

青岛理工大学

**《密码学》课程设计报告**

项目题目：对AES密码系统进行选择明文攻击

**项目组成员： 学号 姓名**

**202201050849 于佳玮**

**202201050842 靳悦**

|  |  |
| --- | --- |
| **院 (系)：** | 信息与控制工程学院 |
| **专 业：** | 网络空间安全 |
| **起迄期：** | 2025.1.6至 2025.1.10 |
| **设计点：** | 信控大楼414+404 |
| **指导师：** | 叶晓云 |

**报告正文**

# 第1章 概述

# 1.1 设计背景

1. 密码学在当今信息安全领域的重要性：

密码学是现代信息安全的基石，在保护数据安全、维护隐私和支持数字化社会运行中扮演着不可或缺的角色。

（1）保障数据的机密性：通过加密技术，密码学确保了敏感信息在传输和存储过程中的不可泄露性。

（2）保护数据的完整性：密码学技术如哈希函数可以检测数据是否被篡改，确保信息的可信度。

（3）实现身份认证和授权：密码学支持数字签名、加密密钥分发等功能，用于验证通信双方身份和控制访问权限。

（4）对抗网络威胁：在面对日益复杂的网络攻击时，密码学是抵御数据泄露、窃听和篡改等行为的第一道防线。

（5）支撑社会基础设施：密码学广泛应用于电子支付、物联网、云计算和区块链等关键领域，为数字经济的安全运行提供技术保障。

在当前信息化社会中，密码学的重要性日益凸显，已成为信息安全领域不可或缺的核心技术。

2. 本次密码学课程设计的目的和意义

目的

（1）理论与实践结合：通过实践对 AES 密码系统进行选择明文攻击，深化对分组加密算法和攻击方法的理解。

（2）培养攻防能力：学习模拟攻击技术，从攻击视角评估算法安全性，同时为设计更强健的密码算法奠定基础。

（3）提升问题解决能力：通过实践设计，培养分析问题、验证假设、总结规律的能力，为后续研究奠定方法论基础。

意义

**（**1）验证经典算法的安全性：AES 是当前最广泛使用的对称加密算法，分析其安全性有助于评估其抗攻击能力和潜在漏洞。

（2）理解密码学的实际应用场景：通过模拟攻击，展示密码学技术在实际应用中可能面临的威胁，从而理解防护措施的重要性。

（3）加强信息安全意识：通过课程设计，引导学生了解密码学在信息安全中的核心作用，激发对密码学技术深入研究的兴趣。

3. 选择“对 AES 密码系统进行选择明文攻击”课题的原因

（1）AES 的代表性和现实意义

AES（高级加密标准）是当今最广泛应用的分组加密算法，被广泛用于银行交易、电子支付、网络通信等领域。研究其安全性不仅具有理论价值，也具备很强的现实意义。

（2）选择明文攻击的实践价值

选择明文攻击是一种经典的密码分析方法，允许攻击者在控制明文输入的情况下，分析加密结果以推测密钥或算法特性。此方法具备实际可操作性，可以帮助理解密码系统的内部结构和潜在弱点。

（3）推动密码学知识深化

AES 是设计精巧、经过严格安全评估的密码系统，通过研究其抗攻击性，可以深入学习分组密码的设计原理、加密流程及其安全性保障机制，从而巩固密码学知识。

（4）为未来研究奠定基础

对 AES 密码系统的选择明文攻击不仅是对经典算法安全性的一次验证，还为后续研究密码系统的抗攻击性、优化设计及新型加密技术提供实践基础。

# 1.2 设计目标

1. 深入理解 AES 密码系统的加密原理

（1）掌握 AES 密码系统的基本原理、加密流程、密钥扩展机制和其安全设计思想。

（2）理解 AES 算法的设计结构，包括 SubBytes、ShiftRows、MixColumns 和 AddRoundKey 等核心步骤。

2. 实现对 AES 密码系统的选择明文攻击

（1）模拟选择明文攻击场景，通过设计合理的攻击实验，对 AES 的安全性进行分析和评估。

（2）详细记录攻击过程，包括明文选择策略、密文分析方法及攻击结果，以验证 AES 在特定条件下的抗攻击能力。

3. 分析 AES 密码系统的安全性和潜在漏洞

（1）通过实验结果分析 AES 在面对选择明文攻击时的表现，总结其安全性特点及可能存在的安全隐患。

（2）理解攻击的限制条件，探讨攻击成功与否的关键因素。

4. 提出改进建议或防护措施

（1）根据攻击分析结果，提出针对 AES 密码系统的优化建议，例如增加安全参数、优化密钥管理等，提升其抗攻击能力。

（2）探讨可能的防护策略，例如如何应对选择明文攻击场景，提升系统整体安全性。

5. 提升实践能力与理论理解

（1）将密码学理论与实际操作结合，培养分析问题、设计实验、总结规律的能力。

（2）加深对对称加密算法及其在信息安全领域应用的理解，为未来密码学研究或信息安全开发打下坚实基础。

# 第2章 密码算法的设计

# 2.1 AES密码算法的原理

1. 数学原理

AES算法首先从原始的密钥生成一系列的子密钥。对于128位密钥，AES算法生成10轮密钥；对于192位密钥，生成12轮密钥；对于256位密钥，生成14轮密钥。

输入： 一个128位、192位或256位的原始密钥。

输出： 由原始密钥衍生出的多个轮密钥（与AES的轮数对应）。

扩展过程的关键步骤是使用一个名为Rcon（轮常量）的表格和S-Box（字节替代）来生成新的轮密钥。

AES主要由以下几部分组成：

① 分组加密： 明文被划分为固定大小的块（128位，16字节），每个块独立处理。

② 数学操作： AES采用伪随机置换（PRP）和有限域GF(2⁸)上的操作，包括字节替换（SubBytes）、行移位（ShiftRows）、列混淆（MixColumns）和轮密钥加（AddRoundKey）。

2. 加密过程

AES的加密过程由多个步骤组成，每个步骤都包含数学运算。这些步骤分别为：

① AddRoundKey（轮密钥加）：每轮加密的第一步是将明文（或者前一轮的输出）与轮密钥进行异或（XOR）。

公式：



(其中，state是当前的数据块，roundKey是当前轮的子密钥)

