



南昌大学实验报告

学生姓名： 邹渠成 学 号： 8007124124 专业班级： 网安244班

实验类型： ☐ 验证 ☒ 综合 ☐ 设计 ☐ 创新 实验日期： 9.25 实验成绩：

一、实验项目名称

对称加密、非对称加密、数字签名、哈希

二、实验目的

1. 理解对称加密、非对称加密、数字签名、哈希算法
2. 掌握对称加密、非对称加密、数字签名、哈希的应用场景。

三、实验任务

- 1、对称加密：自己用记事本或其他文件，写上一段需要加密的消息,如“明日凌晨3点发起总攻”，通过对称加密算法，对该消息进行加密，并通过其他方式告知收信人（您学号对应的下一位同学）密钥，收信人收到消息后进行解密。（传输方式可以通过email进行）。
- 2、非对称加密：你通过类似PGP相关工具，生成非对称密钥环，并将自己的公钥公布到学习讨论区。
 - （1）试图采用使用收件人公钥，对收件人发送密文，待收件人使用私钥进行解密，并且核对实验结果。
 - （2）采用数字签名方式，给收件人发送密文，收件人收到密文后进行验证，得出验证结果。（该过程为发件人通过自己的私钥对需要发送的明文进行加密，而后将密文进行发送，对方收到密文后，进行签名验证）
- 3、对密文或者某文件进行哈希值计算，对比修改密文或文件内容后，观察实验结果。
- 4、自己探索这几种加密方式的组合应用。

实验参考工具：

[在线AES加密解密工具_蛙蛙工具](#)

四、主要仪器设备及耗材

计算机mac环境、无耗材。

五、实验步骤

1. 实验一：

对称加密：自己用记事本或其他文件，写上一段需要加密的消息,如“明日凌晨3点发起总攻”，通过对称加密算法，对该消息进行加密，并通过其他方式告知收信人（您学号对应的下一位同学）密钥，收信人收到消息后进行解密。（传输方式可以通过email进行）。

首先我在mac的文本编辑里面写了

```
hello who are you
```


通过AES加密我得到了

密文：EPgXe6e25EPA0CTfsO0QI9KCuHPeD6DfNVVZUM4A/MA=

密钥：caccf74588a87aad0da008f7064bff9048f234dbfe9242b0819e131e1729df2e

偏移量：8967ee259a50816136891d74786f8a36

我发给她之后顺利解出来了



青蛙工具

便捷的在线工具网站

[首页](#)[文本工具](#)[语言工具](#)[财务工具](#)[日期时间](#)[换算工具](#)[图像工具](#)[便民查询](#)[开发工具](#)[编码解码](#)

[Base64编码解码工具](#)[MD5加密工具](#)[AES加密解密](#)[SHA1在线加密工具](#)[Ascii与Native编码转换](#)[全角半角转换工具](#)[URL解码编码](#)[摩斯电码编码解码](#)[随机密码生成工具](#)[随机数生成器](#)[GUID生成工具](#)[网络工具](#)

AES加密解密

点赞 (296)收藏

EPgXe6e25EPAOCTfs00Ql9KCuHPeD6DfNVZUM4A/MA=

加密解密复制清空

hello who are you

JSON格式化Unicode转中文复制

选项设置

加密模式: CBC 默认

填充方式: PKcs7 默认

密 钥: caccf74588a87aad0da008f7064bff9048Hex随机168一键生成

偏移量IV: Hex随机168

亚马逊云科技

注册亚马逊云服务免费套餐 可享高达200美元抵扣金 包括免费使用精选服务

产品免费体验

相关知识

AES (Advanced Encryption Standard) 是一种对称加密算法，也是当前最流行的加密算法之一，由美国国家标准和技术研究所 (NIST) 标准化，已经成为了国际标准。它的加密密钥长度可以为 128 位、192 位或 256 位。其中 128 位密钥版本最为流行。AES 是一种分组密码，将明文分成 128 位一组，然后分别进行加密，加密方式包括替换、置换、线性变换等基本操作。通过多轮迭代加密，在满足密钥安全性的前提下，能够提供很高的加密强度，以防止恶意攻击者的攻击。在许多场景下，AES 已经被广泛应用，例如数据传输、文件加密、数据库加密等等。

