信息安全的艺术　实验三

PB23010356 张竞一
学院：数学科学学院

2025 年 11 月 3 日

1　实验目的

1. 掌握对称密码算法原理
2. 掌握非对称（公钥）密码算法原理及应用
3. 了解哈希函数的原理及使用
4. 掌握数字签名的功能特点和处理流程
5. 掌握密码破解的基本方法

2　实验原理

2.1　对称加密算法

对称加密算法是使用相同的密钥进行加密和解密的算法。常见的对称加密算法包括 DES、3DES、AES 和 RC4 等。

2.2　哈希函数

哈希函数是一种将任意长度的输入数据转换为固定长度的输出值的过程。常见的哈希算法包括 MD5、SHA-1、SHA-256 等。

2.3　非对称加密算法

非对称加密算法使用两个不同的密钥：公钥和私钥。RSA 是最常用的非对称加密算法。

2.4　数字签名

数字签名是一种基于密码学技术的电子标识，用于验证数字信息的来源、完整性和不可否认性。

2.5　已知明文攻击

已知明文攻击是一种密码分析攻击方法，攻击者拥有一些明文及其对应的密文，试图推导出加密密钥或算法。

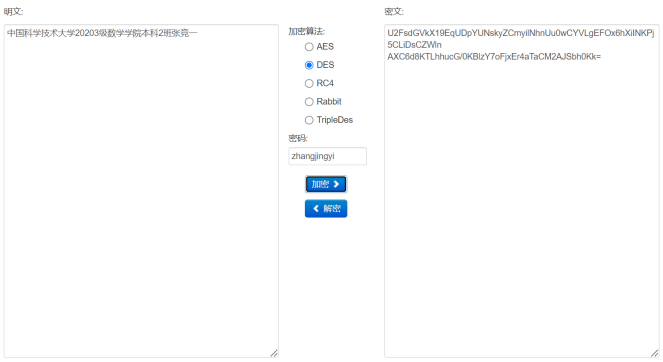
3 实验环境

操作系统：Windows 11
在线加密工具：`https://tool.oschina.net/encrypt`
RSA 加密工具：`https://tools.jb51.net/password/rsa_encode`
数字签名工具：`https://www.metools.info/code/c82.html`

4 实验结果

4.1 对称加密

- DES 加密结果：
  - 明文：中国科学技术大学 20203 级数学学院本科 2 班张竞一
  - 密钥：zhangjingyi
  - 密文：U2FsdGVkX19EqUDpYUNskyZCmyilNhnUu0wCYVLgEFOx6hXiINKPj5CLiDsCZWln AXC6d8KTLhhucG/0KBlzY7oFjxEr4aTaCM2AJSbh0Kk=



- 3DES 加密结果：

  密文：U2FsdGVkX1/vUpG/nImAuup4bMEfjWLqZyCcn4EE1tijGjaUe2cfY9MCR7/5jr/x gYvKnw9n4XMY2T2RsGTm+KoviDuiOqI/6wHygrfEx6c=

- AES 加密结果：

  密文：U2FsdGVkX181RPQhmHvvmJBCBYwQcI41buk5h7g4m6bbRT3cUicPvNHrzTDB0I0Q
  nL9Hp6vlDAVQFMYtDfOEhXYclOdwFd+Cb3iCuOMyeT4=

- RC4 加密结果：

  密文:U2FsdGVkX18eMGdmLvbHUWdRcyuRyfXTDZiNuiMPHxlv/fPARJ7j1FXOP7R83lH5 afUGJqaWv/swf-BmVaMLkwj/EsXTvmIO/zpf32SyC2w==

4.1.1 解密分析

解密时除了需要完整的密文外，还需要以下信息：
1. 加密算法（DES、3DES、AES 或 RC4）
2. 密钥（必须与加密时使用的密钥完全相同）

4.1.2 安全风险分析

在传递这些信息时存在以下安全风险：
1. 密钥传输风险：如果密钥通过不安全的通道传输，可能被第三方截获
2. 算法选择风险：某些算法（如 DES）已被证明不够安全
3. 密钥管理风险：多人共享同一密钥时，难以追踪密钥泄露源