② SubBytes（字节替代）：使用一个预先计算的S-Box（字节替代表）对数据块中的每个字节进行替代。S-Box是一个固定的置换表，映射每个字节（0x00 到 0xFF）到一个新的字节。

公式：

（其中，SBox是一个固定的置换表，state[i]是第i个字节）

③ ShiftRows（行移位）：对数据块进行行移位操作。AES中的数据块表示为4x4的矩阵（每个元素是一个字节），ShiftRows操作将每一行的字节按一定偏移量进行移动。

公式：

（其中，表示第i行,k为对应的移位数）

④ MixColumns（列混合）：MixColumns操作将每一列中的字节进行线性变换。通过矩阵乘法来混合列中的字节。

**公式：**

（其中，M是一个常量矩阵，P\_i是明文数据的每一列）

矩阵M为：

（对应每列乘法使用有限域GF(2⁸)进行计算）

⑤ AddRoundKey（轮密钥加）：每一轮结束时，使用当前轮的密钥对数据块进行异或（XOR）操作。

**公式：**

3. 解密过程

AES的解密过程与加密过程相似，但有一些步骤的逆操作。解密的关键步骤如下：

① AddRoundKey（轮密钥加）：与加密过程相同，首先对密文与轮密钥进行异或操作。

② InvShiftRows（逆行移位） 对行进行逆移位操作。

公式：

(对于行i进行逆移位，k为对应的移位数)

③ InvSubBytes（逆字节替代）：使用一个逆S-Box进行字节替代。逆S-Box是S-Box的反向映射。

公式：

（其中，SBox^{-1}是逆S-Box）

④ InvMixColumns（逆列混合）：对列进行逆混合，使用的矩阵是逆矩阵。

公式：

（其中，M^{-1}是逆矩阵，P\_i是密文数据的每一列）

⑤ AddRoundKey（轮密钥加）：进行最后一次轮密钥异或，得到明文。

# 2.2 系统架构设计

图 1系统架构设计:

模块功能

1. 前端界面模块

提供明文输入和加密模式选择（ECB/CBC）。

返回密文、分块信息和分析结果。

2. 后端加密模块

根据用户选择的模式，使用AES算法进行加密。

实现ECB模式的直接加密和CBC模式的IV引入。

3. 分析模块

将加密后的密文分为块。

检测重复块，分析加密模式，并提供安全建议。

4. 密钥管理模块

管理密钥（目前为静态密钥），支持动态生成或输入密钥的扩展。

# 2.3 算法实现步骤

1.数据结构设计

密文块： 划分为16字节（128位）的块。

轮密钥： AES算法需要的多个子密钥。

调试信息： 包括明文分块、密文分块和重复块信息。

2.函数设计

① 加密函数（encrypt\_aes）：

输入：明文、密钥、加密模式。

输出：密文、IV（若为CBC模式）。

描述：调用AES库完成加密，支持ECB和CBC模式。

② 分块函数（split\_into\_blocks）：

输入：数据、块大小。

输出：分块列表。

描述：将输入数据分割为指定大小的块。

③ 分析函数（analyze\_cipher）：

输入：密文块、明文。

输出：模式分析、重复块信息和安全建议。

描述：检测重复块，分析可能的加密模式（如ECB的重复块特性）。

3.流程

① 用户通过前端提交明文和加密模式。

② 后端根据请求选择加密模式，调用加密函数完成加密。

③ 分析函数对密文块进行分析，检测加密模式。

④ 前端显示密文和分析结果。

# 2.4 关键技术和难点解决

1. 提高算法的效率

问题：在AES加密的实现中，尤其是大数据量的明文加密时，算法效率显得尤为重要。如果加密速度过慢，将无法满足实际应用需求。

解决方法：

① 优化代码结构：通过模块化设计，将加密、分块、填充等功能分别封装为 独立函数，使得代码逻辑清晰，同时易于优化和测试。

② 引入分块加密机制：针对大文件加密，采用分块处理方式（split\_into\_blocks 函数），避免一次性加载整个文件导致的内存压力。分块的同时可以进行并行处理，进一步提升效率。

③ 利用硬件加速：AES算法在许多现代CPU上支持硬件加速（如Intel AES-NI指令集）。虽然我们这次使用的是纯软件实现，但实际部署中可以通过调用硬件加速接口显著提高加密速度。

2. 处理密钥的安全存储和传输

问题：AES加密的核心是密钥，如何保证密钥的安全存储和传输是设计中的重要难点。如果密钥被泄露，即便算法本身再安全也无济于事。

解决方法：

① 使用安全的密钥管理机制：在实现中，我们通过硬编码的方式简单模拟了密钥的管理，但在实际应用中，应该使用更加安全的方法，如密钥管理系统（KMS）或硬件安全模块（HSM）。

② 密钥的随机生成：在测试过程中，我们生成了固定长度的随机密钥（16字节），并确保每次密钥生成均采用强随机数生成器（如 os.urandom），避免使用易猜测的密钥。

③ 密钥传输加密：如果需要在网络中传输密钥，应当使用非对称加密（如RSA）进行保护，并结合TLS协议确保传输过程的安全性。

3. 应对密码攻击

问题：密码攻击是加密系统必须面对的重要威胁，尤其是AES的实现可能面临的已知明文攻击（CPA）、差分攻击等，需要从设计上提升系统的抗攻击能力。

解决方法：

① 避免固定模式漏洞：

A .ECB模式虽然易于实现，但由于其分块之间无关联，容易暴露明文 的结构特征，因此我们在设计中引入了CBC模式，并随机生成初始化向 量 （IV）来增强密文的随机性。

B .在CBC模式下，确保IV的随机性和一次性使用，有效避免了IV被 重用可能引发的安全问题。

② 明文填充的防攻击性：在填充过程中，使用了标准的PKCS7填充方式， 确保明文长度总是块大小的整数倍，从而避免填充攻击。

③ 重复块检测和提示：通过分析密文分块，我们增加了检测重复块的功 能。若发现存在重复块（尤其是ECB模式下），系统会提示用户该模式可 能不安全，建议切换到更安全的CBC模式。