AES 加密模式对比

模式	是否需要 IV	特点	安全性	推荐程度
ECB (Electronic Codebook)	✗	最简单，每个分组独立加密	✗ 极低 (同块明文→同块密文)	🚫 不推荐
CBC (Cipher Block Chaining)	✓	每个分组依赖上个分组，常用	✓ 高	★★★★★ 推荐
CTR (Counter)	✓	将 AES 变成流加密，支持并行	✓ 高 (若计数器不重复)	★★★★★ 推荐
CFB (Cipher Feedback)	✓	类似流加密，能处理小于分组的数据	✓ 较高	★★★ 一般
OFB (Output Feedback)	✓	与 CFB 类似，易受同步攻击	⚠ 较低	★ 一般，不推荐

AES 填充方式对比

填充方式	说明	优点	缺点	推荐程度
PKCS#7	用缺少字节数 N 填充 N 个字节	✓ 通用，几乎所有库默认	✗ 占用额外字节	★★★★★ 推荐
ZeroPadding	用 0x00 填充	✓ 简单	✗ 末尾有 0x00 时可能解密不准	★★ 特殊场景
NoPadding	不填充，需手动保证数据长度是 16 的倍数	✓ 无额外字节	✗ 使用受限	★ 仅限固定长度

关于 · 赞赏 · 反馈 · 会员 · 搞个摊 · 文章

Copyright © 2025 青蛙工具 - 网用的技术分享 版权所有 粤ICP备18006158号

2. 实验二：
非对称加密

(1) 试图采用使用收件人公钥，对收件人发送密文，待收件人使用私钥进行解密，并且核对实验结果。

实验思路：

我拿到了同学的公钥

-----BEGIN PGP PUBLIC KEY BLOCK-----

xsBNBGjUyEMBCACgTcwxcy8Zk8pi1Rz21gq+nuFZV+wTPmJFJ1/j9Z8XQdrjJDJf
97gg9DRwH200m/SfjWbjx4EJZSeka9sF5A6E/SwrKlVqHs0a12zFPzmY/xSrQrHA
gmKYjMvAJIKpY6ynIdZYB1arI1epsD/DsynDo8c0vEGG4ibx00Fb071nDhyBRUt2
4Ibbaig83RoDvejZX6Rrx2L6ZPF7ZV0ysut7cW5jTXeIvB1eY+RN8rmZ3m7RuWRM
pxa3ku8bUHyPp6Td2eGmB9iIUlhMkhatx1h4Jjmj4ISGCgVAUIEFsSh6lJbitYsB
Fm6AdDX0hQA7N97DLQTNgybSMGHb17V1B9pdABEBAAHNHW5vYm9keSA8bm9ib2R5
QGVtYwlsLmFkZJHlc3M+wsB6BBMBCgAkBQJo1MhDAhsvAwsJBwMVCggCHgECF4AD
FgIBAhkBBQKAAAAAAAJEGUuVlcjK4V5txQH/jwShvJl7VcW5KgKnYQfpBsDSiiZ
38U7LDAwd9FWo7vvhSxDpdlZ5hu8Wo9n7C276njruAFcTXA6YD0bY29ZXxfEFVv0
/XasR6Y1oGioUJ9gtejZByqplZt07kxIWk+9Bj9wqV5MxswdQqUREkm04xnp6hwS
s0vqr0S2x4lC6vUDnFmL8uYZaexktV8WesFGHkrL04Ad/w32R9cB8S7WVMLWuIR2
2LR/OKImnr+R2aV+MN4cPePou0bBwW/0ZxJuo0QBENUC2CN1HlooWjqsI530q5k8
pX2BFeIS/yapuUnPvimjYNU7BsbCsXinF0manojce+Z9JVvbpBwUtM+l4v0wE0E
aNTIQwEIANyzTwfJ5SBr18QK2YGWdfRCaZTDHwC/nXxwy77W+h2N2KF2Do1P0ogq
uyl3fyCPXxYd8I165s3haztMQUIL02vQDRVXMC0Wr03l1PPTUhnX79aTLYIvcnNF
svnRaF324owl4Aw2oYtGIFnd5mx99rx0qEYprpcXUm8to86T4o07IWI5AZAYF7T4
yKbx1MdAl929/izNkRX2f8ZrHxeZWH8i/GyX/CCvFCEtRU16xc58z5PSSe1z2KZs
STR9HCTRNNa3rgGRJFTcfjmw55gG2Bcl0QuBJlavJbEXH4c+au9lDoDJvle0iWM8
mLwrxC3lEqsq/9F8byyimxtG9IPNGGkAEQEAAcLBhAQYAQoADwUCaNTIQwUJAAAA
