CRYPTOGRAPHY

Module-20

Aniket Sunil Pagare

🔐 Table of Contents

- **Introduction to Cryptography**
  - 1.1 What is Cryptography
  - 1.2 Objectives of Cryptography
  - 1.3 Cryptographic Process
  - 1.4 Types of Cryptography
    - 1.4.1 Symmetric Key Cryptography
    - 1.4.2 Asymmetric Key Cryptography
    - 1.4.3 Hash Functions

- **Public Key Infrastructure (PKI)**
  - 2.1 Definition of PKI
  - 2.2 Main Purpose of PKI
  - 2.3 Core Components of PKI
    - 2.3.1 Certificate Authority (CA)
    - 2.3.2 Registration Authority (RA)
    - 2.3.3 Key Pairs
  - 2.4 How PKI Works
  - 2.5 Signed Certificate
  - 2.6 Self-Signed Certificate
  - 2.7 Digital Signatures

- **Secure Communication Protocols**
  - 3.1 Secure Socket Layer (SSL)
  - 3.2 Transport Layer Security (TLS)

- **Cryptanalysis**
  - 4.1 Cryptanalysis Methods
    - 4.1.1 Brute Force Attack
    - 4.1.2 Frequency Analysis
    - 4.1.3 Side-Channel Attack

- **Private Encryption Tools**
  - 5.1 DataLock Using Private Encryption Application
    - 5.1.1 Definition
    - 5.1.2 Performing Activity Using Private Encryption

- **VeraCrypt – Confidential Data Encryption**
  - 6.1 Definition
  - 6.2 Key Features
  - 6.3 Use Cases
  - 6.4 How to Download VeraCrypt
  - 6.5 Performing Encryption Activity

EXTRA ACTIVITY

- **Cryptography Tools – Advanced Activities**

○ **CryptoForge**

- 7.1.1 Definition
- 7.1.2 How to Download CryptoForge
- 7.1.3 Performing Encryption Activity

○ **AxCrypt**

- 7.2.1 Definition
- 7.2.2 How to Download AxCrypt

- 7.2.3 Performing Encryption Activity

○ **SensiGuard**

- 7.3.1 Definition
- 7.3.2 How to Download SensiGuard
- 7.3.3 Performing Encryption Activity

○ **OpenSSL**

- 7.4.1 Definition
- 7.4.2 Performing Activity Using OpenSSL

○ **CrypTool**

- 7.5.1 Definition
- 7.5.2 Main Objectives
- 7.5.3 How to Download CrypTool
- 7.5.4 Performing Activity Using CrypTool

○ **AES Crypt**

- 7.6.1 Definition
- 7.6.2 How to Download AES Crypt
- 7.6.3 Performing Encryption Activity

○ **HashCalc**

- 7.7.1 Definition
- 7.7.2 Key Features
- 7.7.3 How to Download HashCalc
- 7.7.4 Performing Hashing Activity

○ **CyberChef**

- 7.8.1 Definition
- 7.8.2 Key Features
- 7.8.3 Performing Activity Using CyberChef

Cryptography

Cryptography is the practice and study of techniques for **securing communication and data** in the presence of adversaries. It ensures that information is **kept confidential**, **unmodified**, and **authentic** when being stored or transmitted.

📖 **Definition**:
Cryptography is the science of converting **plaintext** into **ciphertext** using mathematical algorithms to protect it from unauthorized access.

🎯 **Objectives of Cryptography (C.I.A.A.N)**

🔒 Objective	✅ Description
1. Confidentiality	Ensures that only authorized individuals can access the data.
2. Integrity	Assures that the information has not been altered during transmission or storage.
3. Authentication	Verifies the identity of the sender or receiver of the information.
4. Authorization	Ensures only permitted users can access specific data or perform certain actions.
5. Non-Repudiation	Prevents parties from denying their actions, such as sending a message or signing.

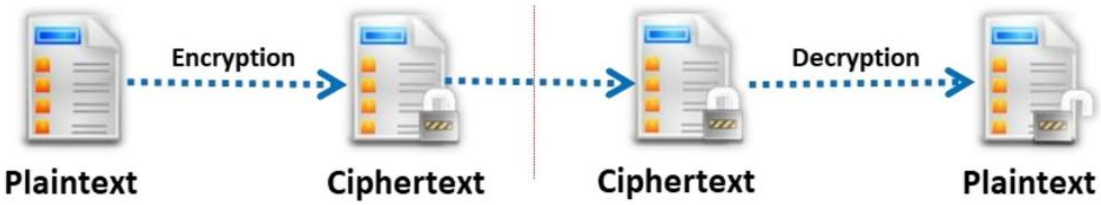
✅ **Simple Example**

Imagine you're sending a message:
Plaintext: I love cryptography
After encryption (AES):
Ciphertext: Y34Dd92@kLmXZ3qWs81!

Only someone with the correct **key** can turn it back into the original message.

🔄 Cryptography Process:

Cryptography follows a well-defined **process** to convert readable data (plaintext) into an unreadable format (ciphertext) and then back to readable format using encryption and decryption.



Example Of Cryptography

🔐 Types of Cryptography

Cryptography is broadly divided into **four main types**, each serving different purposes based on how keys are used and data is protected.

🧱 1. Symmetric Key Cryptography (Secret Key Cryptography)

🔸 Description:

- The **same key** is used for both encryption and decryption.
- It's **faster** but less secure for large-scale communication (because key sharing is risky).

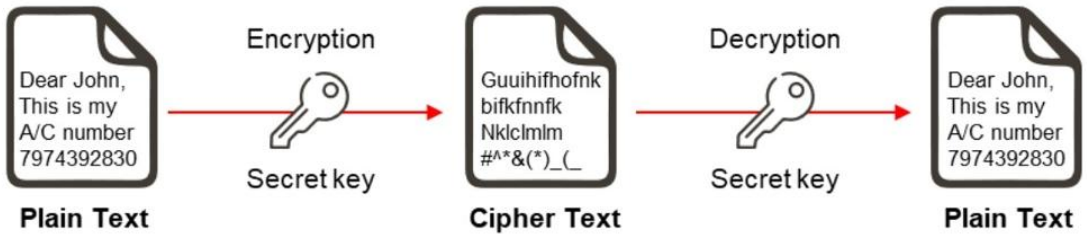
🔐 Examples:

- **AES (Advanced Encryption Standard)**
- **DES (Data Encryption Standard)**
- **RC4, RC5, RC6**
- **Blowfish, Twofish**

✅ Use Cases:

- File encryption
- Disk encryption (e.g., BitLocker, VeraCrypt)

- Secure VPN tunnels (IPSec)



Symmetric Key Encryption

🔐 2. Asymmetric Key Cryptography (Public Key Cryptography)

🔸 Description:

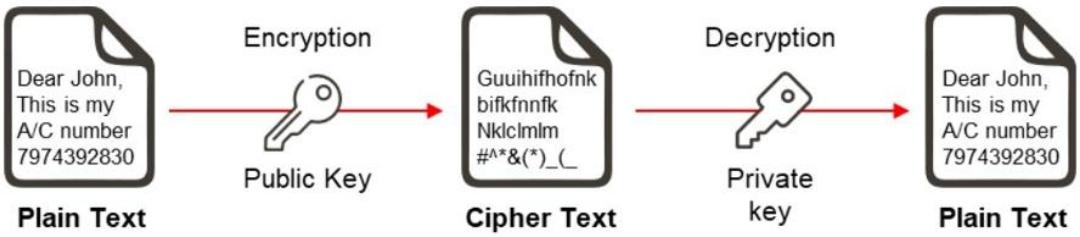
- Uses **two keys**: a **public key** for encryption and a **private key** for decryption.
- It's **secure**, but **slower** than symmetric encryption.

🔐 Examples:

- **RSA (Rivest–Shamir–Adleman)**
- **ECC (Elliptic Curve Cryptography)**
- **Diffie-Hellman** (for key exchange)
- **ElGamal**

✅ Use Cases:

- Digital signatures
- Email security (PGP)
- SSL/TLS for secure websites



Asymmetric Key Encryption

🔢 3. Hash Functions (One-Way Cryptography)

🔸 Description:

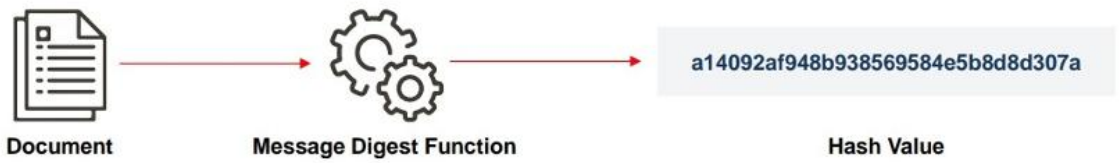
- Converts data into a **fixed-length hash value**.
- It's **non-reversible** (cannot get the original data back from the hash).
- Mainly used for **integrity checking** and **password storage**.

🔐 Examples:

- **MD5** (deprecated due to collision vulnerabilities)
- **SHA-1** (deprecated)
- **SHA-256, SHA-512 (SHA-2 family)**
- **SHA-3**

✅ Use Cases:

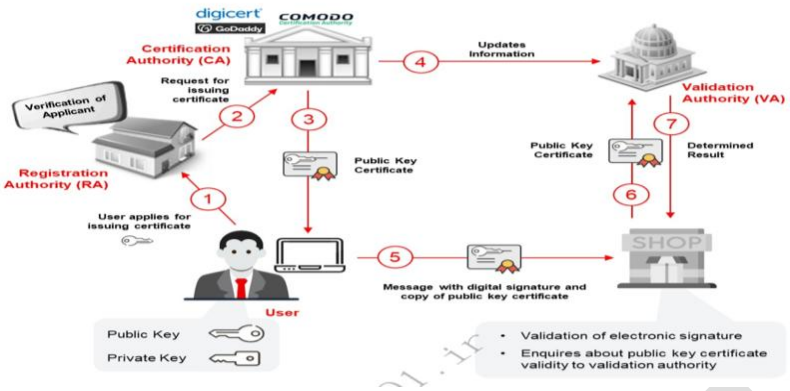
- Password storage (hashed & salted)
- File integrity verification
- Blockchain (mining & transactions)



One Way Hash Function

🔐 Public Key Infrastructure

PKI (Public Key Infrastructure) is a **framework of policies, procedures, hardware, and software** used to **secure digital communication** using **public key cryptography**. It enables users and systems to securely exchange information over untrusted networks (like the internet).



Public Key Infrastructure

🎯 Main Purpose of PKI

To manage **digital certificates** and **public/private key pairs** to ensure:

- **Authentication** (verifying identity)
- **Confidentiality** (encrypting communication)
- **Integrity** (ensuring data is unaltered)
- **Non-repudiation** (proof of origin)

🧱 Core Components of PKI

Component	Description
Certificate Authority (CA)	Issues and signs digital certificates (e.g., DigiCert, Let's Encrypt)
Registration Authority (RA)	Verifies user identity on behalf of the CA
Digital Certificates (X.509)	File that binds a public key with an identity
Public & Private Keys	Used for encryption, decryption, and signing

Component	Description
Certificate Revocation List (CRL)	List of revoked certificates
OCSP (Online Certificate Status Protocol)	Checks certificate status in real-time

---

🔑 How PKI Works (Process Flow)

📩 1. User Requests Certificate

- A user generates a key pair (public/private).
- Submits a **CSR (Certificate Signing Request)** to the CA.

🏛️ 2. CA Verifies Identity

- The CA or RA validates the requester's identity.

📜 3. CA Issues Certificate

- The CA signs the user's public key with its **private key**.
- The user receives a **digital certificate (X.509)**.

🔐 4. Secure Communication

- Others use the **public certificate** to encrypt data or verify digital signatures.
- The recipient uses their **private key** to decrypt or sign data.

📜 Example: HTTPS (SSL/TLS) Using PKI

- When you visit https://example.com, your browser checks the site's **digital certificate**.
- The certificate is issued by a **trusted CA**.
- If valid, a **secure TLS connection** is established using asymmetric encryption first, then symmetric for speed.

🛡️ Benefits of PKI

- ✅ Secure websites (HTTPS)
- ✅ Secure emails (S/MIME)
- ✅ VPNs & Wi-Fi access control
- ✅ Digital signatures for documents (PDF, MS Word)
- ✅ Authentication in Zero Trust networks

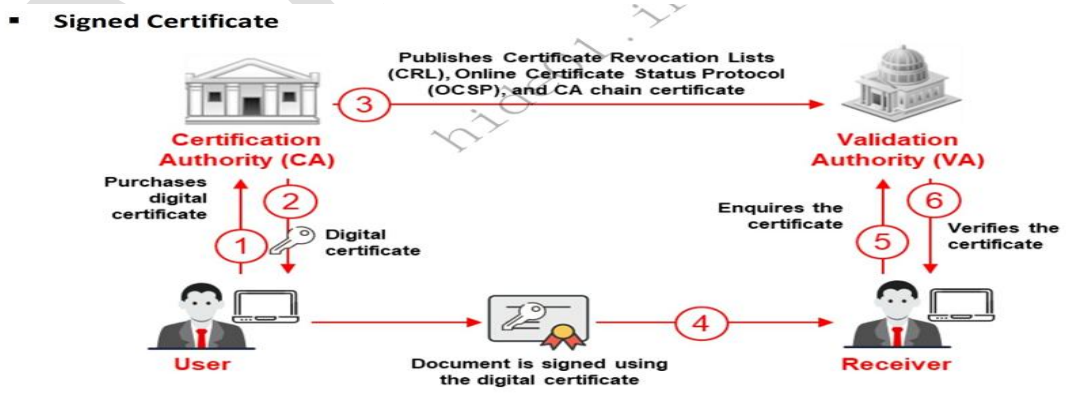
📜 What is a Signed Certificate vs. a Self-Signed Certificate?

Both types of certificates are used to **secure digital communications** (like HTTPS), but they differ in **who signs** them and how **trust** is established.

✅ 1. Signed Certificate (CA-Signed Certificate)

🔐 Definition:

A **signed certificate** is a digital certificate that has been **issued and signed by a trusted Certificate Authority (CA)**, such as DigiCert, Let's Encrypt, or GoDaddy.



Signed Certificate

📜 Key Features:

Feature	Description
Signed By	Certificate Authority (CA)
Trusted?	Yes, browsers and OS trust major CAs by default
Verification	CA verifies the identity of the organization/domain
Use Cases	HTTPS websites, VPNs, email security, code signing
Cost	Usually **paid** (except for Let's Encrypt, which is free)

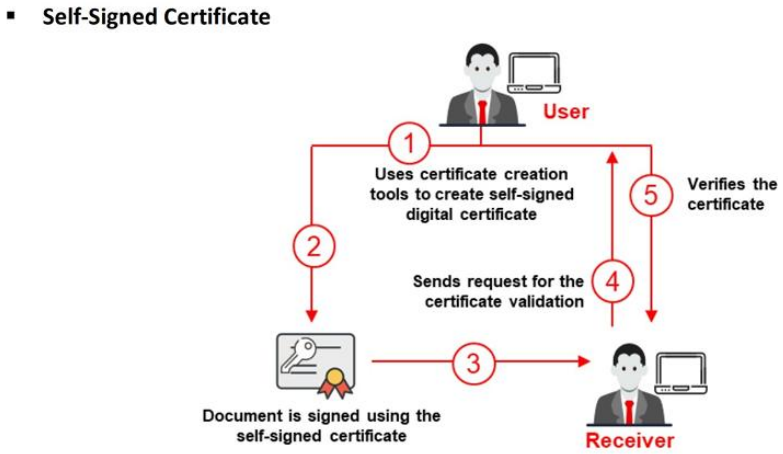
🛡️ Why it's trusted?

- Your browser already has a list of trusted CAs.
- If a site uses a CA-signed certificate, browsers show a **secure lock** icon.

❌ 2. Self-Signed Certificate

🔐 Definition:

A **self-signed certificate** is signed **by the same entity** that owns it — not by a third-party CA.



Self-Signed Certificate

📜 Key Features:

Feature	Description
Signed By	The certificate owner itself
Trusted?	❌ **Not trusted by default** by browsers or OS
Verification	No external verification or identity check
Use Cases	Internal testing, lab environments, dev servers
Cost	Free

Tra

🖋️ Digital Signature

A **digital signature** is a **mathematical technique** used to validate the **authenticity**, **integrity**, and **origin** of a message, software, or digital document.

It acts like a **virtual fingerprint** — unique to both the document and the signer — and is widely used in **cybersecurity**, **secure email**, **e-documents**, and **blockchain**.

🎯 Objectives of Digital Signature

Objective	Description
Authentication	Confirms who signed the data (proves identity).
Integrity	Ensures data hasn't been altered after signing.
Non-Repudiation	The signer cannot deny having signed the message/document.

