

第 1 关：基本测试

根据 S-AES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是 16bit 的数据和 16bit 的密钥，输出是 16bit 的密文。

(1) 加密算法测试结果如下：

明文	密钥	密文
1010101010101010	1010101010101010	2F6B2F6B2F6B2F6B2F6B2F6B2F6B
1010101010101011	1010101010101011	2F6B2F6B2F6B2F6B2F6B2F6B6F6F
1101010101010101	1110101010101010	662E6B6E6B6E6B6E6B6E6B6E6B6E
0000000000000000	1110101010101010	2B6A2B6A2B6A2B6A2B6A2B6A2B6A
1111111111111111	0000000000000000	6F6F6F6F6F6F6F6F6F6F6F6F6F6F

页面演示：

导航栏

二进制/字符串加密

二重加密

三重加密

CBC模式加密

二进制/字符串加密

明文/密文

密钥

加密解密

导航栏

二进制/字符串加密

二重加密

三重加密

CBC模式加密

二进制/字符串加密

1111111111111111

0000000000000000

加密解密

第 2 关：交叉测试

考虑到是“算法标准”，所有人在编写程序的时候需要使用相同算法流程和转换单元(替换盒、列混淆矩阵等)，以保证算法和程序在异构的系统或平台上都可以正常运行。

设有 A 和 B 两组同学(选择相同的密钥 K)；则 A、B 组同学编写的程序对明文 P 进行加密得到相同的密文 C；或者 B 组同学接收到 A 组程序加密的密文 C，使用 B 组程序进行解密可得到与 A 相同的 P。

以二进制加密为例测试算法是否在异构的系统或平台上都可以正常运行：

同学 A 加密结果如下：

二进制/字符串加密

<input type="text" value="1111111111111111"/>	<input type="text" value="0000000000000000"/>	<input type="button" value="加密"/>	<input type="button" value="解密"/>
<div>6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F</div>			

同学 B 加密结果如下：

导航栏

二进制/字符串加密

二进制加密

三重加密

CBC模式加密

二进制/字符串加密

6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F6F

得到密文结果结果相同，所以该算法可以在异构的系统或平台上都可以正常运行

第 3 关：扩展功能

考虑到向实用性扩展，加密算法的数据输入可以是 ASII 编码字符串(分组为 2 Bytes)，对应地输出也可以是 ACII 字符串(很可能是乱码)。

下面是对明文“helloworld”加密和解密的结果：

页面展示:

导航栏

二进制/字符串加密

二重加密

三重加密

CBC模式加密

二重加密

1010101010101010

1010101010101010101110101010101010

1110101010101010101010101010101010

加密

解密

E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708

(2) 解密算法测试结果如下:

[illegible]

页面展示：

导航栏

二进制字符串加密

二重加密

三重加密

CBC模式加密

二重加密

E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708E1AB1708

111010101010101010101010101010101010

101010101010101010101110101010101010

加密

解密

1010101010101010

3.4.2 中间相遇攻击

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用中间相遇攻击的方法找到正确的密钥 Key (K_1+K_2)。

一个密钥的长度为 16 位，要遍历两个密钥（共 32 位），需要的搜索空间非常大，时间复杂度过高。对于一般计算机来说，找到正确密钥所需的时间过长，因此我们认为不可能通过中间相遇攻击的方法找到正确的密钥组合 K_1+K_2 。

3.4.3 三重加密

将 S-AES 算法通过三重加密进行扩展，我们选择按照 32 bits 密钥 Key (K_1+K_2) 的模式进行三重加密解密。

(1) 加密算法测试结果如下：

明文：1234

密钥 1：zjww

密钥 2：abxd

结果如下：

导航栏

- 二进制/字符串加密
- 二重加密
- 三重加密
- CBC模式加密

三重加密

明文: 1234

密钥 1: zjww

密钥 2: abxd

加密 解密

008C33F30B64C53D0D34B3F215382F28

(2) 解密算法测试结果如下：

密文：008C33F30B64C53D0D34B3F215382F28

密钥 2: zjww

密钥 1: abxd

结果如下:

第 5 关: 工作模式

基于 S-AES 算法, 使用密码分组链(CBC)模式对较长的明文消息进行加密。注意初始向量(16 bits) 的生成, 并需要加解密双方共享。在 CBC 模式下进行加密, 并尝试对密文分组进行替换或修改, 然后进行解密, 请对比篡改密文前后的解密结果。

(1) 加密算法测试结果如下:

明文: pycharm

密钥 1: 1010101010101010

初始化向量(共享): 1010101010101011

输出密文结果如下:

导航栏

二进制/字符串加密

二重加密

三重加密

CBC模式加密

CBC模式加密

pycharm

1010101010101010

1010101010101011

加密解密

11FB2F789442BF1E

(2) 解密算法测试结果如下：

密文：11FB2F789442BF1E

密钥 1：1010101010101010

初始化向量（共享）：1010101010101011

输出明文结果如下：

导航栏

二进制/字符串加密

二重加密

三重加密

CBC模式加密

CBC模式加密

11FB2F789442BF1E

1010101010101010

1010101010101011

加密解密

pycharm\$

更改密文 11FB2F789442BF1E 最后一位，使其为 11FB2F789442BF10，得到明文如下：

CBC模式加密

11FB2F789442BF10

1010101010101010

1010101010101011

加密

解密

033887BB0518D5FC8CDB03984E38E84B

由此可见，篡改密文前后的解密结果差异显著，显示了 CBC 模式的加密安全性，使得暴力破解更加困难。