

ST. FRANCIS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
SECURITY LAB

Experiment – 11: Implementation of Email security

Aim: To implement Email security.

Objective: After performing the experiment, the students will be able to understand security methods to achieve Email security.

Lab objective mapped: L502.6: Students should be able to apply network security basics, analyze different attacks on networks and evaluate the performance of firewalls and security protocols, such as SSL, IPSEC, and PGP, and authentication mechanisms to design secure applications.

Prerequisite: Basic knowledge of network security.

Requirements: Windows OS, Gpg4win, Kleopatra

Pre-Experiment Theory:

Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories, whole disk partitions and to increase the security of e-mail communications.

OpenPGP is a non-proprietary format for authenticating or encrypting data, using public key cryptography. It is based on the original PGP (Pretty Good Privacy) software. Beginning in 1997, the OpenPGP Working Group was formed in the Internet Engineering Task Force (IETF) to define this standard that had formerly been a proprietary product since 1991. Over the past decade, PGP, and later OpenPGP, has become the standard for nearly all of the world's signed or encrypted email. OpenPGP also defines a standard format for certificates which, unlike most other certificate formats, enables webs of trust.

GnuPG (also known as GPG) is a complete and free implementation of the OpenPGP standard as defined by RFC4880. GnuPG allows you to encrypt and sign your data and communications. It features a versatile key management system, along with access modules for all kinds of public key directories. GnuPG is a command line tool with features for easy integration with other applications. A wealth of frontend applications and libraries are available. GnuPG also provides support for S/MIME and Secure Shell (ssh).

Gpg4win is a Windows version of GnuPG featuring a context menu tool, a crypto manager, and an Outlook plugin to send and receive standard PGP/MIME mails. The current version of Gpg4win is 4.2.0.

Implementation:

For implementation of Email security through GPG we will use Kleopatra. Kleopatra is a certificate manager and GUI for GnuPG. The software helps to store OpenPGP certificates and keys. It is available for Windows and Linux.

1. Download and Install Gpg4win from its official website [1]. Choose 'Kleopatra' as a key manager component during installation.
2. Open Kleopatra interface from desktop icon.
3. Create a new key pair (choose open PGP key pair if prompted)
4. Save the keys on the desktop and observe the keys.
5. Import your secret communication partner's (your friend's) key in Kleopatra.
6. Write your secret message in any word processing software. (e.g. Microsoft Word, notepad, WordPad etc.)
7. Encrypt this message file with recipient's public key. Save and send it to the recipient through any preferred communication medium.
8. Decrypt and observe the message received by you.

Output:

Attach following screenshots (SS) as the output. Write a brief explanation for each.

1. Welcome window of Kleopatra interface.
2. SS with new key pair created.
3. SS of private and public key.
4. SS with imported public key.
5. SS with message and encrypted message.
6. SS with message decryption.

Post Experimental Exercise- *(to be handwritten on journal sheets.)*

Write answers to following questions.

1. What is PGP?
2. What is S/MIME?

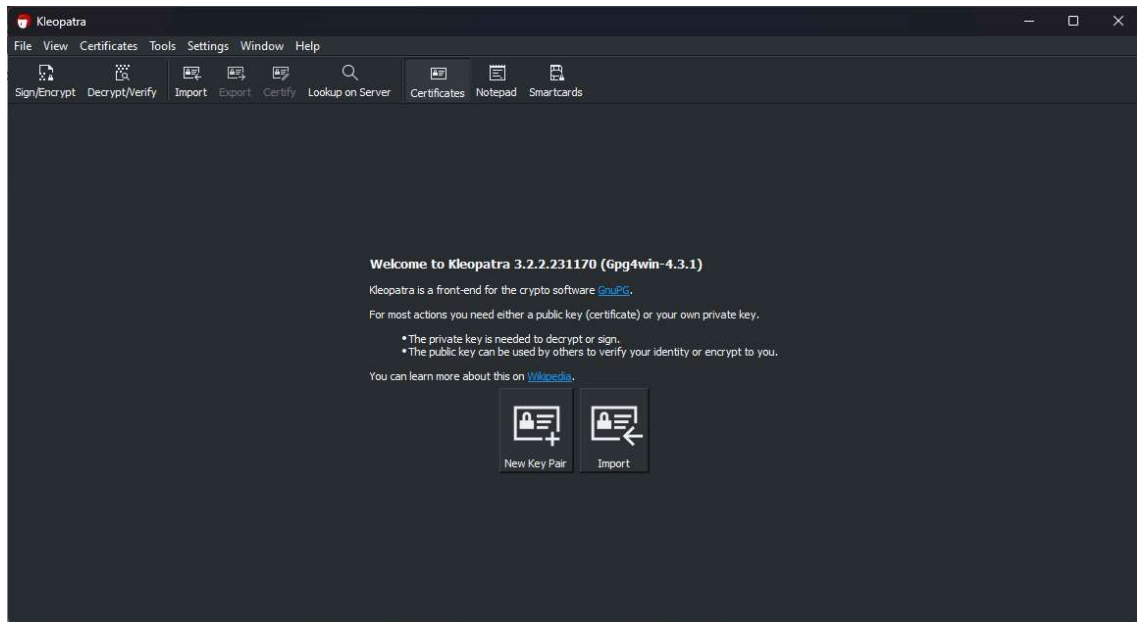
Conclusion:

GPG is used for authentication and privacy to messages over the internet. GPG was originated to address the security concerns of plain e-mail or text messages. Gnupg is used to demonstrate usage of GPG. Kleopatra helps to store OpenPGP certificates and keys.

References:

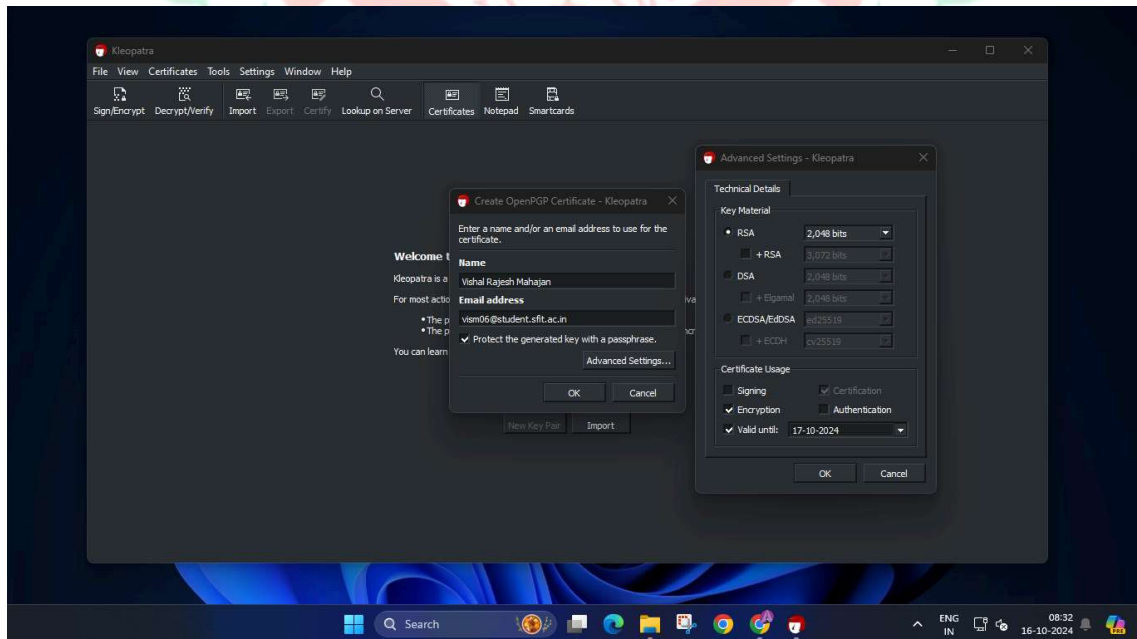
- [1] "Gpg4win - a secure solution", <https://www.gpg4win.org>
- [2] "Kleopatra", <https://www.openpgp.org/software/kleopatra/>
- [3] "Pretty Good Privacy", https://en.wikipedia.org/wiki/Pretty_Good_Privacy
- [4] "The Complete PGP Encryption Tutorial | Gpg4win & GnuPG", <https://youtu.be/CEADq-B8KtI>