4.1.3 解决方案

1. 使用非对称加密算法（如 RSA）加密对称算法的密钥，实现安全的密钥传输

2. 选择强度更高的加密算法（如 AES-256）
3. 实施密钥轮换机制，定期更换密钥
4. 建立安全的密钥分发机制，如使用 PKI（公钥基础设施）

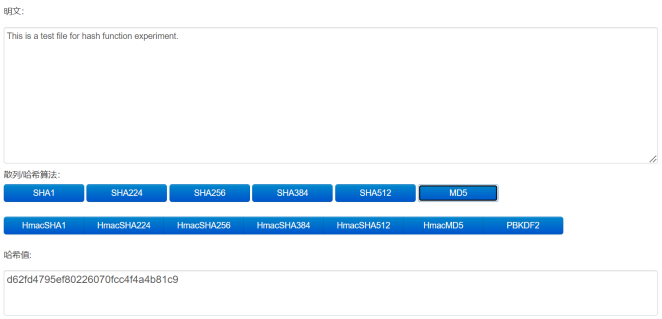
4.1.4 小结

本实验我成功使用了多种对称加密算法对数据进行加密和解密，理解了对称加密的工作原理和应用场景。实验表明，对称加密算法操作简便，加密速度快，但密钥的安全传输和管理是其主要挑战。在实际应用中，通常需要结合非对称加密技术来解决密钥分发问题，形成混合加密系统。

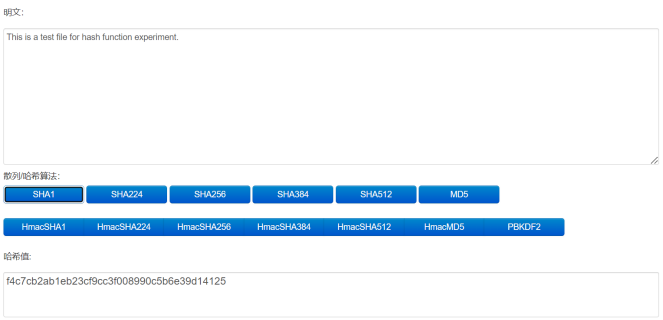
4.2 哈希函数

明文：This is a test file for hash function experiment.

- 在线哈希工具计算结果：
  - MD5：d62fd4795ef80226070fcc4f4a4b81c9



  - SHA-1：f4c7cb2ab1eb23cf9cc3f008990c5b6e39d14125

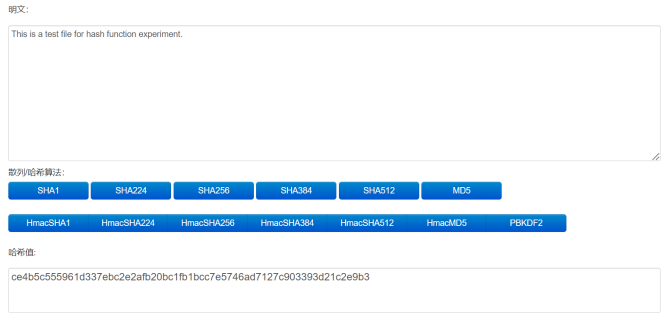


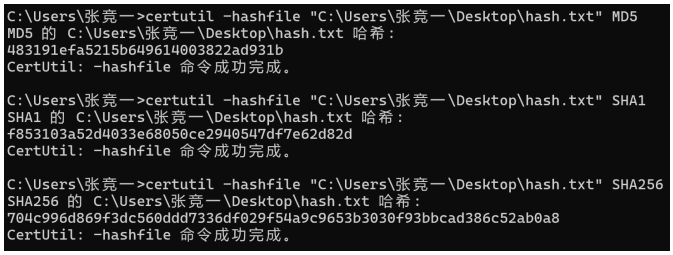
  - SHA-256：ce4b5c555961d337ebc2e2afb20bc1fb1bcc7e5746ad7127c903393d21c2e9b3

- Windows 命令行工具计算结果：

4.2.1 比较分析

通过比较不同工具计算的哈希值，我们发现：
1. 同一文件使用相同的哈希算法，无论使用哪种工具计算，得到的哈希值都是一致的
2. 不同哈希算法计算的结果长度不同：MD5 为 32 个十六进制字符，SHA-1 为 40 个十六进制字符，SHA-256 为 64 个十六进制字符





4.2.2 哈希函数应用场景

1. 文件完整性校验：通过比较下载文件的哈希值与官方提供的哈希值，可以验证文件是否被篡改或损坏
2. 密码存储：将用户密码哈希后存储，避免明文存储密码
3. 数据去重：使用哈希值作为数据的唯一标识符

4.2.3 小结

通过本实验，我成功使用不同工具计算了文件的哈希值，并验证了哈希函数的一致性和平台无关性。哈希函数作为一种单向加密技术，在信息安全领域有着广泛的应用。实验表明，哈希函数能够将任意长度的输入转换为固定长度的输出，且同一输入总是产生相同的输出，这使其成为验证数据完整性的理想工具。