🔐 How Digital Signature Works

🔁 Digital Signature Creation Process

1. The sender creates a **hash** of the original message (e.g., using SHA-256).
2. The hash is **encrypted with the sender's private key**.
3. The encrypted hash is attached to the original message as the **digital signature**.

📩 Verification Process

1. The receiver decrypts the digital signature using the **sender's public key** to get the original hash.
2. The receiver hashes the received message again.
3. If both hashes match ✅, the message is **authentic and unmodified**.

📜 Simple Example

1. 🔑 **Sender**:
   - Message: Hello Aniket
   - Hash → Encrypted with **private key** = Digital Signature
2. 📩 **Receiver**:
   - Decrypts signature with **public key**
   - Compares hashes
   - ✅ If equal: Signature is valid

🧠 Key Concepts

Concept	Meaning
Hash Function	Creates a fixed-length digest from a message
Private Key	Used to **sign** the message
Public Key	Used to **verify** the signature

Concept	Meaning
Digital Certificate	Binds identity to public key (via PKI)

🛠️ Digital Signature Algorithms

Algorithm	Description
RSA	Most common public-key algorithm
DSA	Digital Signature Algorithm by NIST
ECDSA	Based on Elliptic Curve Cryptography
EdDSA	Modern fast signature scheme

📚 Use Cases

Area	Example
Email Security	S/MIME signed emails
Software Publishing	Code signing to prevent tampering
PDF/Word Docs	Signed contracts & legal agreements
Blockchain	Transaction signing in Bitcoin, Ethereum
Web Security	TLS certificates are digitally signed

🔍 Digital Signature vs Electronic Signature

Feature	Digital Signature	Electronic Signature
Based on	Cryptography (PKI)	Image/Text/Click
Legally Recognized	✅ Yes (under IT laws, e.g., IT Act India)	✅ Yes (in many countries)

Feature	Digital Signature	Electronic Signature
Security	Very High	Depends on implementation
Verification Method	Mathematical/Public Key	Depends (OTP, email, etc.)

🔐 Secure Socket Layer (SSL)

SSL (Secure Socket Layer) is a **cryptographic protocol** designed to **secure communication over the internet**, especially between **web browsers and servers**.

📌 Today, SSL has been replaced by a more secure version called **TLS (Transport Layer Security)**, but people still commonly refer to it as "SSL."

🎯 Purpose of SSL

Objective	Description
Confidentiality	Encrypts data during transmission to protect against eavesdropping
Authentication	Verifies the identity of the website (via certificates)
Integrity	Ensures data hasn't been altered in transit

📶 How SSL Works (Simplified)

🔁 SSL Handshake Process

1. **Client Hello**
    - Browser sends: Supported SSL/TLS versions, cipher suites, random data.
2. **Server Hello**

- Server responds: Chosen cipher, SSL certificate (includes public key).

3. **Certificate Verification**
   - Client verifies the server's SSL certificate (via CA trust).
4. **Session Key Generation**
   - Both generate a **shared symmetric session key** using public-private key exchange (like RSA or Diffie-Hellman).
5. **Secure Communication Begins**
   - All data is now encrypted using that session key.

🔐 SSL Certificate

An **SSL certificate** is a digital certificate issued by a **Certificate Authority (CA)** that proves a website's identity and contains its **public key**.

Field in Certificate	Example
Common Name (CN)	www.example.com
Issuer	Let's Encrypt Authority X3
Valid From / To	2025-01-01 to 2026-01-01
Public Key	Used during key exchange
Signature	Signed by CA

📜 SSL vs TLS

Feature	SSL (Deprecated)	TLS (Current Standard)
Version	SSL 2.0 / 3.0	TLS 1.0 → TLS 1.3
Security	Outdated & weak	Modern and secure
Use Today?	❌ No	✅ Yes

Feature	SSL (Deprecated)	TLS (Current Standard)
Supported By	None (mostly disabled)	All modern browsers

✅ **TLS 1.3** is the latest version (faster and more secure).

🌐 Real-World Examples of SSL/TLS Use

Application	Role of SSL/TLS
HTTPS Websites	Encrypts browser-server traffic
Email Servers (SMTP/IMAP)	Secures email transmission
VPN Connections	Ensures encrypted communication
Online Banking	Secures financial transactions
VoIP / Messaging	Protects voice/video/text traffic

🔍 How to Identify SSL on Websites

- URL starts with https://
- Padlock 🔒 icon in the browser address bar
- Click it to view certificate details

🔐 Transport Layer Security (TLS)

TLS (Transport Layer Security) is the **successor to SSL** and is the **most widely used cryptographic protocol** today for securing data over the internet. It ensures **privacy, integrity, and authentication** between communicating applications (e.g., web browsers and servers).

📌 What TLS Does

Function	Description
Encryption	Prevents eavesdropping by encrypting data in transit
Integrity	Ensures that the data is not modified or tampered with
Authentication	Confirms the identity of the communicating parties

---

🔁 How TLS Works (TLS Handshake)

📜 TLS Handshake Steps (Simplified):

1. **Client Hello**
    - Browser sends supported TLS versions, cipher suites, and a random number.
2. **Server Hello**
    - Server selects the TLS version, cipher suite, and sends its digital certificate (with public key).
3. **Certificate Verification**
    - Client checks the certificate's validity and CA signature.
4. **Key Exchange**
    - Both client and server agree on a **shared secret key** using methods like **Diffie-Hellman** or **ECDHE**.
5. **Session Encryption Starts**
    - All further communication is **encrypted** using a **symmetric key**.

✅ This hybrid use of **asymmetric encryption** (for handshake) and **symmetric encryption** (for data exchange) balances **security and performance**.

🔄 TLS Versions Overview

Version	Status	Key Features
TLS 1.0	Deprecated	Initial version, now insecure
TLS 1.1	Deprecated	Minor improvements, also insecure
TLS 1.2	Widely used	Secure, supports modern ciphers
TLS 1.3	Latest (2024+)	Faster, more secure, removes old features

✅ **TLS 1.3** removes support for outdated algorithms like RSA key exchange and focuses on forward secrecy.

🔐 TLS vs SSL

Feature	SSL (Deprecated)	TLS (Current)
Security	Weak	Strong (TLS 1.2 / 1.3)
Versions	SSL 2.0, 3.0	TLS 1.0 → 1.3
Use Today?	❌ No	✅ Yes

🌐 Where TLS is Used

Application	Role of TLS
HTTPS (Web)	Encrypts web traffic
Email (SMTP/IMAP/POP)	Secures email transfer
VoIP	Secures voice communication
VPN	Encrypts network tunnels (e.g., OpenVPN)
Messaging	Secures messages (e.g., WhatsApp uses TLS)

🧠 Cryptanalysis Methods

Cryptanalysis is the art and science of **breaking cryptographic systems**—analyzing ciphertexts, algorithms, and keys to recover the original data **without access to the secret key**.

It's essentially the **opposite of cryptography** — while cryptographers build secure systems, cryptanalysts attempt to break them.

🔍 Types of Cryptanalysis Methods

🔢 Method	🔎 Description
1. Brute Force Attack	Tries **every possible key** until the correct one is found. Time-consuming.
2. Dictionary Attack	Uses a **precompiled list of possible keys or passwords**.
3. Frequency Analysis	Analyzes how often certain characters/words appear. Effective on **substitution ciphers**.
4. Known-Plaintext Attack (KPA)	Attacker has access to **plaintext and corresponding ciphertext**. Tries to find the key.
5. Chosen-Plaintext Attack (CPA)	Attacker can **choose plaintexts** and get their ciphertexts. Useful for breaking symmetric encryption.
6. Ciphertext-Only Attack (COA)	Only ciphertext is available. Tries to guess the key or plaintext.
7. Chosen-Ciphertext Attack (CCA)	Attacker can **choose ciphertexts** and get the decrypted plaintexts. Often targets asymmetric cryptosystems.
8. Side-Channel Attack	Exploits **physical characteristics** like timing, power usage, or electromagnetic leaks.

🔢 Method	🔎 Description
9. Differential Cryptanalysis	Analyzes **how differences in input affect output** in block ciphers.
10. Linear Cryptanalysis	Uses linear approximations to describe the behavior of the block cipher.
11. Man-in-the-Middle Attack (MITM)	Intercepts communication to **read or alter** data in real-time.
12. Algebraic Attack	Solves the mathematical equations behind a cipher (used against ECC and AES).
13. Replay Attack	Captures encrypted messages and **resends them** to trick a system.
14. Birthday Attack	Targets hash functions to find **collisions**.

📌 Classifying Cryptanalysis by Attack Type

Type of Attack	Example
Classical	Frequency analysis, Caesar cipher cracking
Modern	Differential, Linear, Algebraic attacks
Physical (Side-Channel)	Timing attacks, power analysis
Adaptive	Chosen plaintext/ciphertext

🧪 Common Targets of Cryptanalysis

- **Weak encryption algorithms** (e.g., DES, RC4, MD5)
- **Poor key management**
- **Predictable keys/passwords**

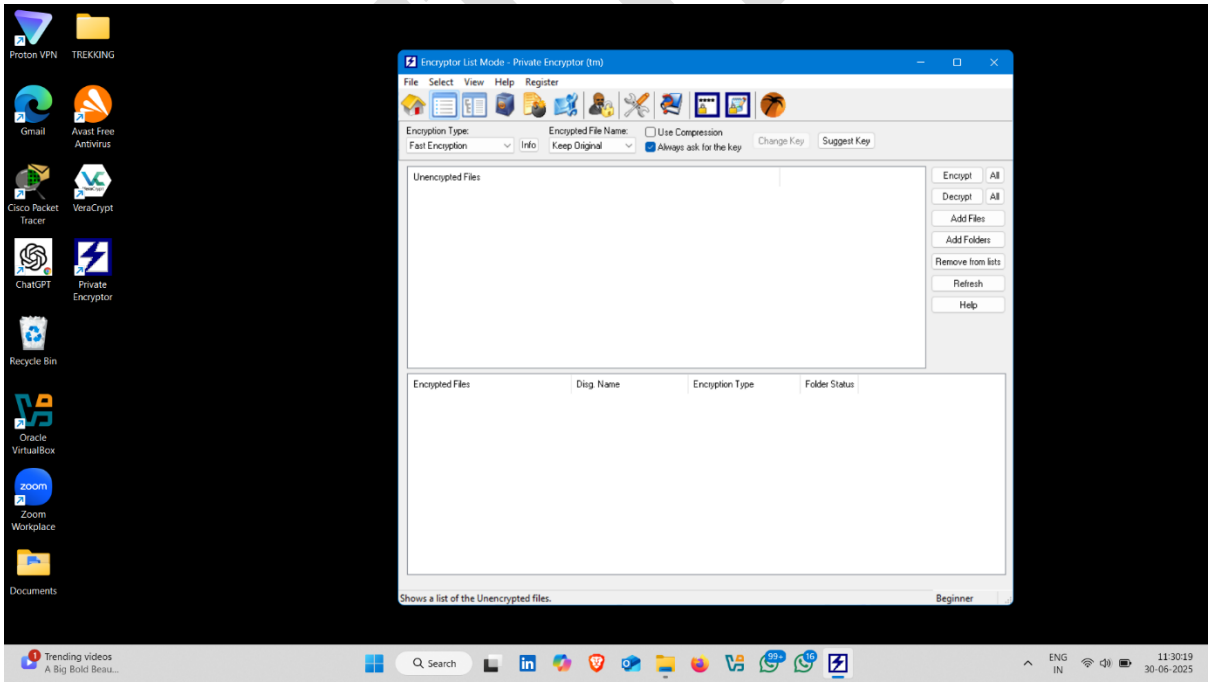
- **Improper implementation of secure algorithms**

DataLock Using Private Encryption Application

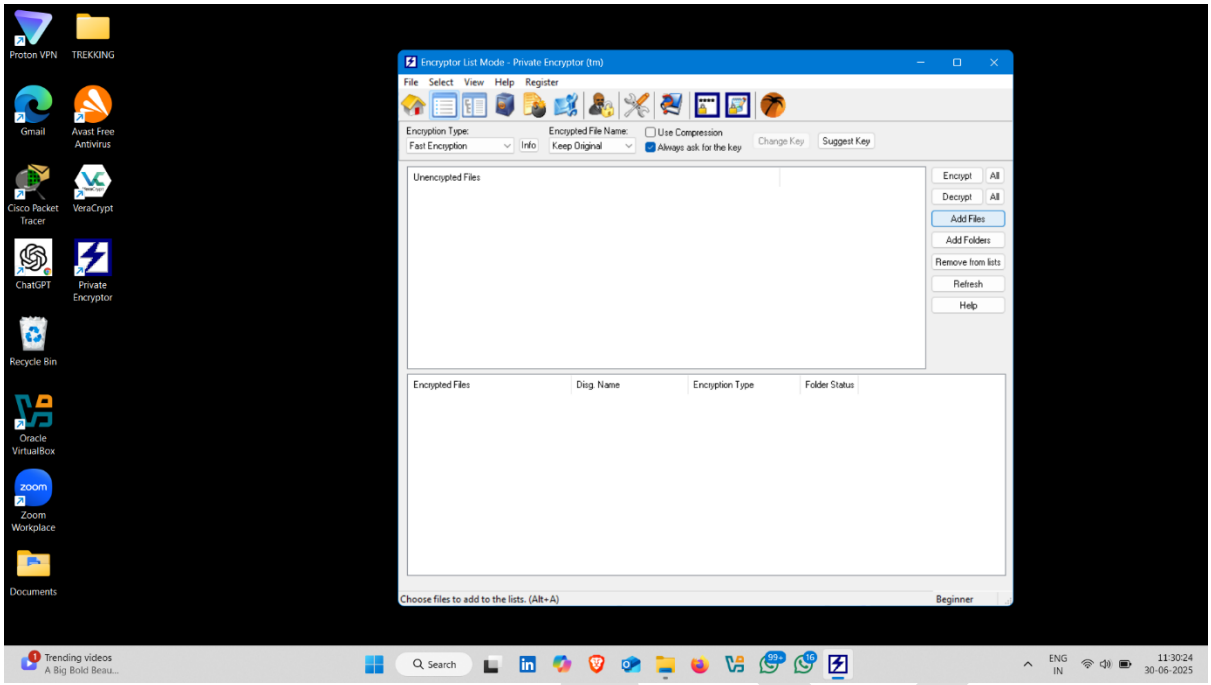
Private Encryptor is a file encryption application designed to protect sensitive or personal data by converting readable information into an unreadable format using strong cryptographic algorithms (such as AES, RSA, etc.). Only users with the correct password or private key can decrypt and access the original content. It ensures confidentiality, prevents unauthorized access, and is typically used for securing documents, folders, and personal files on a local machine.

