**实验报告**

专业： 信息工程

姓名： 林嘉宁

学号： 3150100830

日期： 2017.11.10

地点：

课程名称： 计算机组成与设计 指导老师： 沈继忠 成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验名称： 对RAR5加密压缩文件的解密算法 实验类型： 探究

一、实验目的

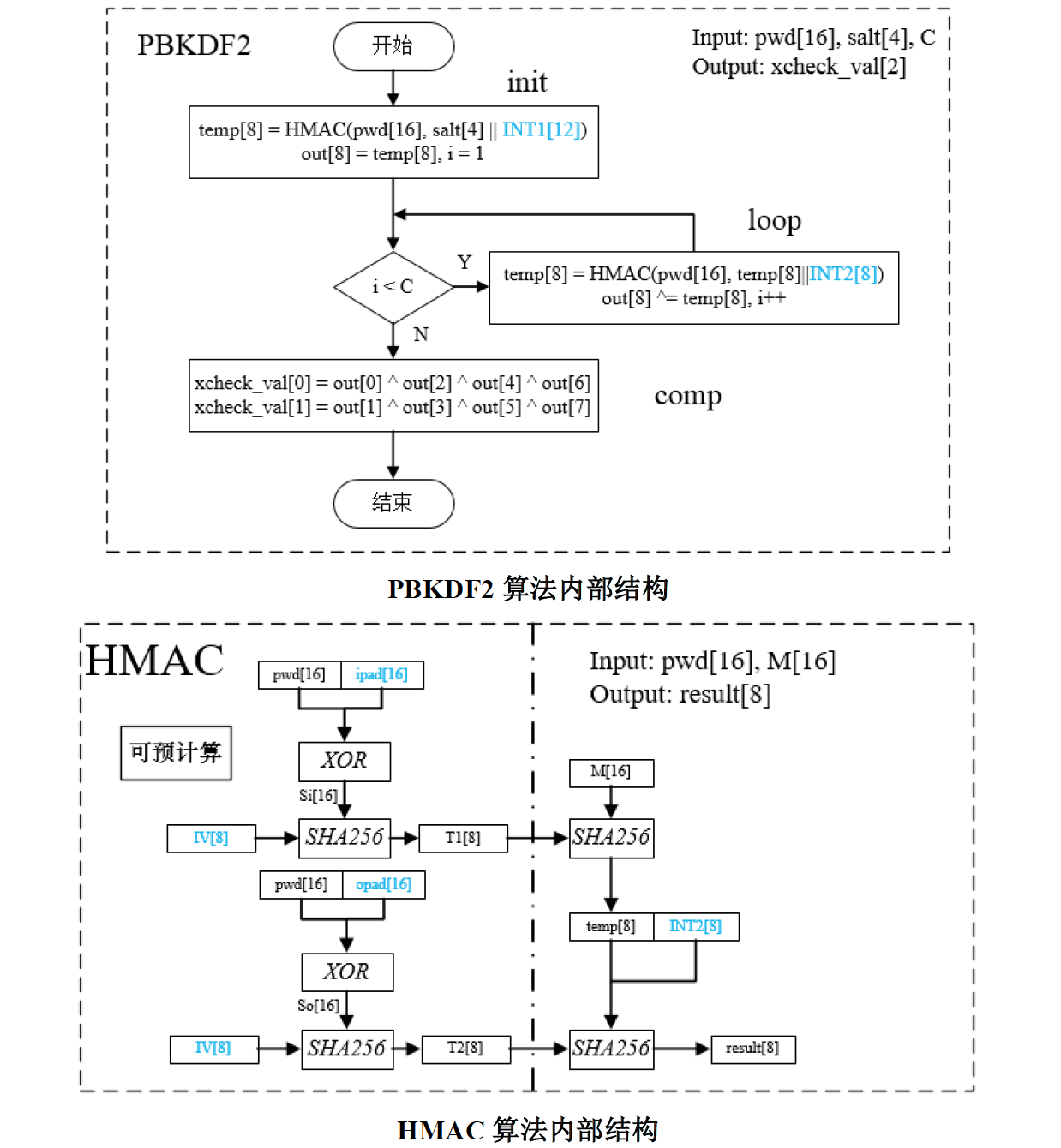
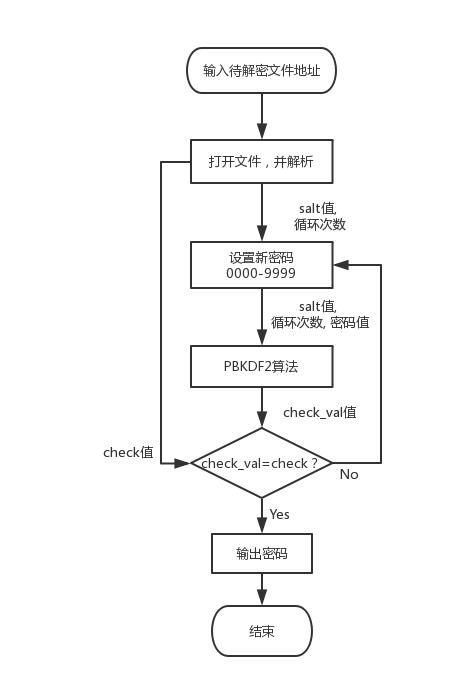
探究RAR5的解密算法，实现利用C/C++语言对加密的RAR5压缩文件进行解压缩的操作，同时不能使用现成的解压缩库。因此，需要从加解密原理层面出发，实现RAR5的解密任务。