4.3 RSA 非对称加密实验报告

- 生成的 RSA 密钥对：
  - 公钥：

-----BEGIN PUBLIC KEY-----
MIIBITANBgkqhkiG9w0BAQEFAAOCAQ4AMIIBCQKCAQB8OboQA6Lq7Nc79Qfoy7lB
iKCpHBdMhHLVarRMv5ta8vgKO/rFERv+Jd63vi/ZLCY14xseCC+0wM+5DxW5+jYd
+f0BNRRioYZ7hrT1oyZ6lYtdlb/n0xGrEVm50GnXATaX8V1EDeCItOkVu8BfOw87
kMnLuaASHMuaOJ8U0S3pd+zpjZlUNzu/2m596raCbTIFb4QQBMpYvqPID1wfJL3U
1S4VwUvr0tGB9JqjKYGOTs5Xv+Y8IRoeGzXURDOSS13hiDxjGX5lATCp/oVCHKVT
EypsJyZ16GXRnUR4m3HTtvidblzIIQ2NXtcjVpNqOFIoo32v0Dgm0zC7yESrcp2x
AgMBAAE=
-----END PUBLIC KEY-----

  - 私钥：

-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQB8OboQA6Lq7Nc79Qfoy7lBiKCpHBdMhHLVarRMv5ta8vgKO/rF
ERv+Jd63vi/ZLCY14xseCC+0wM+5DxW5+jYd+f0BNRRioYZ7hrT1oyZ6lYtdlb/n
0xGrEVm50GnXATaX8V1EDeCItOkVu8BfOw87kMnLuaASHMuaOJ8U0S3pd+zpjZlU
Nzu/2m596raCbTIFb4QQBMpYvqPID1wfJL3U1S4VwUvr0tGB9JqjKYGOTs5Xv+Y8
IRoeGzXURDOSS13hiDxjGX5lATCp/oVCHKVTEypsJyZ16GXRnUR4m3HTtvidblzI
IQ2NXtcjVpNqOFIoo32v0Dgm0zC7yESrcp2xAgMBAAECggEAV1F0WaFUuPD1cW2Q
u6/HbisNUxEtx1kqJp00UuYz20AB3Z+/axnVpVUVMe57T5na2fHis79pWRJWcGwM
B4e61mNL1bdrttB26QkM1cRyXaQULBoApIEWYDp36UZ3vutZZn//4MAoXa9cTyyD
0ex/joz9azf1LgYTjznRiB5ouTvTX9ORuf1w7pmBV26ZNn45WxB+5cZtTYKPz65v
vO9oWr+3+uwMU/4o1kdYSu27Hy7BqlUeq+o226A2RCT2OKNpadfqi6tWmtT1iGPa
FUu42feM3AdPTIIwxxa07xzRmoFYymm1OiJfIZ5u5RE4gH3h+O+LQuc0yE8FakSC
r85wAQKBgQDgpO2Z4+AMZ8LymIW2mmZykOgUsi7DIynhjMqix2mMOhjqbSK87GVm
JVNCcvFjWvlYpI7uHt4376iSiSKHXprk0orbQxMgTINBvIWtm+R0d+oODNbx6klg
3pAxkr/2BYSr3CrB3k+iNxpsVdBQ+n4+vri9ozqkC0exRfW3MdD7MQKBgQCNkJkv
5oOxfpnbdAbvI79kyYLg2A4Bfy6WRbnOpxURyGSwbr7+rQhzZSpIbS6ai1miVGcO
zej2iG+U/LZFdjkDmtMqFqUXj+tl2zQHqyqhXqo7/tD+GLYY030hRRlJjKvaDgSZ
XotsoFYs4CyGBnWbpoyZKDZZ8rvNkNHp704qgQKBgQDKpt4VEoW+iyhJUyBkzzqW
JXTHnsoijSkt6KWwXLYxowNxD0zkI1mB4TAP5MMvvxTrS2AJHd2XlmJ69q0oc0bG
H8zV6ddqLrvF5HyvVc+s5bRGQnzxNUpG0eBAaEM1fFrn8rC7vYIXMyBqFKvdMWwL
zF+563NuIyLkn9j/DziYMQKBgG7SmSmnkscb7gSfo2IIoSecnUm5udUwJF9dvGTr
5hGiyDhaqkijWu9DbkN+vbIcjxIGcxti1RJSos+GuoZMU7vfs5nKWnnEoCp/3Q53
0Aq+GhSXmjKhU8TBo76s4mvypQIyYSVTn2BtcKj24cQWQHGWg5DOqyXlYWI/EiXJ
wm6BAoGAG/9Bo30stiu2WWO7IjCqSlV9bkPVrqplYbSU9LiYZvGVOGj9lt6kt0Vw
2BQvzE/sBdF76cYkzmpr//E0rwcw09TzWGOflHFeewAC4p1TvO0pDbvnF8ykd18l
3+dVeMZL9twAvrL9MOnSl6nLaS2X89SU8b4T159WVukI2Vrivzs=
-----END RSA PRIVATE KEY-----