How to use it -:

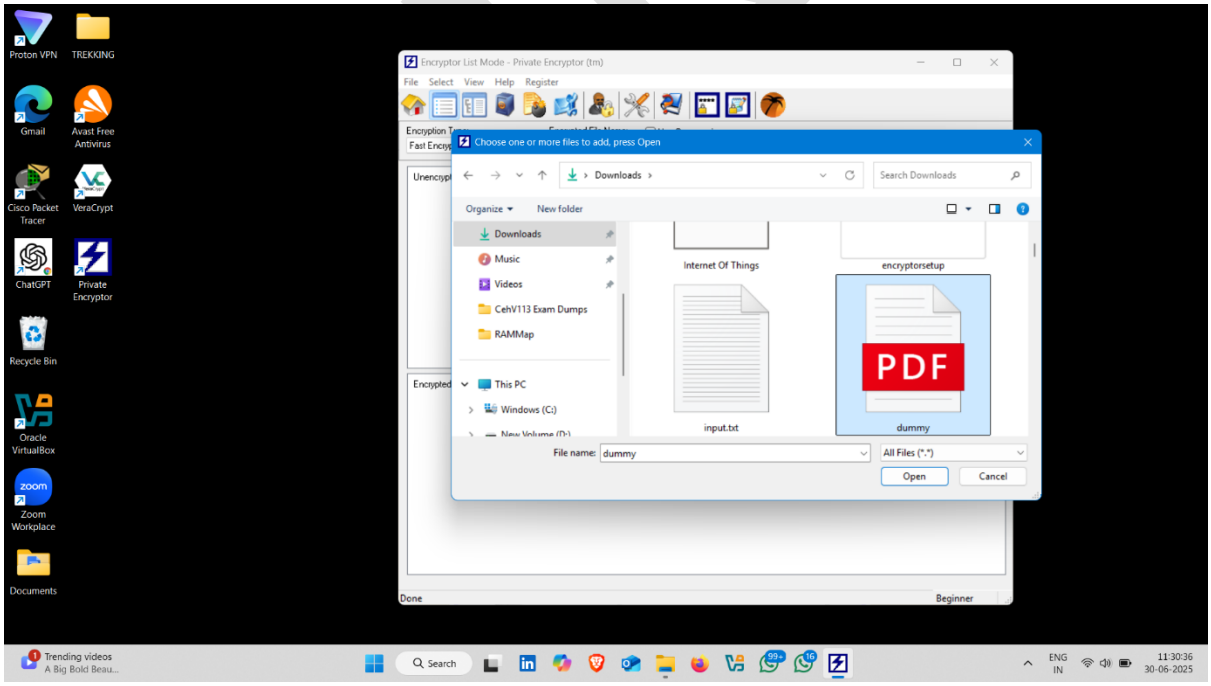
- Private encryption interface



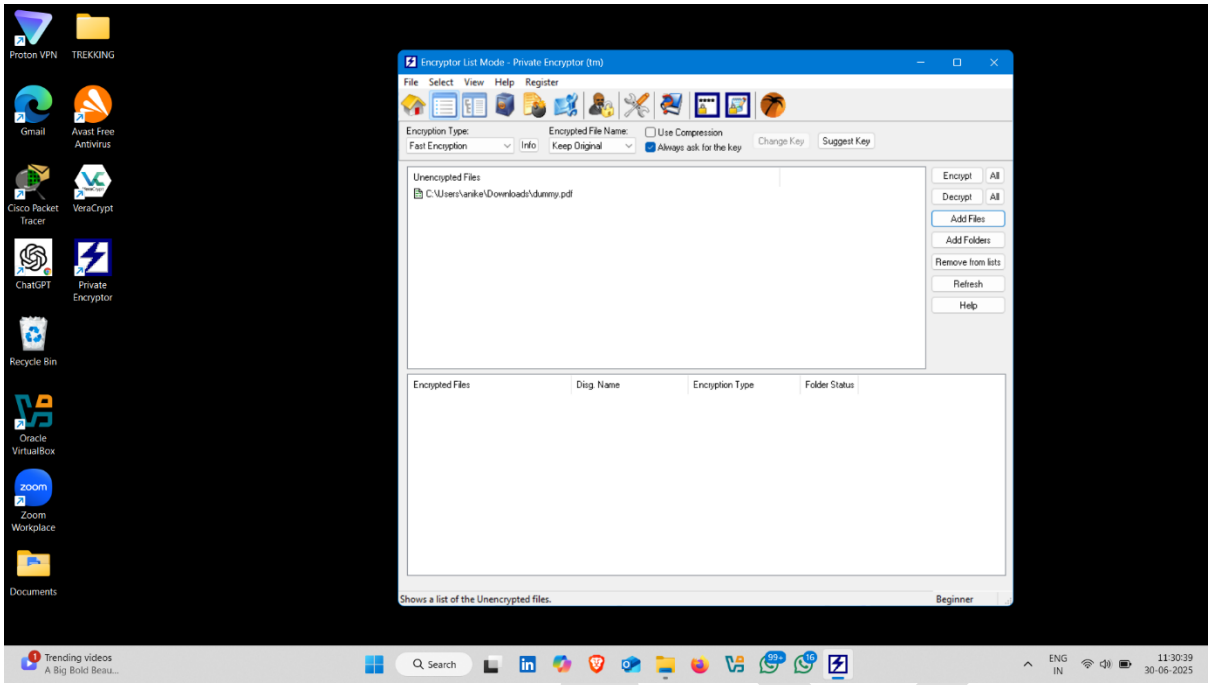
- click on **Add files**



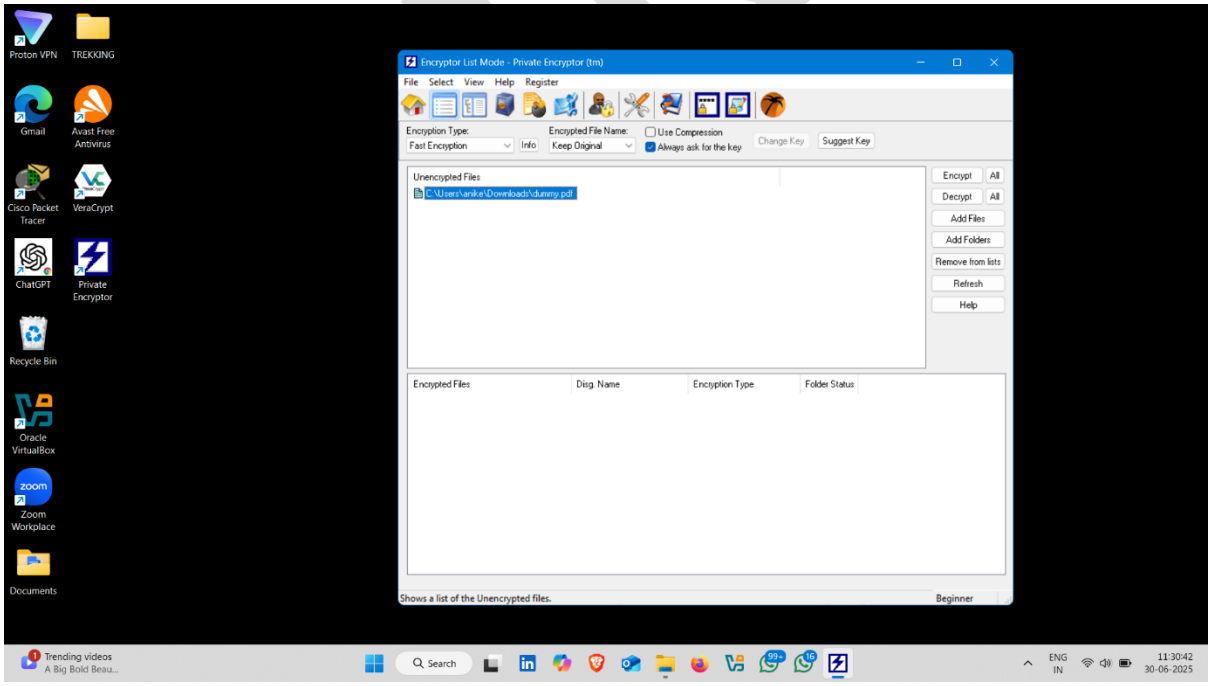
- select **file** and click on **open**



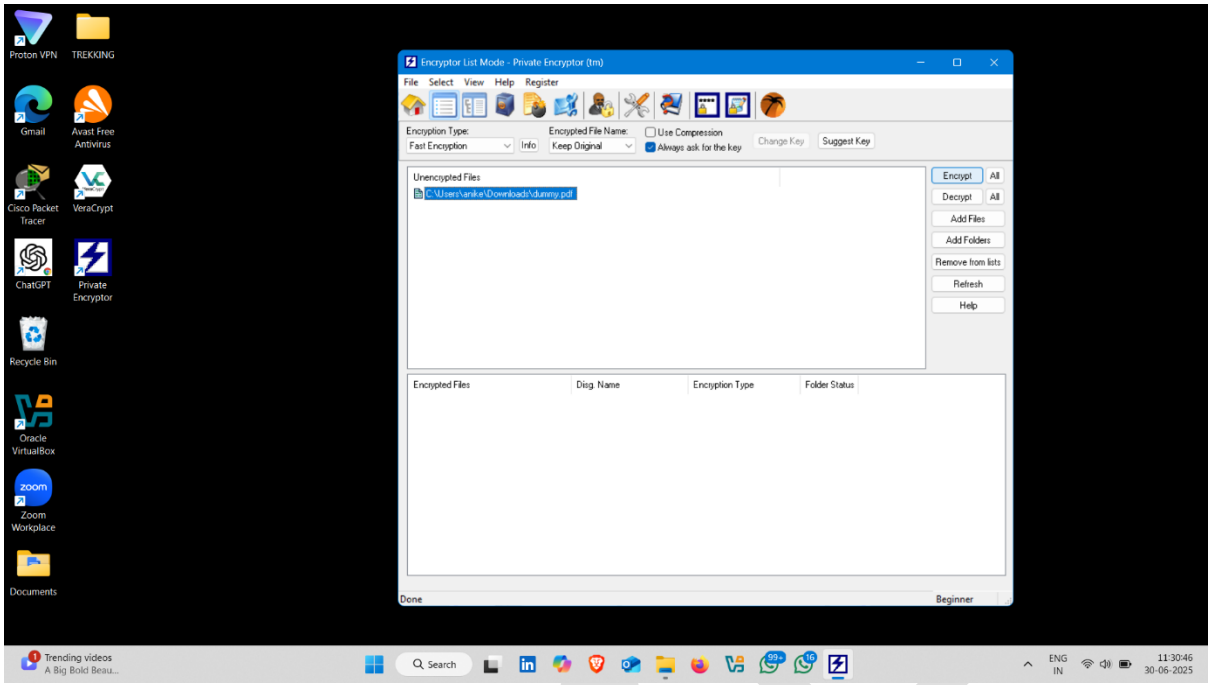
- file added **successfully**



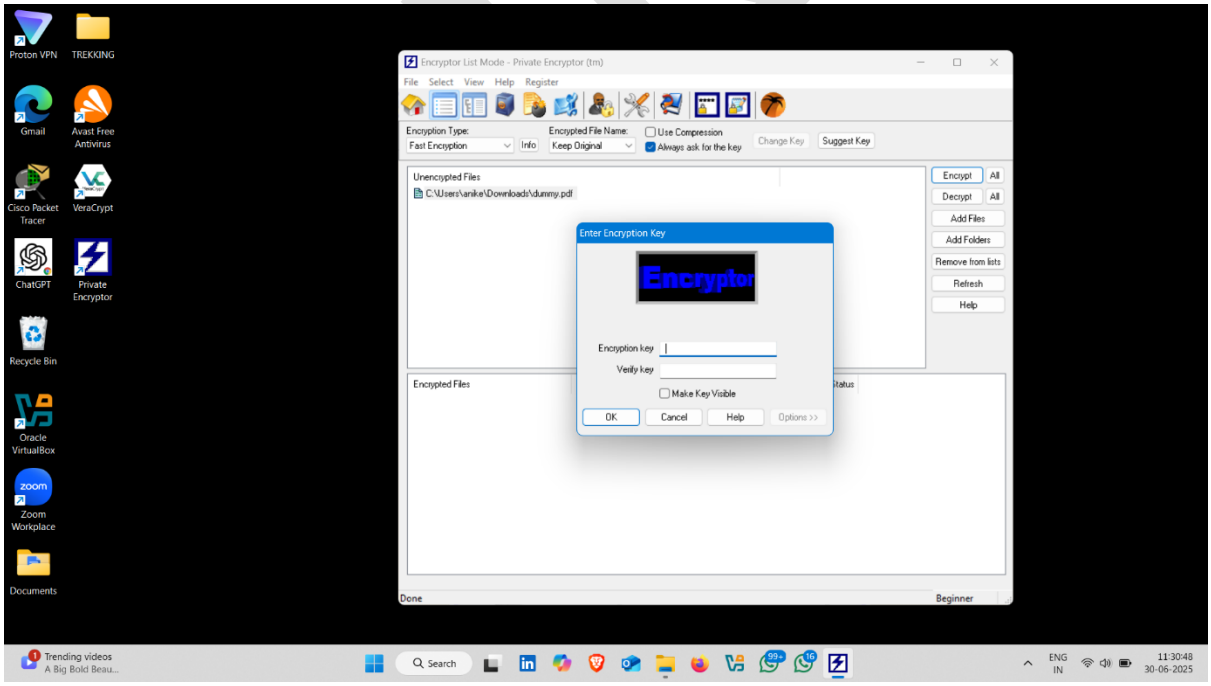
- click on file



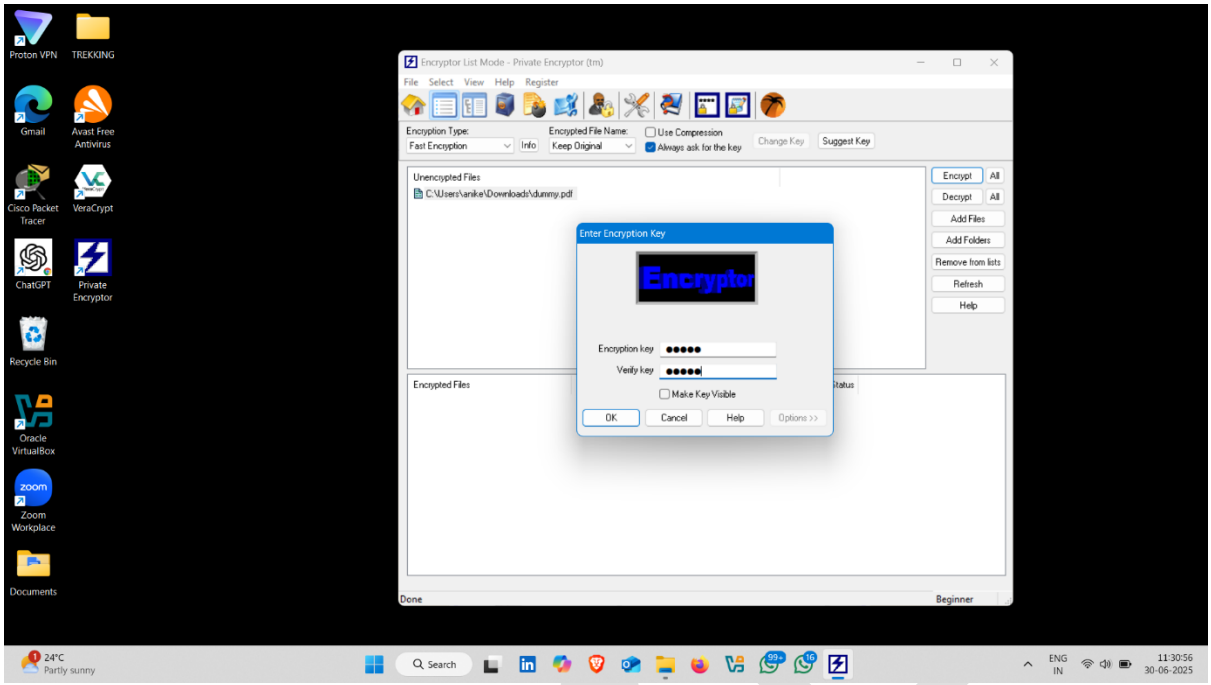
- and then click on **encrypt**



