徐程程

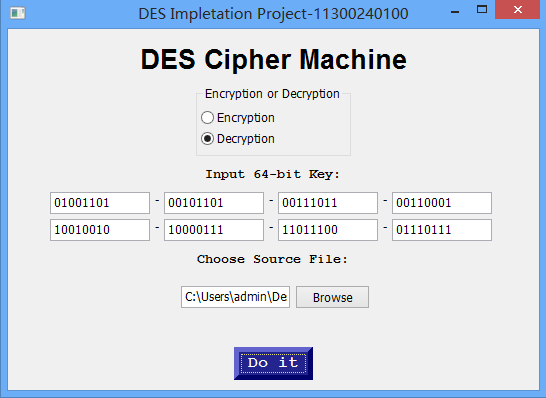
11300240100@fudan.edu.cn

摘要

本文介绍了Project 1-1“6轮DES的编程实现”的  
基本情况、完成细节、相关原理和实验测试。

6轮DES的编程实现

2013-2014秋季学期《信息安全原理》Project 1-1

­­

## **Project需求**

1. **基本信息**

Project 1-1是信息安全原理的课程Project，目标是使作者通过亲自编程实践了解DES编码算法的全过程，为后续Project打下基础。

Project完成时间为2013年10月8日至2013年11月3日。

1. **需求分析**

Project内容是实现DES的加解密过程，其中有以下几点要求：

* **核心部分**  
  输入1：64位明文和64位密钥  
  输入2：64位密文  
  输入1：64位密文和64位密钥  
  输入2：64位明文
* 加解密过程采用**CBC模式**
* **对话界面**  
  输入：  
  ——明文和密文文件：txt文件，内容为二进制字符串)  
  ——密钥：64位二进制字符串  
  ——模式选择：加密/揭秘

## **Project完成细节**

1. **语言及工具选择**

**Python**

考虑到Python的以下几点优势，我选择用Python实现Project1-1：

* Python的lambda表达式处理加解密过程中的置换及矩阵操作非常简洁、方便
* Python有足够的类库资源工具来支持对话界面的实现
* Python语法删繁就简，编程速度快，使我能够集中精力于算法本身。

**GUI图形库wxpython**

Python的GUI图形库有许多，根据我们小程序的需求，网友推荐wxpython和tkinter，由于wxpython比较新，更流行，我选择使用wxpython。

**py2exe**

由于python本身是解释型语言，需要使用py2exe工具产生可执行码。

1. **源代码结构**

源代码由两个python源代码文件组成DES\_GUI.py和 DES\_Encryptor.py分别是GUI对话界面和加解密核心部分的实现。

其中DES\_Encryptor.py由三个类构成：

DES\_Encryptor, FeistelFunctionImplement, SubkeysGenerator分别为DES加密算法主流程、轮函数F的实现和子密钥产生算法的流程。

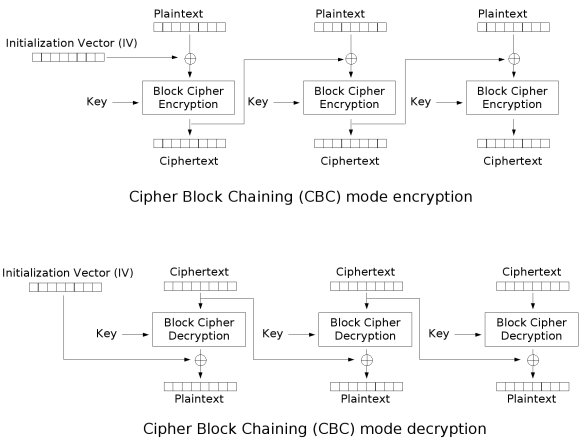
## **相关原理**

除了原来的加密轮数由16轮变为6轮外，project实现的DES加密算法主流程与课本所述无出入，概不赘述。

这里解释一下需求中的CBC模式。

CBC是Cipher Block Chaining的缩写，是四种常用分组密码加密模式之一（其他三种为ECB，CFB，OFB）。

下图解释了CBC模式的流程。我们将明文划分成64bit的小块后，依次加密，每次将前一块的密文与当前块的明文做XOR操作后作为加密算法的输入。第一块则与初始化向量IV做抑或。



