**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.

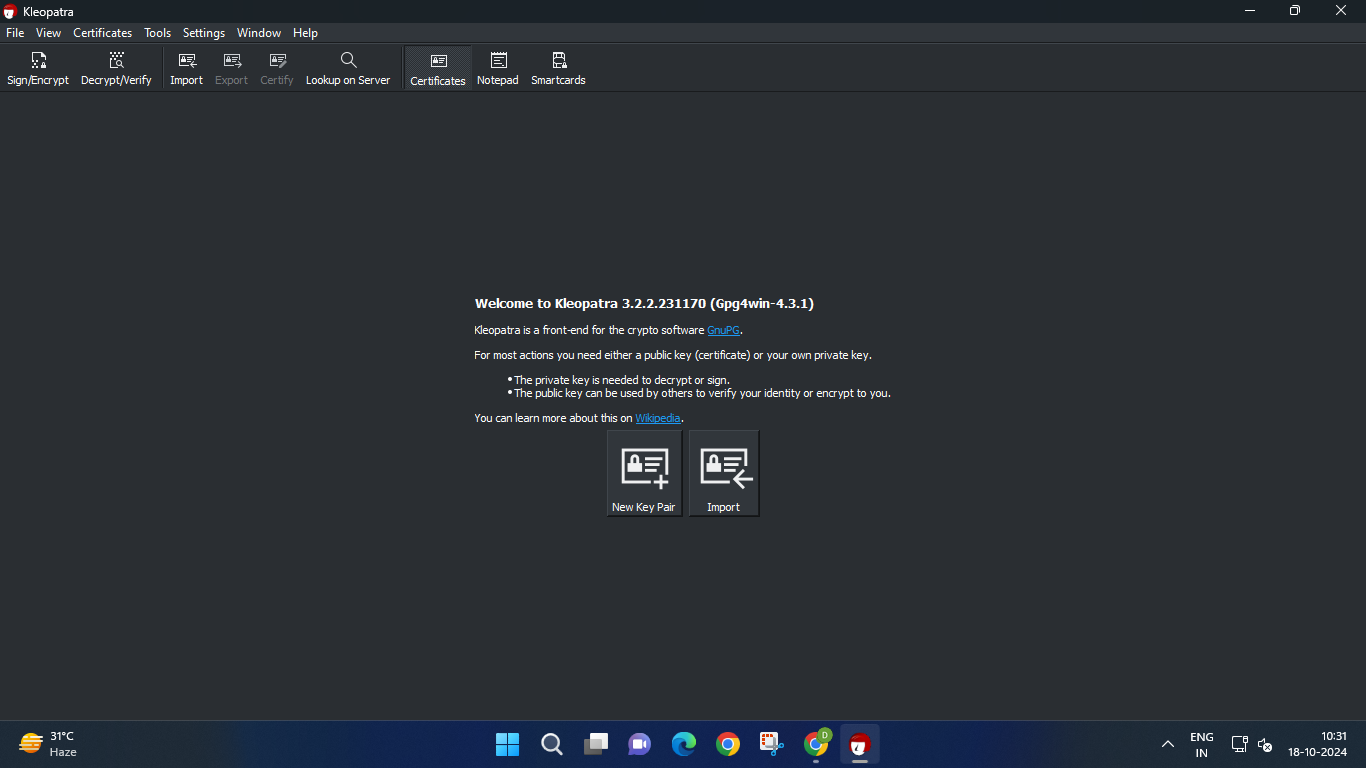
1. What is PGP?
2. What is S/MIME?