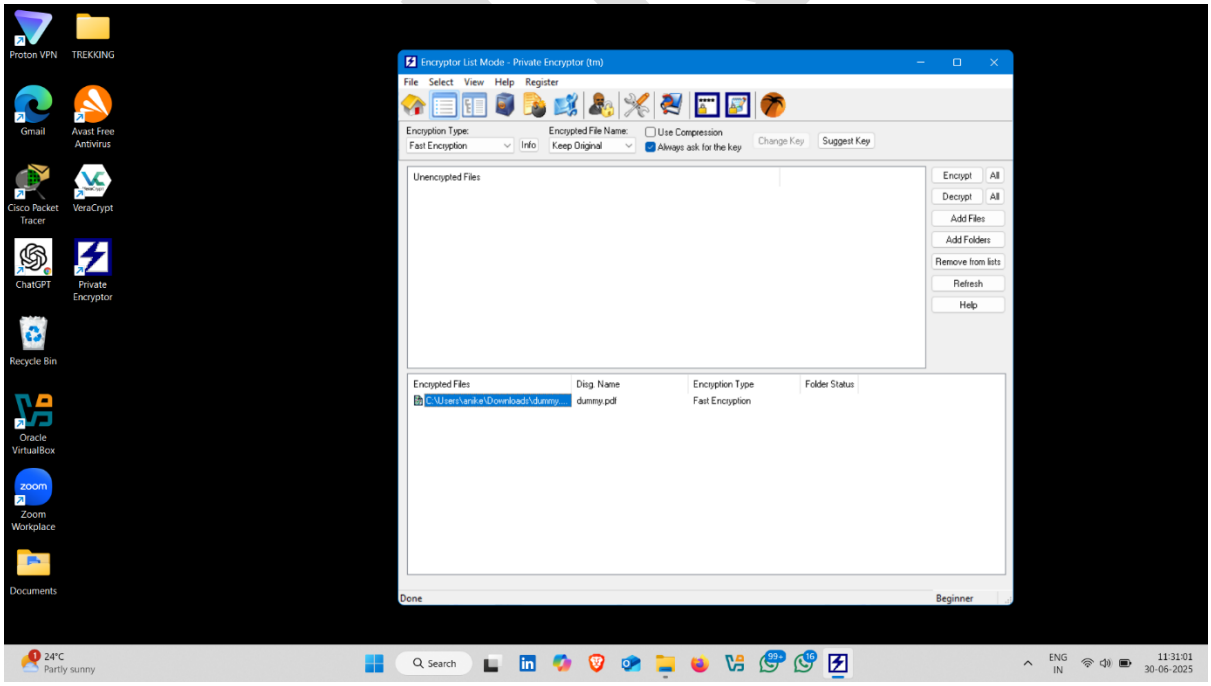
- set strong password for file



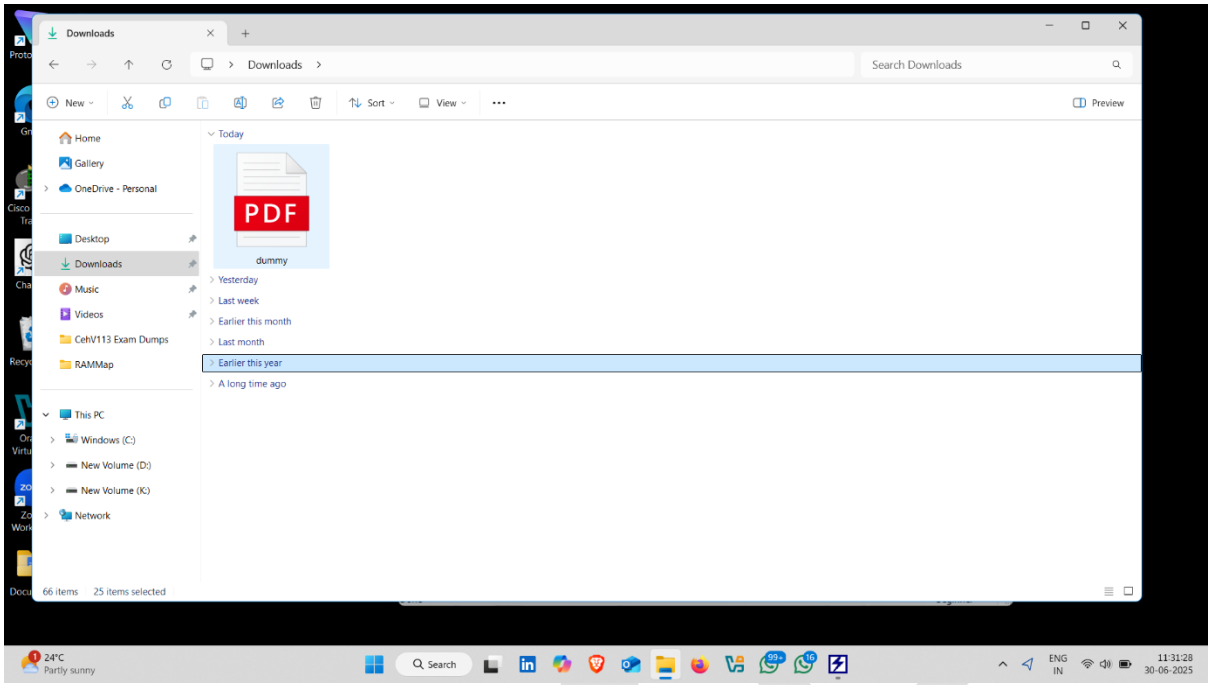
- click on **ok**



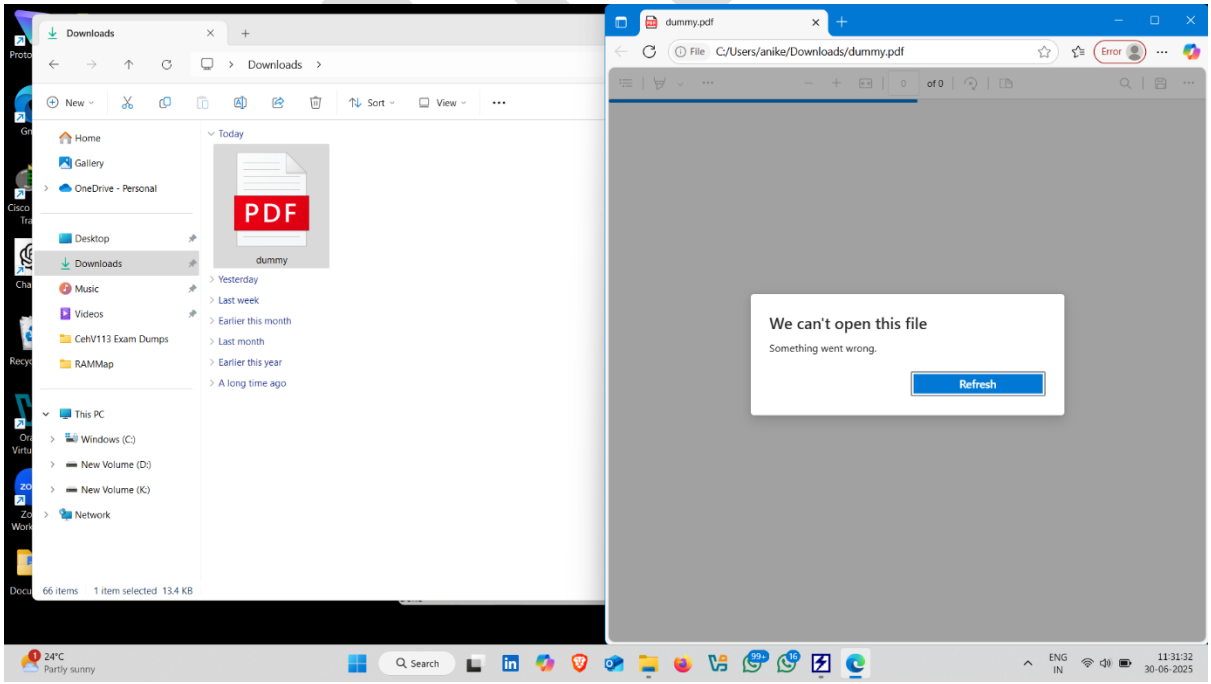
- File encrypted



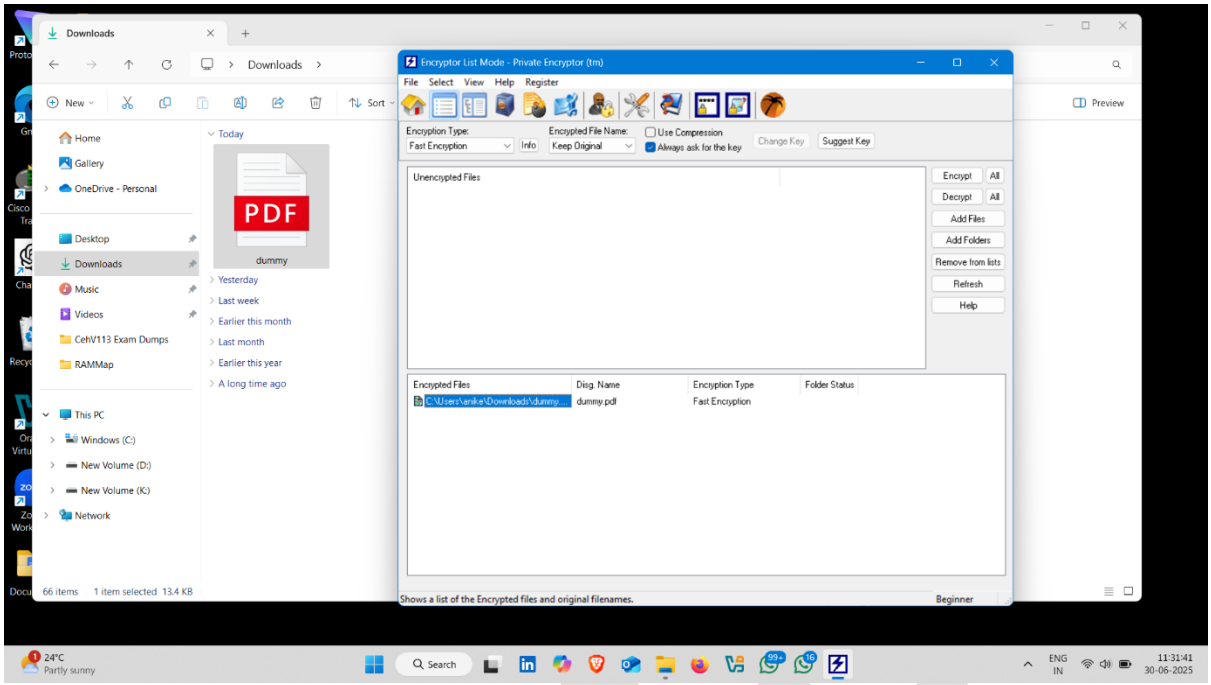
- Now open file to check its encrypted or not



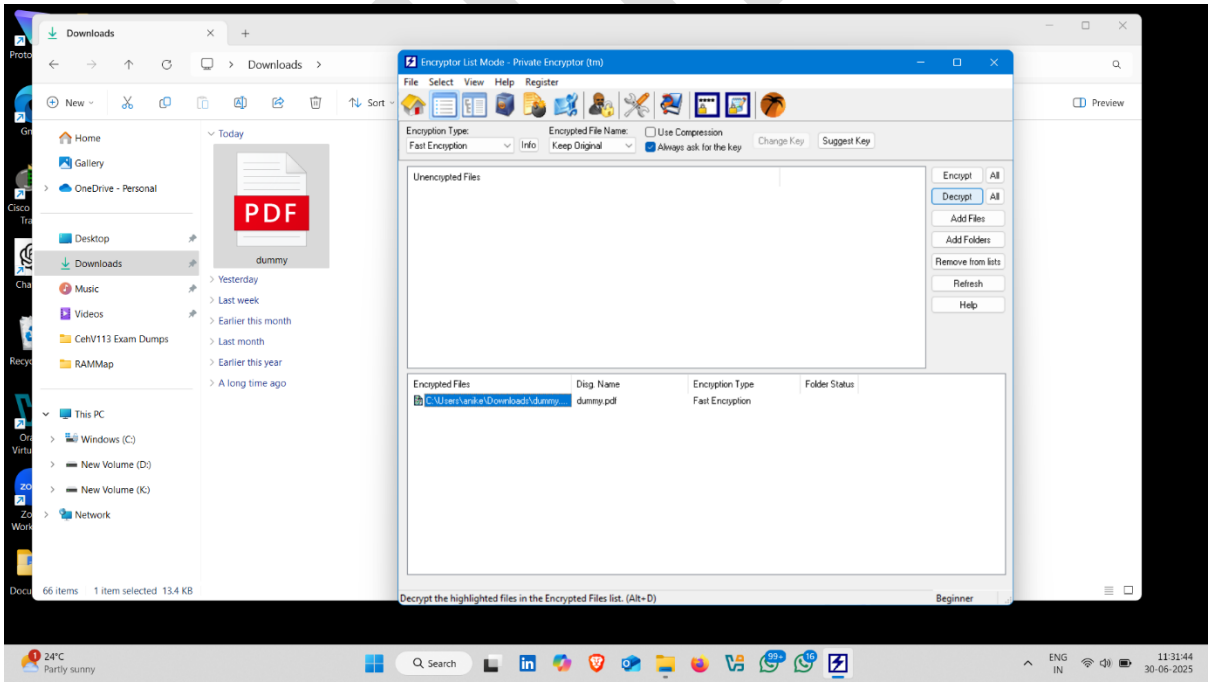
- File encrypted 👇—the main thing of this application,it not show file is encrypted it show something else



