DES 的 CBC 模式

DES 加密/解密

在本项目中，我们将在 CBC（密码分组链接）模式下使用 DES 对文件进行加密和解密。“tempdes.c”是用于加密/解密固定的 64 位块的核心文件。在本作业中，您需要通过实现 DES-CBC 操作模式，扩展此框架代码以获取任意大小的输入文件并对其进行加密/解密。实际上，您必须实现 CBC 模式，并且除了 tempdes.c 中的功能外，不允许使用任何内置功能。您可以在您的课本中找到有关 DES-CBC 的信息。

可以通过对照输入文件“test.txt”检查您的工作。如果您正确实现了算法，则应该在“test.des”中获得输出。

要求

a. 只需使用 tempdes.c 中出现的内置函数
b. 您的代码应生成以下形式的可执行文件：

```
./tempdes   iv   key   inputfile   outputfile
```

参数说明如下：
-iv：要使用的实际iv：必须表示为仅由十六进制数字组成的字符串。
-key：要使用的实际密钥：它必须表示为仅由十六进制数字组成的字符串。
-inputfile：输入文件名
-outputfile：输出文件名

示例：

```
./tempdes fecdba9876543210  0123456789abcdef  test.txt  test.des
```

如果存在无效的参数，代码应向用户返回适当的消息。请务必考虑密钥无效的这种情况。

内置功能信息：

我们将简要描述允许您使用的内置功能。您可以在下面的示例源代码中找到有关我们正在使用的大多数内置函数的信息 http://www.openssl.org/docs/crypto/crypto.html

- des_encrypt1

您可以使用名为des_encrypt1的内置函数来执行实际的des加密/解密。您可以通过查看tempdes.c文件了解如何使用此函数。

```
des_encrypt1(long *data, des_key_schedule *ks, int enc)
```

a. data: 此参数是指向long（4字节）类型的两元素数组的指针，该数组将包含您将从文件中读取但打包在long类型变量中的数据。

注意：字符以小端格式加载到此函数中。例如，字符串{0xA0，0xB7，x07，0x08}

是小端格式的08 07 B7 A0（最低有效位优先）。

b. ks: 指向实际键数组的指针。不要担心这个参数数据类型。
c. enc: 进行加密操作时此值为1，进行解密操作时此值为0。

- des_set_key_checked
　　此函数将检查传递的密钥是否为奇数奇偶校验，并且不是一个弱密钥或半弱密钥。如果奇偶校验错误，则返回-1。如果密钥是弱密钥，则返回-2。如果返回错误，则不会生成密钥编排（key schedule）。

des_set_key_checked(const_des_cblock *key, des_key_schedule *schedule)
a. key：指向实际键数组的指针。不要担心这个参数数据类型。
b. schedule：是新密钥，将用作函数des_encrypt1的输入参数

在此文件夹中，您将找到以下文件：
-tempdes.c：是使用DES加密/解密的示例代码

使用的Key和IV值如下。
- Key = 40fedf386da13d57 （十六进制值）
- IV = fedcba9876543210 （十六进制值）

可以使用以下命令编译和执行所有c代码：

- 在linux命令行中，执行"gcc -o tempdes tempdes.c -lcrypto"，其中tempdes.c是源代码，tempdes是要生成的可执行文件的名称。此命令将编译代码并生成可执行文件。

- 要执行刚刚创建的程序（在我们的示例中为tempdes），请使用linux命令行写入“./tempdes”