**Conclusion:**

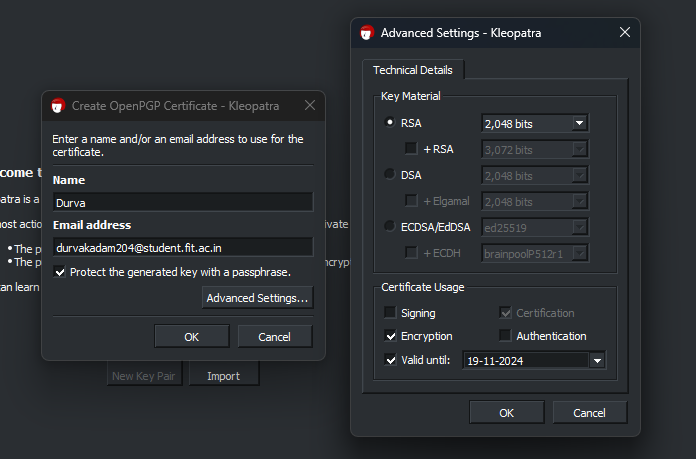
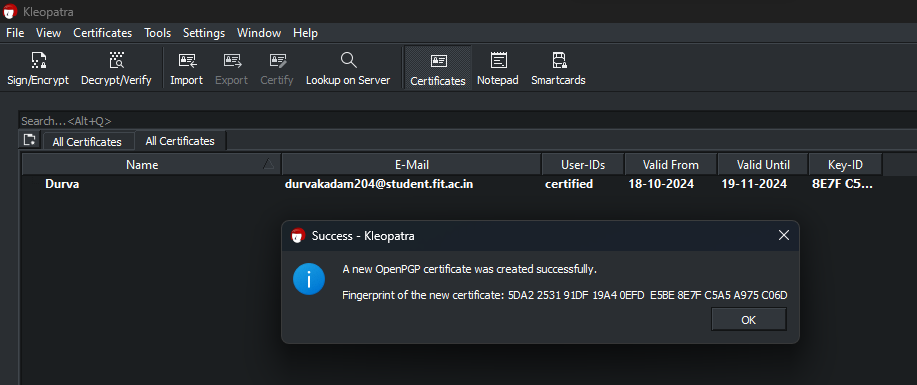
GPG is used for authentication and privacy to messages over the internet. GPG was originated to address the security concerns of plain e-mail or text messages. Gnupg is used to demonstrate usage of GPG. Kleopatra helps to store OpenPGP certificates and keys.

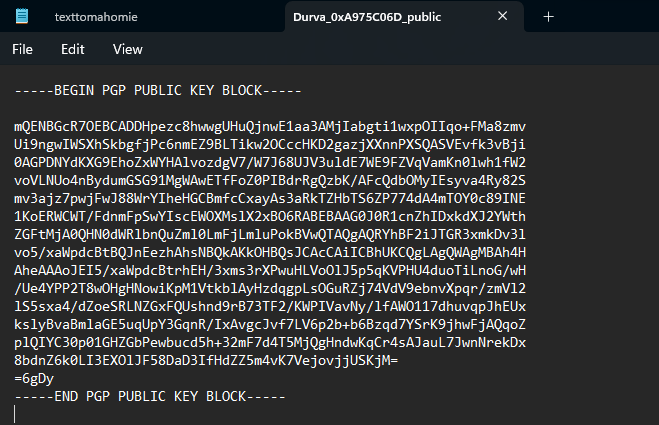
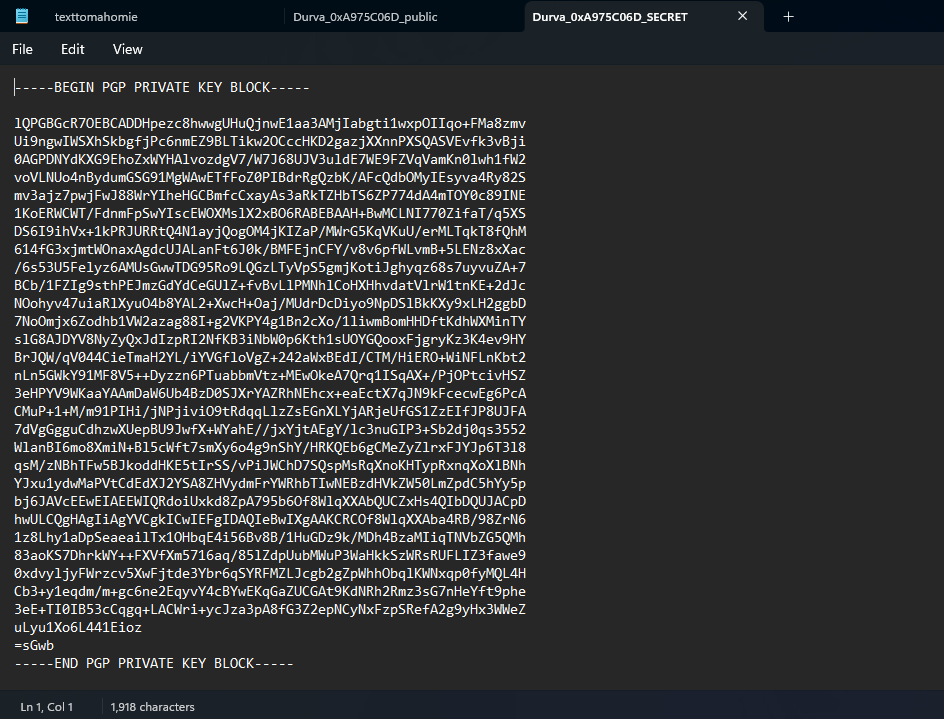
**References:**

