Github（或者 Coding）账号：
https://github.com/tsrigo/xdu_crypto_exps

个人博客关于密码学实验的链接：

1. https://blog.csdn.net/weixin_45574854/article/details/134237072
2. https://blog.csdn.net/weixin_45574854/article/details/134237085
3. https://blog.csdn.net/weixin_45574854/article/details/133282915
   https://blog.csdn.net/weixin_45574854/article/details/133324316
4. https://blog.csdn.net/weixin_45574854/article/details/134237344

实验题目（中文）：

1.
多次加密
2.
PA1 选做题
3.
(1)将十六进制转换为 base64
(2)固定 XOR
(3)单字节异或密码
(4)检测单字符异或
(5)实现重复键异或
(6)断开重复键异或
4.
MTC3 破解 sha1 哈希密码

实验摘要（中文）：

关于密码学实验的说明

1. 密码学实验将进行四次，每次实验，需按要求上传提交代码截图、相关结果等。
2. 请建立自己的技术博客或者其它记录载体，简单记录每次实验内容，所遇到的问题以及心得（建议）。
3. 因学校要求提交实验报告以给出成绩，我们只交一次纸质版实验报告，内容 4 次实验任选。
4. 最终提交时间 11 月 30 日晚 23:00 前。
5. 请建立自己的代码托管账号，Github 或 Coding 或其他托管平台均可，建立合理的文件目录托管代码，请清晰命名，给出必要注释；
6. 电子版提交至 63307507@qq.com，需按要求时间，提交四次实验报告邮件命名为“姓名_学号_密码学实验”，提交 pdf 版附件命名“姓名_学号_密码学实验”。

本实验主要涉及四个密码学相关的题目，分别是：

- Many Time Pad，要求利用异或运算的性质，破解使用相同密钥的流密码加密的密文，揭示其中的秘密消息。
- PA1 option，要求编写一个程序，破解使用类似维吉尼亚密码的字节异或加密的密文，恢复明文。
- http://www.cryptopals.com/sets/1，要求完成一系列的密码学挑战，涉及编码转换、异或运算、单字符异或密码、重复键异或密码、AES 加密等。
- MTC3 Cracking SHA1-Hashed Passwords，要求在给定 SHA1 哈希值的情况下，揭示明文密码，利用已知的密码特征。

题目描述（清楚描述题目中文，写出自己的理解，请勿复制原题目）

**1.**

**Many Time Pad**

让我们看看当一个序列密码密钥被多次使用时会发生什么。下面是 **11** 个十六进制编码的密文，它们是用流密码对 **11** 个明文进行加密的结果，它们都使用相同的流密码密钥。你的目标是解密最后一个密文，并将其中的秘密消息作为解决方案提交。

提示:将密文一起异或，并考虑当空格与**[a- za -z]**中的字符异或时会发生什么。

**2.**

**PA1 option**

编写一个程序，让你能够“破解”使用类似 **vigenere** 的密码生成的密文，其中

使用的是按字节异或运算，而不是求模 26 的加法运算。

3.

http://www.cryptopals.com/sets/1

(1)将十六进制转换为 base64

(2)固定 XOR

(3)单字节异或密码

(4)检测单字符异或

(5)实现重复键异或

(6)断开重复键异或

4.

MTC3 Cracking SHA1-Hashed Passwords

日志含义基于密码的认证是指用户发送明文密码到服务器，服务器根据明文密码计算 hash 值，并与存储的 hash 值进行比较。这项挑战的目标是在给定 SHA1 哈希值的情况下揭示明文密码。关于原始密码，我们知道了一些信息。隐藏详细信息…

过程（包括背景，原理：必要的公式，图表；步骤，如有必要画出流程图，给出主要实现步骤代码）

1.

步骤 1:理解题目要求

这是一个基于异或运算的单字节流密码的破解题。给出了 11 个加密消息 MSGS[0-10],目标是破解出 MSGS10 的明文。提示中提到可以将 TARGET 与其他 MSGS 依次异或,根据结果来推断 TARGET 的明文。

步骤 2:实现异或运算

代码中已经给出了实现异或运算的 strxor()函数,可以直接调用。将 TARGET 与其他 10 个 MSGS individually 异或。

步骤 3:解析异或结果

观察异或结果,如果某位出现大写字母,则该位 TARGET 和 MSGS 之一为小写字母,另一个为空格。根据该规律可以推断出 TARGET 的大致明文。

步骤 4:调整明文

根据推断出的 TARGET 明文,存在一些词语错误。添加标点符号冒号和逗号进行调整,再次异或验证,明文更加通顺。

步骤 5:得到最终明文

通过多次调整验证,最终得到 TARGET 的正确明文:“The secret message is: When using a stream cipher, never use the key more than once”。

2.

