

陕西科技大学

《计算机网络安全》实验报告



实验二： 对称加密技术

学 生： _____

学 院： 电子信息与人工智能学院

专 业： 网络工程

指导教师： 张楠

2022 年 6 月 2 日

实验二 对称加密技术

班级：_____

实验预习报告

一、实验目的

- 1、掌握对称加密算法的思想和方法
- 2、了解经典的对称加密算法：凯撒密码
- 3、使用高级语言完成凯撒密码的加解密

二、实验要求

使用编程语言完成对称加密的算法，掌握对称加密体制的特点和密钥的管理方法，掌握数据加密技术的一般实现步骤，尝试将加密运用到实际问题当中，理解密码分析的特点。

三、实验原理

1、对称加密运用范围广泛，可以加密不同的对象和数据，方便起见，可以在此处使用字符串作为加密对象，通过复盘凯撒密码的流程，使用高级编程语言来实现凯撒密码的加解密。

2、为了实现凯撒密码的加解密，此处需要设计两个不同的程序，分别完成加密和解密，它们可以是不同的函数、模块或者是对象、方法，等。

四、实验预习内容

1、对称加密和非对称加密

对称加密和非对称加密是两种常见的加密方式，它们的主要区别是：

对称加密是指加密和解密用的是同一个密钥或者同一套逻辑的加密方式。对称加密的优点是速度快，缺点是密钥的管理和分发比较困难，如果密钥泄露，就会带来很大的安全风险。

非对称加密是指加密和解密用的是一对密钥，一个叫做公钥，一个叫做私钥。

使用公钥加密的数据，只能用私钥解密，反之亦然。非对称加密的优点是安全性高，因为不存在密钥泄露的问题，公钥即便被知道也没关系。缺点是计算复杂度高，所以一般来说非对称加密的速度相对于对称加密慢很多。

现今最流行的对称加密的主要算法有：DES、3DES、AES 等，这些加密算法太过于复杂，此处可以选用古典对称加密算法凯撒密码作为实现的目标。

2、凯撒密码

凯撒密码是一种古老且简单的替换密码，由罗马军事指挥官凯撒在古罗马帝国时期使用。它通过将字母按照固定的位移量进行替换来实现加密和解密，通常，我们将字母移位的距离作为密钥，然后对明文进行加密。

3、凯撒密码的详细加解密过程

凯撒密码的加密过程如下：

- (1)确定一个位移量（通常为正整数），例如 3。
- (2)将待加密的消息中的每个字母，按照位移量进行右移。例如，字母 A 右移 3 位后变成字母 D，字母 B 变成字母 E，以此类推。
- (3)对于非字母字符（如数字、空格、标点符号等），保持原样不变。