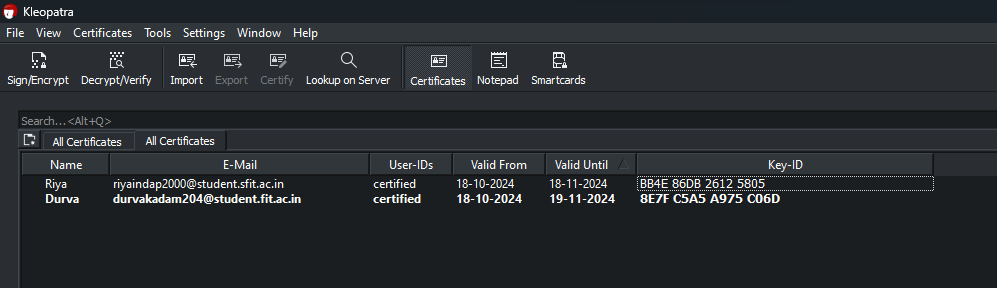
1. “Gpg4win - a secure solution”, <https://www.gpg4win.org>
2. “Kleopatra”, <https://www.openpgp.org/software/kleopatra/>
3. “Pretty Good Privacy”, <https://en.wikipedia.org/wiki/Pretty_Good_Privacy>
4. “The Complete PGP Encryption Tutorial | Gpg4win & GnuPG “, <https://youtu.be/CEADq-B8KtI>

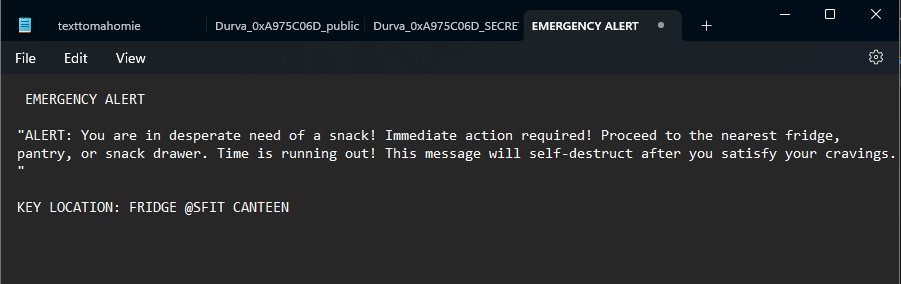
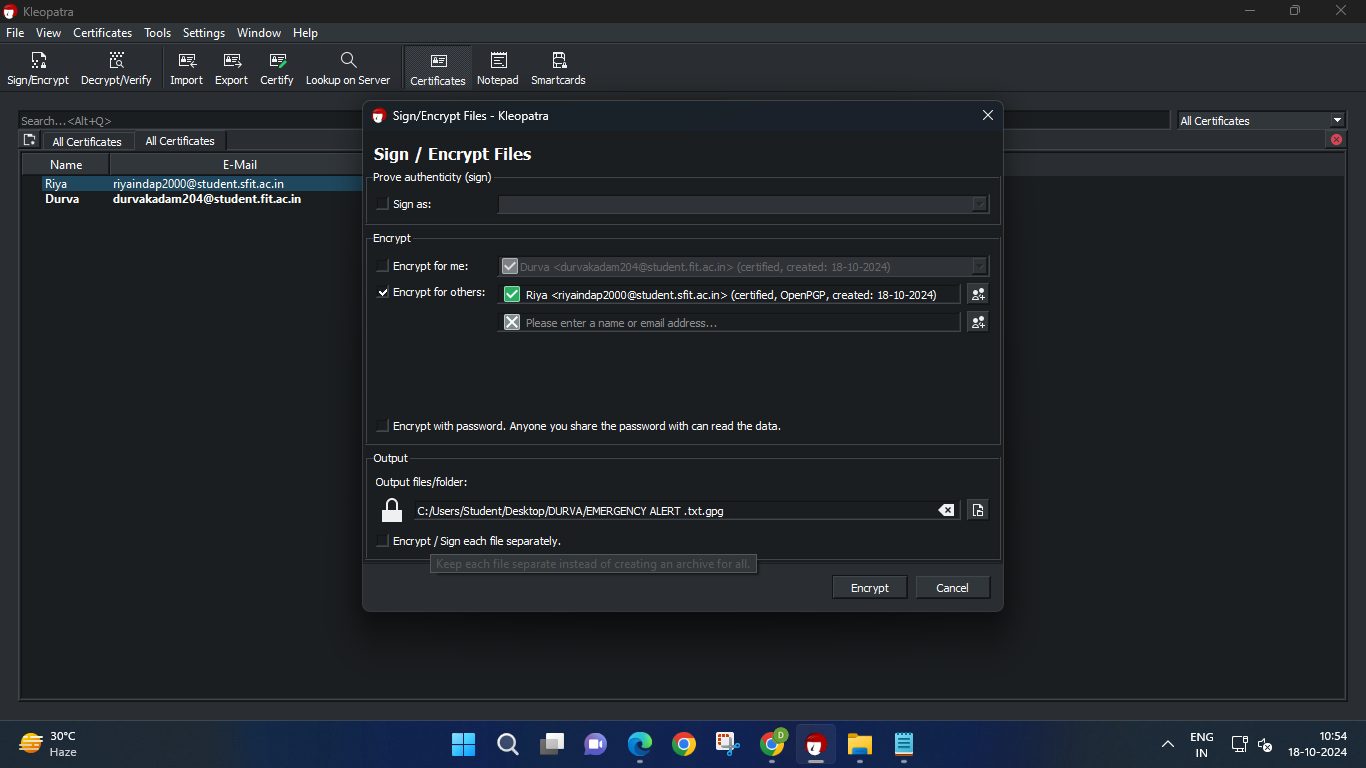
**1. Welcome window of Kleopatra interface.  
  