步骤 1:理解题意

这是一道基于维吉尼亚密码变种的密文攻击题,给出了使用固定长度密钥的字节异或加密的密文,需要恢复明文。

步骤 2:分析密文特点

密文是 16 进制表示的字节序列,密钥长度未知但在 1-13 之间,明文包含字母、空格和标点符号,不包含数字。

步骤 3:枚举密钥长度

枚举可能的密钥长度,对每个长度下提取密码分组,暴力枚举异或密钥,判断解密结果是否含非法字符,得到可能的密钥空间。

步骤 4:搜索密钥空间

遍历所有可能的密钥组合,解密并验证明文,找到正确的密钥。

步骤 5:优化搜索

可适当减小明文字符范围,缩小密钥空间,使搜索更快收敛。

步骤 6:解密明文

使用得到的正确密钥解密密文,恢复明文。

总的来说,这类固定密钥长度的字节流密码可通过分组枚举攻击,选择合适字符集合可以大幅优化密钥搜索。

3.

第一题:使用 Python 中的 int、chr、join 和 base64 等函数,可以轻松实现 base64 编码和解码。

第二题:与第一题类似,同样利用 Python 内置函数,实现十六进制和 base64 之间的转换。

第三题:通过暴力枚举所有可能的密钥,并设计一个评分函数作为判断标准,对解密结果进行评分,找到评分最高的密钥。

第四题:基于第三题的代码,进行暴力枚举攻击。需要注意,题目没有说明异或后的字符就是字母,这是一个陷阱。

第五题:调用之前实现的固定异或函数,但需要注意字符串与二进制、十六进制编码之间的转换需要对齐长度,不足补零。

第六题:首先实现汉明距离计算函数作为评分标准,读取 base64 编码的密文,按照给定方法计算不同密钥长度的评分,确定最有可能的密钥长度。然后对密文分组进行单字符异或破解,逐位得到密钥,并最终解密。

第七题:根据提示不建议使用 OpenSSL,我直接用 Python 代码实现 ECB 模式的解密。

第八题:检测 ECB 模式是否使用的方法是,统计密文分组的重复出现频次,ECB 模式下同一密文块重复加密会出现相同的密文,从而检测到 ECB 模式。

4.

步骤 1: 初始化。定义一个空的路径和给定的字符集。

步骤 2: 深度优先搜索。从字符集的第一个字符开始，对于每个字符，选择其两种可能的形式（例如，大写或小写），并将其添加到路径中。

步骤 3: 递归。对于字符集中的下一个字符，重复步骤 2。如果已经处理了所有的字符，那么就得到了一个可能的密码组合。

步骤 4: 检查。对于得到的每一个可能的密码组合，生成其所有可能的排列，并计算每个排列的 SHA1 哈希值，与给定的哈希值进行比较。

步骤 5: 找到答案。如果找到了一个匹配的哈希值，就打印出对应的密码和搜索所用的时间，并停止搜索。

总结（完成心得与其它，主要自己碰到的问题和解决问题的方法）

实验过程中，我使用了 Python 语言，利用了其内置的函数和模块，实现了各种密码学相关的算法和工具。

实验过程中，我学习了许多密码学的基本概念和技巧，如异或运算、哈希函数、加密模式、评分函数等。我也掌握了一些密码分析的方法，如枚举、分组、频率分析、汉明距离等。我还提高了我的编程能力和逻辑思维能力，解决了一些实际的问题。

实验中，我也遇到了一些困难和挑战，如如何确定密钥长度、如何缩小密钥空间、如何处理不同的编码格式、如何优化搜索效率等。我通过查阅资料、参考代码、尝试不同的方法、比较不同的结果等方式，逐步克服了这些困难，找到了合适的解决方案。

这是一个有趣而富有挑战性的实验，让我对密码学有了更深入的了解和兴趣，也锻炼了我的实践能力和创新能力。我希望能够继续学习和探索更多的密码学知识和应用，为网络安全和信息保护做出贡献。

参考文献（包括参考的书籍，论文，URL 等，很重要）

https://github.com/Morrandir/Crypto001_Week1

https://www.cnblogs.com/elpsycongroo/p/7669786.html