MIT World Peace University

Information and Cybersecurity
Second Year B. Tech, Semester 1

Email Security using - PGP

Lab Assignment 8

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

April 30, 2023

Contents

1 Aim

Demonstrate Email Security using - PGP or S/MIME for Confidentiality, Authenticity and Integrity.

2 Objectives

To learn authentication technique for access control

3 Theory

3.1 PGP

1. PGP (Pretty Good Privacy) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications.
2. Phil Zimmermann developed PGP in 1991. PGP and similar software follow the OpenPGP standard (RFC 4880) for encrypting and decrypting data.
3. PGP encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and, finally, public-key cryptography; each step uses one of several supported algorithms. Each public key is bound to a user name and/or an e-mail address. The first version of this system was generally known as a web of trust to contrast with the X.509 system, which uses a hierarchical approach based on certificate authority and which was added to PGP implementations later. Current versions of PGP encryption include both options through an automated key management server.
4. PGP encryption should only be used with data that is transferred via file transfer applications that use secure connections. PGP should not be used with email applications that send and receive data in plain text. PGP encryption is not compatible wi

3.2 Steps to Send an EMail using PGP

3.2.1 Generate a key Pair

The first step is to generate a key pair consisting of a private key and a public key. The private key is kept secret and is used for decrypting messages that are encrypted using the corresponding public key. The public key is shared with others so they can encrypt messages that only the owner of the private key can decrypt.

3.2.2 Share public keys

In order to exchange encrypted messages, each person needs to share their public key with the other person. This can be done by sharing the key through a key server, sending the key as an email attachment, or sharing it in person.

3.2.3 Encrypt the message

Once the public keys have been exchanged, the sender can encrypt the message using the recipient's public key. The encrypted message can only be decrypted by the recipient using their private key.

3.2.4 Sign the message

The sender can optionally sign the message using their private key. This adds a digital signature to the message, which provides a way for the recipient to verify that the message was actually sent by the claimed sender, and that it has not been altered in transit.

3.2.5 Send the message

The encrypted and optionally signed message can now be sent to the recipient through email or another messaging service.

3.2.6 Decrypt the message

Upon receiving the message, the recipient can decrypt it using their private key. If the message was also signed, the recipient can verify the digital signature using the sender's public key.

These steps are shown below in Input and outputs.

4 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code

5 Input and Output

5.1 Generated Keys

```
-----BEGIN PGP PRIVATE KEY BLOCK-----

lQIGBGQ3mhsBBAC00HioVaOXhVbOpo7Gw9/LlM4K9LvmiOdjRaQdJs8kTagJGJkP
yQk2Dpt0oxN3FY4/HeutrKnn/EYDnuTmglXvo7Vbessfbn9h6NQ9ZyCJmln/SAQT
Vz14PtaXymiii+puG7QC0rIYA4acNc2kWiGoC0kyqtljAL0+5K8qxrGKhwARAQAB
/gcDAnqziXIUqkxp+emZF2o8uw7JRWCyLUedfNztqU+Jy8XcQJeSEhqobqM3p8X6
zOlJ9QHm6lr6BduvzulxBnkjuqPyCOUmJz2C7368uq2mNgYwb7w9tfYhI6O8wWVs
LCBa/f0mLVhAqCBhj26HNYkPYkt1PcKgYU/Qx2R0bDhMavNMwWxmE9IAkySQpQjb
mOtcYva53lj3jnq2LmWinvvD88P2uHvd4fXGmomqIzeRK2rADIcY4z8s4o72BLVL
zCGITg4YLChjJttxtUlJSobApla//ckJFNdnuJbsjNColrN+J8XYPMsI/C/aTwxR
X21wJcwf7tVn1Ps9BXY3IGZurrbGbwg9I8V/091hb5od3J5IP3CJrs0i8CTwu7/l
FaKWTDSl3aIpFbqdeClCReJF4fMBbe1WwECdAgYEZDeaGwEEALeH73HJ8lyR6pN1
I7pT+FNgutYL49S5OXPyPh4jcQ3EqV4Sm96Xy6YBpFEsHvokWHSBrmiMe3OEGKdY
go8g0tKBMG1sByiPSmE/ktLChWae2EV/taWkynhNwisrAonLrYCVo/FRR/PIBaT8
qWC8P6w4XIRUpKIi2QEhd+1sIGvbYUKUbkeadnDunDgyg/7gb/H2OTXaL6XvMyzT
QjWlWBy+GMuL1fSCB3Kh1cRVHVsjdjj1V2c1zx0lilToO8PNV2X6NvDNI/6yfuRX
NeNUq//u3WJd1Ayq/2SE91b0M0LLKg5/42ZUoQetc6VEIcjL7c2/XQ9rH6vgy/BP
6A378Y1ZZo3Z0zvCyVh5NMhgKYi2BBgBCAAgFiEEseSJ53V8cfAaI9i3JiEHouMD
/UIFAmQ3mhsCGwwACgkQJiEHouMD/UImEgP8D/hMV0JBlSWERGuGqthbsslrWtKB
uFnGobSJuHCjRtUlzsRLcKixulMXyJiuTMtqpjEl1i2BqhpZqz6IDojFn8Q0KsWw
g9Hw/cFGwTPNPgujchRWZFkOqu+BBzzgl/bS1+EpbkAD7oxkJZxZI7yjQGNTrJJ5
3rJ6wrgXJLQyuJs=
=hg+7
-----END PGP PRIVATE KEY BLOCK-----
```