**

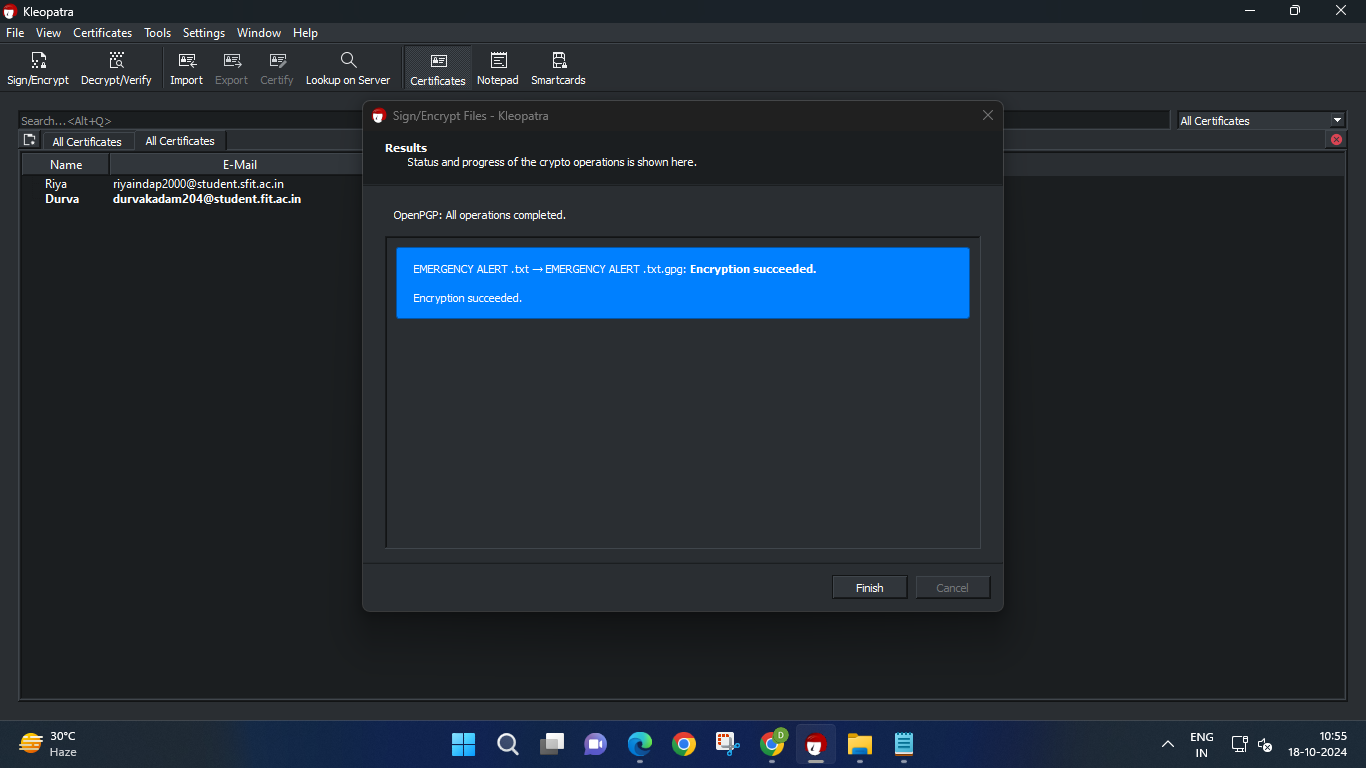
This is the Kleopatra graphical user interface for managing cryptographic keys, part of the Gpg4win suite. It allows users to easily encrypt, decrypt, sign, and manage public key certificates and private keys. Kleopatra's interface is designed to simplify cryptographic tasks for users who may not be familiar with command-line tools, making encryption and key management more accessible.

