Somaiya Vidyavihar University
K. J. Somaiya College of Engineering
Department of Computer Engineering

Batch:__C2______ Roll No.:16010121110

Experiment No. 6

Title: Email security using PGP implementation (Pretty Good Privacy).

Objective: Email security using PGP implementation (Pretty Good Privacy).

Expected Outcome of Experiment:

CO	Outcome
2	Apply various cryptographic algorithms for software security

Books/ Journals/ Websites referred:
https://www.youtube.com/watch?v=xtiWwvHL7p0
https://www.youtube.com/watch?v=_GxpeZa-uZ8

Abstract:-

PGP stands for Pretty Good Privacy. It's a data encryption and decryption program that provides cryptographic privacy and authentication for data communication. PGP is often used for securing email communications, but it can also be used to secure files, directories, and whole disk partitions.

1. Key Generation: PGP uses a public key system, meaning each user has a pair of keys: a public key and a private key. The public key can be shared with anyone, while the private key is kept secret. The keys are generated using a complex algorithm.

2. Encryption: To send an encrypted message, the sender uses the recipient's public key to encrypt the message. Once encrypted, the message can only be decrypted by the recipient's private key.

3. Decryption: The recipient uses their private key to decrypt the message. Since the private key is kept secret, only the recipient can decrypt the message.

4. Digital Signatures: PGP also supports digital signatures, which allow the sender to sign a message using their private key. The recipient can then verify the signature using the sender's public key, ensuring that the message has not been altered and indeed comes from the claimed sender.

PGP is widely used for secure communication, particularly in situations where privacy and authenticity are crucial, such as in government communications, business transactions, and personal messaging.

Related Theory: -

Key Generation:

Public Key: A user generates a public/private key pair. The public key is meant to be shared with others and can be freely distributed.

Private Key: The private key is kept secret and is used for decrypting messages that were encrypted with the corresponding public key.

Encryption:

When a sender wants to send an encrypted message to a recipient, they obtain the recipient's public key.

The sender then uses the public key to encrypt the message. This process is done using a symmetric encryption algorithm (such as AES) for the actual message data, and an asymmetric encryption algorithm (such as RSA) for encrypting the symmetric key used for the message data.

Once encrypted, the message can only be decrypted by the recipient's private key.

Decryption:

The recipient uses their private key to decrypt the message. This involves first decrypting the symmetric key used for the message data, and then using that key to decrypt the actual message.

Digital Signatures:

PGP also supports digital signatures, which provide authentication and integrity checking for messages.

To create a digital signature, the sender uses their private key to generate a unique hash of the message. This hash is then encrypted with the sender's private key and attached to the message.

The recipient can verify the signature using the sender's public key. They decrypt the attached hash using the sender's public key and compare it to a freshly generated hash of the received message. If the two hashes match, the signature is valid, indicating that the message has not been altered since it was signed and that it indeed came from the claimed sender.

PGP's strength lies in its use of both symmetric and asymmetric encryption, as well as its support for digital signatures, which together provide strong security and privacy for communications.

Implementation Details:

Step: 1 Make keys

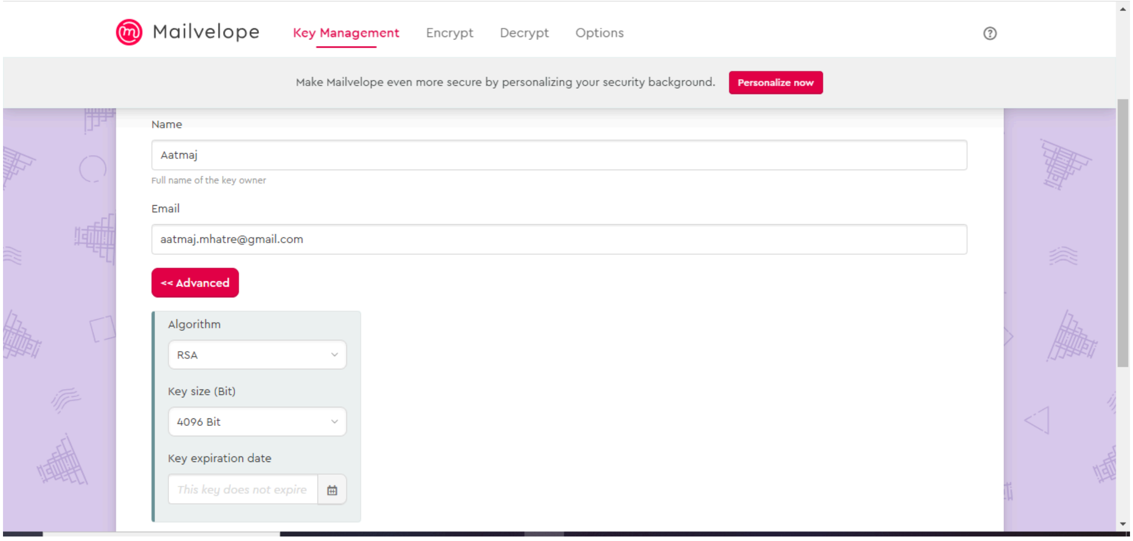
Step: 2 send public key to trusted party

Step: 3 Get public key of receiver

Step: 4 Encrypt message using receivers public key

Step: 5 Send encrypted message

Step: 6 Receiver decrypts using private key



Make Mailvelope even more secure by personalizing your security background.