Listing 1: "Krish Private Key"

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

mI0EZDeoGgEEANBxkECodUnki0ouPXtisjdTOb6Z6ASGjaaYltwlNp1PjuIE1E1E
/U+FXB1yPnzQXYJNzZinyKALznNHPA5u1q3kfkxHx0Bsz2Jr79Ly8VcFOXi621c0
Lw50mFDBYCotKHqrMQa4V4ovoG1toJn2RDEbYYo5zpj/cVIS4R9M+3qFABEBAAG0
NktyaXNobmFyYWogKGljcyBhc3NpZ25tZW50KSA8a3B0LmtyaXNobmFyYWpAZ21h
aWwuY29tPojOBBMBCAA4FiEE00HVO8avaij3dBuGrN2zFzRsyHgFAmQ3qBoCGwMF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AACgkQrN2zFzRsyHgE0AQAg+kQmV7DKOX1
gOzyW0oDcbyLhnTodytDT2RZnNi96d+cFjRfXDB3ET26gBKVzn7b8QG5JO7jSZw2
6noliMFlbM7JFJDP881Dr5SeSqwQZnB2Mozc0gqrlFUYmlShppoJ0wPY5y2ME8U8
g2u1rx3EUsu4GxyQUguyO7S+u4DDLO24jQRkN6gaAQQArO87+003qyBMi88xSx9u
ktouHOso25kAR3tl3tjB2hCstRZoQgGJCbKnt6k0fhRmEqYFwgUKULYvODs+GnWM
ah/E2shSlcSPpOdAjovHTSncOUJmU6qZDLUk78j7NMiFwf3M1vLU2KkBg6Jqw5QT
rpyXDIYMV2RQ3cO/nS7bMd8AEQEAAYi2BBgBCAAgFiEE00HVO8avaij3dBuGrN2z
FzRsyHgFAmQ3qBoCGwwACgkQrN2zFzRsyHgE4AP/WNEAzJuQLh/eyPNW/CzOwkQK