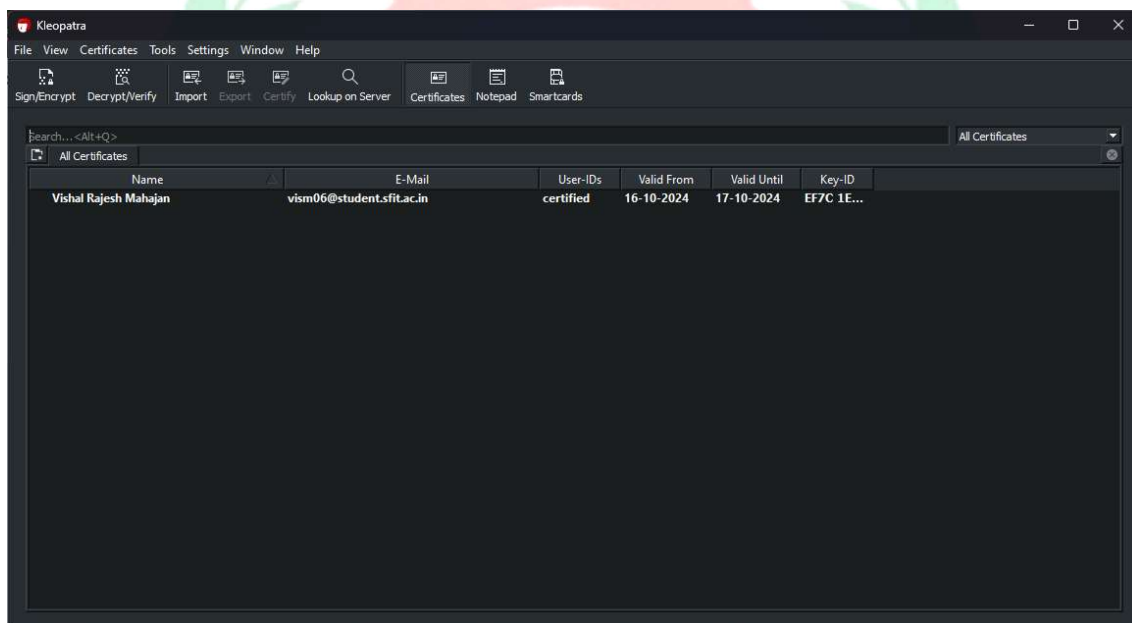
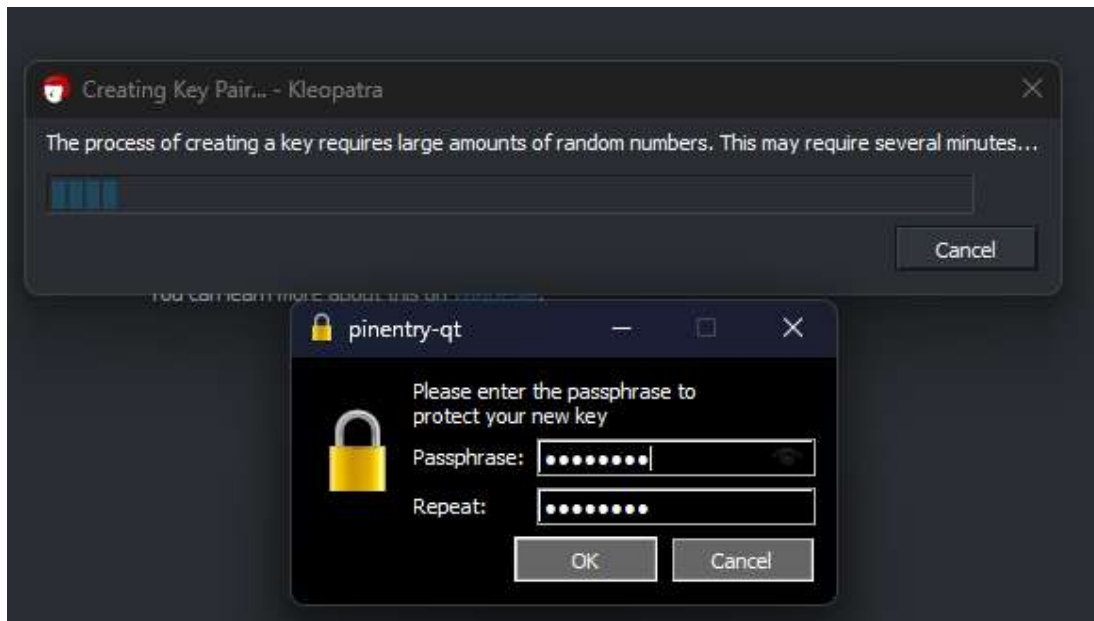
1. Welcome window of Kleopatra interface.



The Welcome window of the Kleopatra interface introduces users to the application, highlighting its key features like key management and document encryption. It provides navigation options to set up a keyring, import keys, or access help resources, making it user-friendly. This initial screen aims to facilitate a smooth onboarding experience for new users.

2. SS with new key pair created.





The screenshot displaying a newly created key pair in Kleopatra typically shows the key details, including the key ID, type (e.g., RSA), and expiration date. Users can see the public and private keys associated with their identity, along with options to export, manage, or use the keys for encryption and signing. This visual representation confirms the successful generation of the key pair, essential for secure communications.