Name: Aatmaj

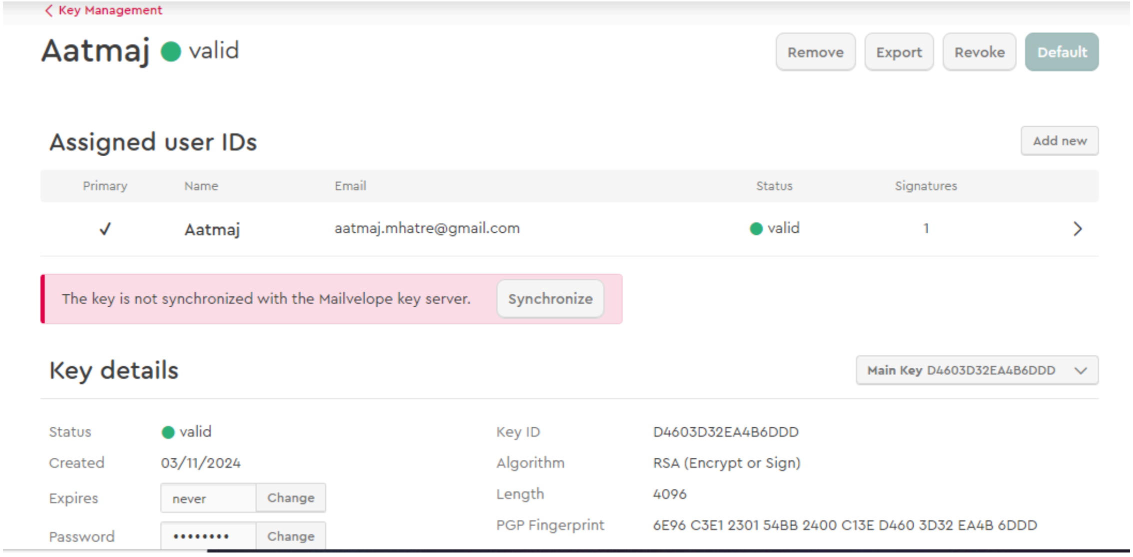
Full name of the key owner

Email: aatmaj.mhatre@gmail.com

Algorithm: RSA

Key size (Bit): 4096 Bit

Key expiration date: This key does not expire



Aatmaj ● valid

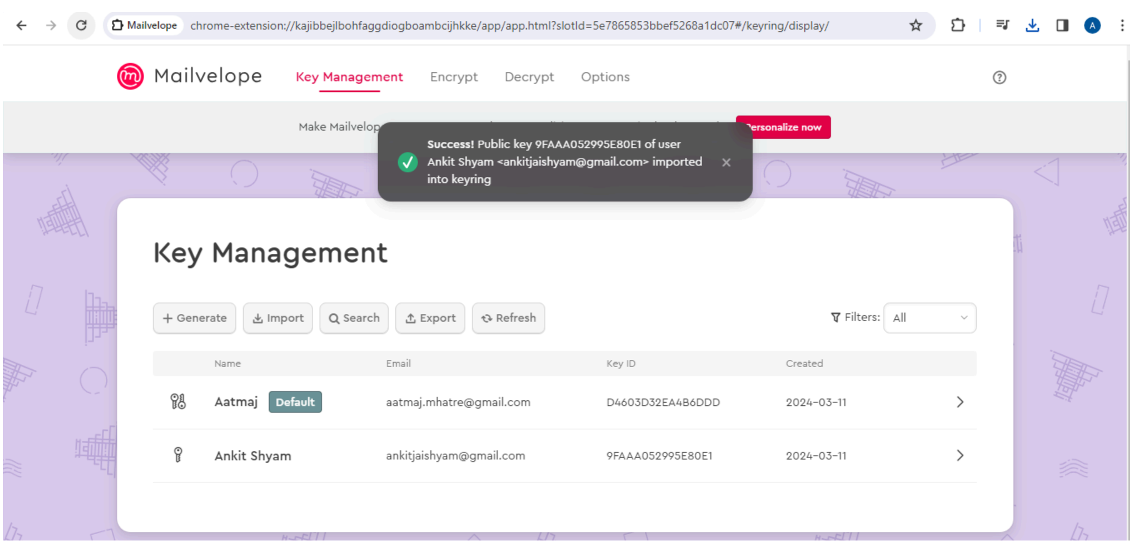
Assigned user IDs

Primary	Name	Email	Status	Signatures
✓	Aatmaj	aatmaj.mhatre@gmail.com	● valid	1

The key is not synchronized with the Mailvelope key server.

Key details

Status	● valid	Key ID	D4603D32EA4B6DDD
Created	03/11/2024	Algorithm	RSA (Encrypt or Sign)
Expires	never Change	Length	4096
Password	•••••••• Change	PGP Fingerprint	6E96 C3E1 2301 54BB 2400 C13E D460 3D32 EA4B 6DDD