REeu1xj0pFGdXfPytoe1GT+xZ6oDd0WHPzMb3oa7NDC4DbzbRw7AfLJj2Qo4PdYG
vmEHmlON7NY4Wv5W73bxiR8HJGThL6SS4TJXF/etW/Jxs/LYaEnifa77quKeX0Sh
hI+UmhljipkecYXakA4=
=Qz5/
-----END PGP PUBLIC KEY BLOCK-----
```

Listing 2: "Krish Public Key"

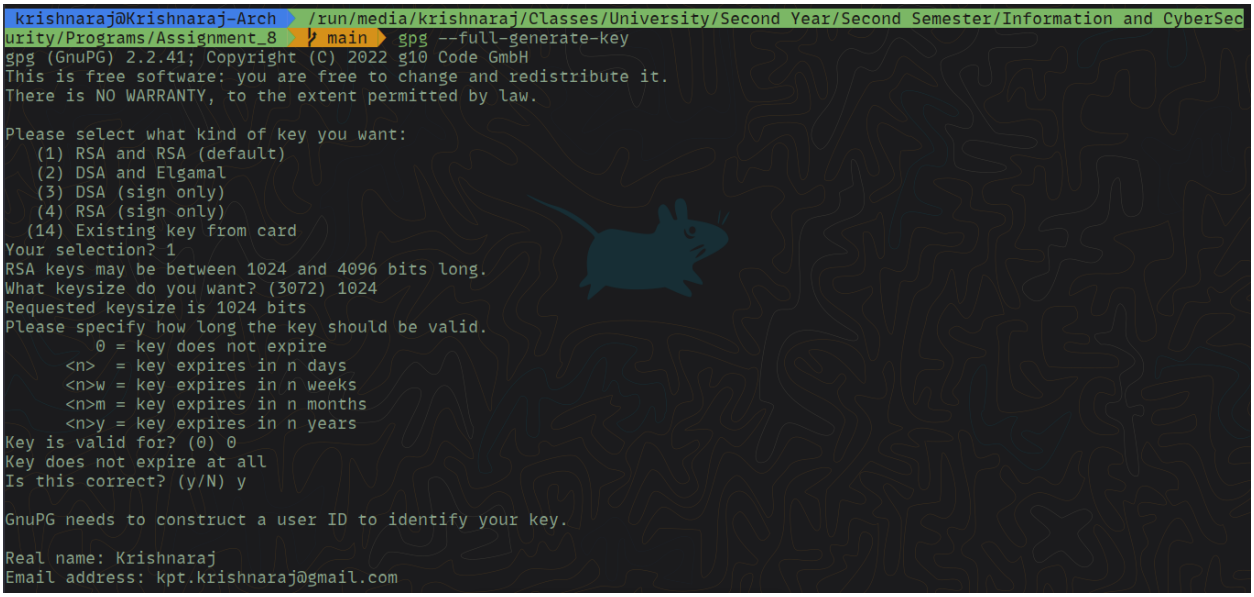

Figure 1: Generating Key Pairs

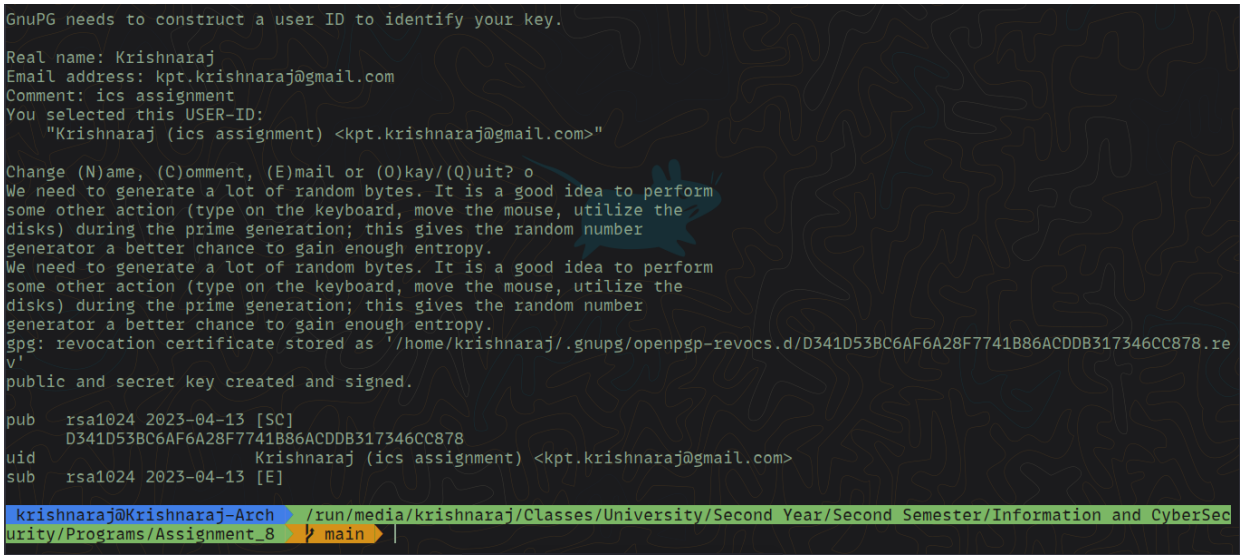

Figure 2: Generating Key Pairs continued

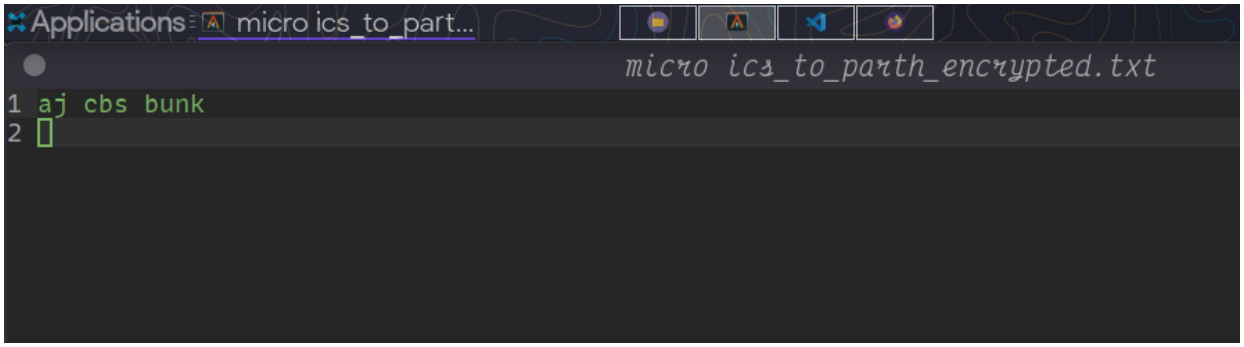

Figure 3: Secret Message to Parth - Recipient - "Aj CBS Bunk"

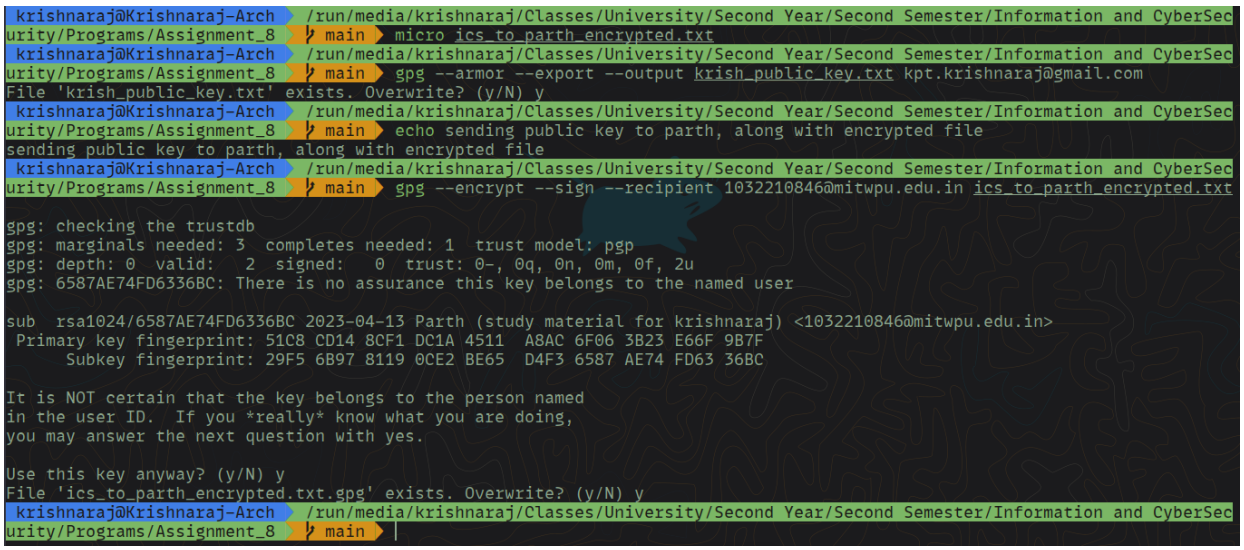

Figure 4: Signing Message to Send Parth using Parth's Public Key

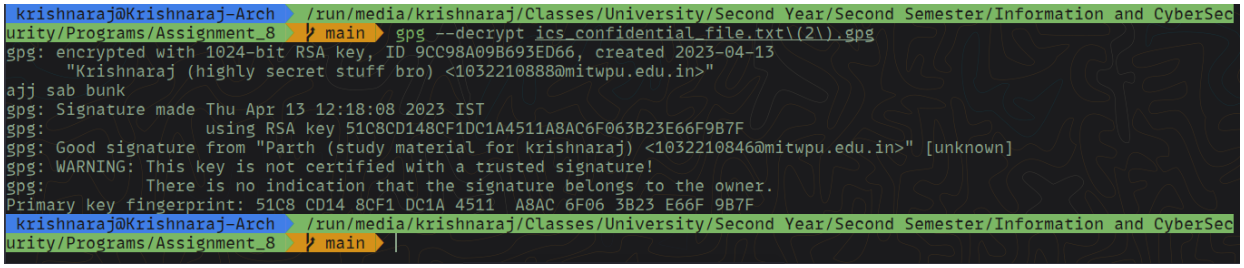

Figure 5: Decrypting Message from Parth using his Public key - "Ajj Sab Bunk"

6 Conclusion

Thus, we have successfully implemented Email Security using - PGP or S/MIME for Confidentiality, Authenticity and Integrity.

7 FAQ

1. **How email security is provided through PGP?**

