

Welcome to S-AES !

信息安全导论作业2：S-AES算法实现；本次作业由**欢乐斗地组**利用Java+swing完成了一个简单的关于S-AES加解密、双重加密、三重加密和CBC的可视化程序。针对作业要求，成功完成了1-5关。

1. Overview

整个程序的整体框架如下所示，分别有：



1.普通程序 (s_aes)

- 加密
- 解密

2.双重加密 (2s_aes)

- 加密
- 解密
- 中间相遇攻击

3.三重加密 (3s_aes)

- 加密
- 解密

4.CBC模式

- 加密
- 解密

2. 基本测试和扩展功能

基本测试：在这里根据S-AES算法编写和调试程序实现了输入可以是16bit的数据和16bit的密钥，输出是16bit的密文或密文。

扩展功能：以及实现了对任意2bit以上的ASCII字符的加解密。

2.1 加密界面

在这里可以实现16bit**二进制明文**和**任意2bit以上的ASCII字符**的加密，但是同时只支持一种形式的加密，否则会报错；对于十六位二进制的密钥，可以选择**随机生成**或者**手动输入**，但是需要满足16位。具体展示如下：

- 对16bit二进制明文加密

S-AES

加密 解密

二进制密钥(16位) 随机 1010001110100011

二进制明文(16位) 1111111111111111

ASCII码字符串明文

加密 全部重置

二进制明文加密结果: 1101111001011011

- 对ASCII字符加密



- 同时对16bit二进制明文和ASCII字符加密，会报错，提示只支持一种形式的加密



2.2 解密界面

在这里可以实现16bit**二进制密文**和**任意bit的ASCII字符**的解密，但是同时只支持一种形式的加密，否则会报错；对于十六位二进制的密钥，可以选择**随机生成**或者**手动输入**，但是需要满足16位。具体展示如下：

- 对16bit二进制密文解密, 可以发现输入在加密界面输入的密钥和加密出的密文，解密出了正确的明文1111111111111111



- 对ASCII字符解密，可以发现输入在加密界面输入的密钥和加密出的密文，解密出了正确的明文abc



- 同时对16bit二进制密文和ASCII字符解密，会报错，提示只支持一种形式的解密



3. 交叉验证

- **组内交叉验证** 针对加解密的结果（二进制和ASCII码）进行交叉验证，发现可以完成逆向验证，这里只展示二进制的交叉验证，如下图所示：
- **组间交叉验证** 我们同其他组（荔枝组（张芷芮，刘俐莹））针对加解密的结果（二进制和ASCII码）进行交叉验证，发现依然可以完成逆向验证。

4. 多重加密

4.1 2-DES

在这里可以实现输入两个十六位二进制的密钥和明密文，实现对16bit二进制明密文的加解密

- 加密

2 S-AES

加密 解密 中间相遇攻击

二进制密钥1 (16位) 随机 0000000000000001

二进制密钥2 (16位) 随机 0000000000000000

二进制明文 (16位) 1111111111111111

加密 全部重置

二进制明文加密结果: 0001001010111001|

- 解密

2 S-AES

加密

解密

中间相遇攻击

二进制密钥1 (16位)

随机

0000000000000001

二进制密钥2 (16位)

随机

0000000000000000

二进制密文 (16位)

0001001010111001

解密

全部重置

二进制明文解密结果: 1111111111111111

4.2 中间相遇攻击

在这里可以输入一对或多对使用相同密钥的明、密文对，使用中间相遇攻击的方法可以找到正确的密钥Key(K1+K2)。

- 输入一对明密文对会出现多对可行的密钥

2 S-AES

加密

解密

中间相遇攻击

二进制明文(16位)

二进制密文(16位)

使用

1111111111111111	0001001010111001	<input checked="" type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

开始破解

耗时: 0.676 秒

攻击结果:

0000000000000001 0000000000000000,
0111100010010100 0000000000000010,
1110011010110101 0000000000000011,
1000110011000001 0000000000000100,
1100011001111001 0000000000000110,
0000111110110000 0000000000001000,
0111011111110010 0000000000001011,
0111100011101101 0000000000001101,
0001100000000000 0000000000001110,

- 输入多对明密文对出现正确的一对密钥

2 S-AES

加密

解密

中间相遇攻击

二进制明文(16位)

二进制密文(16位)

使用

1111111111111111	1000011110111000	<input checked="" type="checkbox"/>
1111111111111110	1100011110111010	<input checked="" type="checkbox"/>
1111111111111100	0001011110111010	<input checked="" type="checkbox"/>