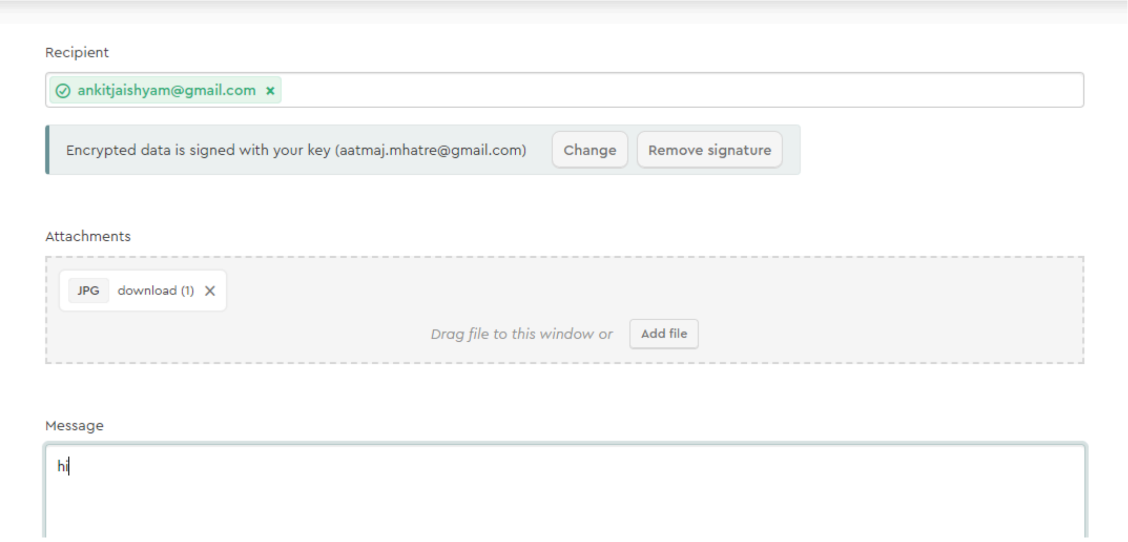


chrome-extension://kajibbejlbohfaggdiogboambcijhkke/app/app.html?slotId=5e7865853bbef5268a1dc07#/keyring/display/

Success! Public key 9FAAA052995E80E1 of user Ankit Shyam <ankitjaishyam@gmail.com> imported into keyring

Key Management

Name	Email	Key ID	Created
Aatmaj Default	aatmaj.mhatre@gmail.com	D4603D32EA4B6DDD	2024-03-11
Ankit Shyam	ankitjaishyam@gmail.com	9FAAA052995E80E1	2024-03-11



Recipient

ankitjaishyam@gmail.com ×

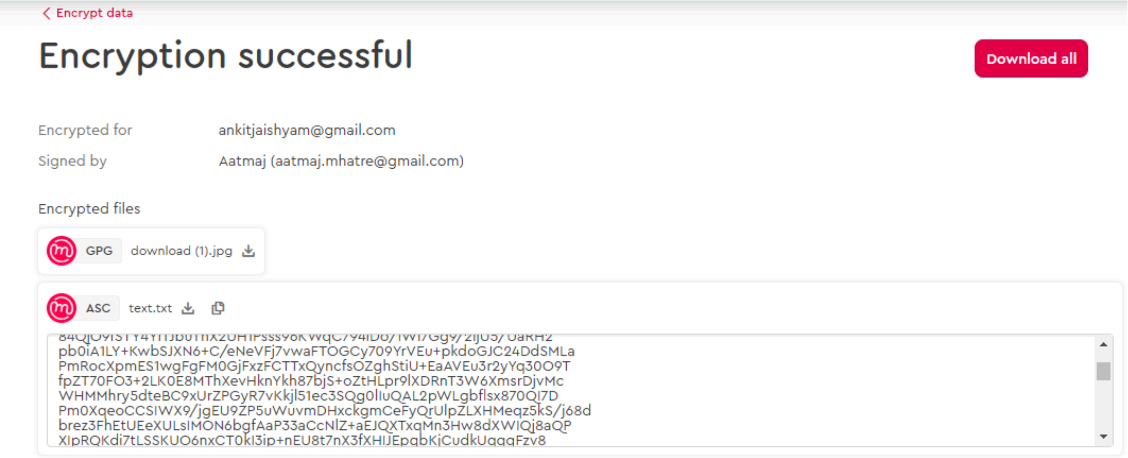
Encrypted data is signed with your key (aatmaj.mhatre@gmail.com)

