计算机系统安全实验二

Julius Karl

2024 年 6 月 26 日

1 实验步骤：

1.1 软件下载

Ubantu 默认下载 Gunpg

1.2 创建密钥

```
juli@DESKTOP-SQ9JMQ4:~$gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
Requested keysize is 3072 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: juliuskarl
Email address: juliuscar2333@outlook.com
Comment:
You selected this USER-ID:
    "juliuskarl <juliuscar2333@outlook.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key A9A7700D2D558DCA marked as ultimately trusted
gpg: revocation certificate stored as '/home/juli/.gnupg/openpgp-revocs.d/5E9F4E81AAC2F9E047846BA8A9A7700D2D558DCA.rev'
public and secret key created and signed.

pub   rsa3072 2024-03-24 [SC]
      5E9F4E81AAC2F9E047846BA8A9A7700D2D558DCA
uid                      juliuskarl <juliuscar2333@outlook.com>
sub   rsa3072 2024-03-24 [E]
```

创建完成！

gpg -ao public-key.txt –export linus 导出公钥

gpg -ao secret-key –export-secret-key 99F583599B7E31F1! 导出主私钥

gpg -ao sign-subkey –export-secret-subkeys FDB960B857D397F6! 导出有 [S] 标识、签名用子私钥

gpg -ao encrypt-subkey –export-secret-subkeys 6FE9C71CFED44076! 导出有 [E] 标识、加密用子私钥

1.3 加密文件

1.4 解密文件

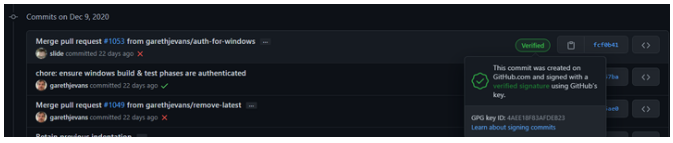
gpg –decrypt encrypt.txt –output decrypt.txt 新建解密后的文件

gpg -d encrypt.txt 输出到终端直接查看

2 实验总结:

PGP 使用对称密钥加密算法保护数据机密性，使用公钥加密算法保护对称密钥的安全性，使用数字签名技术验证消息的完整性和身份。这种结合了对称密钥和公钥加密的方法，可以在安全性和效率之间取得平衡。PGP 已经成为一种被广泛应用的数据加密和数字签名的标准，保护了用户的隐私和安全。

PGP 能保证一条信息是你相信的人发的，除了你俩之外别人无法解密，而且这条消息在传送时中间没有经过任何哪怕是一个标点一个字节的修改。可以用密钥来验证我的 Git committer 身份，并且增加我的安全



联系方式。

可以用来代替 Openssh 进行 ssh 操作。