**2. SS with new key pair created.**  
 ****We click on the new key pair option and fill in our details and open the advanced settings tab for additional settings. In the advanced tabs, we choose the RSA encryption format and set the number of bits and a date until when the key pair certificate will be valid. This allows for control over the key's lifecycle, ensuring it’s only usable for a set period before requiring renewal.We can see that the certificate has been created and successfully. We can now see our certificate and the key id on the gui. From here we can select the certificate to perform actions like exporting the public key to share with others, signing data, or encrypting files.  
The certificate and its details makes it simple to manage encryption keys and ensure their proper usage.

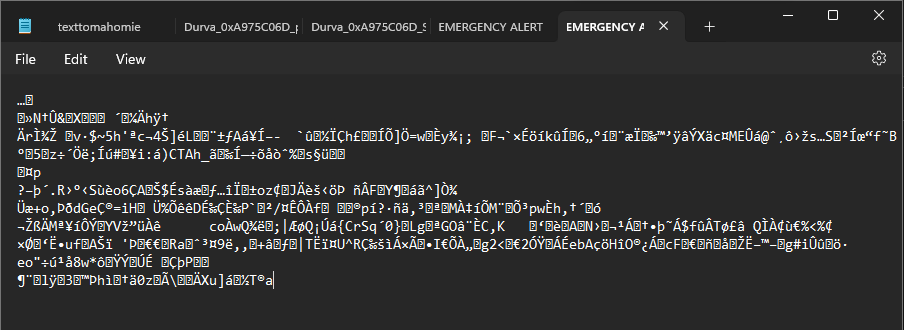
**3. SS of private and public key.**  
The public key can be exported from Kleopatra by selecting the key from the list, right-clicking, and choosing the export option. The public key is generally shared with others so they can encrypt messages for you. Sharing the public key is safe since it can't be used to decrypt data. It also enables others to verify digital signatures you create, ensuring the authenticity of your communications.  
 The private key can be exported by selecting it, but it’s kept secure and not shared publicly. This key is used to decrypt the messages encrypted with your public key. During the export process, you’ll be prompted to choose a location to save the keys in .asc or .gpg format.  
Unlike the public key, which is shared with others for encryption purposes, the private key is kept confidential and should never be shared publicly.

**4. SS with imported public key.**  
  
After receiving our friend’s public key file via email, we saved it to our local storage. To import it into Kleopatra, we opened the application and clicked on the "File" menu, then selected "Import Certificates." We navigated to the location where we saved the public key file, selected it, and clicked "Open." Then we could see our friend's public key listed in the Certificates section.

