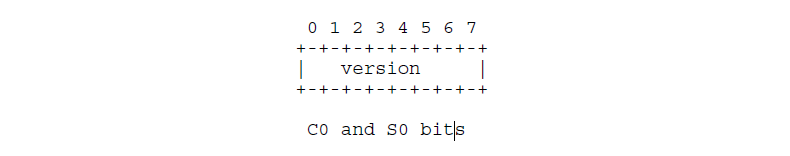
RTMP 握手过程介绍

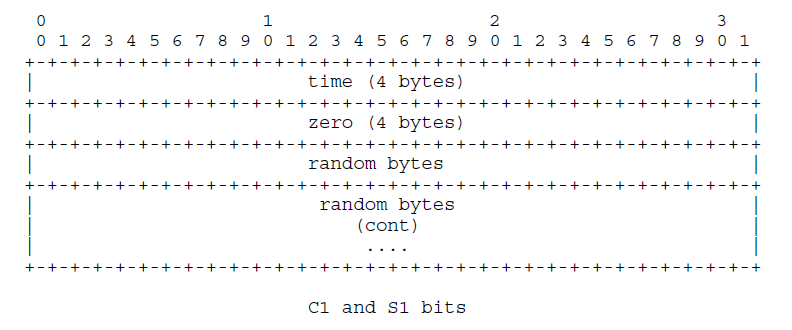
Rtmp的握手协商，其实也可以看做是三次握手，客户端首先发送c0c1给服务器；服务器收到c0c1后，根据c0c1的内容，生成s0s1s2，将s0s1s2发送给客户端；客户端再根据收到的s0s1s2生成c2，将c2发送给服务器，服务器接受c2完成握手。握手过程简图如下：



**c0**: 只占一个字节，表示版本信息，其值为0x03;



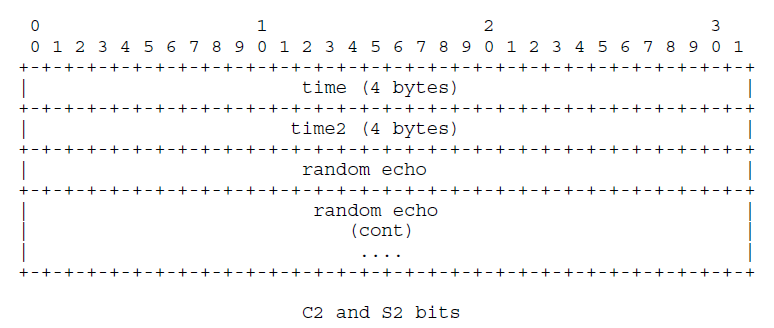
**c1,s1**: c1和s1的格式相同，占1536个字节，详细结构如下：



time：c1或者s1发送时，所在服务器的系统时间；

zero：简单握手时，该值为0。加密握手时，该值为版本信息；

random：简单握手时为1528字节的随机值。加密握手时，该字段由两部分组成：digest(764 bytes) + key(764 bytes)；



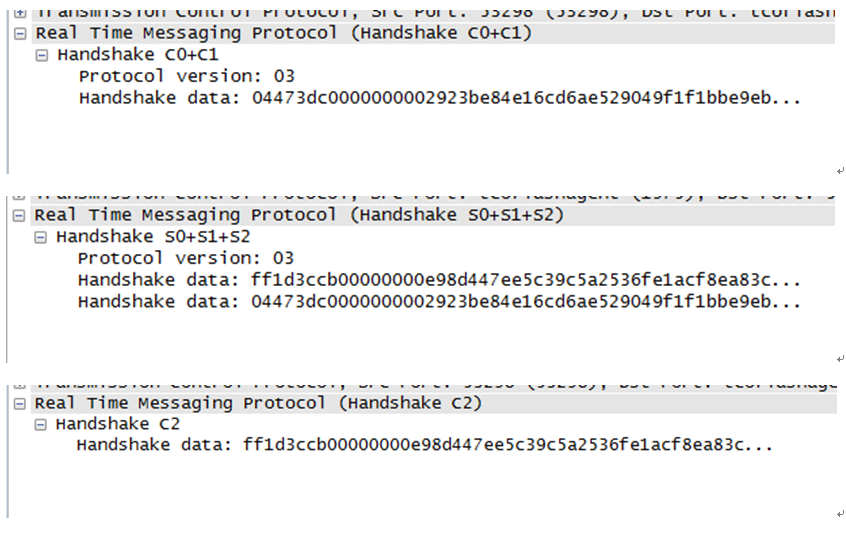
**c2,s2**：c2和s2的格式相同，占1536字节；

time：发送c2或者s2时的系统时间。加密握手时，可用随机值填充；

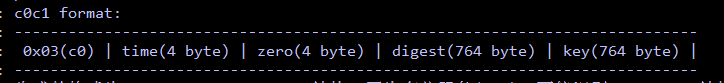
time2：c2时为s1中time的值，s2时为c1中time的值。加密握手时，可用随机值填充；

random：简单握手时，为1528字节的随机值。加密握手时，为rand(1496 byte) + digest-data(32 byte);