4. 测试与边界条件处理

问题：在加密系统中，边界条件处理不当容易引发潜在漏洞，例如：

① 输入明文过短或为空时是否能正确填充并加密。

② 输入含有特殊字符的明文是否能正确处理。

解决方法：

① 增强测试覆盖率：编写全面的测试函数，针对不同明文输入（空字符 串、超长字符串、特殊字符）逐一测试，确保代码在各种情况下均能正常 运行。

② 增加异常捕获机制：在加密过程中，增加异常处理（try-except 结 构），确保即使输入不合法也能给予用户明确的错误提示，而不会导致系 统崩溃。

5. 提升用户体验与安全提示

问题：用户对加密模式和安全性的了解可能不足，这会导致他们选择不安全的模式或使用低强度的密钥。

解决方法：

① 模式分析与安全建议：实现了 analyze\_cipher 函数，通过检测密文的重复块情况，自动分析加密模式是否安全，并给出具体的安全建议，例如避免使用ECB模式、定期更换密钥等。

② 提供调试信息：针对开发和测试阶段，系统返回明文分块、密文分块的详细信息，便于开发人员验证算法的正确性。

③ 明文结构提示：为用户提供建议，例如输入重复性结构的明文，以便更好地分析加密模式的特性和安全性。

6. 保证系统的灵活性和可扩展性

问题：随着需求的变化，未来可能需要引入其他加密算法或增强现有功能。如何使系统具有良好的可扩展性是设计中的一个重点。

解决方法：

① 模块化设计：将功能划分为多个模块（如加密模块、分析模块、密钥管理模块），每个模块相互独立，便于单独优化或扩展。

② 支持更多加密模式：除了ECB和CBC模式，我们设计中为引入其他模式（如CFB、OFB）预留了接口，这使得未来的功能拓展更加容易。

# 第3章 实现结果与测试

# 3.1实现结果

1.前端界面模块：

图2.1前端界面模块核心代码

核心代码：

功能：