AAIbLgEpCRBlLlZXIyuFecBdIAQZAQoABgUCaNTIQwAKCRAKn0UZ96WxiLtlCADQ
4QSCaJxmgoV01vc5Peeehscq9nXVPRjXvLxXANwwdvWuYuEaZfV4a3aEIfklQwuH
PeHJ0LiW6/5ZCx5tYIURwbFJiFeeH/KGxvg93qpj/3A/InQ9JkSQfrg9acyxe7yQ
A35uHC2VTWvqNDRWus1ca9S/qNoyfwugStZP/xB6GPD0QvsFffH9yi++yzf+z0zG
QhWaPX60tmXMMMrayJLUHCGFKmd08sHk4ZEd4ghpp3FU1uK28oZ+oUqPMXSJGMpA
IgGhi6IigJj2U04Dxkyep6RH7l9qzsfx9zxT4AwgZ1QsamCiTiVsS3hGiubUZ5ff
2U8/IdrJ5U/8s0Xkn+iMYg4H/AqpWmWRBzMUH+nP2GsN0d1a0CDBGy0n34Aao5Q8
P2N8wH53HCLBFxpQvkd0Ki60l9ynsYSMH/5b+ndj1PXfH0SAzL+AHGyixBTh+57Q
an0qswXXSbH4IxWcZNYsXhr+ARI12NQn8vumR+hjNyJGpDFU4ypVDgNY++oqfimG
70SINvnHs040dVxsFuKplWhw5br86BMD84XkauKzDsZqgzBpaVxShgLbQtWkM9X
LV3WIpNhZiYqs8AafP2Gk0CKIZfWtCK6zi8Czf73FpIWWXHLs8iysxFCsgerJJ7
q98HHT51B4E76amH21AF54hjrdjD3WBG5prvZyqB6w58CI70wE0EaNTIQwEIAPe2
86vgDr7vFm0F4UCUi52mFGDeuesVEmvZ33n1K8aHSiKHx6TKwgpaxz5AqLcJcwze
F8I8sLvnZ3uC37dHFCVy1/HNG0CGlz2svtXNM8LdrJ4ooSJTN1ccs8Ld2SH1hAD+
jSCxT+TpbLDjmT0Lv44y6q0v8kkYXVSKzbzFd0C1hzcK47xJkplqBJnjK9axE5k
hMTtYkgFef5EZ07lTsTiffN8Yne071L9zAa1eTibHylHEoa+T1E1mzPdBZmt1/zs
g4GtvaBqAwbawLpIB9da9N7CJ5+mb+yiC31USBqNCD2xpEQXvcG8AahRvZSK+znT
ZwLmp5qWQC+T/ZvBj5EAEQEAAcLBhAQYAQoADwUCaNTIQwUJAAAAAAIbLgEpCRBl
LlZXIyuFecBdIAQZAQoABgUCaNTIQwAKCRCuIlRKt04enjXDCACG45Ich/ScsmDH
DhlebtHdaJ9ln3VkmPpzW0PhejrkC/5IXGoE93Lsw/J0puoyk5kzfidG70IAeAwu
dh4bWgflar/dg7meY4aP+kk4Lxqo1Bl5zwvNbHCJCHdiEATkYWX4mrmZ96lCiJBk
eq6DJX7UY9Z6RIX2lPumQIoXfrz2E9/GWuCbqzBNLHLegL90ZR2ZQagC1J8vLee8
89WV0GuRkaHw4+/SV1Uh6jMfFcPHEJAeDHk7twNMiBvymS1SVd3vSQah+hKF3h1
3s4MYz0WKXrbukKzpXwA8zsI5X4cV818MHwCTM28Wf/DGcLrHBm05hwK8sLRR7W2
eBLerX0hYt8H/3261ZMiSwwWBXU0owTEIUUv4c2luqfhKW1dbSYHZl7rnq0DH8LI

```
ZBQZynWq7i5wSMgN4zai1kPcBZ0lwootMHor4/T48eGnBEEEOjh4dPppGnU84Woi
Ho2A/0ffYBg6Zv8Ld2j10hsqDfiG0340hfoVtqbcDhw327CQi7ByP0i9v/1TALVi
xM5kUwY/1Pw2XnjNxA9QykTRSRsXiTQX6l5oWRLExndgz9LNVy3/XBVEsNkfby3h
+4Lm8+UxyElknzapoPyQHqDjKvZrFQ0VX3aQTKnKBXXfgfACjEfAuATVrpTvCOLr
CnZfTrCheBwfsVsjskUlVvqLt9x7+jgV8Q=
=wrfl
-----END PGP PUBLIC KEY BLOCK-----
```