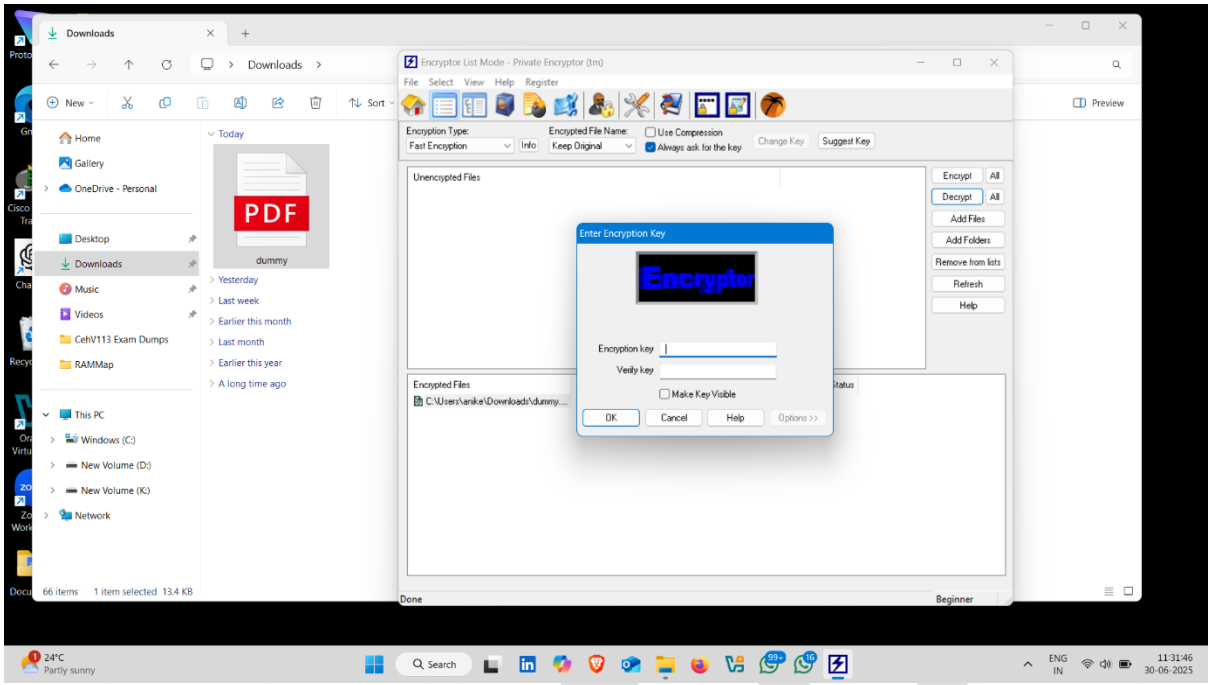
- Click on fille



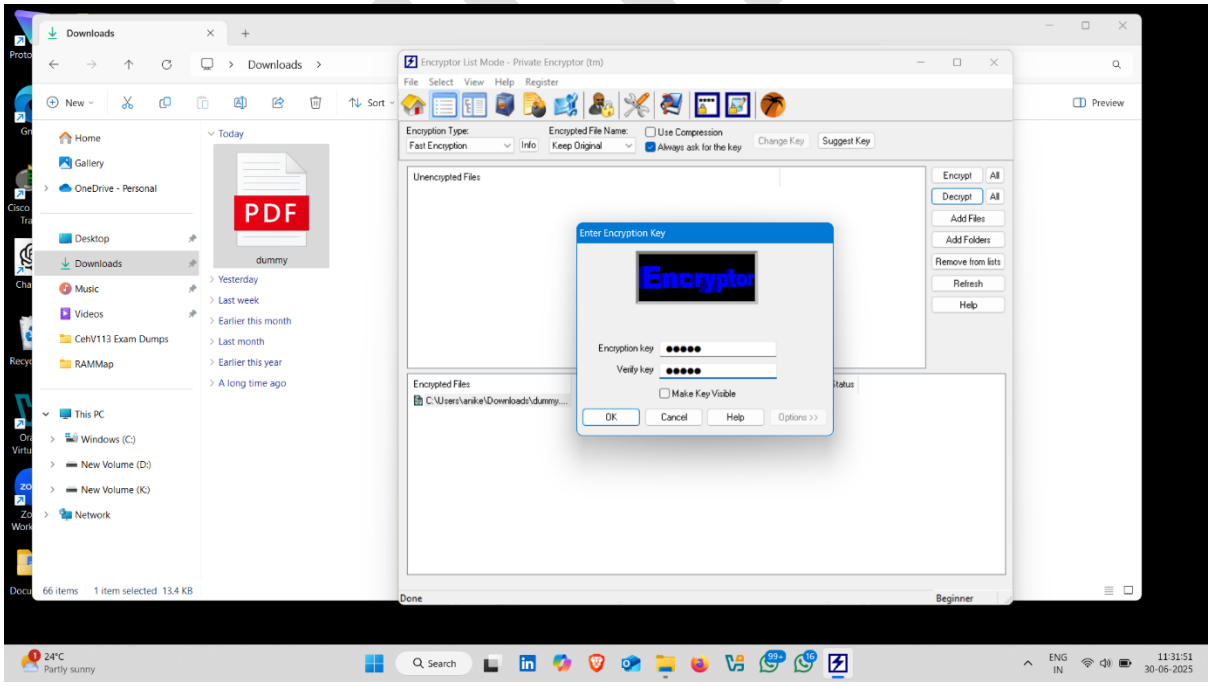
- Click on **Decrypt**



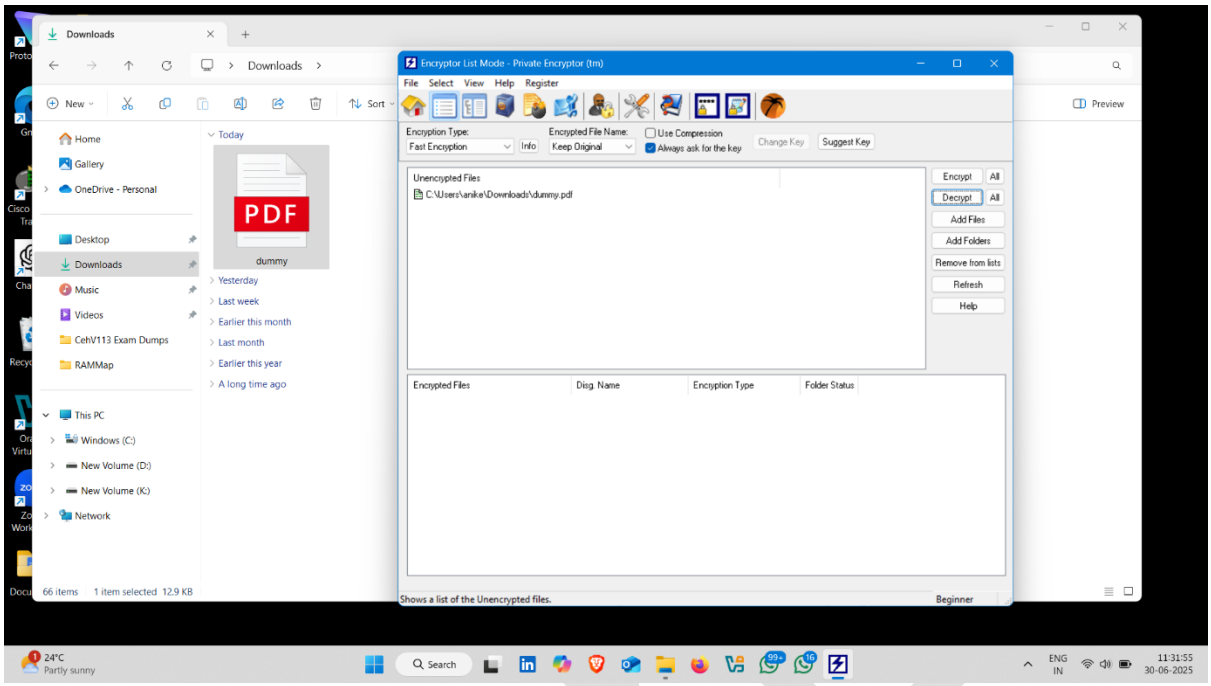
- Enter a password that you set during encryption



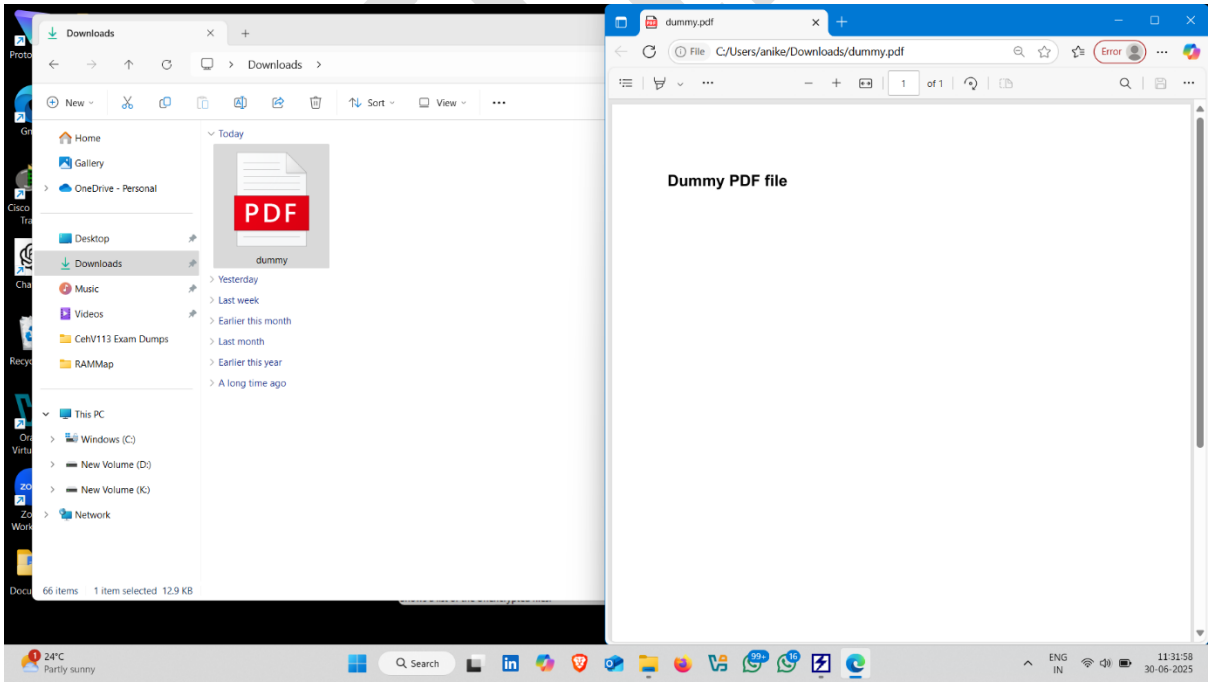
- Click on **ok**



- File **decrypted** 👇



- **Result** 👇✅



Confidential Data Encryption with VeraCrypt on Windows

VeraCrypt is a **free, open-source disk encryption software** used to **secure data** by encrypting entire storage devices, partitions, or files. It is a successor to the discontinued TrueCrypt project, offering stronger security algorithms and enhancements.

🔍 Key Features:

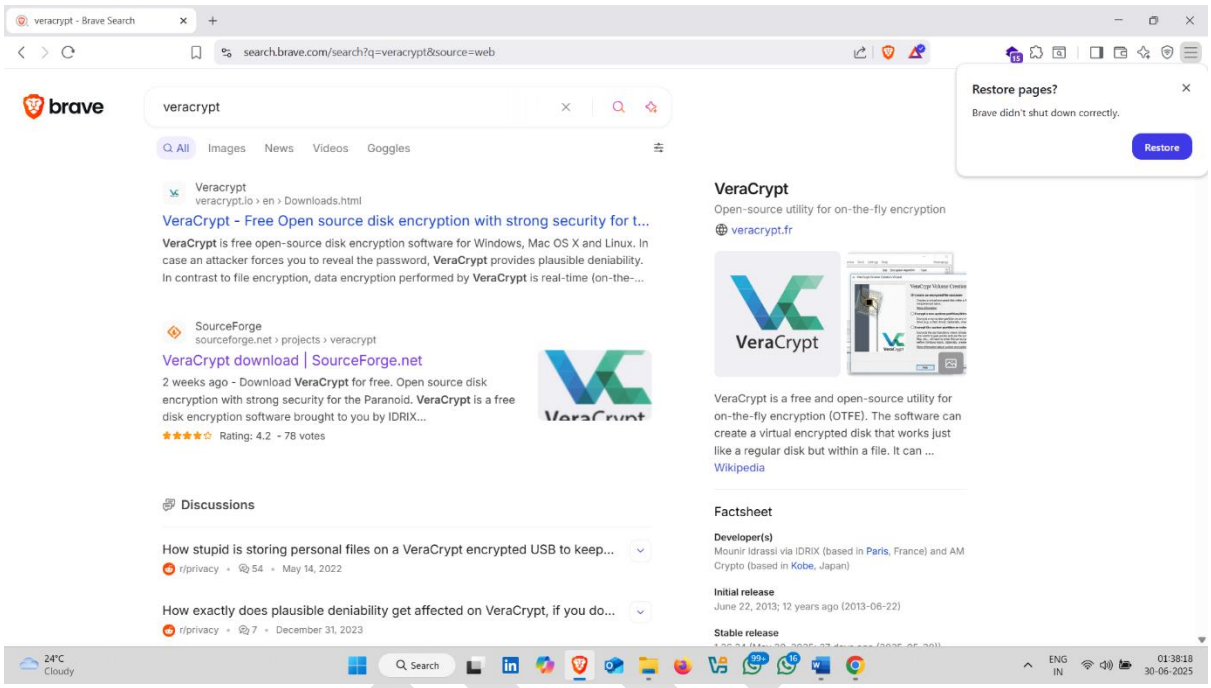
- ✅ **Strong Encryption Algorithms**
  Uses AES, Serpent, Twofish, and combinations (e.g., AES+Twofish+Serpent).
- 🔐 **On-the-Fly Encryption**
  Data is automatically encrypted and decrypted as it's read or written.
- 🧱 **Create Encrypted Volumes**
  You can create a **virtual encrypted disk** that appears as a real disk.
- 📁 **Hidden Volumes**
  Helps prevent detection through **plausible deniability**.
- 💾 **Encrypt Partitions or Entire Drives**
  Supports encrypting internal/external drives and USBs.
- 💼 **Portable and Cross-Platform**
  Works on Windows, macOS (experimental), and Linux.

🎯 Use Cases:

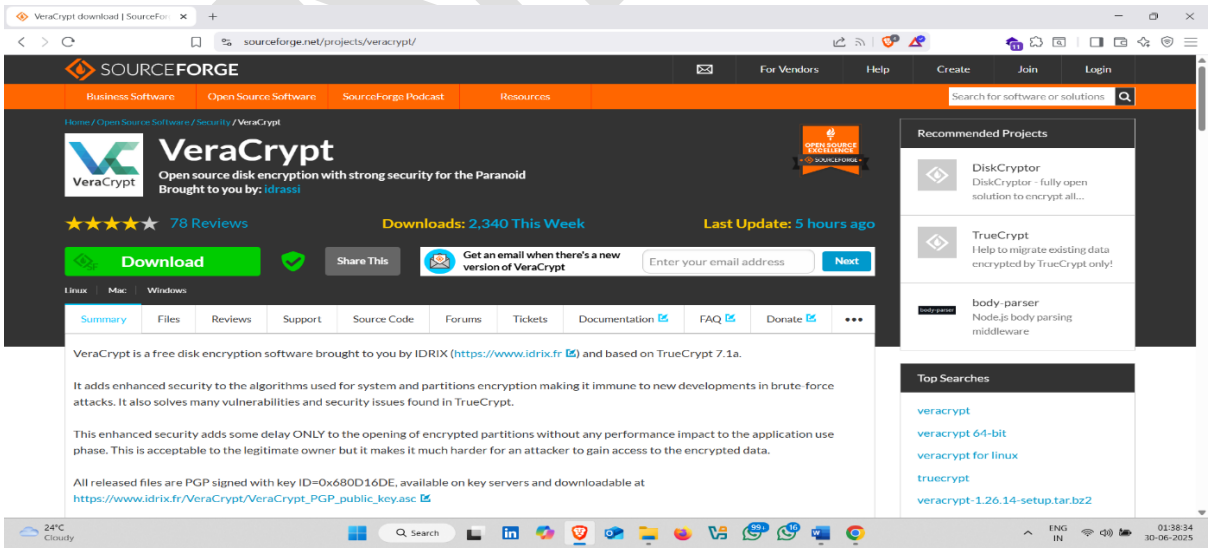
- Securing personal files or backups
- Encrypting USB drives or portable media
- Protecting sensitive data in case of device theft or loss
- Setting up hidden, deniable storage

How to download it -:

- Open Browser and search **Veracrypt**
- Click on second website – **Source Forge**