3. SS of private and public key.

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

mQENBGCPLMsBCAC+862/a5GHSS/bhIXfW8WpBT131BPcRVJPY9mTYV9Dz5y9ALNu|
qQWTO1R9fjWGR3Y1PQnwGsEQ3VKsNtnGUetW/tQLe3g6YTP0egRtgoXJtt8JXlyi
LmpFlDQsrnUhwK2YE8A03MHdI+ZWBTZQW54DyqaPWLf+QjF+JKPWP3ro+dJ8jeOC
Tu4FoqknGJKda1Q8unPZzOsAsYzXWjEewWI7ApYD/judi8Gj6s06b+L022HntcFB
qZUBlytTjWYMPZBSbuCe33xurRdCpIwbszHMP0qKHZDp/CB5r9e3Dbx5/tJO+gCB
t9mWliVfCEdFUKQd6EVVAa6dNx0o35zRQXgJABEBAAG0MvZpc2hhbCB5Wp1c2gg
TWFOYWhpbhA8dmlzbTA2QHh0dWR1bnQuc2ZpdC5hYy5pbj6JAVcEEwEIAEewIQSY
2IQ+q5Mb1LFDrvLvF4ytBNfXAUCZw8sywIbDQUJAAAGCHQULCQgHAGIiAgYVCgKI
CwIEFgIDAQIeBwIXgAAKCRDvf84ytBNfXBR8B/40+6sWBXkqzFckDgU9M/D3djYn
MWhNeLhmz68UHj761RUIQ081XU/xisJQo6gyMQW0xQ7A6F0YzE5wx90J76f0JRnf
2WPA4QaeaZP050jccUggz7bGmlcv1yUgV1Lro8B0SMs/rWZy6Uy3Kk1AgotnFY6v
wpD8h9t1YLilyZxU8ZbnnxHnZv8f5RmD/5j1LhpDr+213T5cQleYHJ6SLXZ6mo5c
p7QKPLU0MIiMD5T5ye0DcTnQUBY1vNay0xX6BSoHJR6GATb8/Ohf7vmf1H4Nt5Dx
2GYu3q3uLsrMkzft4bG85KKOYFHAx8a2UcVMzc8B5bbyqC1AX0BfsEzkl1
=hB85
-----END PGP PUBLIC KEY BLOCK-----

Ln 3, Col 65  985 characters  100%  Windows (CRLF)  UTF-8
```

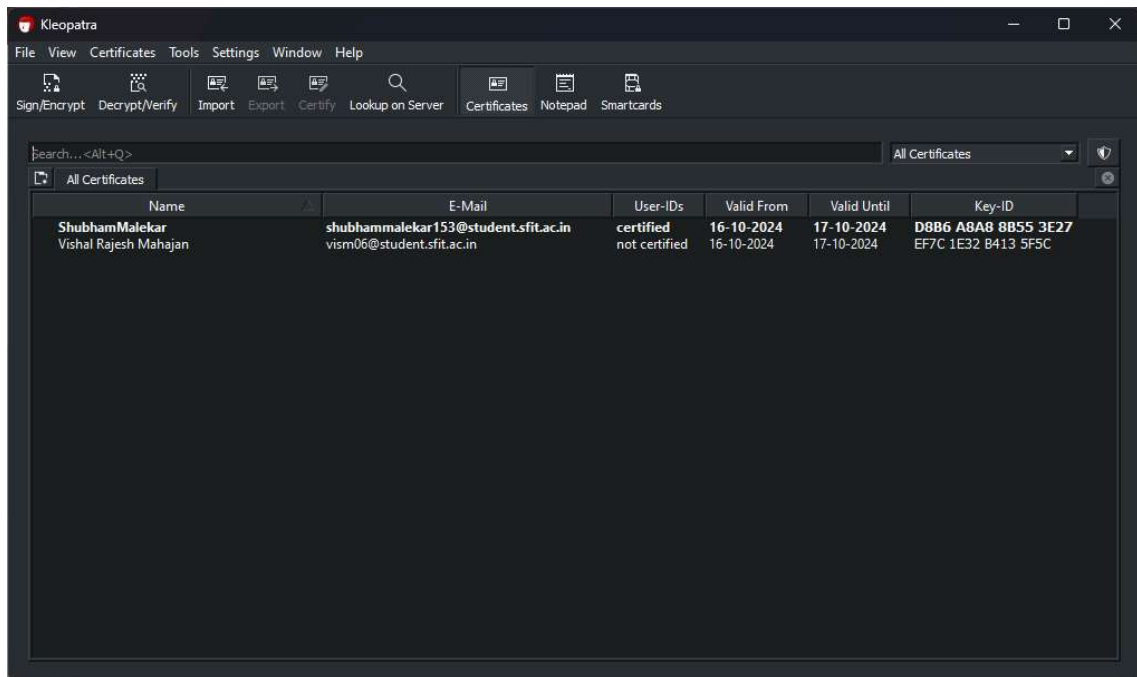
```
-----BEGIN PGP PRIVATE KEY BLOCK-----

