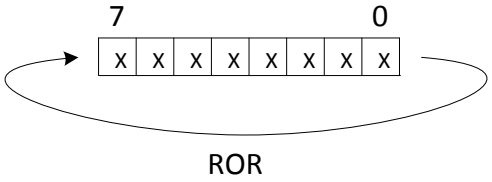


图 3 ROR3 示意图

示例：

如果安全访问第一等级的常数数组是 [0x33, 0x44, 0x55, 0x66]，那么第二等级的常数数组是 [0x66, 0x88, 0xAA, 0xCC]，第三等级的常数数组是[0xCC, 0x11, 0x55, 0x99]。

5 安全访问运算法则

5.1 函数

以下是用 C 语言编写的运算函数，定义了如何计算密钥：

```
void encipher(unsigned int num_rounds, uint32_t v[2], uint32_t const k[4])
{
    unsigned int i;
    uint32_t v0=v[0], v1=v[1], sum=0, delta=0x9E3779B9;
    for (i=0; i < num_rounds; i++)
    {
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + k[sum & 3]);
        sum += delta;
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + k[(sum>>11) & 3]);
    }
    v[0]=v0; v[1]=v1;
}
```

注：供应商可不用 C 语言编程，但密钥运算方法必须与上述一致。

5.2 输入参数

5.1节的函数的输入参数定义如下：

num_rounds = 2;

v[0] = {seed0;seed1;seed2;seed3}; (seed0 as the MSB)

v[1] = {Not(seed0); Not(seed1); Not(seed2); Not(seed3)};

k[0] ={00;00;00;SA_constant_0};

k[1] ={00;00;00;SA_constant_1};

k[2] ={00;00;00;SA_constant_2};

k[3] ={00;00;00;SA_constant_3};

注：参数“SA_constant_n”指的是constant[n]的取值。constant[n]与安全访问等级相关，具体定义见4.8节中的描述。

示例：

如果 seed0 = 0x11, seed1 = 0x22, seed2 = 0x33, seed3 = 0x44

那么 v[0] = 0x11223344

v[1] = 0xEEDDCCBB

如果 SA_constant_0 = 0x11, SA_constant_1 = 0x22, SA_constant_2 = 0x33, SA_constant_3 = 0x44

那么 k[0] = 0x00000011

k[1] = 0x00000022

k[2] = 0x00000033

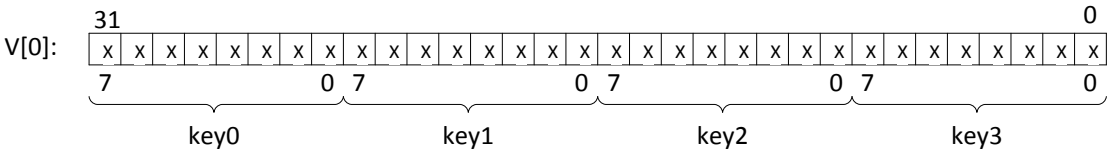
k[3] = 0x00000044

5.3 输出参数

按安全访问运算法则计算后，会得到一个新的v[0]和v[1]，v[0]的取值便是用于安全访问的密钥，密钥的取值如下图所示：

图 4 密钥取值示意图

示例：



如果 v[0] = 0x11223344

那么 key0 = 0x11

Key1 = 0x22

Key2 = 0x33

Key3 = 0x44

5.4 示例

示例 1：

当 SA_constant_0 = 0x11

SA_constant_1 = 0x22

SA_constant_2 = 0x33

SA_constant_3 = 0x44

如果 seed0 = 0x11

Seed1 = 0x22

seed2 = 0x33

seed3 = 0x44

那么 key0 = 0xC3

Key1 = 0x13

Key2 = 0xBD

Key3 = 0x44

示例 2：

当 SA_constant_0 = 0x33

SA_constant_1 = 0x5A

SA_constant_2 = 0xB7

SA_constant_3 = 0x98

如果 seed0 = 0x11

Seed1 = 0x22

seed2 = 0x33

seed3 = 0x44

那么 key0 = 0xC3

Key1 = 0x13

Key2 = 0x69

Key3 = 0xAA