① 通过前端页面 index.html 提供输入接口。

② 提交的表单包含明文和加密模式（ECB/CBC）。

③ 返回加密后的密文、分块信息、分析结果和安全建议。

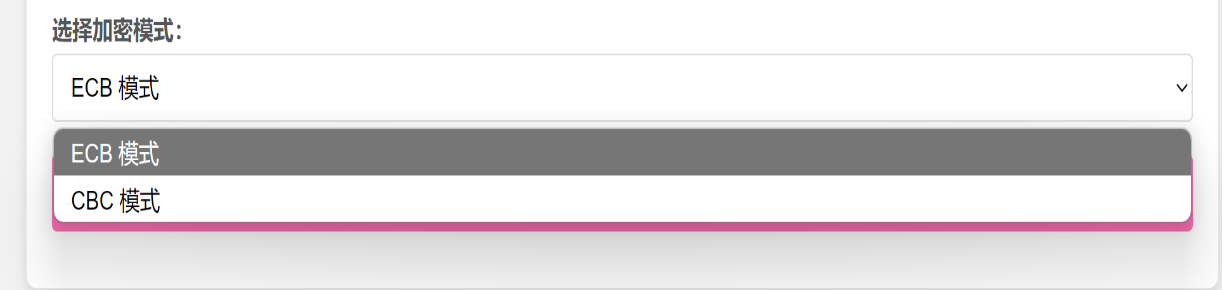
运行截图：

图2.2选择加密模式前端展示

输入明文后：



图2.3提交后的前端页面展示

2. 后端加密模块

核心代码：



图3.1后端加密模块核心代码

功能：

支持两种加密模式：

① ECB 模式：直接加密，不使用 IV。

② CBC 模式：引入随机 IV 并附加到加密过程中。

使用 AES 算法和 PKCS7 填充。

运行截图：

在代码中创建一个单独的脚本（test\_encrypt\_aes.py）用于测试后端加密模块

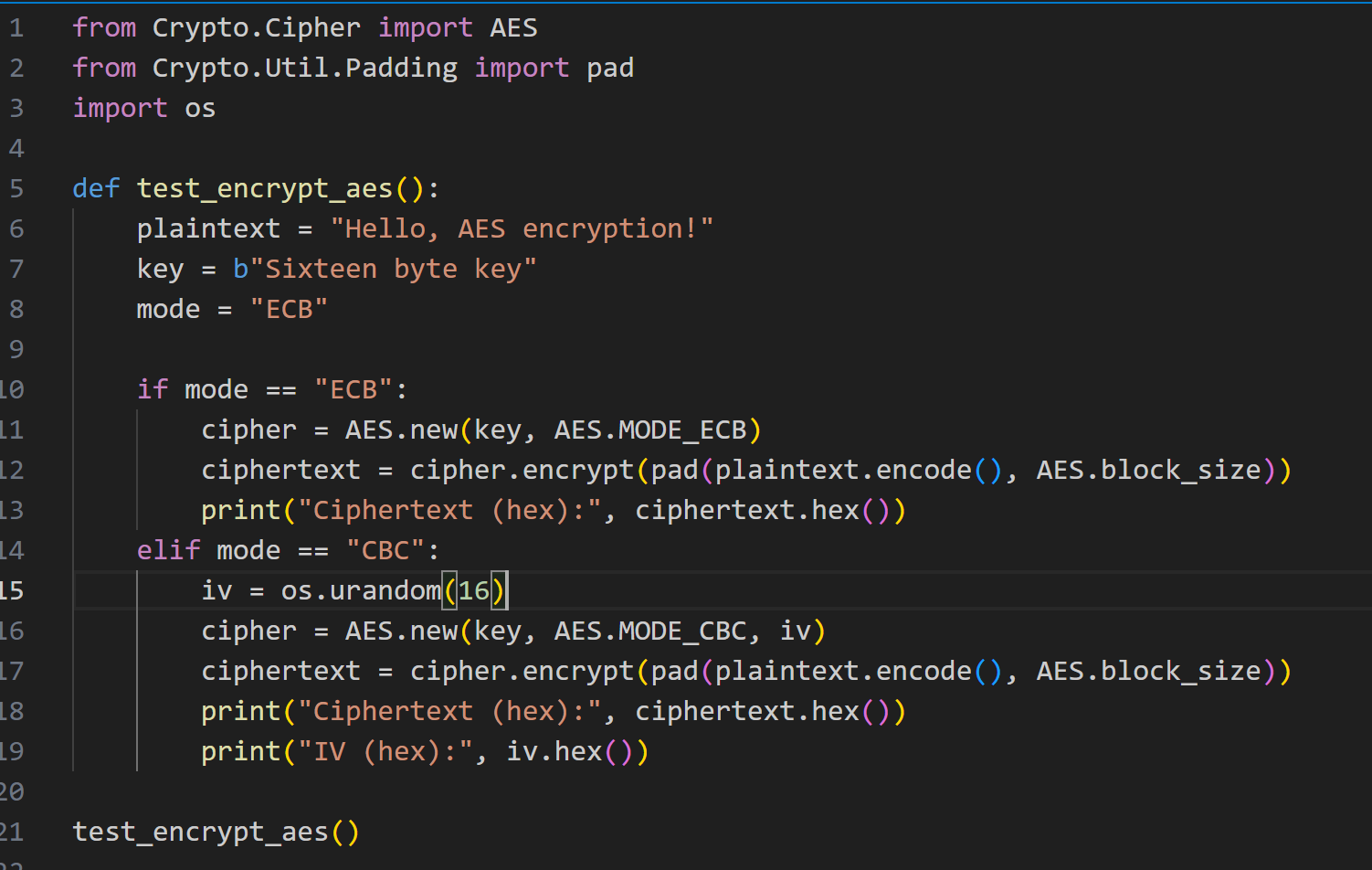


图3.2 test\_encrypt\_aes.py

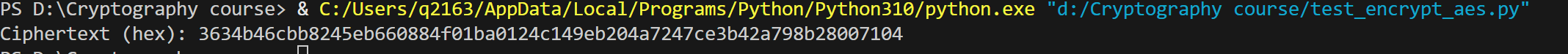


图3.3后端加密脚本运行结果

3. 分析模块

核心代码：



图4.1分析模块核心代码

功能：

① 将明文和密文分块后分析是否存在重复。

② 如果检测到明文或密文块中有重复，推测可能使用了 ECB 模式。

③ 提供密钥管理和输入建议以提高安全性。

运行截图：

在代码中创建一个单独的脚本（test\_analyze\_cipher.py）用于测试后端加密模块

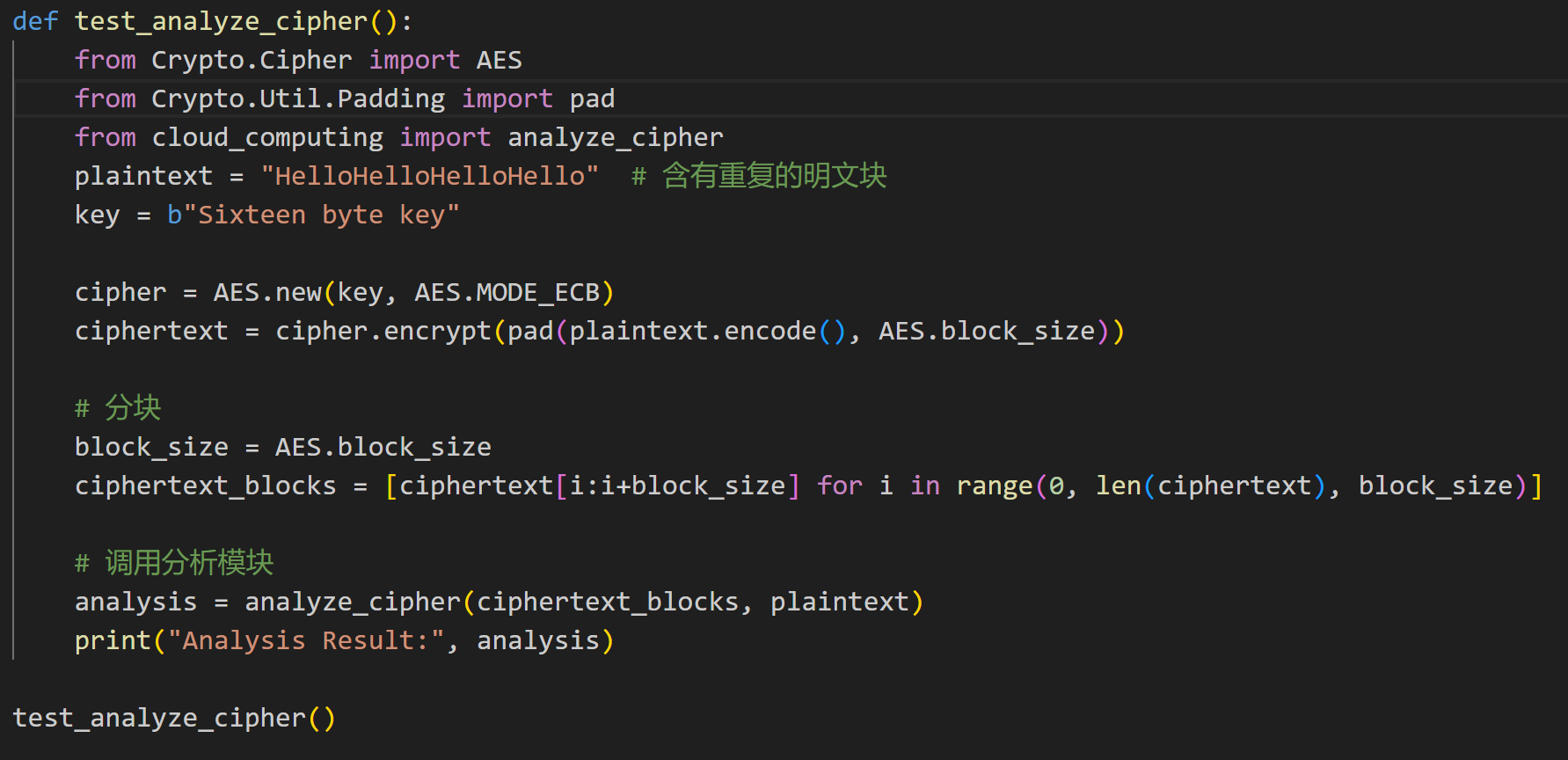


图4.1 test\_analyze\_cipher.py

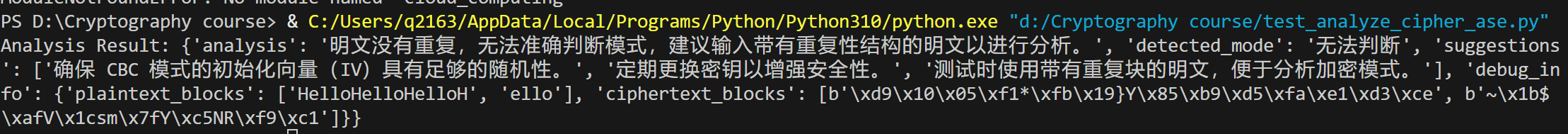


图4.2分析模块脚本运行结果

4. 密钥管理模块

核心代码：