- 加密结果：
  - 明文：This is a confidential message for RSA encryption test.
  - 使用公钥加密后的密文：

DyphrlsTl0drZ+32/JrSJCBY4W78UUvhVNEws5pkRLOFWKO4aaF9cRDi0ml0cJUO

X0CbicnzoeZu1tHgHHp4ziiDfbnH4SAvhMPclMWJJsaDUPny7eWZNe7NomSRh6X1

m7T8sgeA8GKX++C7z30SV8bd4x7VABHJv7K3aQBSkRvrCX5kwTuJhY6AzwdHvmTY

p9iglE9XsfvcidAcwr6X9R+448ieSP2lyKEDhWVjNF1Q477N3MPgI7kPNARJsNLB

4DunuYkk0dFZ898j4qMolW5mwlNp7D84XK2US/n0eLiIY6g/1FiNUpK+WrTlTbZ0

DR2p939GM0NXBFyTA2hHDg==

- 解密结果：
  - 使用私钥解密后的明文：This is a confidential message for RSA encryption test.

4.3.1 小结

通过本实验，我们成功使用 RSA 算法生成了密钥对，并完成了加密和解密过程。实验表明，RSA 非对称加密算法通过使用不同的密钥进行加密和解密，有效解决了对称加密中密钥分发的问题。公钥可以安全地公开分发，而私钥需要严格保密。这种机制使得即使攻击者获取了公钥和密文，也无法在合理的时间内解密获取明文，从而保证了信息的安全传输。

4.4 数字签名

- 使用的 RSA 密钥对：
  - 公钥：

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4HvaczVUVOd4uUE1Ghu9
SeZ+q3iKLcwpN+3IefpGJgJUGrcx8J9Ehwu5Xet1gdpUjz7XnMhl+Wilg+3K8x0I
P/fqmNJW8kkafBvK2Tk9dvJpwkc9T+/6ZHS5EcUefoEyzRKaNY+XdxsW8i3lWWoj
HhMhFhiCj+UCqsubGEgckSaxWed/UTYxltfkmzfVJghA1qliD03QE3x9FDdYh0kd
5xJChicle4YkjN518P37hWPTLxBXdYF2hLKRuFSi26BgND9LU/pp2FiXdH9DaQ4z
bqGfD/5odVcmsxh7X7OE20RS75JOiwejcUfJgXkgRYB6LY5avZcm1kXHf7X4YLjW
NwIDAQAB
-----END PUBLIC KEY-----

  - 私钥：

-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDge9pzNVRU53i5
QTUaG71J5n6reIotzCk37ch5+kYmAlQatzHwn0SHC7ld63WB2lSPPtecyGX5aKWD
7crzHQg/9+qY0lbySRp8G8rZOT128mnCRz1P7/pkdLkRxR5+gTLNEpo1j5d3Gxby
LeVZaiMeEyEWGIKP5QKqy5sYSByRJrFZ539RNjGW1+SbN9UmCEDWqWIPTdATfH0U
N1iHSR3nEkKGJyV7hiSM3nXw/fuFY9MvEFd1gXaEspG4VKLboGA0P0tT+mnYWJd0
f0NpDjNuoZ8P/mh1VyazGHtfs4TbRFLvkk6LB6NxR8mBeSBFgHotjlq9lybWRcd/
tfhguNY3AgMBAAECggEASMDbM6t9AWdjgObZRvTmjCTfuMMARYd9dFrkNmQeiAMJ
Lv9geBOmKf5qvT2qf8M61JIRXwazxflcaz2OIOyv18uJhK3m7+8xXjfDrlMW0I4Q
KJ7eCCJGjHgn4tJ355gH1t7Ii8NV24w0C5lEU45Kg1y/xGF9LxWRn7kNMvlsCL5e
JimKxjay/3WSzV7f3Vy5SbJ2yUYNTARnsw6ciVT7gvJkJlMaR27RslFlJUoNDOkA
qi2Zls+W76EhKzYLZlky8g3QENBEbASt0MRQVn8HLexjfL2TS9CPELS/R4IEG/4b
xm/xwsOKH9rw3cwgJQ+RIYSE1MuPctXiyM51IsexAQKBgQDylCIKufqHWzbv4xD+