解密过程与加密过程相反：

- (1)使用与加密时相同的位移量。
- (2)将加密后的消息中的每个字母，按照位移量进行左移。
- (3)对于非字母字符，保持原样不变。

实验二 对称加密技术

班级：_____

实验报告

一、实验目的

- 1、掌握对称加密算法的思想和方法
- 2、了解经典的对称加密算法：凯撒密码
- 3、使用高级语言完成凯撒密码的加解密

二、实验要求

使用编程语言完成对称加密的算法，掌握对称加密体制的特点和密钥的管理方法，掌握数据加密技术的一般实现步骤，尝试将加密运用到实际问题当中，理解密码分析的特点。

三、实验原理

1、对称加密运用范围广泛，可以加密不同的对象和数据，方便起见，可以在此处使用字符串作为加密对象，通过复盘凯撒密码的流程，使用高级编程语言来实现凯撒密码的加解密。

2、为了实现凯撒密码的加解密，此处需要设计两个不同的程序，分别完成加密和解密，它们可以是不同的函数、模块或者是对象、方法，等。

四、实验内容

1、编程语言选用

为了能够更加直观的展示加解密的结果，可以尝试将加密的输入和输出放到Html 结构当中，使用 JavaScript 来完成加密和解密的流程。

2、界面功能设计

使用两个文本域显示加密、解密的文本，输入框输入一个数字作为密钥，点击加密、解密按钮在对应的文本框更新结果，如下：



3、实现加密算法的核心代码：

```
// 凯撒密码加密
function caesarEncrypt(str, shift) {
  const uppercaseStr = str.toUpperCase();
  let encrypted = '';

  for (let i = 0; i < uppercaseStr.length; i++) {
    const char = uppercaseStr[i];
    if (char ≥ 'A' && char ≤ 'Z') {
      const charCode = (char.charCodeAt(0) - 65 + shift) % 26 + 65;
      encrypted += String.fromCharCode(charCode);
    } else {
      encrypted += char;
    }
  }

  return encrypted;
}

// 凯撒密码解密
function caesarDecrypt(str, shift) {
  const uppercaseStr = str.toUpperCase();
  let decrypted = '';

  for (let i = 0; i < uppercaseStr.length; i++) {
    const char = uppercaseStr[i];
    if (char ≥ 'A' && char ≤ 'Z') {
      const charCode = (char.charCodeAt(0) - 65 - shift + 26) % 26 + 65;
      decrypted += String.fromCharCode(charCode);
    } else {
      decrypted += char;
    }
  }

  return decrypted;
}
```

```

let pwd = document.getElementById('pwd');
let text = document.getElementById('text');
let cipher = document.getElementById('cipher');
document.getElementById('encode').addEventListener('click', function () {
    cipher.value = caesarEncrypt(text.value, parseInt(pwd.value));
});
document.getElementById('decode').addEventListener('click', function () {
    text.value = caesarDecrypt(cipher.value, parseInt(pwd.value));
});

```

其中，html 结构大致如下：

```

11 <body>
12   <container>
13     <div class="pwd-div">
14       <p>输入密钥: </p>
15       <input type="text" id="pwd">
16     </div>
17     <div class="text-div">
18       <textarea name="" id="text" cols="30" rows="10" placeholder="输入原文"></textarea>
19       <textarea name="" id="cipher" cols="30" rows="10" placeholder="输入密文"></textarea>
20     </div>
21     <div class="btn-div">
22       <button id="encode">加密</button>
23       <button id="decode">解密</button>
24     </div>
25   </container>
26   <script src="index.js"></script>
27
28 </body>
29
30 </html>

```

为了能够展示凯撒密码的核心逻辑，故页面没有太多的 css 样式，以功能的实现为主

4、JavaScript 代码的逻辑

这段代码包含了以下几个部分：

(1) caesarEncrypt 函数

这个函数接受一个字符串 str 和一个整数 shift 作为参数，返回加密后的字符串。它的逻辑是：首先将 str 转换为大写字母，方便处理，然后遍历 str 中的每个字符 char，判断是否是英文字母，如果是英文字母，就将 char 的 ASCII 码减去 65（A 的 ASCII 码），加上 shift，然后对 26 取模（因为有 26 个字母），再加上 65，得到新的 ASCII 码。然后用 String.fromCharCode 方法将新的 ASCII 码转换为字符，拼接到 encrypted 字符串中。如果不是英文字母，就直接拼接到 encrypted 字符串中，最后返回 encrypted 字符串。

(2) caesarDecrypt 函数

这个函数接受一个字符串 str 和一个整数 shift 作为参数，返回解密后的字

字符串，它的思路大致和加密算法相同，只是部分地方存在差异。它的逻辑是：首先将 `str` 转换为大写字母，方便处理。然后遍历 `str` 中的每个字符 `char`，判断是否是英文字母，如果是英文字母，就将 `char` 的 ASCII 码减去 65（A 的 ASCII 码），减去 `shift`，加上 26（防止出现负数），然后对 26 取模（因为有 26 个字母），再加上 65，得到新的 ASCII 码，然后用 `String.fromCharCode` 方法将新的 ASCII 码转换为字符，拼接到 `decrypted` 字符串中。如果不是英文字母，就直接拼接到 `decrypted` 字符串中，最后返回 `decrypted` 字符串。

（3）HTML 部分

这部分定义了一些页面元素和事件监听器。具体来说：`pwd` 是一个输入框，用于输入偏移量（`shift`），`text` 是一个文本域，用于输入明文或显示解密后的结果，`cipher` 是一个文本域，用于输入密文或显示加密后的结果，`encode` 是一个按钮，点击时会调用 `caesarEncrypt` 函数，将 `text` 中的内容加密，并显示在 `cipher` 中，`decode` 是一个按钮，点击时会调用 `caesarDecrypt` 函数，将 `cipher` 中的内容解密，并显示在 `text` 中。

5、凯撒密码实现完成，使用相关数据进行凯撒密码算法的测试

（1）测试编码（加密）

对英文句子 “” 进行加解密测试
，密钥为 3。

加密结果：

可以看到，程序完成了对所给句子的加密，输出为：“
”，再次将这个加密后的句子放到程序
中进行解密，可以正常的还原内容。

(2) 测试解密

使用在线凯撒编码工具 (<https://www.1ddgo.net/encrypt/caesar-cipher>), 将句子 “

…” 进行编码，偏移量 27，将编码后的结果使用算法进行解密。

使用在线工具加密，结果为 “

”，将这个结果放到算法

中进行解密：

可以看到，对于 26 以上的密钥依然可以正常加解密。

五、实验结论

1、凯撒密码是一种最为简单的对称加密技术，由于其太过简单，现在基本上不会在重要的地方使用。

2、对称加密和非对称加密各有优缺点，两者往往结合使用，各取其长，使得密码系统更好用。

3、正如实验内容中的加解密的结果，无论加密解密的算法、平台、工具如何，被加密的句子不会变，它的意思亦不变。