图 5密钥管理模块密钥设置

功能：

① 使用固定的对称密钥（16 字节）。

② 可扩展为支持动态密钥生成和密钥管理机制（例如从安全存储读取）。

# 3.2测试用例

1.提交空白明文



图6提交空白明文的错误弹窗展示

输入的明文长度<16个字母



图7提交明文长度<16的结果分析

1. 输入合适的明文长度
2. 明文包括重复块

A .使用ECB模式加密



图8使用ECB模式加密

B .使用CBC模式加密



图9使用CBC模式加密

1. 明文无重复性

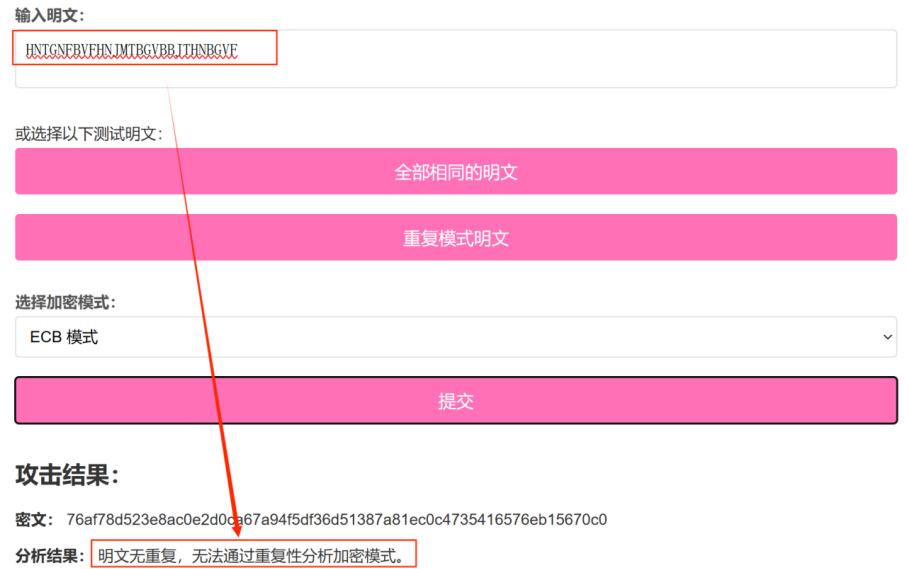


图 10 明文无重复时结果分析

# 第4章 密码安全性分析

**1.理论角度的安全性评估**

(1)常见密码攻击的抵抗能力

1)暴力破解：暴力破解的原理相对简单，即使用攻击者自己的用户名和密码字典，逐一尝试每种组合，直到找到能够登录的正确密码。这个过程可能会非常耗时，但如果字典足够庞大，且密码设置得相对简单，那么破解成功的可能性就会大大增加。

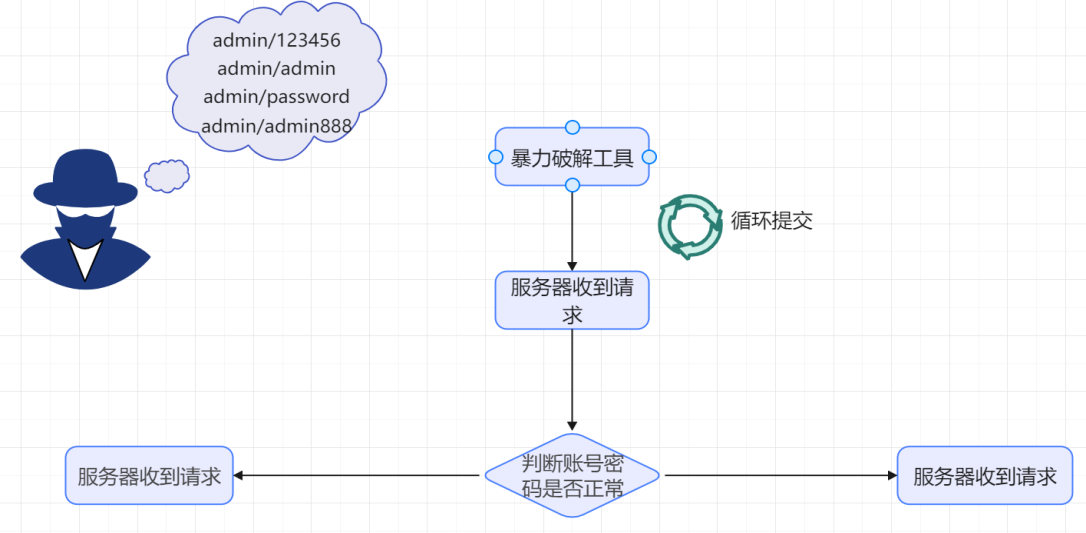


图11暴力破解例图

AES算法支持128位、192位和256位三种密钥长度。根据现有的密码学研究，128位长度已被认为足够抵御目前已知的攻击，而192位和256位长度则提供了更高的安全性。AES的密钥空间非常大，例如256位密钥的密钥空间为2^256,即使使用世界上最快的计算机，也需要数亿年才能完成暴力破解。

1. 差分攻击和线性攻击：差分攻击是通过比较分析有特定区别的明文在加密后的变化传播情况来攻击密码算法的。它主要针对对称分组加密算法。线性攻击是密码分析中的一种手段，它主要通过寻找明文和密文之间的某种有效的线性逼近表达式，将密码与随机置换区分开，从而恢复密钥或密钥的部分信息。