Attachments

JPG download (1) ×

Message

hi



Encryption successful

Encrypted for ankitjaishyam@gmail.com

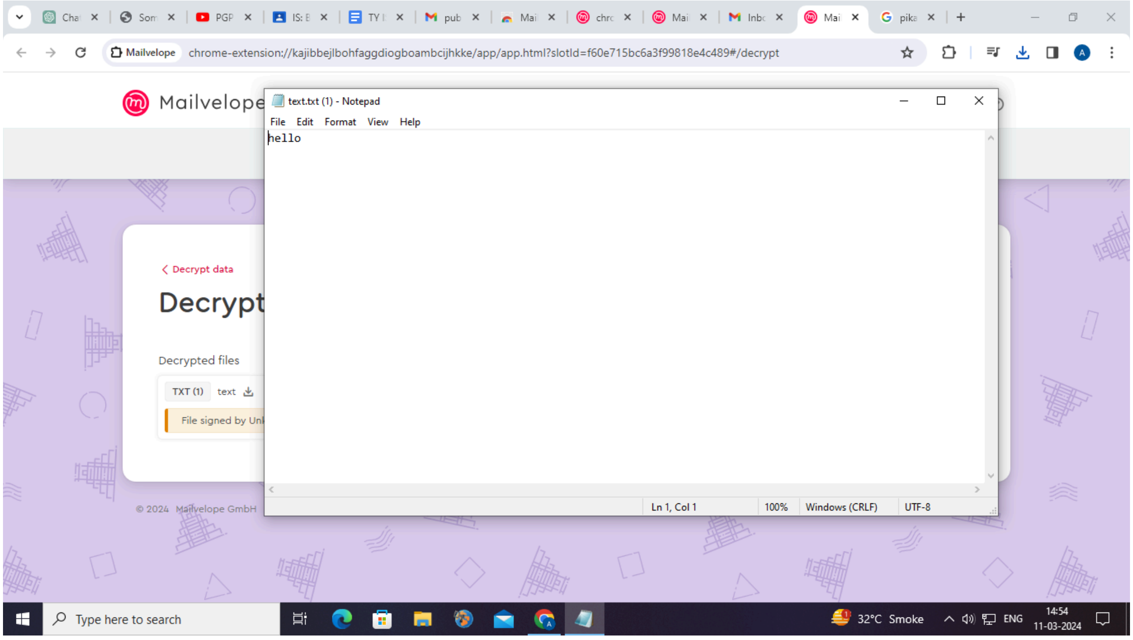
Signed by Aatmaj (aatmaj.mhatre@gmail.com)

Encrypted files

GPG download (1).jpg

ASC text.txt

```
84QjO9lS1Y4YlTJb0TnX2UH1Psss96KWqC794lD6/1W1/Gg9/2ljU5/UaRHZ
pb0iA1LY+KwbSJXN6+C/eNeVFj7vwaFTOGCy709YrVEu+pkdoGJC24DdSMLa
PmRocXpmES1wgFgFM0GjFxzFCTTxQyncfsOZghStiU+EaAVEu3r2yYq30O9T
fpZT70FO3+2LK0E8MThXevHknYkh87bjS+oZtHLpr9lXDRnT3W6XmsrDjvMc
WHMMhry5dteBC9xUrZPGyR7vKkjl51ec3SQg0lIuQAL2pWLgbflsx870QI7D
Pm0XqeoCCSIWX9/jgEU9ZP5uWuvmDHxckgmCeFyQrUlpZLXHMeqz5kS/j68d
brez3FhEtUEeXULsIMON6bgfAaP33aCcNlZ+aEJQXTxqMn3Hw8dXWIQj8aQP
XIpRQKdi7tLSSKUO6nxCT0kI3ip+nEU8t7nX3fXHIJEpqbKjCudkUqqqFzv8
```