lQPGBGcPLMsBCAC+862/a5GHSS/bhIXfW8WpBT131BPcRVJPY9mTYV9Dz5y9ALNu
qQWTO1R9fjWGR3Y1PQnwGsEQ3VKsNtnGUetW/tQLe3g6YTP0egRtgoXJtt8JXlyi
LmpFlDQsrnUhwK2YE8A03MHdI+ZWBTZQW54DyqaPWLf+QjF+JKPWP3ro+dJ8jeOC
Tu4FoqknGJKda1Q8unPZzOsAsYzXWjEewWI7ApYD/judi8Gj6s06b+L022HntcFB
qZUBlytTjWYMPZBSbuCe33xurRdCpIwbszHMP0qKHZDp/CB5r9e3Dbx5/tJO+gCB
t9mWliVfCEdFUKQd6EVVAa6dNx0o35zRQXgJABEBAAG0MvZpc2hhbCB5Wp1c2gg
TWFOYWhpbhA8dmlzbTA2QHh0dWR1bnQuc2ZpdC5hYy5pbj6JAVcEEwEIAEewIQSY
2IQ+q5Mb1LFDrvLvF4ytBNfXAUCZw8sywIbDQUJAAAGCHQULCQgHAGIiAgYVCgKI
CwIEFgIDAQIeBwIXgAAKCRDvf84ytBNfXBR8B/40+6sWBXkqzFckDgU9M/D3djYn
MWhNeLhmz68UHj761RUIQ081XU/xisJQo6gyMQW0xQ7A6F0YzE5wx90J76f0JRnf
2WPA4QaeaZP050jccUggz7bGmlcv1yUgV1Lro8B0SMs/rWZy6Uy3Kk1AgotnFY6v
wpD8h9t1YLilyZxU8ZbnnxHnZv8f5RmD/5j1LhpDr+213T5cQleYHJ6SLXZ6mo5c
p7QKPLU0MIiMD5T5ye0DcTnQUBY1vNay0xX6BSoHJR6GATb8/Ohf7vmf1H4Nt5Dx
2GYu3q3uLsrMkzft4bG85KKOYFHAx8a2UcVMzc8B5bbyqC1AX0BfsEzkl1
=JK8N
-----END PGP PRIVATE KEY BLOCK-----