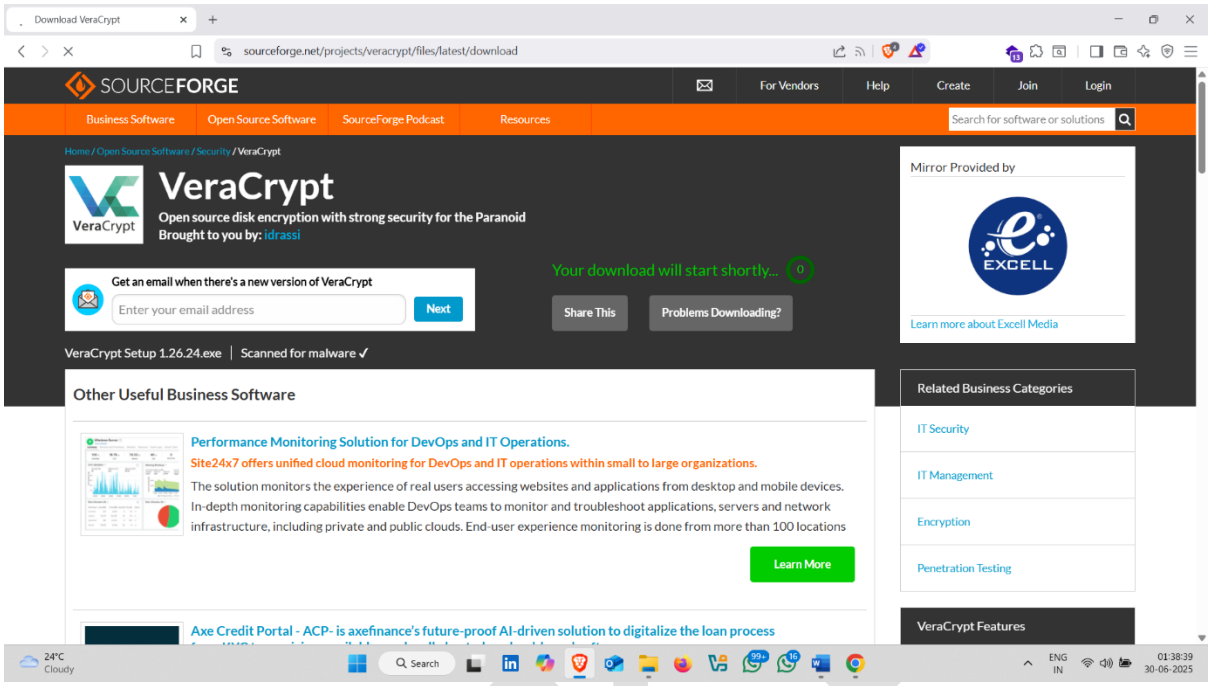


- click on Download



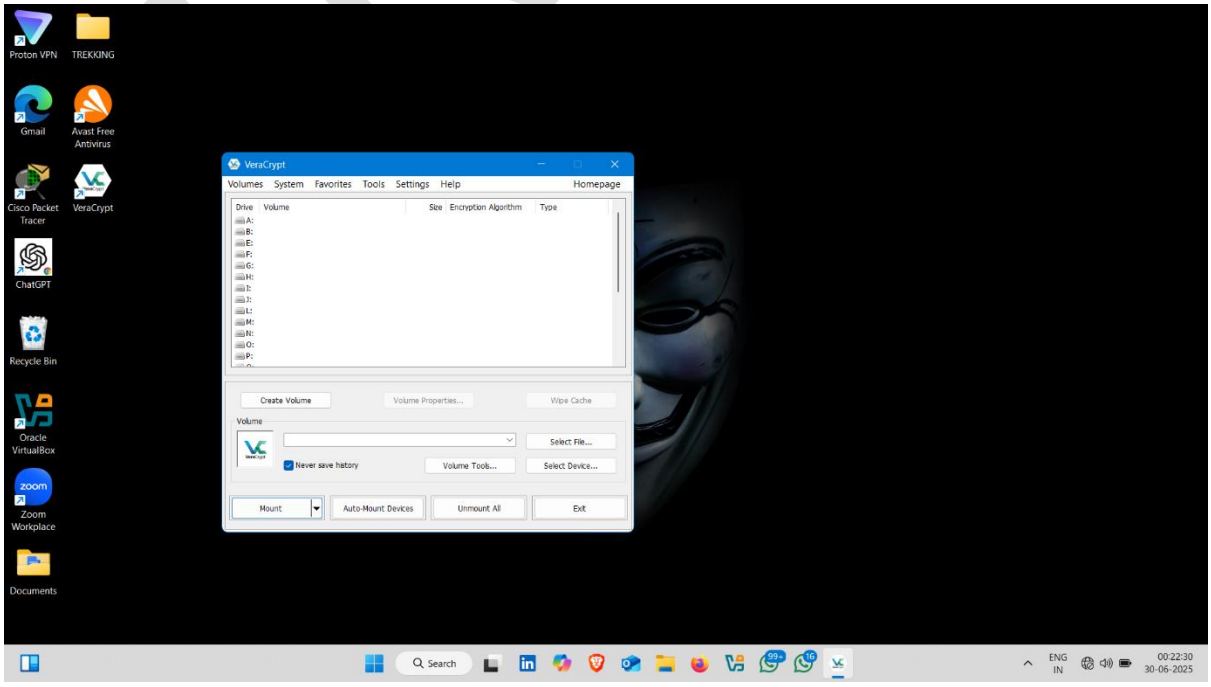
- Download Started automatically

Download Link-: https://sourceforge.net/projects/veracrypt/

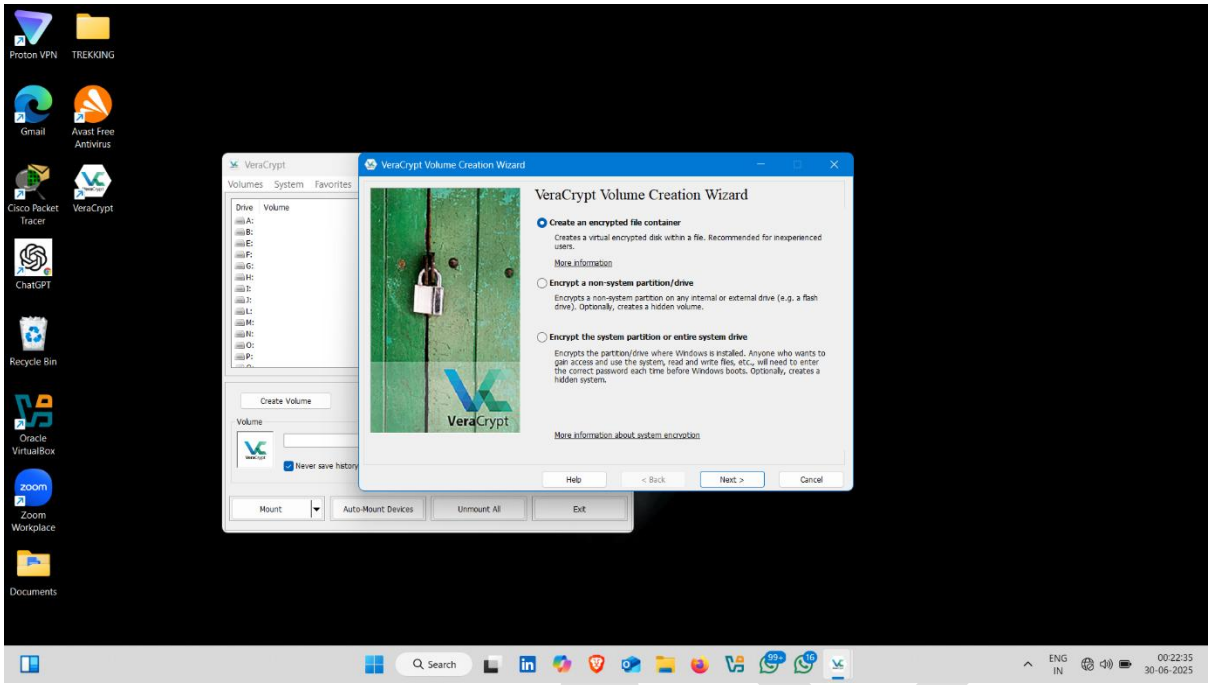


How to use it -:

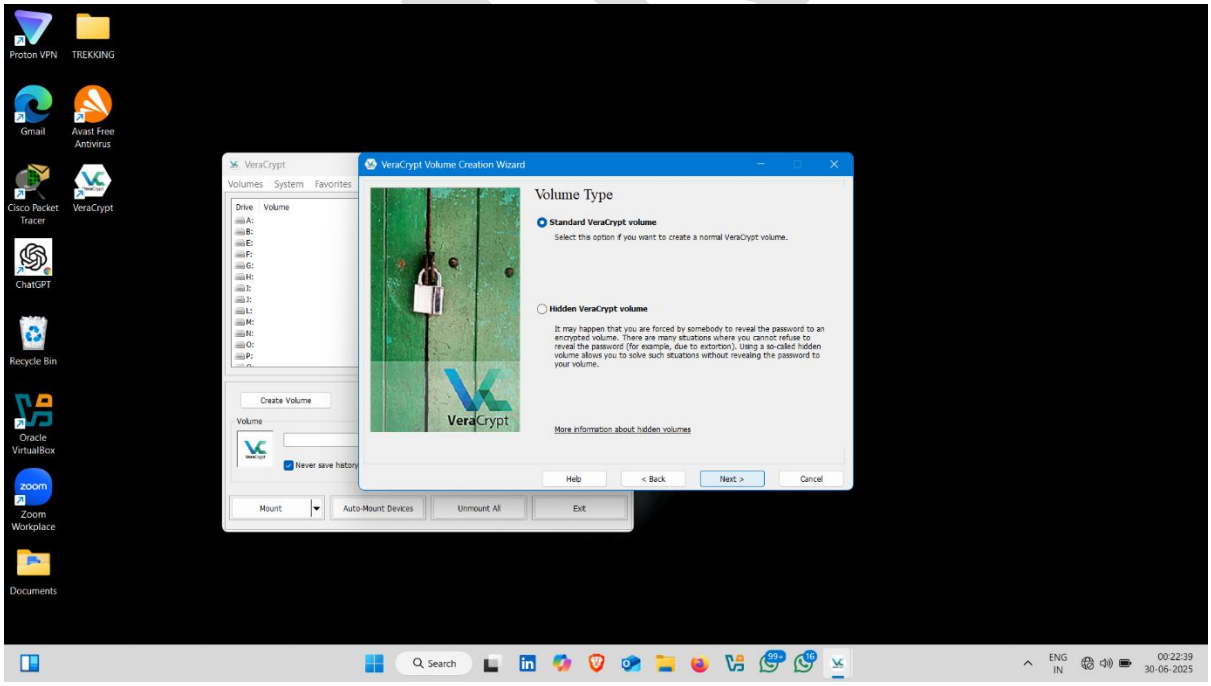
- After setup **veracrypt** open it
- Click on **create volume**



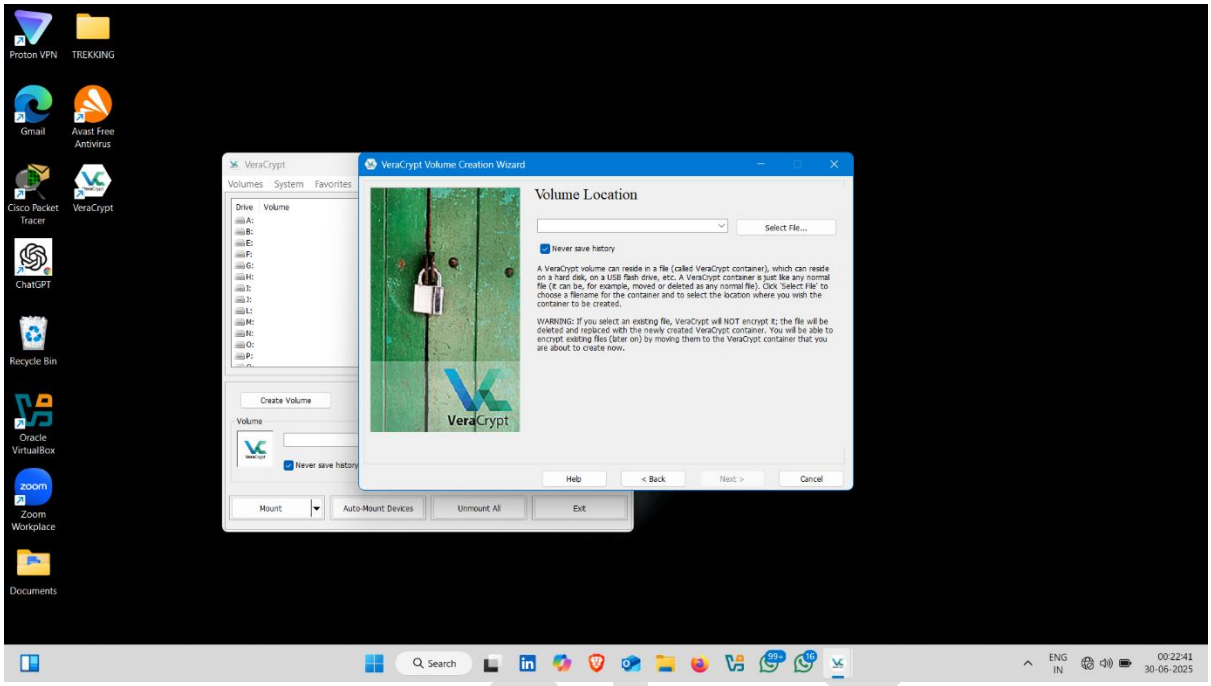
- Select first option – **Create an encrypted file container**



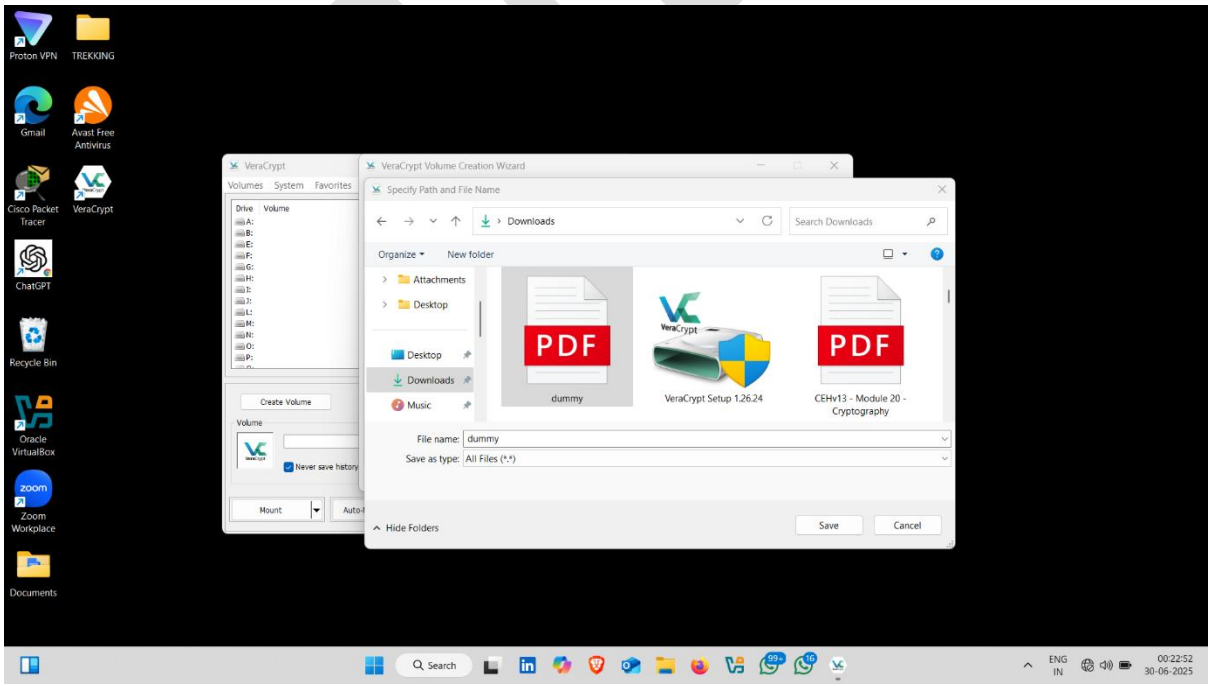
- Now select first option – **Standard VeraCrypt volume**



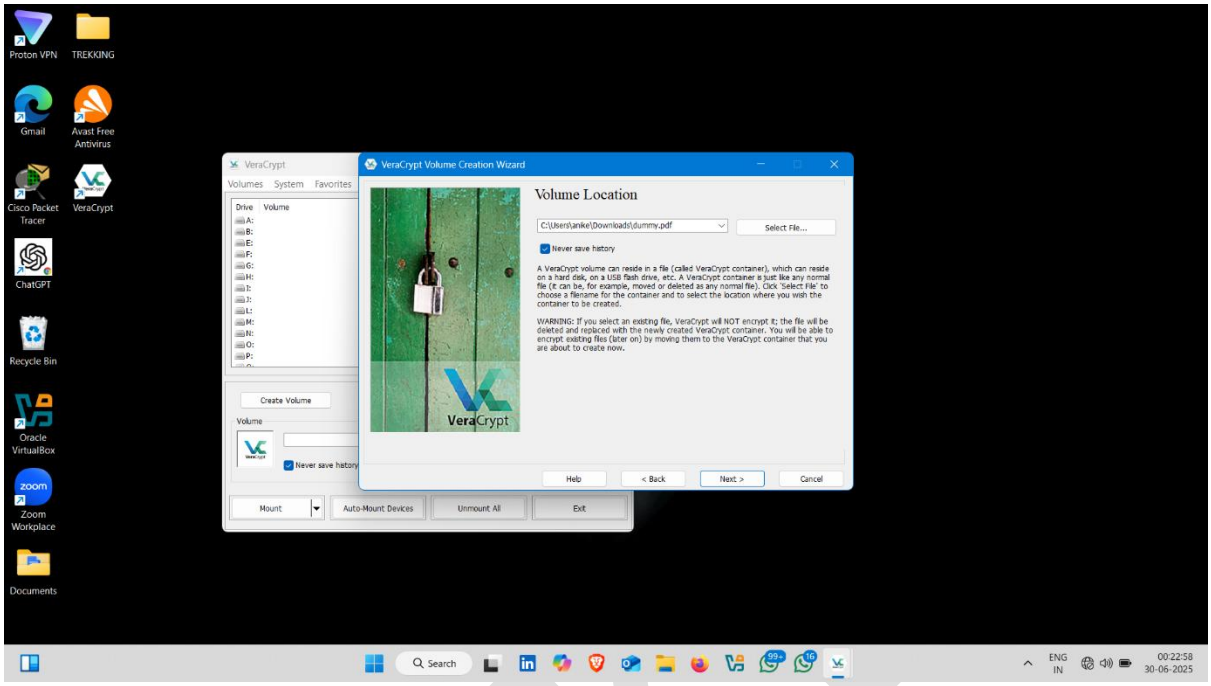
- Select file that you want to encrypted