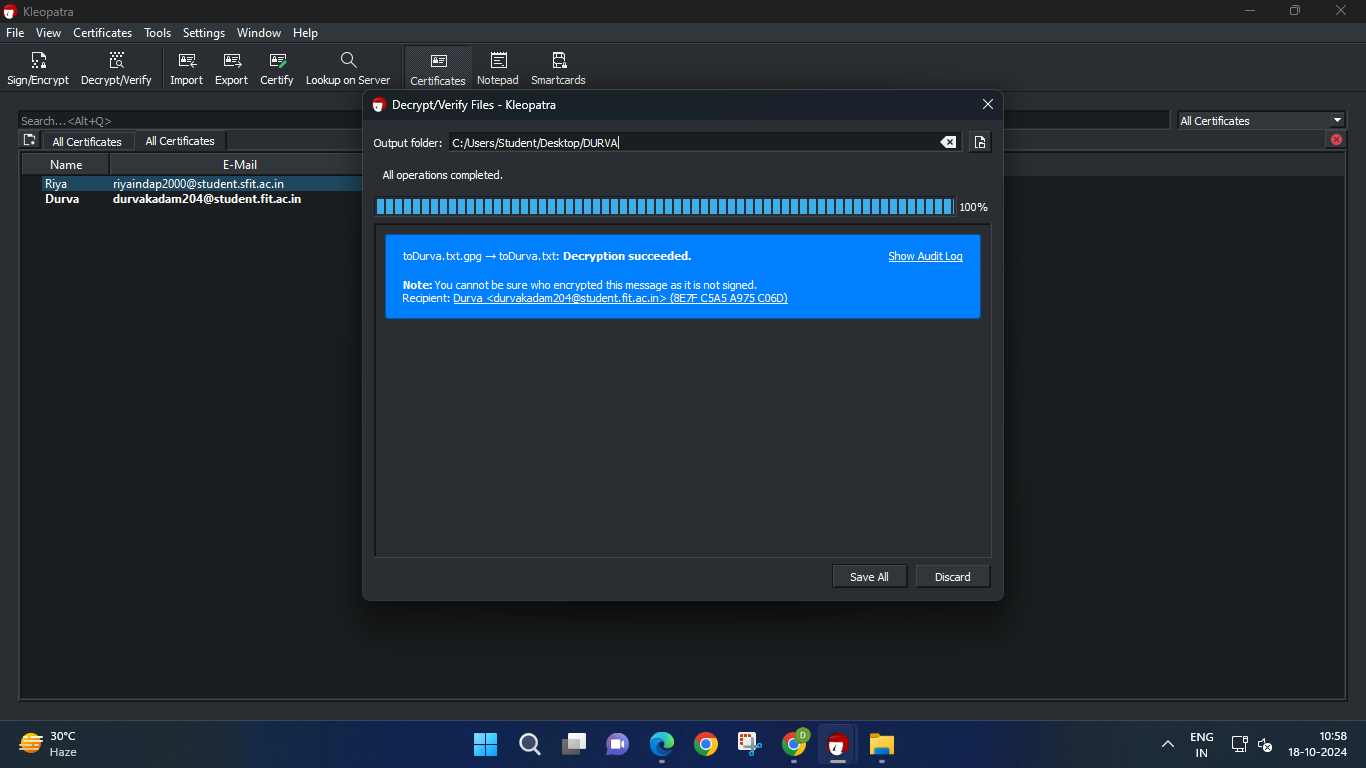
**5. SS with message and encrypted message.**  
  
This is the message that I'm going to encrypt on cleopatra to send to my friend via email. We open Kleopatra and select our friend's public key from the list of certificates. We then use the "Encrypt" function to securely encrypt the message with our friend's public key  
  
.   
  
We select the file we want to encrypt and then right-click on it. From the context menu, we choose the "Encrypt" option. This action prompts us to select the recipient's public key, ensuring that only they can decrypt the file. After selecting the public key, we confirm the encryption process.

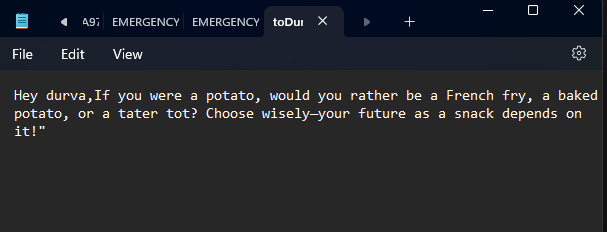


After we complete the encryption process in Kleopatra, a confirmation message appears, indicating that the encryption was successful. This message reassures us that our text file has been securely encrypted using the recipient's public key.

  
The encrypted message in my folder is a securely protected text file, ensuring that only the intended recipient can access its contents. It is encoded using my friend's public key, making it unreadable to anyone without the corresponding private key.

**6. SS with message decryption.**

  
To decrypt the encrypted message that my friend sent me, we start by opening Kleopatra. First, we locate the encrypted file in our folder. Kleopatra will automatically use our private key to decrypt the message. If the private key is protected by a passphrase, we will be prompted to enter it for verification. After entering the passphrase, Kleopatra processes the file and reveals the original message, allowing us to read it securely.

  
  
This is the decrypted message we see after the decryption process. This process ensures that we can access the contents of the encrypted message while maintaining its security throughout the transmission.