Ln 1, Col 1  1,934 characters  100%  Windows (CRLF)  UTF-8
```

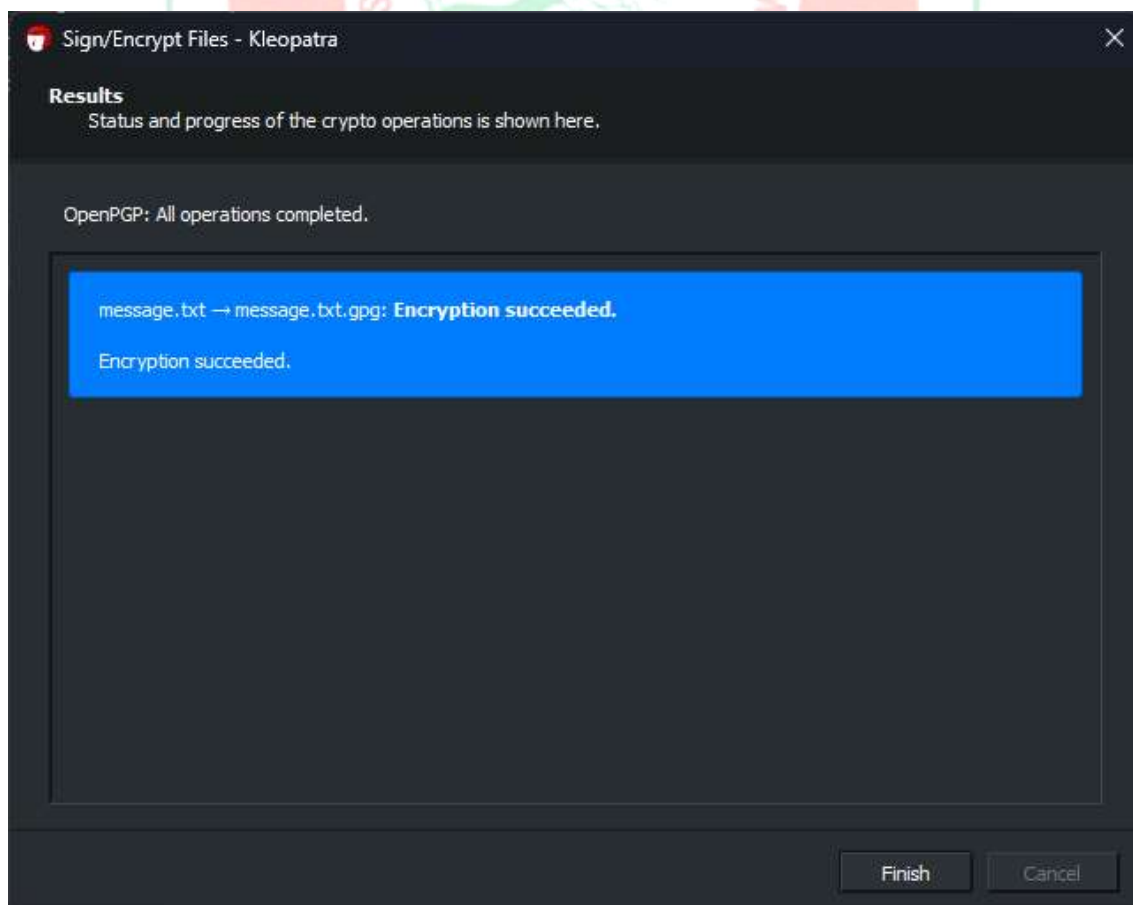
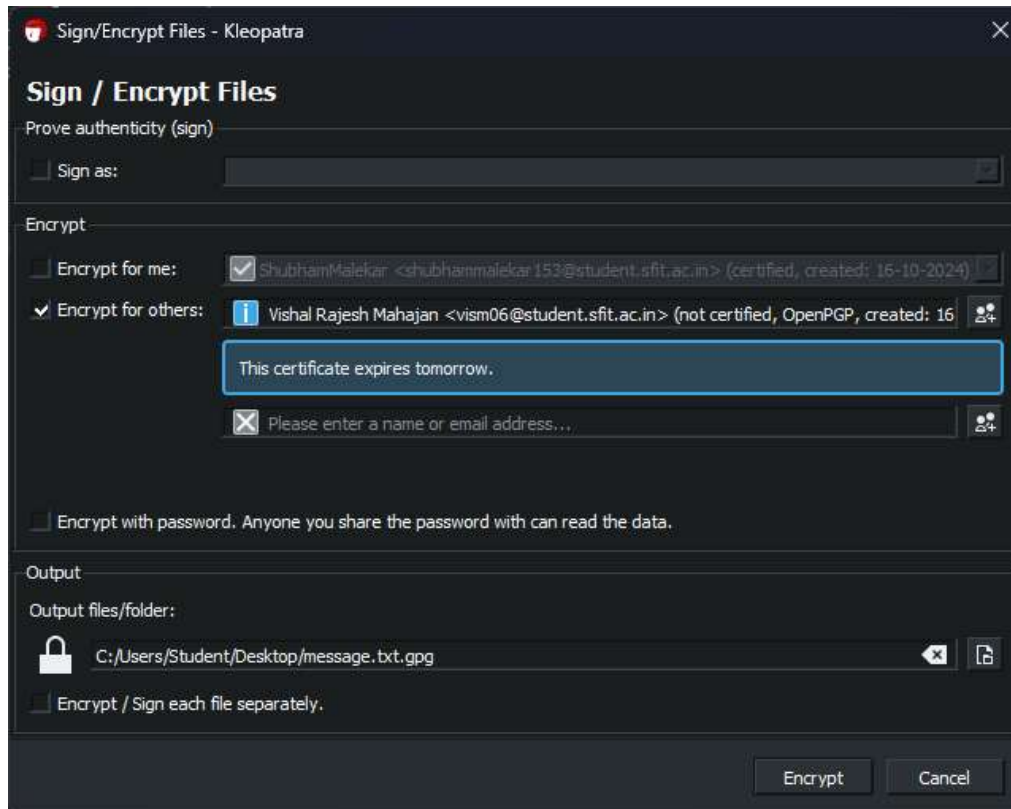
The screenshot showing the private and public keys in Kleopatra typically displays two distinct sections: one for the public key, which is meant for sharing and encrypting messages, and another for the private key, which must be kept secure and used for decryption and signing. Each key section usually includes details such as the key ID, algorithm type, and expiration date, along with options to export or manage the keys. This visual emphasizes the importance of both keys in secure communication and identity verification.

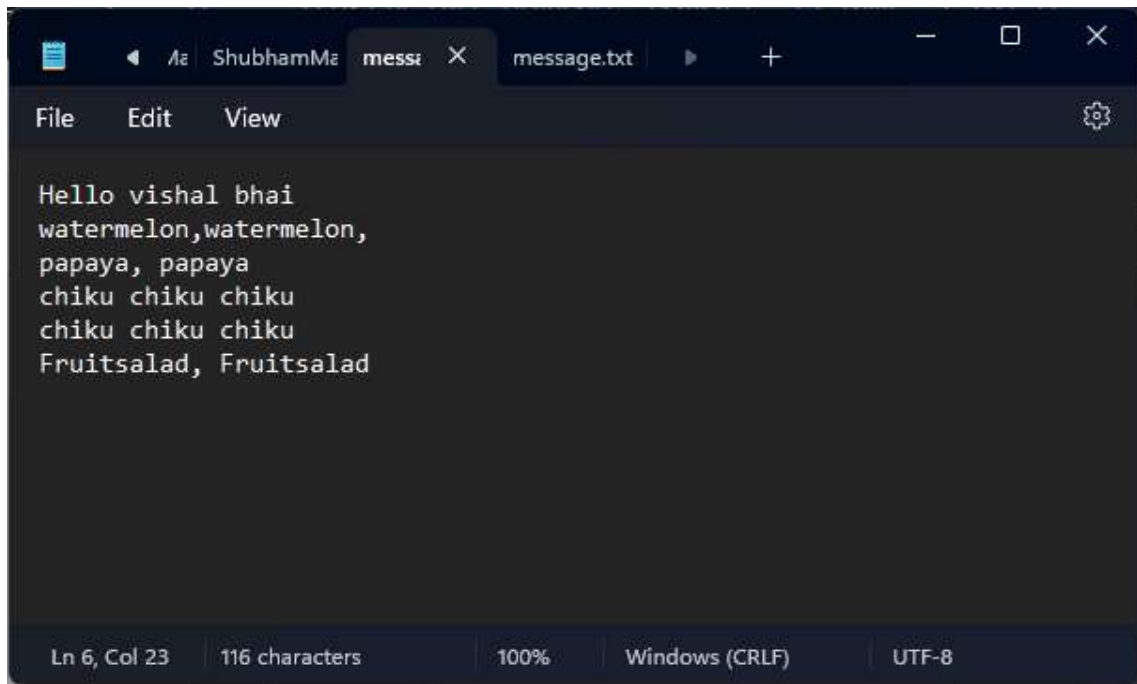