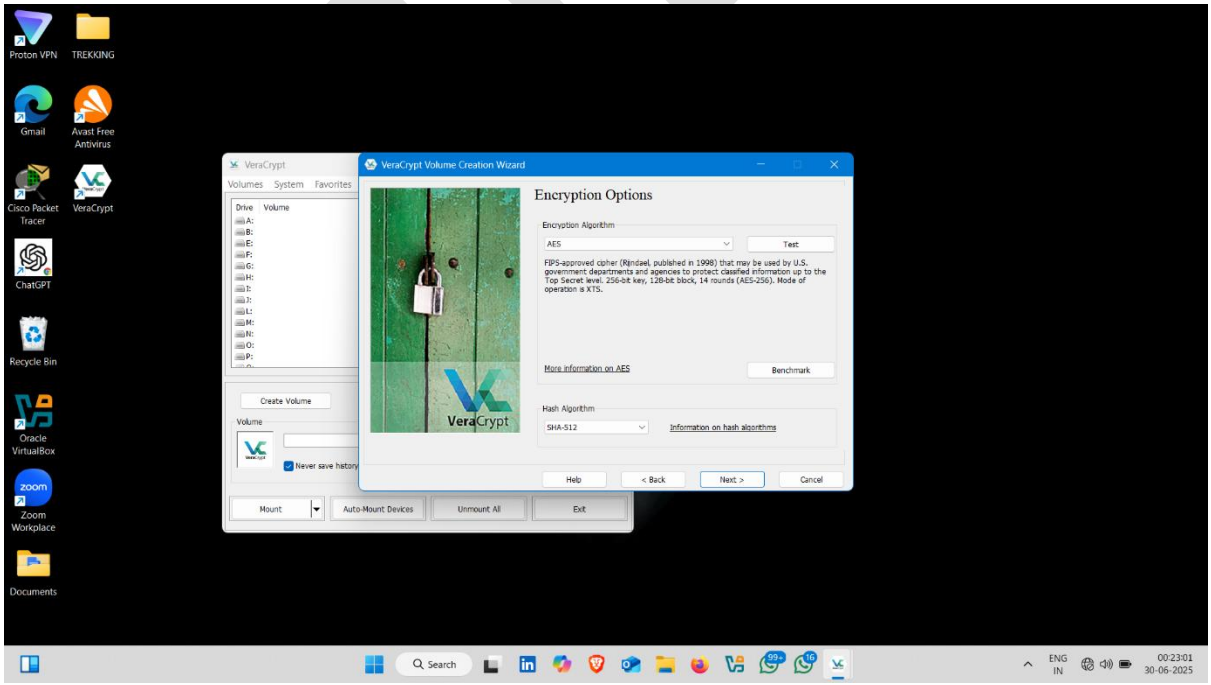
- Click on **save**



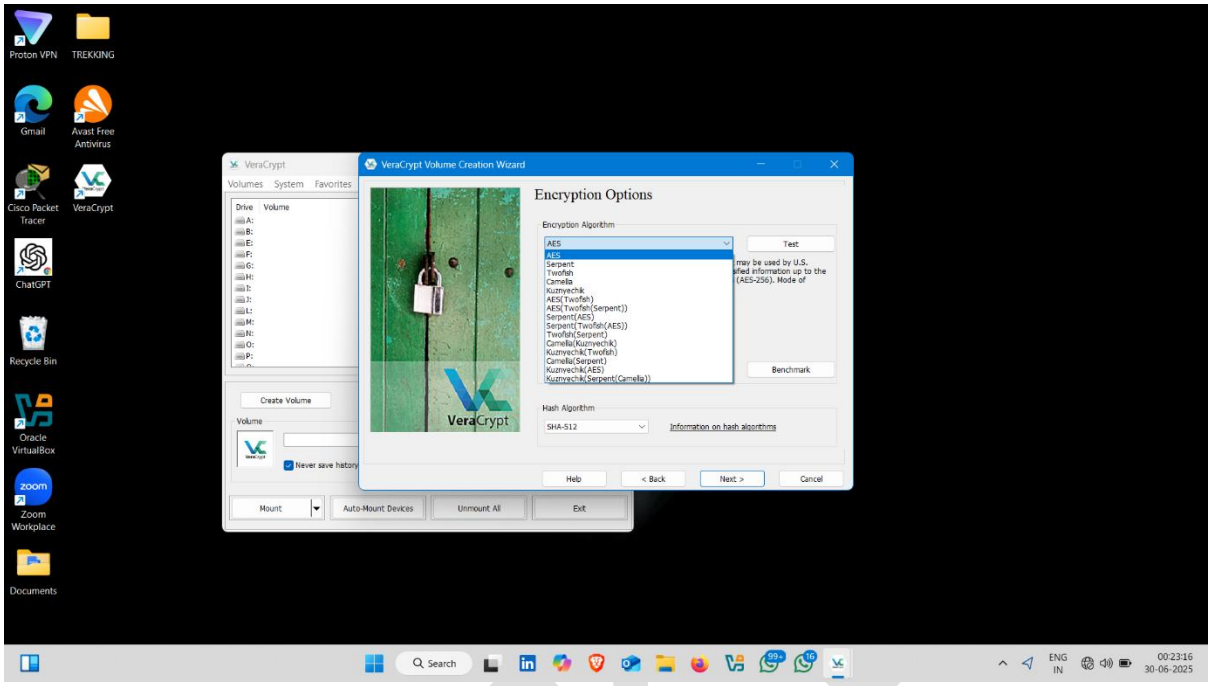
- Click on next



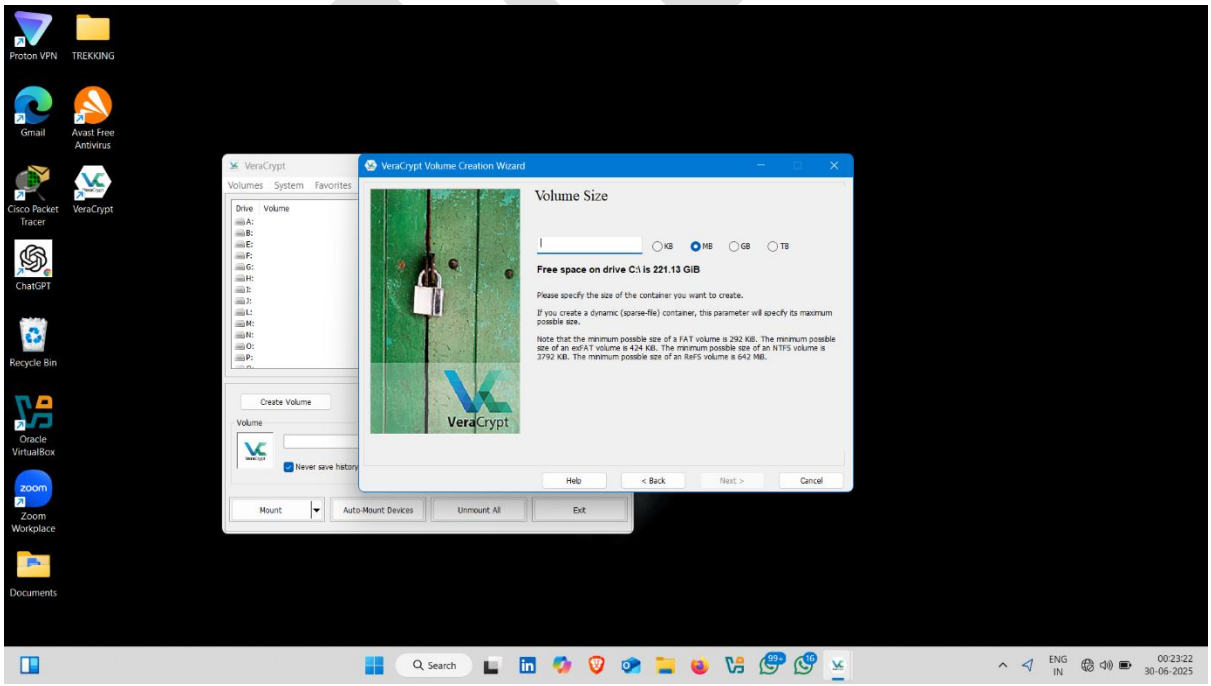
- Click on drop-down arrow 🔽 – to select encryption type