开始破解

耗时: 0.974 秒

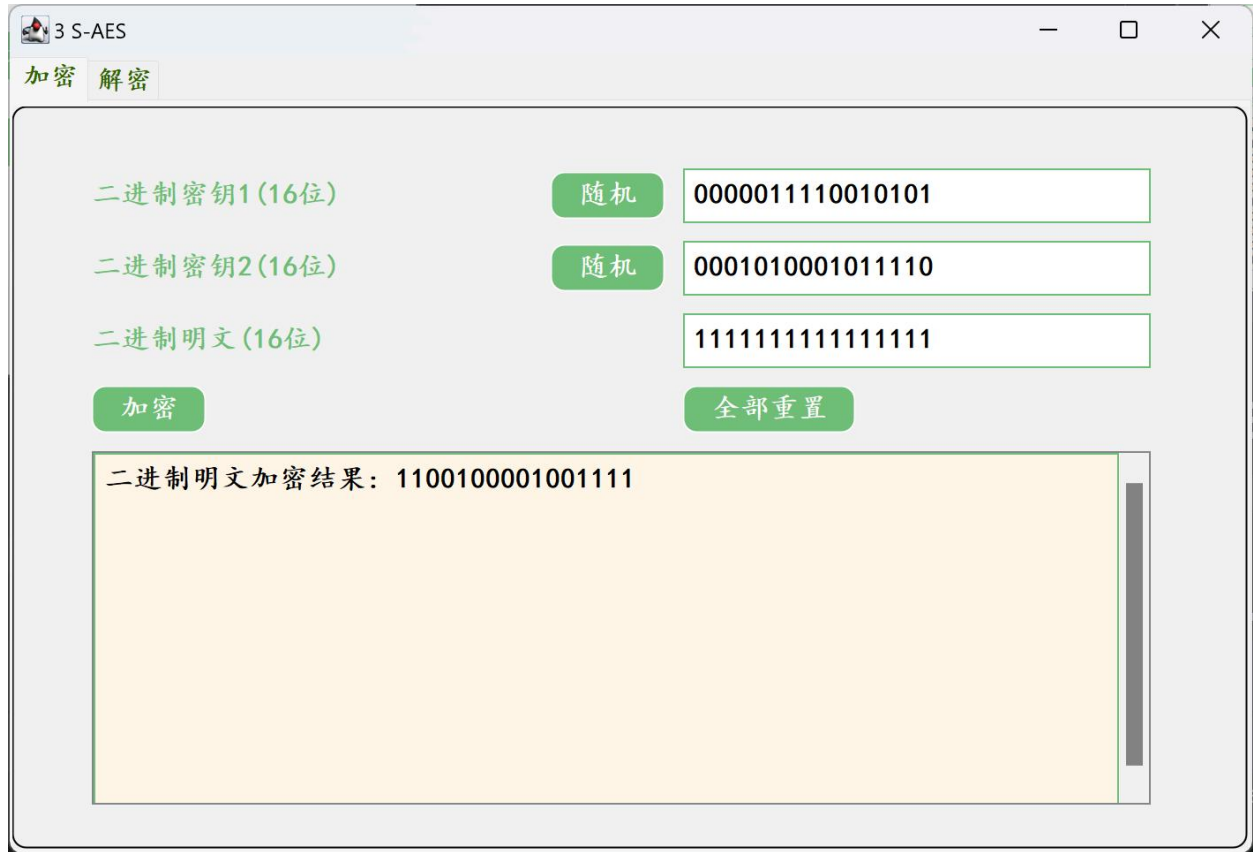
攻击结果:

0011110011011001 1001000100000111

4.2 3-DES

这里按照32 bits密钥Key(K1+K2)的模式进行三重加密解密, $C=E(K1,D(K2,E(K1,P)))$, 输入两个16 bit的密钥和16bit的明密文就能对明密文加解密。

- 加密



The screenshot shows a web-based application titled "3 S-AES". It has two tabs: "加密" (Encrypt) and "解密" (Decrypt), with "加密" currently selected. The interface includes three input fields for 16-bit binary data: "二进制密钥1 (16位)" with a "随机" (Random) button and value "0000011110010101", "二进制密钥2 (16位)" with a "随机" button and value "0001010001011110", and "二进制明文 (16位)" with value "1111111111111111". Below these are two buttons: "加密" (Encrypt) and "全部重置" (Reset All). A large text area at the bottom displays the result: "二进制明文加密结果: 1100100001001111".

- 解密

3 S-AES

—□×

加密解密

二进制密钥1(16位)

随机

0000011110010101

二进制密钥2(16位)

随机

0001010001011110

二进制密文(16位)

1100100001001111

解密

全部重置

二进制明文解密结果: 1111111111111111

5. 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。输入16位的明密文和初始向量(16 bits) 和密钥对明密文加解密。

- 加密

CBC mode

加密

解密

二进制密钥(16位)

随机

1010000001100100

IV(16位)

随机

0010101110010000

二进制明文(any位)

1111111111111111

加密

全部重置

二进制明文加密结果: 1000000111011010

- 解密

CBC mode

加密

解密

二进制密钥(16位)

随机

1010000001100100

IV(16位)

随机

0010101110010000

二进制密文(any位)

1000000111011010

解密

全部重置

解密结果: 1111111111111111

这里在CBC模式下进行加密，并尝试对密文分组进行修改，然后进行解密。可以看到修改密文之前解密得到的结果是Hello,S-AES! 在对密文分组进行篡改最后一个字符之后进行解密得到的结果是Hello,S-AEC(。

发现篡改密文后会引起最后一个块的更改，用篡改后的密文进行解密会影响解密结果的正确性。

原始明文: Hello,S-AES!

原始密文: 001010110000000110001111001001001100010101101000000110110101001101111010111110111101111010100

在加密完成后对密文分组进行篡改最后一个字符的密文: 001010110000000110001111001001001100010101101000000110110101001101111010111110111101111010101

对上述篡改密文进行解密的明文: Hello,S-AEC(

6. 具体实现

- functionalClass文件夹 存放用于实现中间相遇攻击和ASCII加解密的函数式算法，便于在ui界面中直接调用。
- UI文件夹 用于实现程序的ui界面和相关监听器的编写。
- images文件夹 存储了相关关卡测试的图片。
- 如果想要运行程序，可以直接点击src/UI/Main.java程序运行。