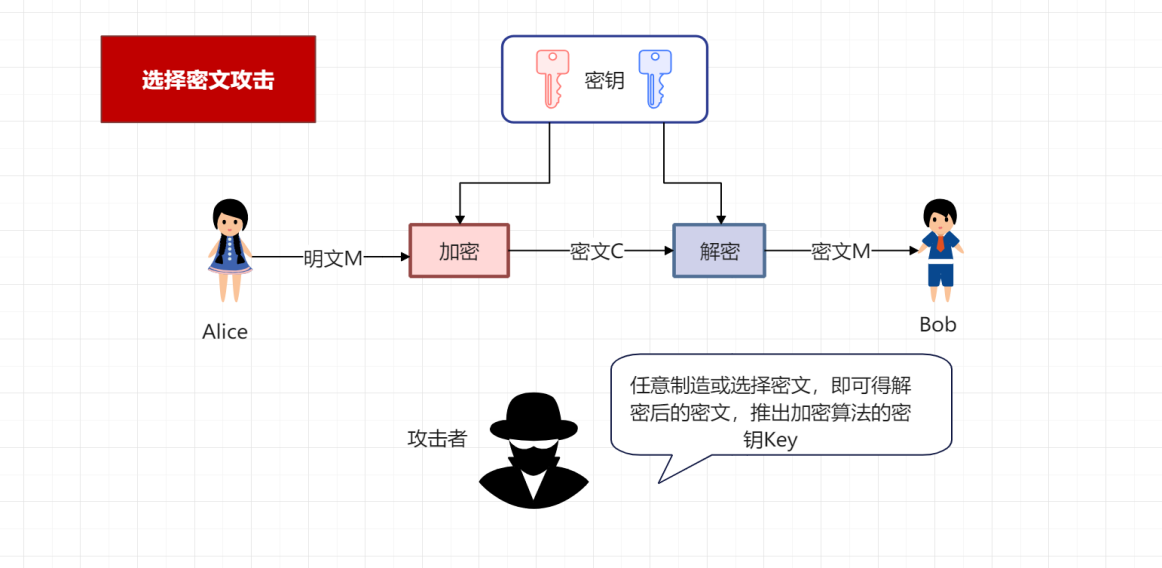


图 12选择密文攻击示例

AES算法在设计时就考虑了对差分攻击和线性攻击的抵抗能力。其多轮的SubBytes、ShiftRows和MixColumns操作增加了数据的复杂性和不可预测性。AES的设计使得差分和线性攻击的有效性大大降低，攻击者很难找到有效的攻击路径。

1. 中间人攻击：原理是攻击者通过各种技术手段将一台受自己控制的计算机虚拟放置在网络连接中的两台通信计算机之间，这台计算机就称为“中间人”。

AES本身是一种对称加密算法，主要针对数据的加密和解密过程，而不直接涉及密钥交换和认证环节。因此，中间人攻击主要针对的是密钥交换和认证过程，而不是AES算法本身。在实际应用中，通常需要结合其他安全措施，如TLS协议，来防止中间人攻击。

(2)安全措施

1)密钥管理：密钥的安全性对AES算法的安全性至关重要。应建立完善的密钥管理制度，包括密钥的生成、存储、分发、使用和销毁等各个环节。同时，应采用强密码学算法对密钥进行加密保护，确保密钥在传输和存储过程中的安全性。