然后加密了一条信息

收信人的私钥 (Private Key)

```
1TQX6i5oWRLExndgz9LNVy3/XBVEsNkfbY3n+4Lm8+UxyE
lknzapoPyQHqDjKvZr
FQOVX3aQTKnKBXXfgfACjEfAuATVrpTvCOLrCnZfTrCheB
wfsVsjskUlVvqLt9x
7+jgV8Q=
=DMG5
-----END PGP PRIVATE KEY BLOCK-----
```

已加密的信息

```
-----BEGIN PGP MESSAGE-----

wcBMA64iVEq07h6eAQgAjX3AF1jNtZu3GFD3VCLBv/14fNcHx1r/5AMAEWk8SvTn
bPDYNxDr4hJASdk9YxLGXyuybal9nPRxH2k1RY8N8xKeSQ2Z5lz+zLbzUpwaX06q
PFKc9WmP42Dsol5Djac3Zxd9kwnw45NJQFOUWcEaGeBK2MEkesEbaWXKU00TLvdE
gQyQAB+AAcw1xEv1N3k5Yp1pw1MFshKkTeaGMzqFNw5XKF7v3SSCcV4/JD5ouStO
1J5AtTEBNYg+vzfPMX8sn9ADz9q8pEq2qzGbHRelDPGHqGG5yQd0bXxf0s9bqf+z
```

解密信息

解密后的文本信息

hello

她用她的私钥解出了

收信人的私钥 (Private Key)

```
1TQX6i5oWRLExndgz9LNVy3/XBVEsNkfbY3n+4Lm8+UxyE
lknzapoPyQHqDjKvZr
FQOVX3aQTKnKBXXfgfACjEfAuATVrpTvCOLrCnZfTrCheB
wfsVsjskUlVvqLt9x
7+jgV8Q=
=DMG5
-----END PGP PRIVATE KEY BLOCK-----
```

已加密的信息

```
-----BEGIN PGP MESSAGE-----

wcBMA64iVEq07h6eAQgAjX3AF1jNtZu3GFD3VCLBv/14fNcHx1r/5AMAEWk8SvTn
bPDYNxDr4hJASdk9YxLGXyuybal9nPRxH2k1RY8N8xKeSQ2Z5lz+zLbzUpwaX06q
PFKc9WmP42Dsol5Djac3Zxd9kwnw45NJQFOUWcEaGeBK2MEkesEbaWXKU00TLvdE
gQyQAB+AAcw1xEv1N3k5Yp1pw1MFshKkTeaGMzqFNw5XKF7v3SSCcV4/JD5ouStO
1J5AtTEBNYg+vzfPMX8sn9ADz9q8pEq2qzGbHRelDPGHqGG5yQd0bXxf0s9bqf+z
```

解密信息

解密后的文本信息

hello

(2) 采用数字签名方式，给收件人发送密文，收件人收到密文后进行验证，得出验证结果。(该过程为发件人通过自己的私钥对需要发送的明文进行加密，而后将密文进行发送，对方收到密文

后，进行签名验证)