EXOboHbLSbFsF4fA8dXUnwwmXlvrFNFOPwRbAfwZLyVtAeovllbdLPVTrfMRFLAi
Vt700IAif/r1Lp7YrVTGkYhhjl7s3QQhdGJAkEbnrMaKxCXDqW7j4MSFT7sNZYeh
z1xOtDFD/a2GUNt1sywYgLdqUQKBgQDs520fKfgnsy9X/5SBT3BoMhpI6hiyD5sE
jbgCs9zfVT+PjTXSSczKNCnp70UyJ2pvAiJvhwO58PlYkmR+ojSIFYoaF5XoOJwG
c0DYHpfg9F6kNFCTEFZxc0jyhthlPS3kvtBMc9D94p8khm3nKUvq274DQbESm4ct
qBN+R/yOBwKBgQCT7h4saGoNbWS9Ob29Eqtrtpl7Mz+NaXQSB+ictzlTlKt9Wr6h
dvCTB0zTltq99TL5PjdOvgIFu8jFG9alulBQ/22EnINQTdP00w8l4M5In+fnjojc
Vfss4E89LTIOobtd7M+BDCsMGSk67DWgSbJDFMQOcH8Hr/uFdXRopOY1YQKBgQCg
nQZOapXj1If0kUKNqN38xnsrlArKLepbW/m1W1wTzigZmxMyTvaY4GTKRX4UUsTG
HhxaoJ7WIZnrgtS/Rhl3aaMHrZXkqvydIuaDn7o4uXSwvwchraHtqMt8Ig/0E1Za
D0QJkTvP5SUgDkPUxRrDT9ezPYZfpzIushUFOKjxYQKBgCCCNl4dsm361QyGPX6a
VU+pdybVsXWV/aHPUrInEZsDUnipeAkrVMN7KpaekpMGzp2J0lKMAduNiC95Nn8P
CeB7K6OTHXHId/57n7t9VJpCs/D4E8BZLs1SqzdFvsWtV6XJu5gDelHSLK+Od73E
k8NMq1bQOq3lw24bXiKHb5di
-----END PRIVATE KEY-----

- 签名结果：

- 验证结果：
  - 使用原始消息和公钥验证签名：验证成功
  - 修改消息后使用公钥验证签名：验证失败

4.4.1 小结

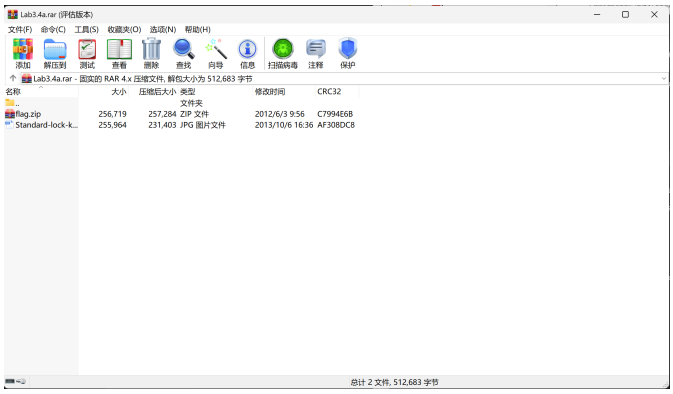
通过本实验，我们成功使用 RSA 算法生成了数字签名，并验证了签名的有效性。实验表明，数字签名能够有效地验证信息的来源和完整性。当消息被修改后，签名验证失败，这证明了数字签名在防止信息篡改方面