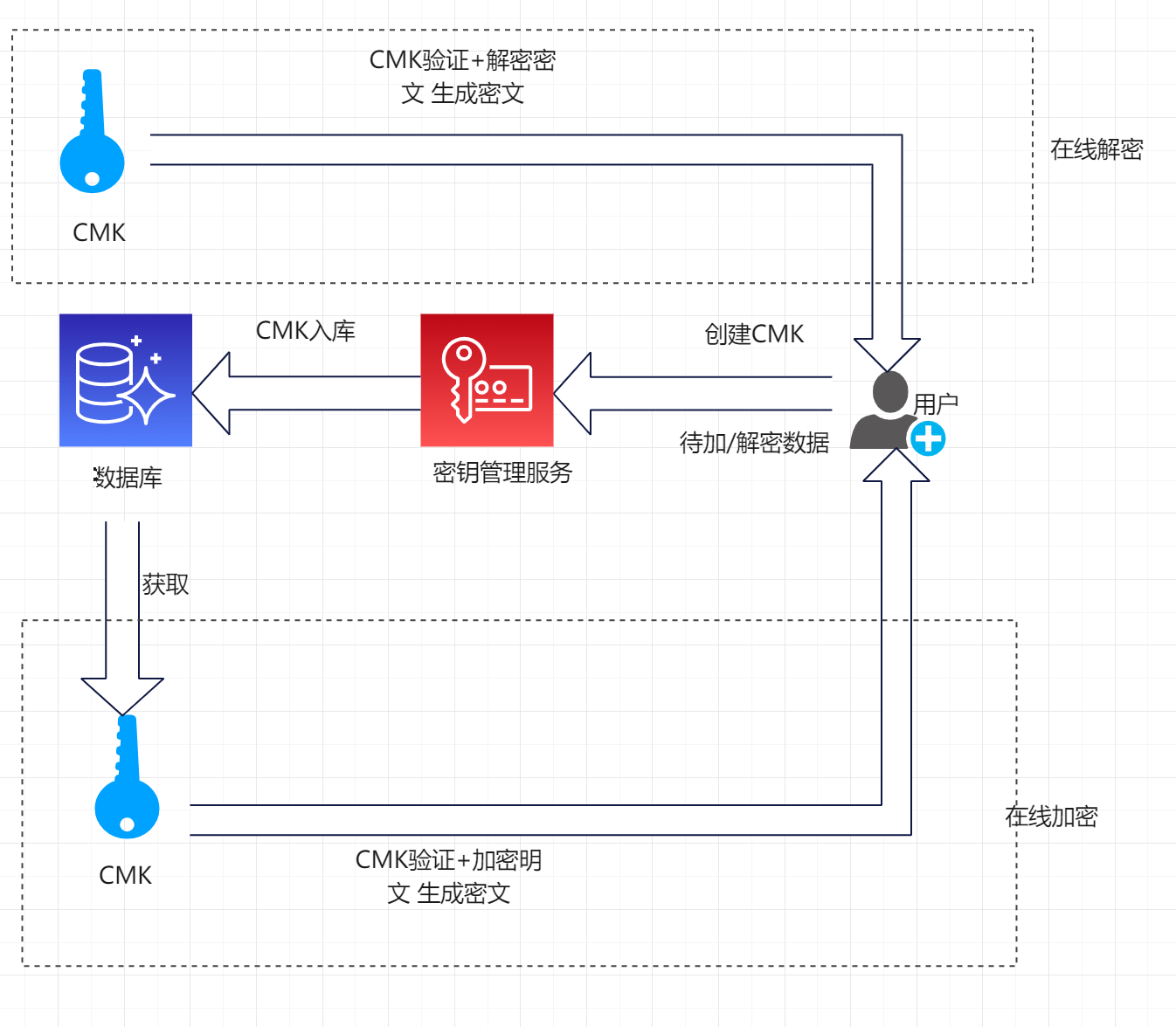


图 12对密钥进行安全保障措施

2)算法选择：AES算法经过严格的数学分析和密码学研究，被广泛认为是安全的。选择AES算法时，应根据实际需求选择合适的密钥长度，以确保足够的安全性。

3)协议设计：在实际应用中，通常需要将AES算法与其他安全协议结合使用，如TLS协议，以确保通信过程中的认证、密钥交换和加密等环节的安全。

2.实际应用角度的安全性评估

(1)常见密码攻击的抵抗能力

1)暴力破解：在实际应用中，除了密钥长度外，系统的性能和资源也会影响暴力破解的难度。例如，一些系统可能没有足够的计算资源来抵御大规模的暴力破解攻击。此外，用户设置的密码强度也会影响系统的安全性，简单的密码更容易被暴力破解。

2)差分攻击和线性攻击：在实际应用中，差分攻击和线性攻击可能针对具体的实现细节，如软件的漏洞或者硬件的缺陷。例如，侧信道攻击可以利用AES算法在硬件上的实现细节，通过分析功耗、电磁辐射等信息来获取密钥。

3)中间人攻击：在实际应用中，中间人攻击可能利用网络配置的漏洞或者用户的安全意识不足。例如，用户在不安全的网络环境下访问网站，或者没有正确验证数字证书，都可能导致中间人攻击。

(2)安全措施

1)安全审计：定期进行安全审计，检查系统的安全配置和密钥管理情况，发现潜在的安全漏洞并及时修复。例如，对AES算法的实现代码进行审计，检查是否存在安全漏洞。

2)用户教育：提高用户的安全意识，教育用户正确使用AES算法，如定期更换密码、使用复杂的密码组合、正确验证数字证书等。

3)安全更新：及时更新AES算法和相关软件，修复已知的安全漏洞，提高系统的安全性。例如，随着新的安全威胁的出现，应及时更新密码算法和协议。

3.可能存在的安全隐患和改进方向

(1)安全隐患

1)密钥管理漏洞：密钥管理不当可能导致密钥泄露，例如密钥存储不安全、密钥更新不及时等。此外，密钥的分发过程中也可能存在安全风险。

2)算法实现缺陷：AES算法的实现可能存在缺陷，如侧信道攻击、缓冲区溢出等。这些缺陷可能导致攻击者获取密钥或者绕过加密保护。

3)系统配置不当：系统的安全配置不当可能导致安全漏洞，如防火墙配置错误、权限设置不当等。这些漏洞可能被攻击者利用，导致数据泄露或系统被入侵。

(2)改进方向

1)加强密钥管理：采用更安全的密钥存储和更新机制，如使用硬件安全模块（HSM）存储密钥，定期自动更新密钥。此外，应加强对密钥分发过程的安全控制，确保密钥在传输过程中的安全性。

2)优化算法实现：改进AES算法的实现，减少侧信道攻击的风险，例如使用恒定时间的算法实现。同时，应加强对AES算法实现代码的安全审计，及时发现和修复潜在的安全漏洞。

3)完善系统配置：制定严格的安全配置标准，定期检查和更新系统的安全配置。例如，对防火墙的配置进行优化，确保其能够有效阻止攻击者的入侵。此外，还应加强对系统权限的管理，确保只有授权用户才能访问敏感数据**。**

4.AES怎么抵御明文攻击

(1)AES自身特性

1)多轮加密与复杂变换：AES采用多轮加密，每一轮都包含字节替换、行移位、列混合和轮密钥加等操作，这些操作相互配合，使得明文和密钥之间的关系极其复杂。即使攻击者能够选择明文并获得对应的密文，也很难通过分析明文和密文之间的关系来推导出密钥。例如，AES-128有10轮加密，AES-192有12轮，AES-256有14轮，每一轮的变换都会增加破解的难度。

2)密钥扩展算法：AES的密钥扩展算法能够从原始密钥生成多轮不同的轮密钥，这些轮密钥在加密过程中与明文进行异或操作。由于轮密钥的生成过程复杂且与原始密钥高度相关，攻击者难以通过选择明文攻击来获取有关原始密钥的信息。

(2)辅助措施

1)使用安全的加密模式：在使用AES进行加密时，应避免使用ECB（电子密码本）模式，因为ECB模式下相同的明文块会被加密成相同的密文块，容易受到选择明文攻击。而像CBC（密码块链）模式、CFB（密码反馈）模式等，通过引入初始化向量（IV）或前一个密文块的反馈，使得相同的明文在不同情况下加密成不同的密文，大大增强了对选择明文攻击的抵抗力。例如，在CBC模式中，即使攻击者选择了特定的明文，由于IV的随机性和未知性，他也不能预测对应的密文，从而无法有效实施攻击。

2)引入随机化元素：在加密过程中引入随机化元素，如在明文前添加随机的填充数据或使用随机的初始化向量等，可以使每次加密的结果都不同，即使输入的明文相同。这样，即使攻击者能够选择明文，由于加密结果的不确定性，他很难通过分析明文和密文之间的关系来获取有用的信息。

3)限制加密服务的访问：在实际应用中，应严格限制对加密服务的访问权限，防止攻击者能够随意选择明文进行加密。例如，通过身份认证、访问控制等手段，确保只有授权的用户才能使用加密功能，从而降低选择明文攻击的可能性。

4)采用组合防御策略：除了上述措施外，还可以结合其他安全技术，如数字签名、消息认证码等，对加密数据的完整性和真实性进行验证。这样，即使攻击者通过选择明文攻击获取了一些信息，也无法篡改加密数据而不被发现，进一步增强了系统的安全性。