实验思路：

我拿到了他的公钥：

-----BEGIN PGP MESSAGE-----

yMCKAnicAT4Bwf7EDQMACgHwoanoCg8UPAHLdNUAaNT08WhlbGxvend5wsBcBAAB
CgAGBQJo1M7xAAoJENahqegKDxQ8AckH/idTd9w/rir7hBUIRCXbpZzdaKI18xYB
c4suz6sUJTlaYx/AP/bYJWe9NdmBj7Sc2nLPtzELoB9aLV/hAnnI1rYDCeRo1rLl
yTE2kskA6fjAAweMasjL4nyDipsVMMpm/DWt4uQ7n5q0vj30qg4/d9uuyseCWLRG
IKfgPdvwXGv592Cti6KCW2j8AVFuoEHNUbWW4vGBLKxk274Gwyb7PMR0lo+2CXbC
GfPih7jGl1LThBHxT15KTcx+8LLzp7yFIEg1UL7FbCuQx4s1EEKoQDTcXoaHqRZ
vsiAx6ELdhcl5H5r/uzTDXfDPDixwu0h+vLEHfHnXn3+yfL+zT7Q9Ftp/J2d
=vGbF

-----END PGP MESSAGE-----

然后我拿到了她用私钥加密的PGP签名：

-----BEGIN PGP MESSAGE-----

wcBMA64iVEq07h6eAQgAjX3AF1jNtZu3GFD3VCLBv/14fNcHx1r/5AMAEWk8SvTn
bPDYNxDr4hJASdk9YxLGXyuybal9nPRxH2k1RY8N8xKeSQ2Z5Iz+zLbzUpwaX06q
PFKc9WmP42Dsol5Djac3Zxd9kwnw45NJQF0UWcEaGeBK2MEkesEbaWXXU00TLvdE
gQyQAB+AAcw1xEv1N3k5Yp1pw1MFshKkTeaGMzqFNw5XKF7v3SSCcV4/JD5ouSt0
1J5AtTEBNYg+vzfPMX8sn9ADz9q8pEq2qzGbHReIDPGHqGG5yQd0bXxf0s9bqf+z
/QPZujxkZdbDqI0qP/dYxd8fhxrSCVLX6K/E7QQYfNJBWmroc4pRoRkiAx0K110
MxJPYZcgmVMsGQcv+GpBPadXizKyMGxKwSWg0JRNx0ukenbMryEL2f5VAp06+UcI
+xs=
=ieu1

-----END PGP MESSAGE-----

原始的讯息和状态

已成功验证对信息的签名。公钥指纹为： b235def660e23e4c66aa806d390ed10276b82fbd ✕

hellozwy

下载讯息文本

下载为二进制文件

非对称加密（又称公钥加密）使用一对数学相关的密钥：**公钥（Public Key）** 和 **私钥（Private Key）**。

公钥可公开分发，用于加密或验证签名；

私钥必须严格保密，用于解密或生成签名。

其实简单来说就是完成以下两个过程：

（1）公钥加密与私钥解密（保密性验证）

发送方获取收件人的公钥，使用该公钥对明文消息进行加密，生成密文并发送。

由于只有收件人持有对应的私钥，因此只有收件人能够成功解密密文，还原原始明文。

（2）数字签名与验证（完整性与身份认证）

发送方使用自己的私钥对明文的哈希值进行加密，生成**数字签名**，并将明文（或密文）与签名一同发送。收件人使用发送方的公钥对签名进行解密，得到原始哈希值，并与自己计算的明文哈希值比对。

注：数字签名并非直接“用私钥加密明文”，而是对明文的摘要（哈希值）进行加密，以提高效率与安全性。

3. 实验三：对密文或者某文件进行哈希值计算，对比修改密文或文件内容后，观察实验结果。

本来如果文件里面是hello

hash——output是

2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

改成hell

hash——output是

0ebdc3317b75839f643387d783535adc360ca01f33c75f7c1e7373adcd675c0b

可以明显看出差异

4. 实验四：组合搭配

我采用了混合加密的方法

用**对称密钥**（如 AES）加密明文，生成密文（效率高，适合大数据）；

用**收件人公钥**加密该对称密钥；

将“加密后的密文 + 加密后的对称密钥”一并发送。

六、实验数据及处理结果

见详细的实验步骤（均解出实验答案）

七、思考讨论题或体会或对改进实验的建议

通过本次实验，我深入理解了对称加密、非对称加密、哈希函数及数字签名在信息安全中的核心作用。对称加密效率高但密钥分发困难；非对称加密解决了密钥交换问题，但计算开销大；哈希函数虽不加密，却是保障数据完整性的基石。三者单独使用各有局限，而**组合应用**（如 PGP 的“加密+签名”机制）才能同时实现**机密性、完整性、身份认证和不可否认性**，这正是现代安全通信（如 HTTPS、电子邮件加密）的设计思想