的有效性。数字签名结合了哈希函数和非对称加密技术，为电子通信提供了可靠的身份验证和数据完整性保护机制。

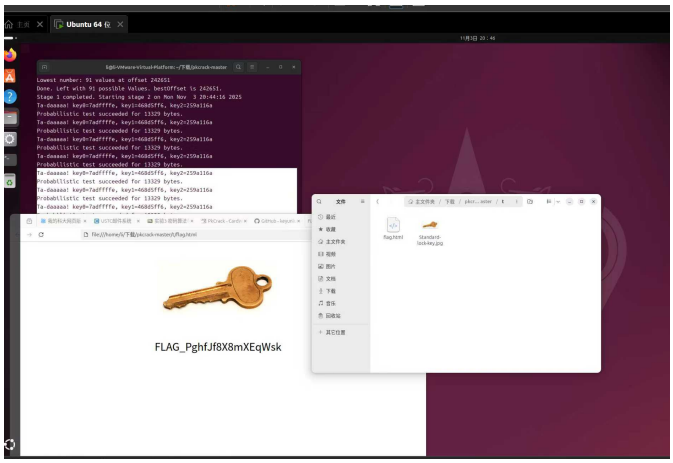
4.5 已知明文破解

压缩包分析：

- 加密压缩包：Lab3.4a.rar
- 加密算法：ZipCrypto Store
- 包含文件：



攻击结果：

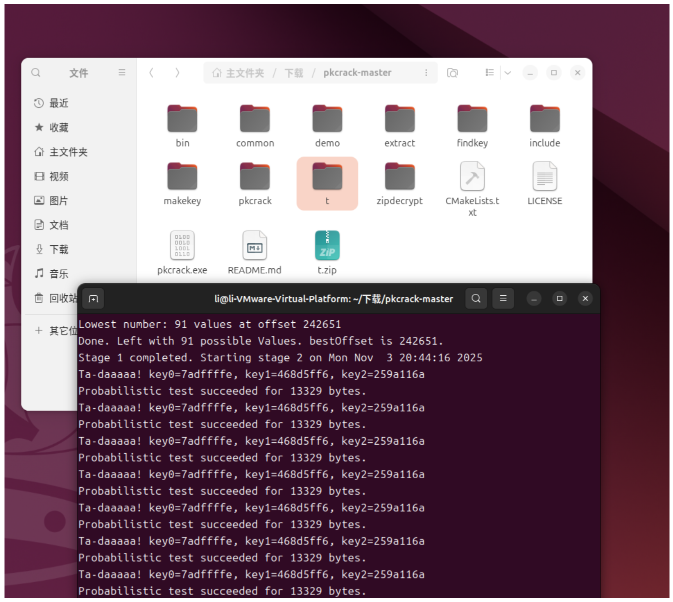


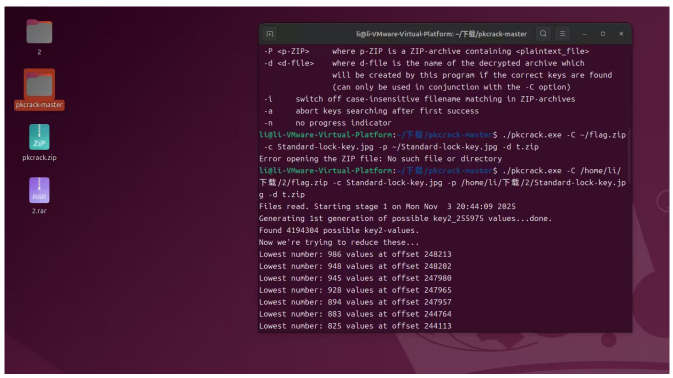
4.5.1 已知明文攻击原理分析

已知明文攻击是一种密码分析方法，攻击者拥有一些明文及其对应的密文，试图推导出加密密钥或算法。对于 ZIP 文件，传统的 ZIP 加密算法（ZipCrypto）存在设计缺陷，使其容易受到已知明文攻击。

ZIP 加密的工作原理是：1. 使用密码初始化一个伪随机数生成器 2. 生成器产生一个密钥流 3. 密钥流与明文进行 XOR 运算，生成密文

如果攻击者知道明文和对应的密文，就可以通过 XOR 运算推导出密钥流。对于 ZipCrypto 算法，一旦获取了足够的密钥流信息，就可以恢复出用于初始化伪随机数生成器的内部状态，从而解密整个压缩包。





4.5.2 实验总结

通过本实验，我成功使用已知明文攻击方法破解了 ZIP 加密压缩包。实验表明，传统的 ZIP 加密算法（ZipCrypto）存在安全漏洞，容易受到已知明文攻击。这种攻击不需要暴力破解密码，而是通过分析明文和密文之间的关系来恢复加密密钥，大大降低了破解的难度和时间。为了防范此类攻击，可以使用更安全的加密算法，如 AES-256 加密，或者使用专业的加密软件对重要文件进行保护。同时，应避免将已知内容的文件与机密文件一起加密存储在同一个压缩包中。