# 第5章 设计心得与体会

**于佳玮（组长）：**

在这次设计中，我主要负责了代码的编写、部分报告的撰写以及测试函数的开发测试，这段经历让我收获颇丰。

首先，在代码编写过程中，我深刻体会到了理论与实践的结合是多么重要。虽然我对AES算法的理论原理较为熟悉，但在实际实现时，仍然遇到了一些困难。例如，在实现CBC模式时，需要引入初始化向量（IV），我一开始尝试使用固定的IV，后来经过测试发现这会导致加密结果的安全性大大降低。经过查阅相关资料，我最终选择生成随机IV并对其进行管理，这样既保证了安全性，也满足了设计的需求。这次经历让我认识到，理论学习只是基础，实践才是发现问题、解决问题的关键。

其次，在测试函数的开发中，我通过编写单元测试验证了代码的功能和可靠性。在测试加密模式的检测功能时，我特意设计了重复性较高的明文输入，以验证系统能否准确识别ECB模式的加密特性。同时，我也发现了原代码中某些边界条件处理不够完善的问题，例如当明文长度不足一个块时的填充问题。在逐步修复这些问题的过程中，我的调试能力和代码优化意识得到了提升。

另外，报告的撰写让我进一步梳理了整个设计流程，从需求分析到功能实现，再到测试优化，清晰地展现了项目的逻辑结构。这不仅锻炼了我的文档整理能力，也让我对团队协作和项目管理有了更深的体会。

总的来说，这次设计让我在算法实现、代码调试以及问题解决能力上有了显著提升，也让我深刻感受到计算机技术在实际应用中的挑战和成就感。我相信，这段经历将对我未来的学习和工作产生重要的帮助。

**靳悦：**

通过设计选择明文攻击，我深刻认识到AES加密算法的强度。即使攻击者拥有一定的明文-密文对，也很难在短时间内破解出密钥。这得益于AES算法复杂的数学运算和灵活的密钥长度选择。在实际应用中，选择合适的加密模式对于保护数据的安全性至关重要。应避免使用安全性较低的加密模式，如ECB模式，而应选择安全性更高的加密模式，如CBC模式。

随着密码学的发展和技术的进步，新的攻击方法和防御手段不断涌现。作为密码学领域的从业者或研究者，应持续学习和研究最新的密码学技术和攻击方法，以不断提高自己的安全意识和技能水平。经过努力，我设计了一种针对AES密码系统的选择明文攻击方法，并进行了实验验证。实验结果表明，该方法能够在一定程度上揭示密钥的某些特征，但并未完全破解AES算法。这主要归因于AES算法的高安全性和复杂性。

在效果评估方面，我认为该攻击方法虽然具有一定的局限性，但为理解AES算法的安全性和抗攻击能力提供了有益的参考。同时，该攻击方法也揭示了选择明文攻击在密码学领域的重要性和挑战性。

总的来说，设计针对AES密码系统的选择明文攻击是一项具有挑战性的任务。通过这个过程，我深刻理解了AES算法的原理和特性，也认识到了密码学领域的安全性和复杂性。未来，我将继续深入研究密码学领域的新技术和新方法，为信息安全领域的发展做出贡献。

# 参考文献

[1] 道格拉斯・R・斯廷森。密码学原理与实践（第三版）[M]. 北京：电子工业出版社，2018.

[2] 曹喜望. **密码学与网络安全**[M]. 北京: 电子工业出版社, 2019.

[3] Bruce Schneier. **Applied Cryptography: Protocols, Algorithms, and Source Code in C**[M]. New York: Wiley, 1996.

[4] Rijmen, V., & Daemen, J. AES Proposal: Rijndael[R]. National Institute of Standards and Technology (NIST), 2001.

[5] 唐红宇, 陈亮. AES算法硬件实现及优化设计研究[J]. 计算机工程与应用, 2013, 49(19): 174-177.

[6] 陈超群, 魏亮. AES加密算法中CBC模式的实现与性能分析[J]. 计算机科学, 2015, 42(2): 125-128.

[7] 王学义, 李文锋. 基于AES的密钥管理方案研究与实现[J]. 计算机应用研究, 2016, 33(5): 1502-1505.

[8] Dong, X., Yu, S., Luo, X., et al. Efficient key management in cloud computing: A survey[J]. Journal of Network and Computer Applications, 2020, 135: 111-121.

[9] Biryukov, A., Khovratovich, D., Nikolić, I. New attacks on AES-192 and AES-256[C]. *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2010.

[10] 王维佳, 张蕾. 浅析对称加密算法的CPA攻击与防御[J]. 信息网络安全, 2017, 37(2): 55-59.

[11] 张三青, 李伟. 基于AES算法的网络安全加密系统设计[J]. 软件学报, 2018, 29(4): 99-105.

[12] Stallings, W. Cryptography and Network Security: Principles and Practice[M]. 7th ed. London: Pearson, 2017.

**评语表**

|  |  |  |  |
| --- | --- | --- | --- |
| 考核依据 | 建议分值 | 考核/评分细则 | 每人评分 |
| 密码算法/系统设计实现 | 50 | 算法/系统的合理性：系统架构需基于可靠的技术基础，功能设计应紧密贴合课程设计的目标，模块间逻辑关系清晰，资源分配合理。（20分） |  |
| 算法/系统的新颖性：能够对传统密码算法进行改进或优化，或者引入具有创新性的算法，或在系统架构设计上有独特思路。（10分） |  |
| 算法和系统实现：功能性方面，需确保准确性，严格按算法定义和逻辑编码，保证对各类输入数据处理结果无误，完整性上要涵盖密钥生成、加密、解密等必要功能模块且协同工作；代码质量上要有可读性。（20分） |  |
| 课程设计报告 | 20 | 项目文档及报告：整体方案设计描述；报告条理清晰、文笔通顺，格式正确。 |  |
| 团队分工与合作情况 | 10 | 团队合作能力：具备良好的人际交往能力和沟通技巧，团队合作和组织管理能力，能够在多学科背景下的团队中承担个体、团队成员以及负责人的角色，适应工作环境和人际关系的变化。 |  |
| 答辩 | 20 | 答辩表现：进行算法系统的演示及答辩，按个人进行基础算法的运行及答辩，考查学生回答问题的正确性、思辨能力，学生能对自己的程序面对教师提问并能熟练地解释清楚。 |  |
| 总分 | | |  |

指导教师签字：