4. SS with imported public key.



The screenshot showing an imported public key in Kleopatra typically displays the key details, including the key ID, owner's name, and expiration date. It may also indicate the status of the key, such as whether it is trusted or verified. This visual confirms that the public key has been successfully added to the user's keyring, allowing for secure communication and encryption with the key's owner.

5. SS with message and encrypted message.

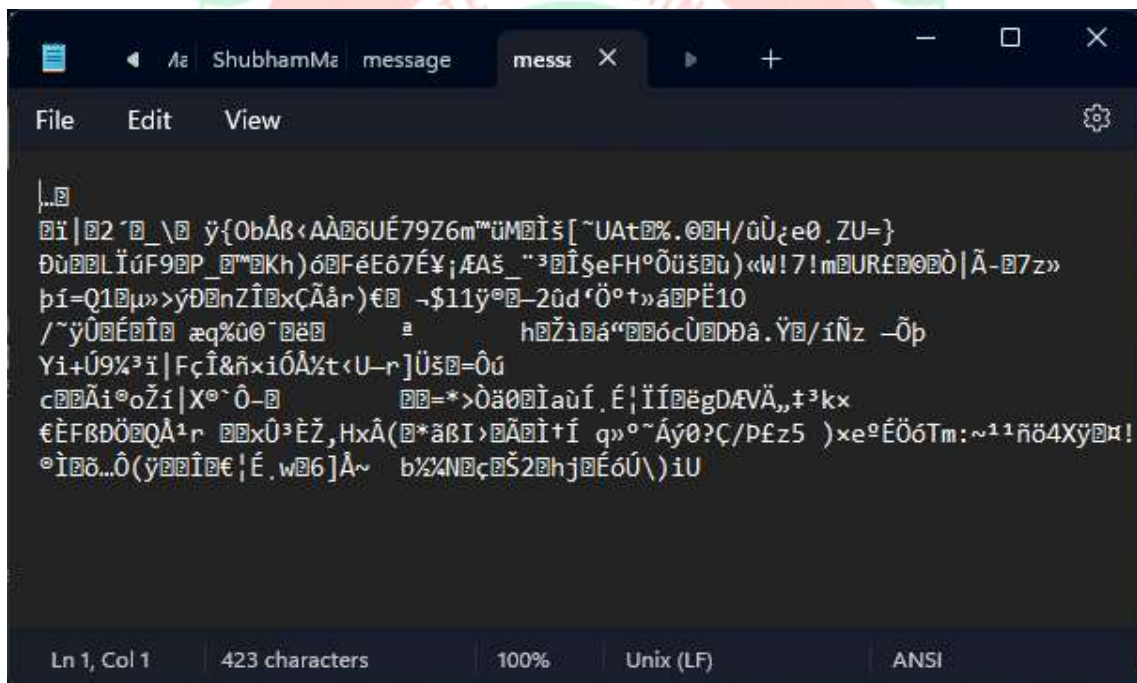




A screenshot of a text editor window with a dark theme. The title bar shows 'ShubhamM...' and 'message.txt'. The menu bar includes 'File', 'Edit', and 'View'. The text content is as follows:

```
Hello vishal bhai
watermelon,watermelon,
papaya, papaya
chiku chiku chiku
chiku chiku chiku
Fruitsalad, Fruitsalad
```

The status bar at the bottom indicates 'Ln 6, Col 23', '116 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.



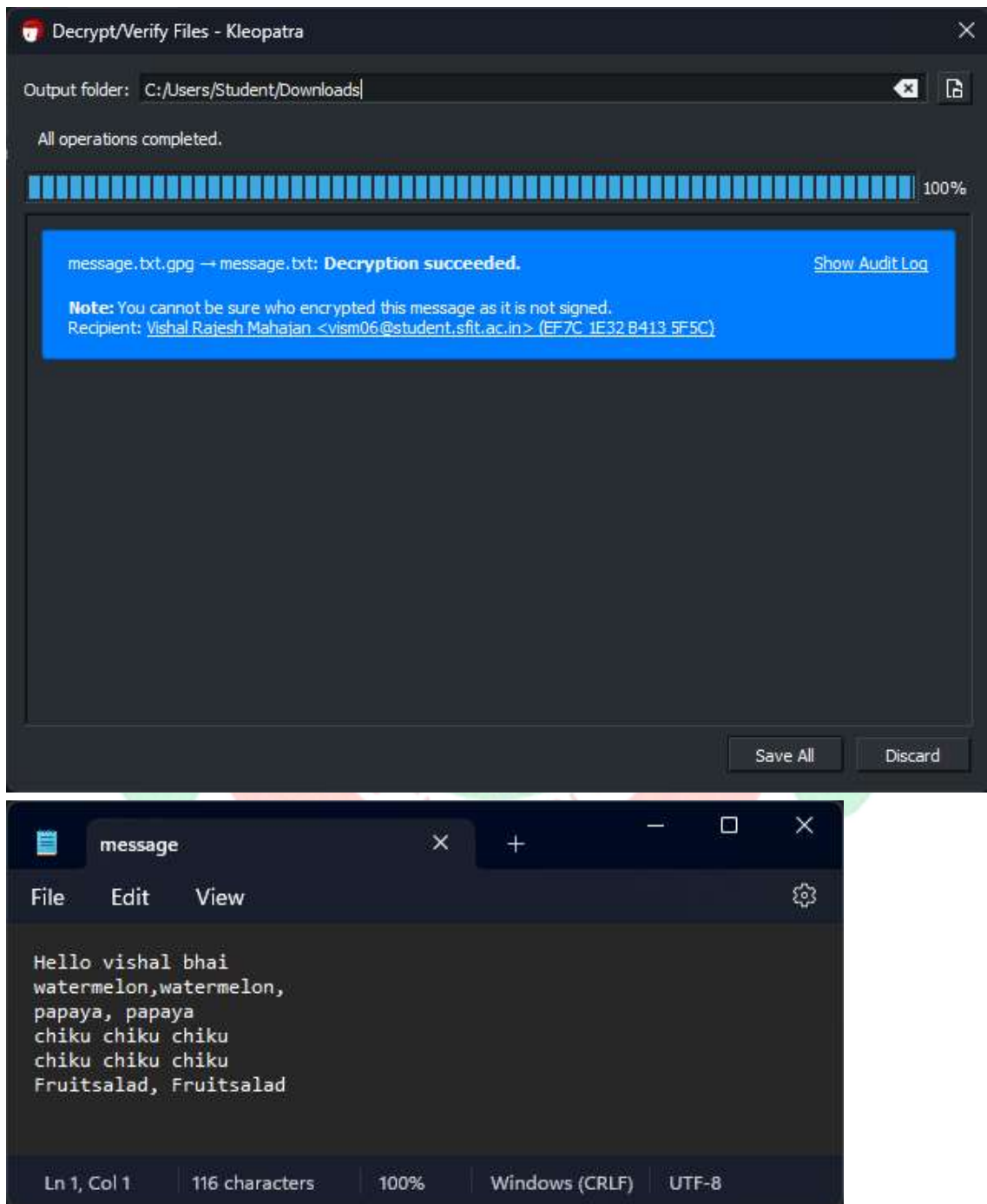
A screenshot of a text editor window with a dark theme. The title bar shows 'ShubhamM...' and 'message'. The menu bar includes 'File', 'Edit', and 'View'. The text content is a single line of encrypted data, appearing as a string of unreadable characters.

```
|..
[Unreadable encrypted text]
```

The status bar at the bottom indicates 'Ln 1, Col 1', '423 characters', '100%', 'Unix (LF)', and 'ANSI'.

The screenshot displaying a message alongside its encrypted version in Kleopatra typically shows the original plaintext message, highlighting its content for easy reference. Below or beside it, the encrypted message is presented, usually as a string of unreadable characters. This visual demonstrates the encryption process, emphasizing how the original message is transformed to ensure confidentiality before being sent securely.

6. SS with message decryption.



The screenshot showing message decryption in Kleopatra typically displays the encrypted message along with the decrypted plaintext result. It may include fields indicating the successful decryption status, the keys used, and any relevant notifications about the process. This visual highlights how the encrypted content is transformed back into a readable format, showcasing the effectiveness of the encryption and decryption mechanism.