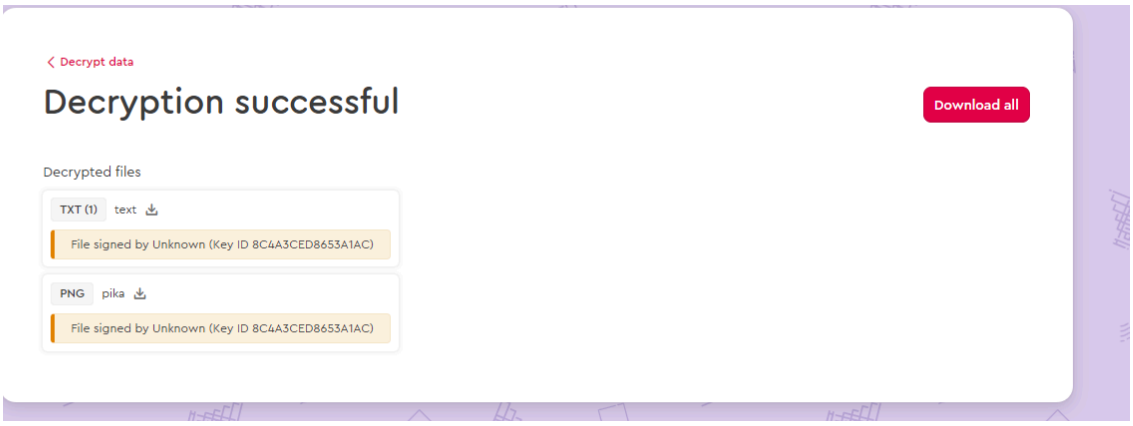
```
hello
```

Decrypt

Decrypted files

TXT (1) text

File signed by Un



< Decrypt data

Decryption successful

Download all

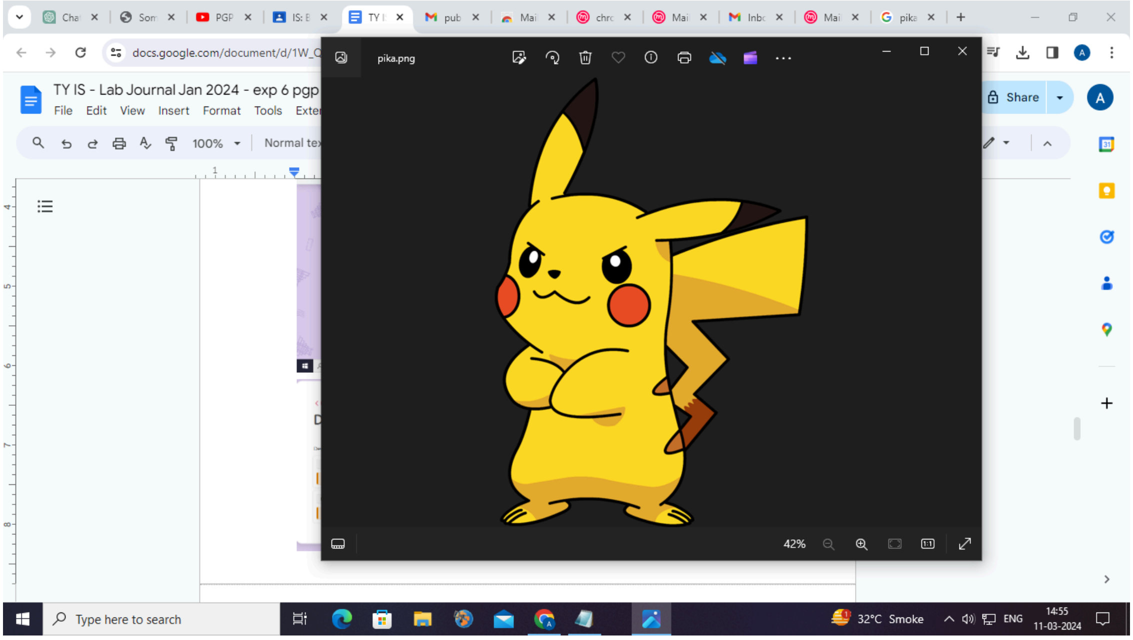
Decrypted files

TXT (1) text

File signed by Unknown (Key ID 8C4A3CED8653A1AC)

PNG pika

File signed by Unknown (Key ID 8C4A3CED8653A1AC)


docs.google.com/document/d/1W_Q
TY IS - Lab Journal Jan 2024 - exp 6 pgp
File Edit View Insert Format Tools Exte
100%
Normal te
Share
pika.png
42%
Type here to search
32°C Smoke
ENG
14:55
11-03-2024

Conclusion:-

Thus we have used pgp to send and receive encrypted data and decrypt data after receiving it. We have understood how pgp works. We have used mailvelope to encrypt and send encrypted messages. Then we used it to decrypt the messages.