**简单未加密握手**：未加密的握手非常简单，客户端生成c0c1，c0值为0x03，c1的time字段为发送c0c1时的客户端机器的系统时间，zero字段填零，后面的1528字节用随机数填充，构造完成c0c1后，发送给服务端；服务端只要收到了1537个字节的数据，并且第一字节为0x03，就认为收到了c0c1。然后构造s0s1s2，发送给客户端，s0s1的生成过程与c0c1相同，s2就是c1的拷贝；客户端收到s0s1s2，向服务端发送c2，c2为s1的拷贝，服务端收到1536字节长度的c2，就完成了rtmp的简单握手。抓包截图如下：



**加密握手**：需要对明文加密，然后进行发送，主要就是对c1/s1的一个32字节长的字段用openssl进行加密。



c0: 0x03

c1: time(4byte) + version(4byte) + digest(764byte) + key(764byte)

time: 系统当前时间;

version: 版本信息;

digest:random\_4(4byte)+random\_offset(byte) + \_digest(32byte) + random\_764-4-offset-32(byte)

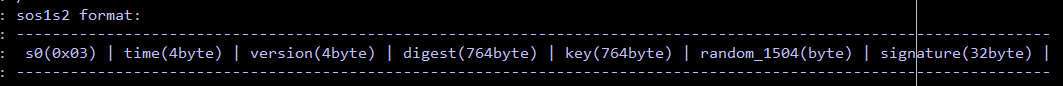
random\_4: 4字节的随机值，根据这4字节计算random\_offset的长度;

random\_offset: 随机值填充;

\_digest: 这个是32字节的加密字段，通过openssl生成;

random\_764-4-offset-32: 随机值填充;

key: 764字节的随机值填充;



s0: 0x03

s1: time(4byte) + version(4byte) + digest(764byte) + key(764byte)

time:发送s1时的系统时间;

version: 版本信息;

digest:random\_4(4byte)+random\_offset(byte)+\_digest(32byte)+random\_764-4-offset-32(byte)

random\_4: 4字节的随机值，根据这4字节计算random\_offset的长度;

random\_offset: 随机值填充;

\_digest: 这个是32字节的加密字段，通过openssl生成;

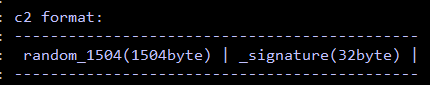
random\_764-4-offset-32: 随机值填充;

key: 764字节的随机值填充, srs代码里128字节的\_key，是由c1的key变换生成的，但是这个值貌似没有什么用，用随机值填充也是可以的，librtmp就是用随机值填充的；

s2: random\_1504(byte) + \_signature(32byte)

random\_1504: 可以用1504个字节的随机数填充；也可以拷贝c1的前1504字节，librtmp就是这样做的；

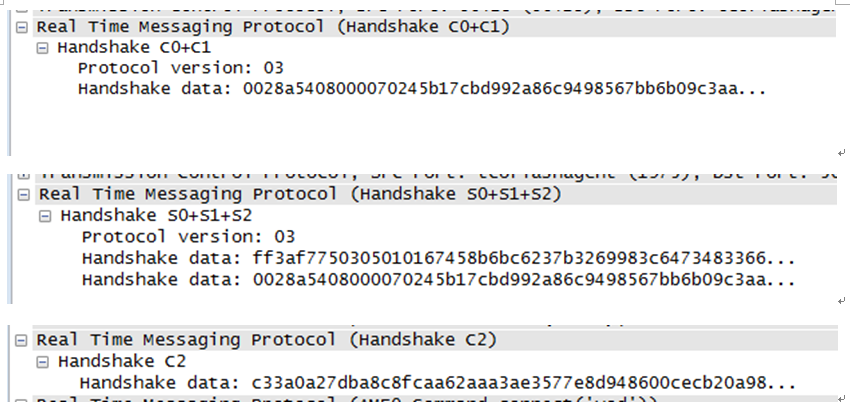
\_signature: 根据c1的\_digest字段生成temp\_key, 再根据temp\_key和1504字节的random生成s2的\_digest；客户端不会校验s2；



c2: random\_1504(byte) + \_signature(32byte)

random\_1504: 1504个字节的随机数填充；

\_signature: 根据s1的\_digest字段生成temp\_key, 再根据temp\_key和1504字节的random生成c2的\_digest；



上图是flash player与librtmp服务器交互数据包的截图，可以看见s2的前面部分与c1相同，而c2为随机生成；下图为flash player与nginx交互数据包的截图，s2为随机数据填充，c2与s1存在相同数据。不同的服务程序，s2和c2的生成是可以不同的，但是c1和s1的生成规则则是固定的，也是最关键的。