**优点：**

1.不容易主动攻击,适合传输长度长的报文,是SSL、IPSec的标准。

**缺点：**

1.不利于并行计算

2.误差传递

3.需要初始化向量IV

我用一个随机64bit串作为IV，IV与密文一起发送给解密端。

明文不足64位时，按照CBC模式的标准，以最后8位为不足位标识符进行补位处理。

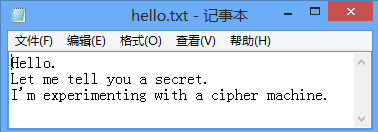
## **实验测试**

为了方便检查加解密过程的正确性，增强可读性，我采用ASCII字符串作为明文，八进制串作为密文。进入加解密算法前先把对应格式转换成二进制串。

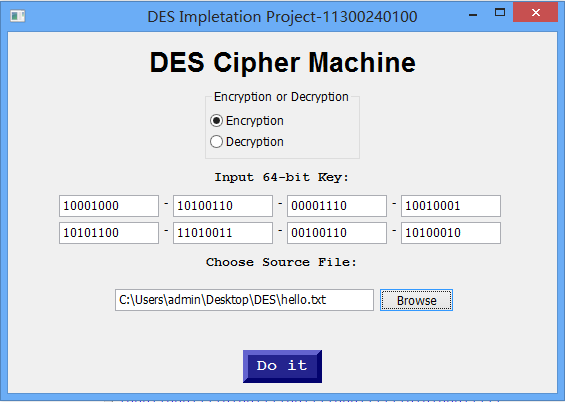
输入原始密钥K为64位二进制串。（时间关系，无输入格式检查。系统自动输入随机64位字符串。）

加解密实验：

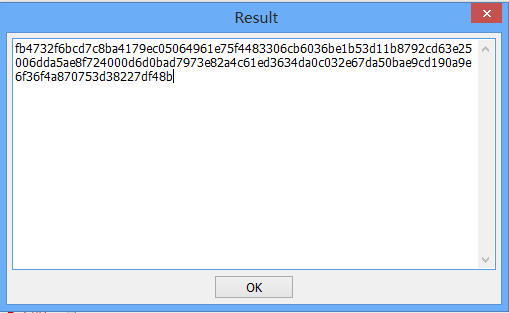
1. 在一个文件中输入明文。

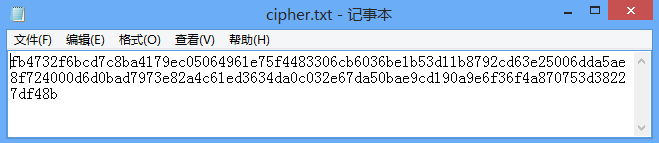


1. 运行DES Cipher Machine，选择加密模式，填入密钥（或使用默认密钥），选择明文文件后，点击按钮。

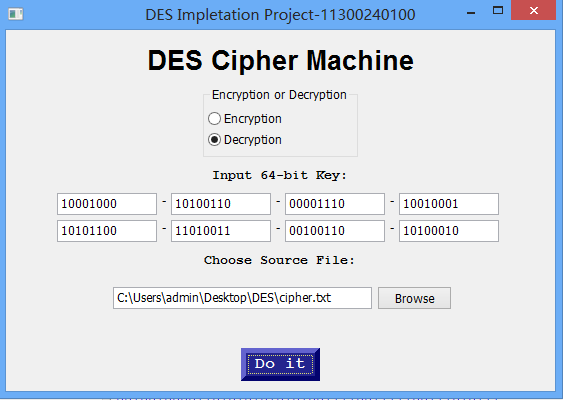


1. 将得到的密文结果存入txt文件。





1. 回到DES Cipher Machine，选择解密模式，注意保持输入密钥与加密初始密钥，选择密文文件cipher.txt后，点击按钮。



1. 得到解密结果。与最开始的原文相同！实验成功。