   (a) PGP (Pretty Good Privacy) provides email security through a combination of encryption, digital signatures, and compression. When a user sends an email using PGP, the message is encrypted using a symmetric key algorithm.

   (b) The symmetric key is then encrypted using the recipient's public key, which is obtained from a key server or a public key directory. The encrypted message and the encrypted symmetric key are then sent to the recipient, who can decrypt the message using their private key.

   (c) PGP also allows users to sign their emails digitally using their private key. The digital signature provides a way for the recipient to verify that the email was actually sent by the claimed sender, and that it has not been altered in transit.

   (d) In addition, PGP can compress the message before encryption, which can reduce the size of the message and make it easier to send over a slow or unreliable connection.

2. **What type of encryption is PGP?**

   PGP uses a combination of symmetric and asymmetric encryption. The symmetric encryption algorithm is used to encrypt the message itself, while the asymmetric encryption algorithm is used to encrypt the symmetric key.

   The symmetric encryption algorithm used in PGP is typically AES (Advanced Encryption Standard), which is a widely used and highly secure algorithm. The asymmetric encryption algorithm used in PGP is typically RSA (Rivest–Shamir–Adleman), which is also widely used and highly secure.

3. **What is the key size allowed in PGP**

   PGP supports a wide range of key sizes, from 512 bits to 4096 bits. The key size determines the level of security provided by the encryption algorithm.

   In general, larger key sizes provide stronger security, but they also require more processing power to encrypt and decrypt the data. For most purposes, a key size of 2048 bits is considered to be sufficient, but some applications may require larger key sizes for enhanced security.