二、实验测试平台

Intel Core i7-4790 (No GPU)

三、实验原理

**1.RAR5的解密算法流程图**



**2.SALT值在SHA256中的设置**

SALT值为4字节数，在输入HMAC前首先要补上一个字节，其内容为0x00000001。

**3.SHA256消息填充规则（即INT1、INT2值的设置）**

根据SHA256算法的对消息填充的规定，若输入的值不等于所要求的消息长度，那么将在消息的后一字节补上0x80000000，最后一字节的填充为：（原消息长度+填充后消息长度（即16字节））\*32。特别注意的是，SLAT值在输入前补上0x00000001之后为5字节因此在最后一字节的填充为：（16+5）\*32。

例如：

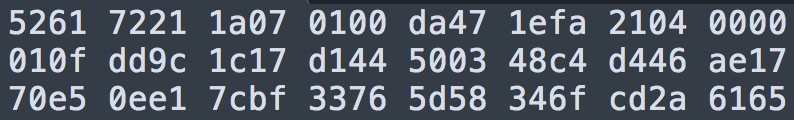
对temp[8]的填充为：0x80000000，0，0，0，0，0，0，（16+8）\*32。

对应的代码如下：

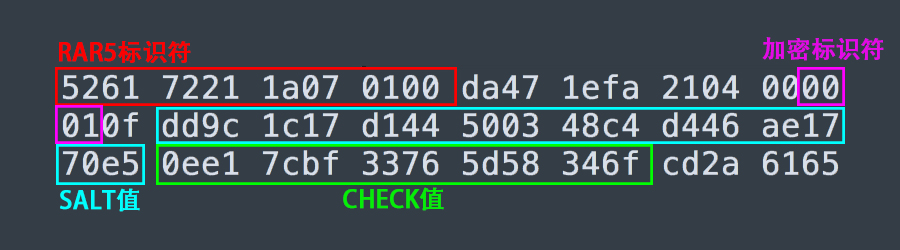


**4.RAR5文件头的结构**

RAR5文件开始区域的数据是有严格规定的，以老师提供的RAR5文件“encrypted.rar”为例，其二进制打开后的文件开始区域如下所示：



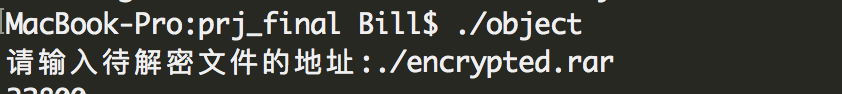
头文件分析如下：



四、实验结果

**测试一（老师提供的RAR5文件）：**

首先输入待解密文件的地址，UI界面如下所示：

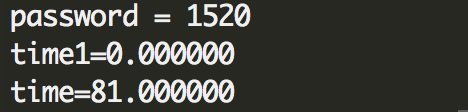


运行过程中显示当前破解的密码，运行完后显示正确的密码，以及解析文件(time1)和解密的时间(time)，单位为秒，如下所示：



我们可以发现程序成功的解出了文件的密码，用时239秒。

**测试二（自己压缩的RAR5文件）：**



经过81秒，成功解出RAR5的密码为1520。

五、算法优化

根据《基于CUDA的RAR5格式文件密码恢复研究》一文中所提到的优化算法，我们可以将密码的SHA256运算预计算好存在一个数组中，在每次循环的里面我们只要调用数组即可，不必再次计算密码的SHA256值。

六、存在的问题

1.改解密算法只能计算RAR5加密的文件，且必须要将文件名加密才可以。因为不加密文件名的RAR文件的头文件格式与此不同。

2.改解密算法只能适用于密码为0000-9999的RAR文件。