开发手册

main.py：

1.KeyExpansion：密钥扩展

功能：基于给定的初始密钥 key 扩展出更多的轮密钥。
参数：key:16bit 字符串
返回值：一个列表，包含三个字符串，分别是前两段原始密钥拼接(w0+w1)，以及通过密钥扩展得到的两个新密钥块拼接(w2+w3)和(w4+w5)

2.Encrypt：加密算法

功能：执行以下步骤来加密数据：
密钥扩展：使用提供的密钥 key 通过调用 KeyExpansion 函数生成多个轮密钥。
初始密钥加：将明文 plain_text 与第一个轮密钥（expanded_key[0]）进行 XOR 操作。
轮函数：依次执行替换操作 SubNib、行移位 ShiftRows 和列混合 MixColumns。
密钥加：再次与第二个轮密钥（expanded_key[1]）进行 XOR 操作。
轮函数（不包括列混合）：继续执行替换操作 SubNib 和行移位 ShiftRows。
最终密钥加：最后与第三个轮密钥（expanded_key[2]）进行 XOR 操作。
参数：
plain_text:16bit 明文数据
key:16bit 加密密钥
返回值：经过加密处理后的密文 cipher_text

3.Decrypt：解密过程

功能：执行以下步骤，使用给定的密钥将密文转换回原始的明文：
密钥扩展：使用提供的密钥 key 生成多个轮密钥。
初始密钥加：将密文 cipher_text 与最后一个轮密钥（expanded_key[2]）进行 XOR 操作。
逆向轮函数：首先执行行移位 ShiftRows，然后执行逆向替换 InvSubNib。
密钥加：与倒数第二个轮密钥（expanded_key[1]）进行 XOR 操作。
逆向列混合：执行 InvMixColumns 操作。
逆向轮函数：再次执行行移位 ShiftRows 和逆向替换 InvSubNib。
最终密钥加：最后与第一个轮密钥（expanded_key[0]）进行 XOR 操作。
参数：
cipher_text:16bit 密文数据
key:16bit 密钥
返回值：解密后的明文 plain_text

extend.py：

1.ascii_to_binary：
功能：将 ASCII 文本字符串转换为其对应的二进制表示形式。
参数：ascii_text:输入的 ASCII 编码的文本字符串
返回值：一个包含了输入文本中每个字符的二进制形式的字符串

2.binary_to_ascii
功能：从一个二进制字符串中提取出 ASCII 字符，从而将二进制表示的文本转换回可读的 ASCII 文本格式。
参数 binary_text:一个由 0 和 1 组成的字符串，表示二进制数据
返回值：一个字符串，包含了从输入的二进制字符串中解析出来的 ASCII 字符

3.ascii_encrypt：
功能：将 ASCII 编码的明文字符串加密。
参数：
plain_text:明文字符串，即需要加密的信息
key:加密密钥，用于加密算法中生成子密钥
返回值：一个字符串，包含了加密后的内容，其中每个字符都是经过加密处理后转换回来的 ASCII 字符

4.ascii_decrypt：
功能：从给定的密文字符串中解密出原始的 ASCII 明文字符串。
参数：
cipher_text:密文字符串，即需要解密的信息
key:解密密钥，用于解密算法中生成子密钥
返回值：一个字符串，包含了解密后的内容，其中每个字符都是经过解密处理后转换回来的 ASCII 字符

multiple_encryption.py：

1.double_encrypt：

功能：对输入的明文字符串进行两次加密，第一次加密使用第一个密钥，第二次加密则使用第二个密钥。

参数：

plain_text:要加密的明文字符串；key1:第一把密钥；key2:第二把密钥

返回值：cipher_text2：经过两次加密后的最终密文字符串

2.double_decrypt:

功能：对经过两次加密的密文进行解密，恢复出原始的明文。它首先使用第二个密钥对密文进行解密，随后使用第一个密钥对得到的中间明文再次解密

参数：cipher_text:要解密的密文字符串；key1:第一把密钥；key2:第二把密钥

返回值：plain_text2：经过两次解密后的最终明文字符串。

3. generate_all_keys：

功能：生成所有可能的 32 位二进制密钥对。该函数通过遍历所有 32 位的二进制数，逐步生成并返回每一对密钥。

4. middle_meet_attack：

功能：利用了明文和密文在中间过程中的相等性，通过已知的明文和密文列表来找出潜在的密钥对。找到一个密钥对后，跳出函数。

参数：

known_plain_text_list: 一个包含已知明文的列表

known_cipher_text_list: 一个包含已知密文的列表

返回值：找到匹配的密钥对，则返回一个列表，列表中包含一个元组 [(key1, key2)]。

5.triple_encrypt：

功能：对输入的明文字符串进行第一次加密：将明文 plain_text 使用 key1 加密，得到 cipher_text1。再进行解密：对 cipher_text1 使用 key2 解密，得到 cipher_text2。第二次加密：最后，再将 cipher_text2 使用 key1 加密，得到最终的 cipher_text3。

参数：plain_text:要加密的明文字符串；key1:第一把密钥；key2:第二把密钥；

返回值：cipher_text3：经过两次加密和一次解密后的最终密文字符串

6.triple_decrypt:

功能：对输入的密文字符串进行第一次解密：将密文 cipher_text 使用 key1 解密，得到 plain_text1。再进行加密：对 plain_text1 使用 key2 加密，得到 plain_text2。第二次解密：最后，再将 plain_text2 使用 key1 解密，得到最终的 plain_text3。

参数：cipher_text: 需要解密的密文字符串或数据；key1: 第一个解密密钥，用于首次解密和最后一次解密；key2: 第二个解密密钥，用于中间的加密操作。

返回值：plain_text3：经过两次解密和一次加密后的最终明文字符串。

CBC.py：

1. CBC_encrypt 函数
功能：实现密码分组链（CBC）模式的加密。
参数：
plain_text：明文字符串，长度必须是 16 的倍数
key：16 位 2 进制密钥。IV：16 位 2 进制初始向量
返回值：加密后的密文字符串

2. CBC_decrypt 函数
功能：实现密码分组链（CBC）模式的解密。
参数：
cipher_text：密文字符串，长度必须是 16 的倍数
key：16 位 2 进制密钥
IV：16 位 2 进制初始向量
返回值：解密后的明文字符串

3. test_CBC 函数
功能：测试 CBC 模式的加密和解密。
参数：无。
返回值：无。

4. tamper_cipher_text 函数
功能：替换或修改密文分组。
参数：无。
返回值：无。