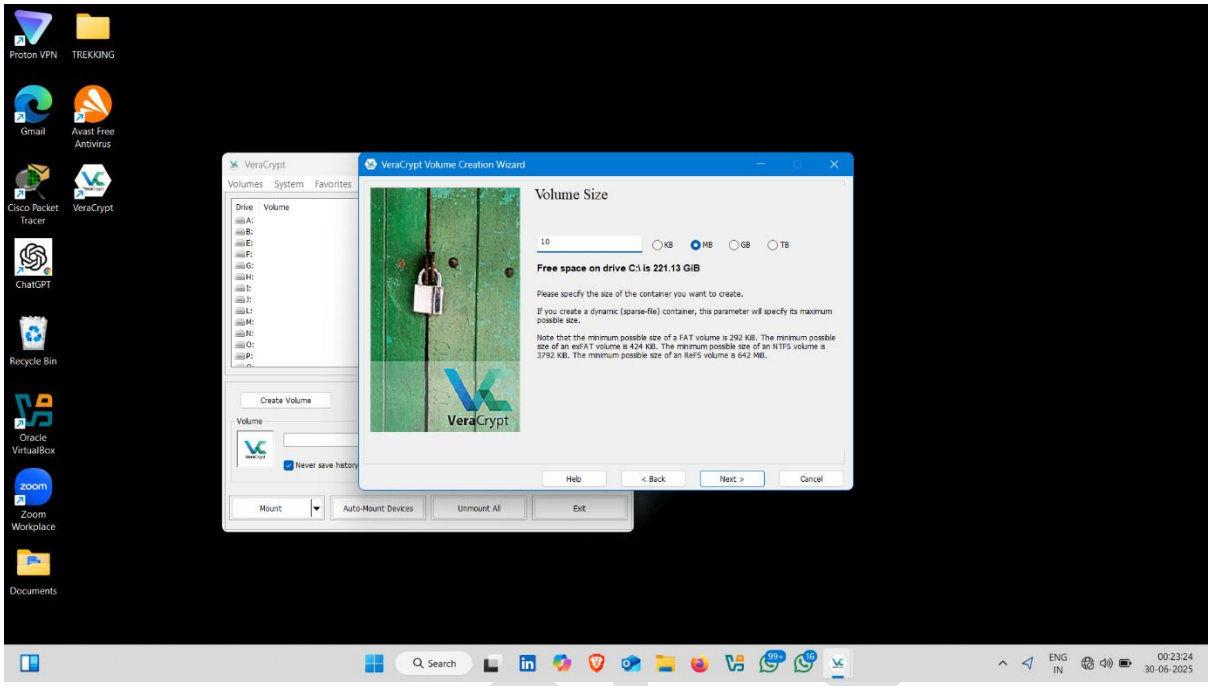
- Click on **next**



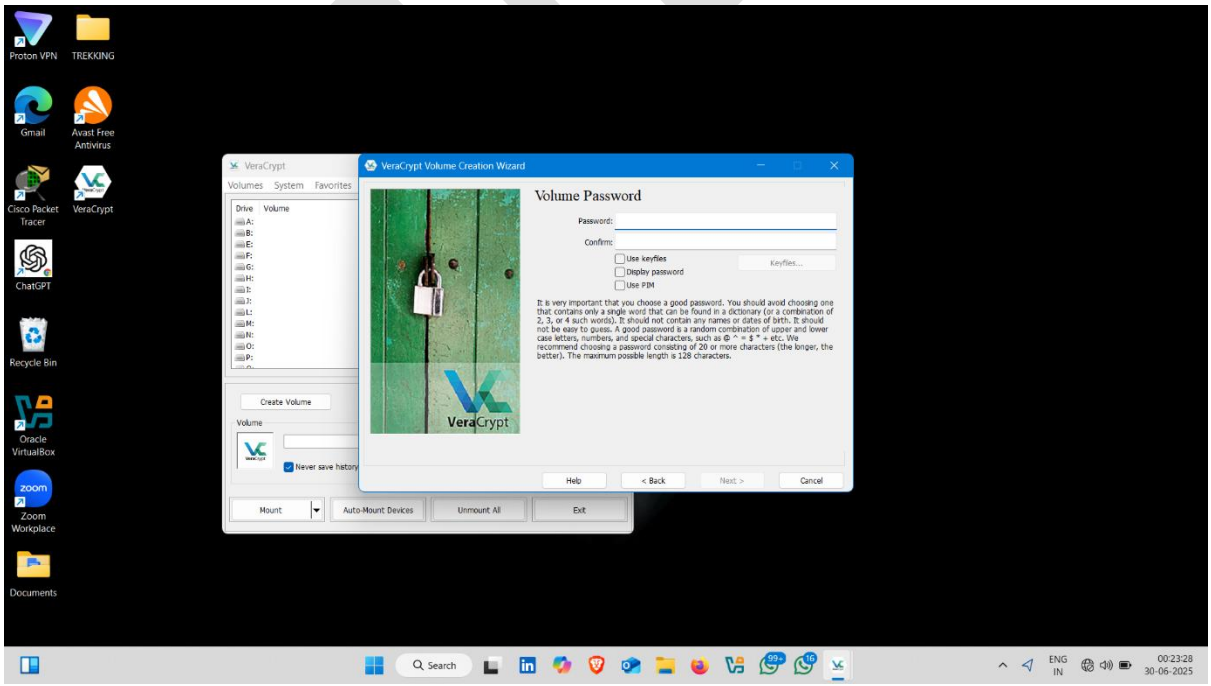
- Select **free space on drive**



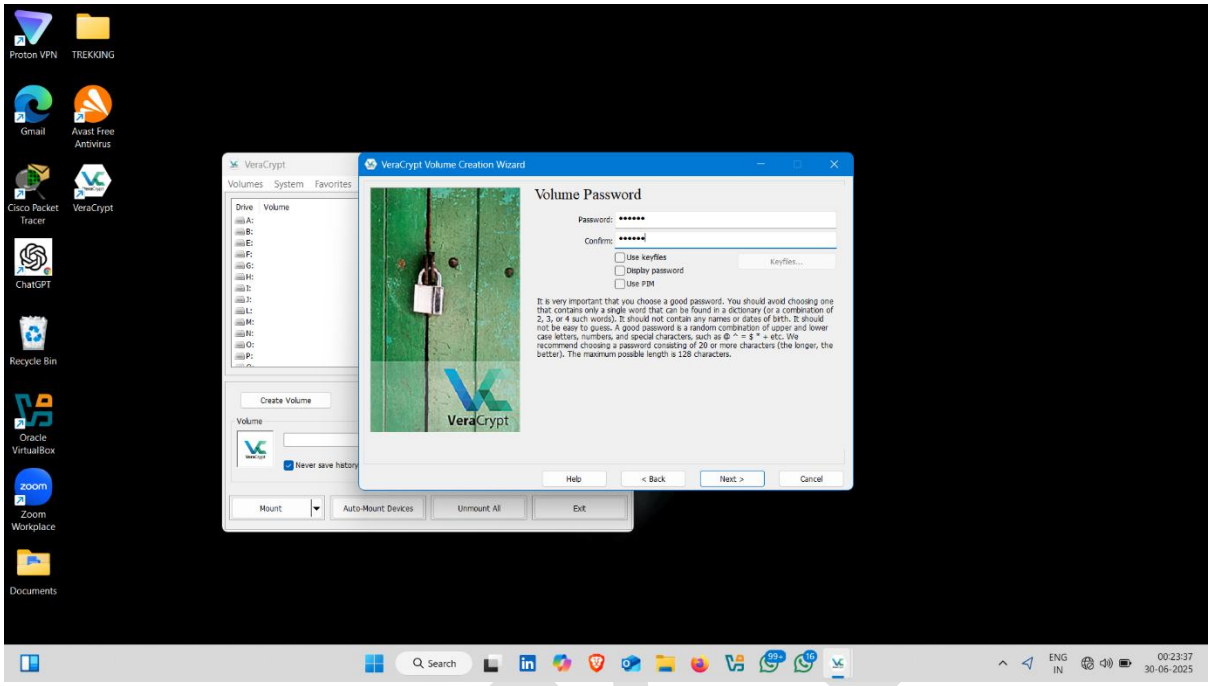
- Click on **next**



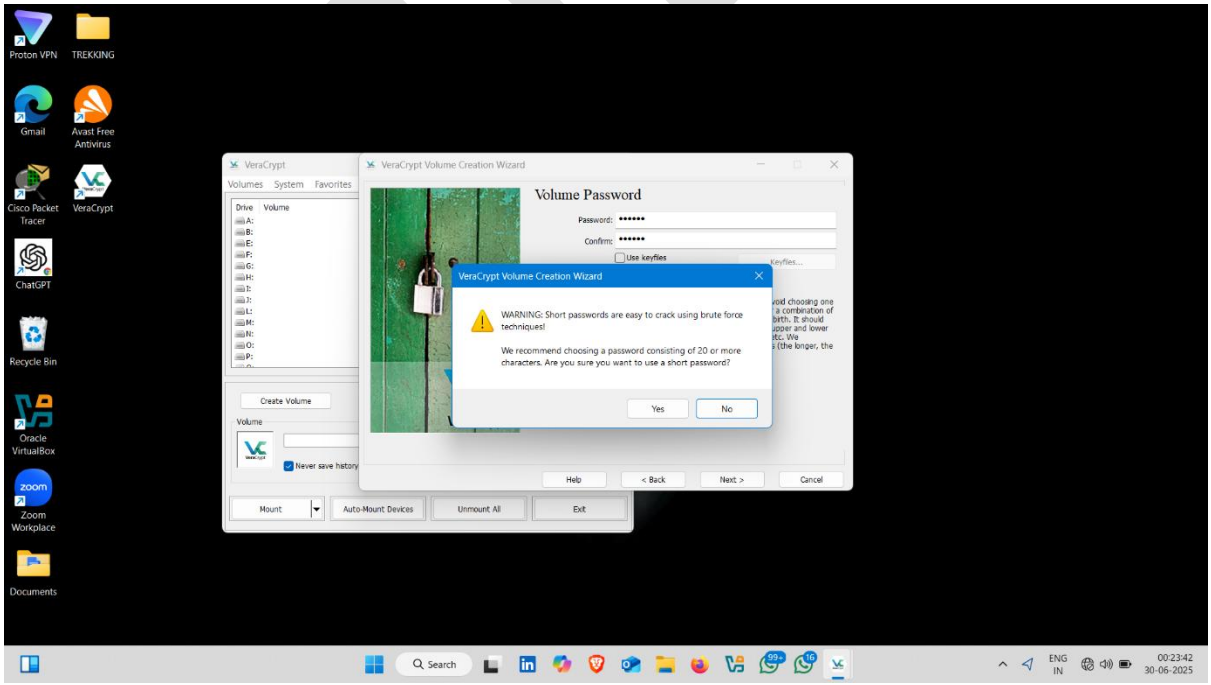
- now set a **password**



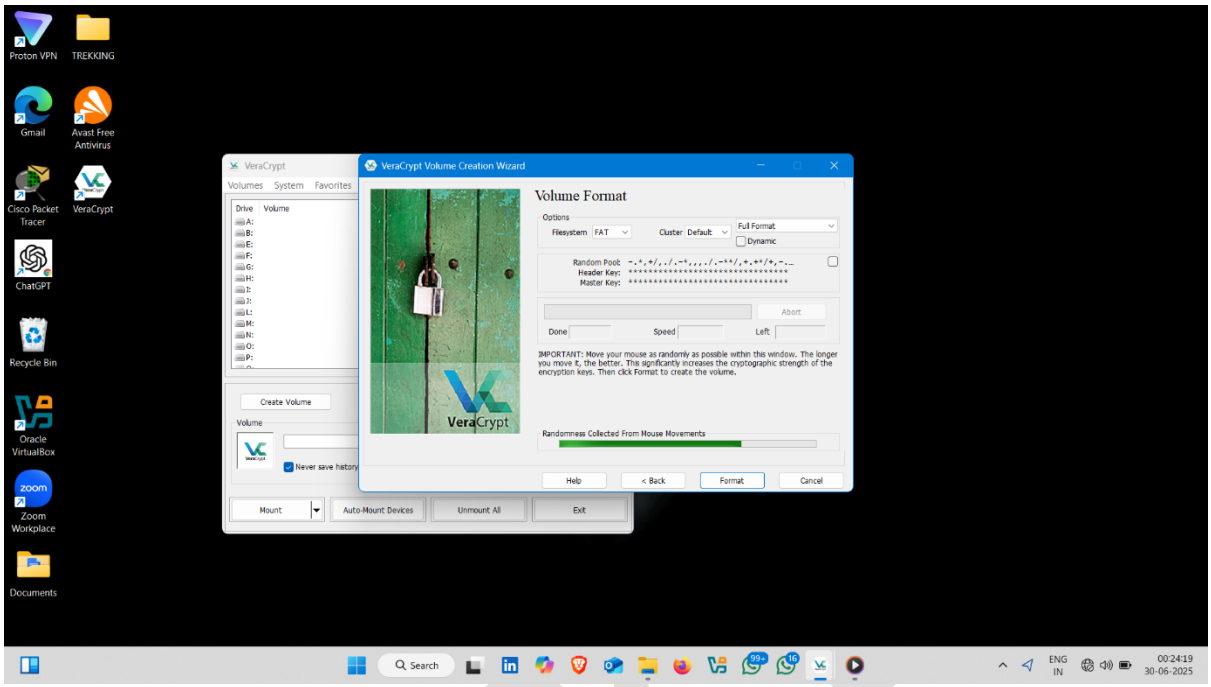
- click on **next**



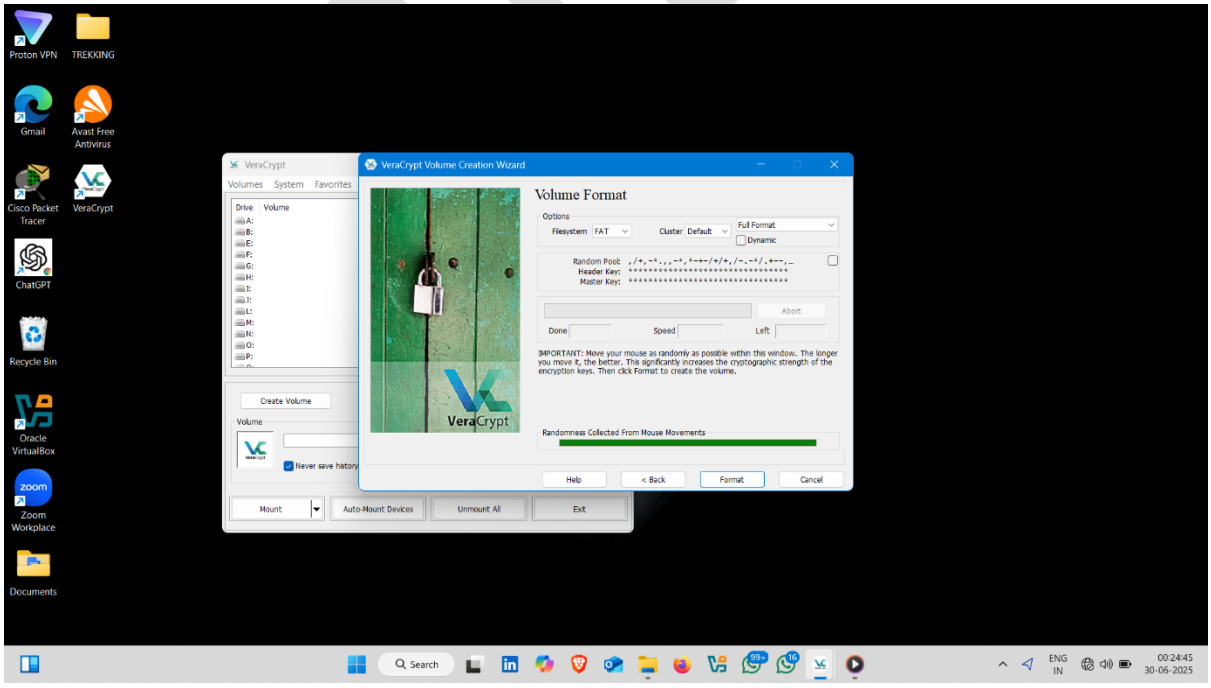
- click on **yes** ✅



- **Encryption started** ✅ 👇



- Click on **format**



✅ What Happens After Clicking "Format":

Once you click the **"Format"** button:

1. VeraCrypt **formats the volume** with the selected file system (here, FAT).
2. It **encrypts the entire volume** using the selected encryption algorithm.
3. A **virtual encrypted file container** is created — this is like a secure vault.
4. You'll receive a **success message**, and the volume will be ready to **mount** and use.
5. You can now **mount the volume** in VeraCrypt, enter your password, and it will behave like a normal drive (you can copy, edit, delete files inside — all encrypted in real-time).

⚠️ Important Note:

- **Be careful:** Clicking "Format" **erases any existing data** in the target volume.
- After the format completes, you should **mount** the volume from the main VeraCrypt window using the correct password.

EXTRA ACTIVITY

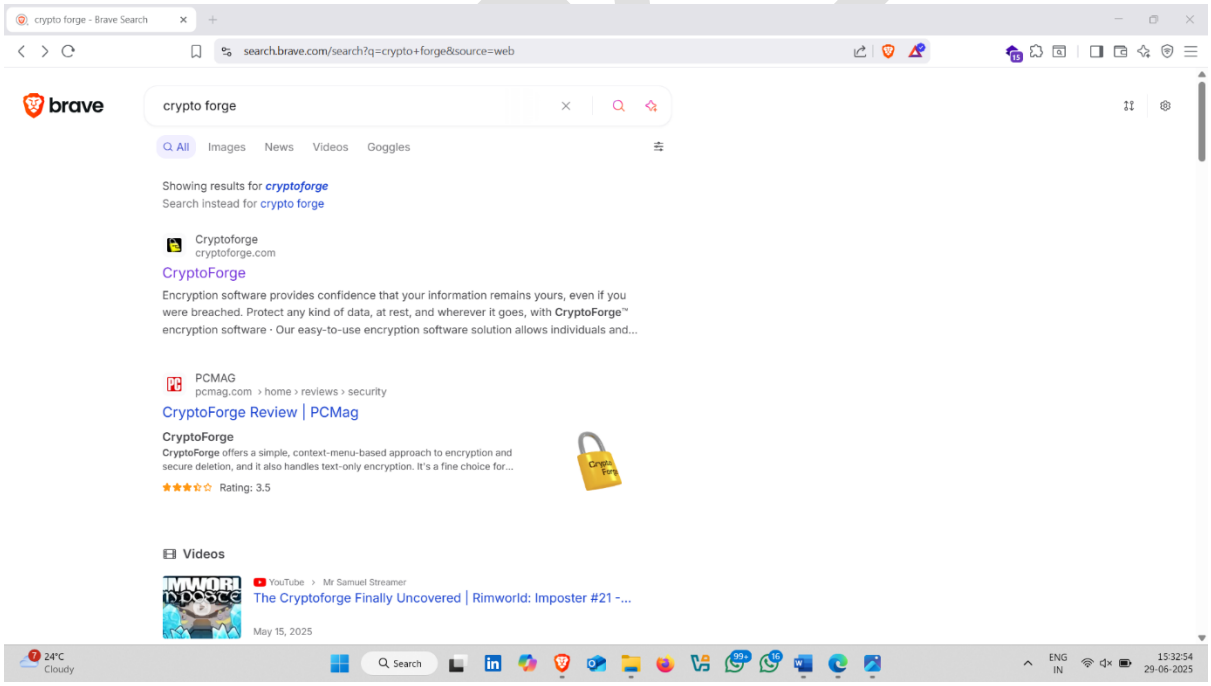
Cryptography Tools

1. Crypto-Forge

CryptoForge is a **Windows-based encryption software** designed for **secure file and text encryption** using strong symmetric cryptographic algorithms. It provides a simple and powerful solution for users who want to **protect sensitive files, folders, and text** from unauthorized access—whether for personal, professional, or business use.

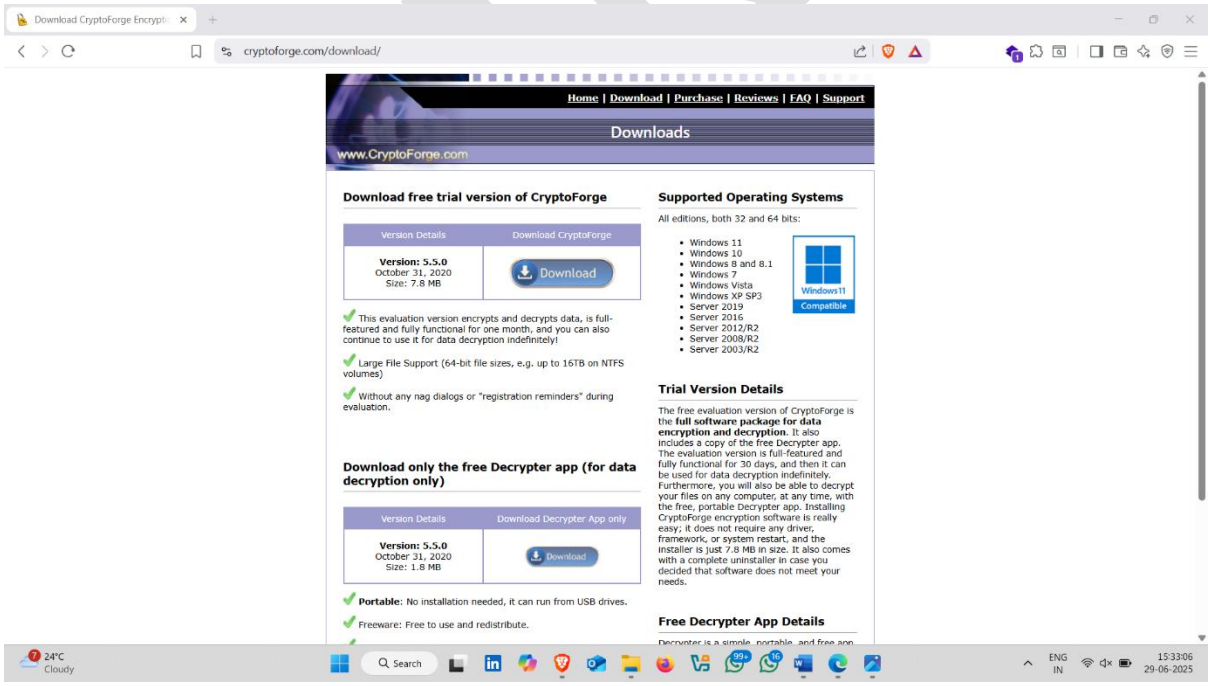
How to download it-:

- Open Browser and search **cyrpto-forage**
- Click on first Website 👇



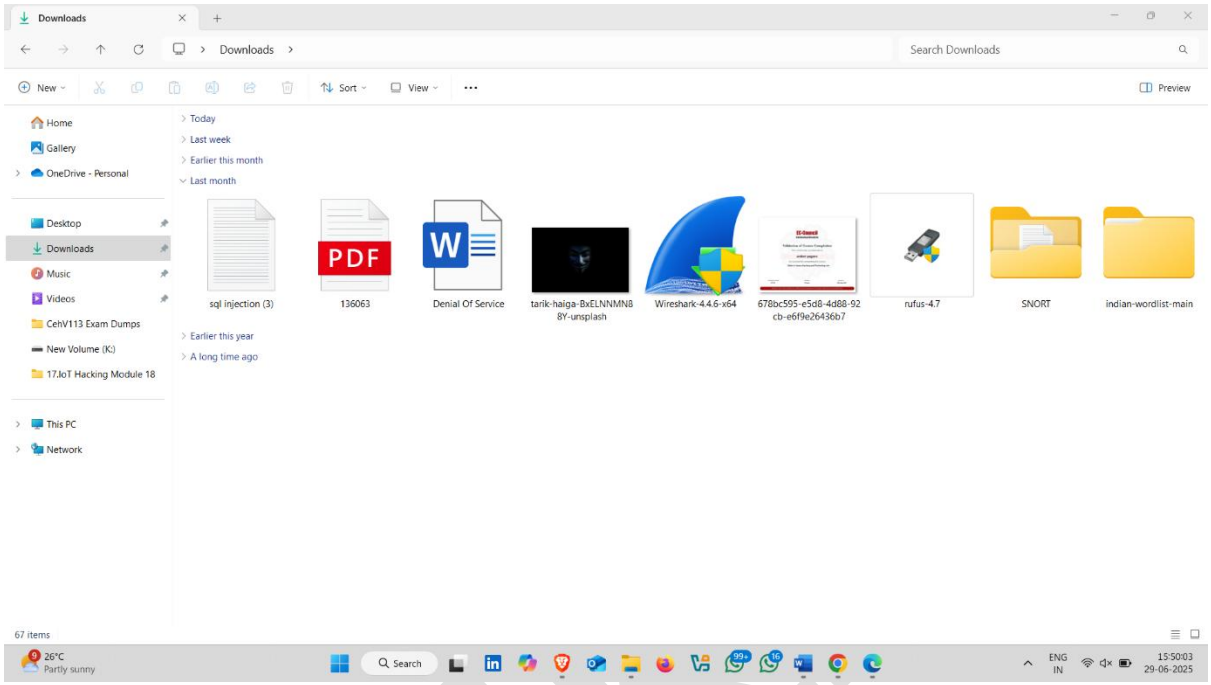
- Click on Download

- Once again , click on Download

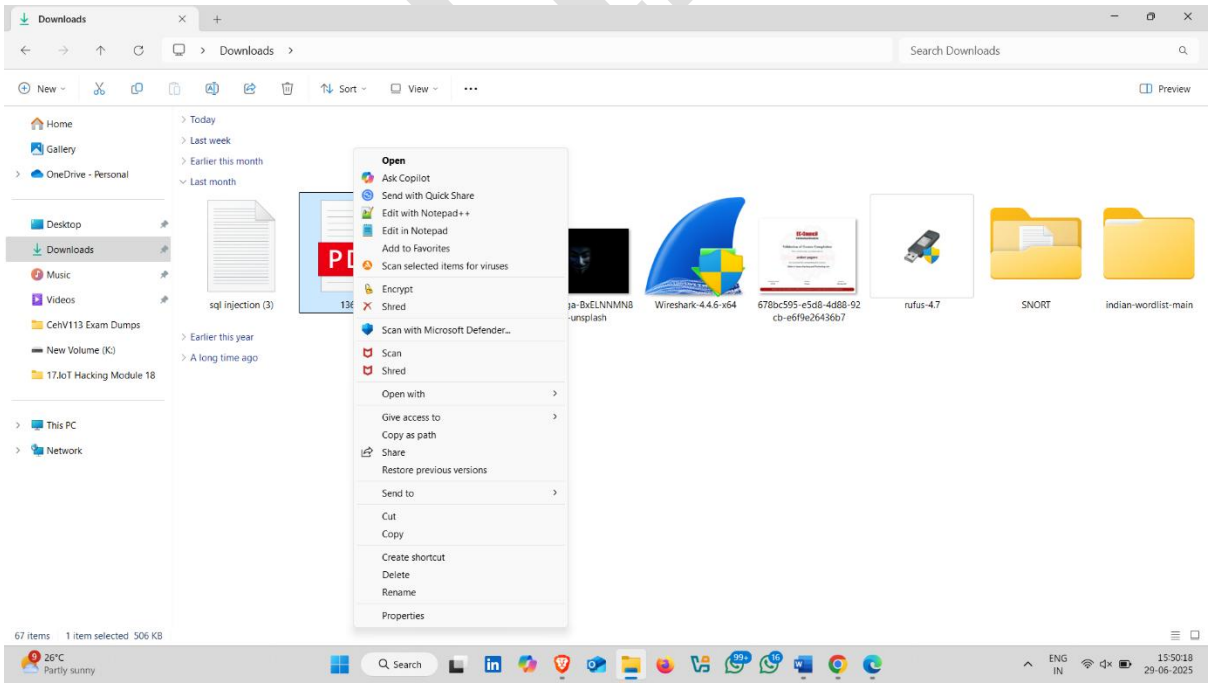


How to use it -:

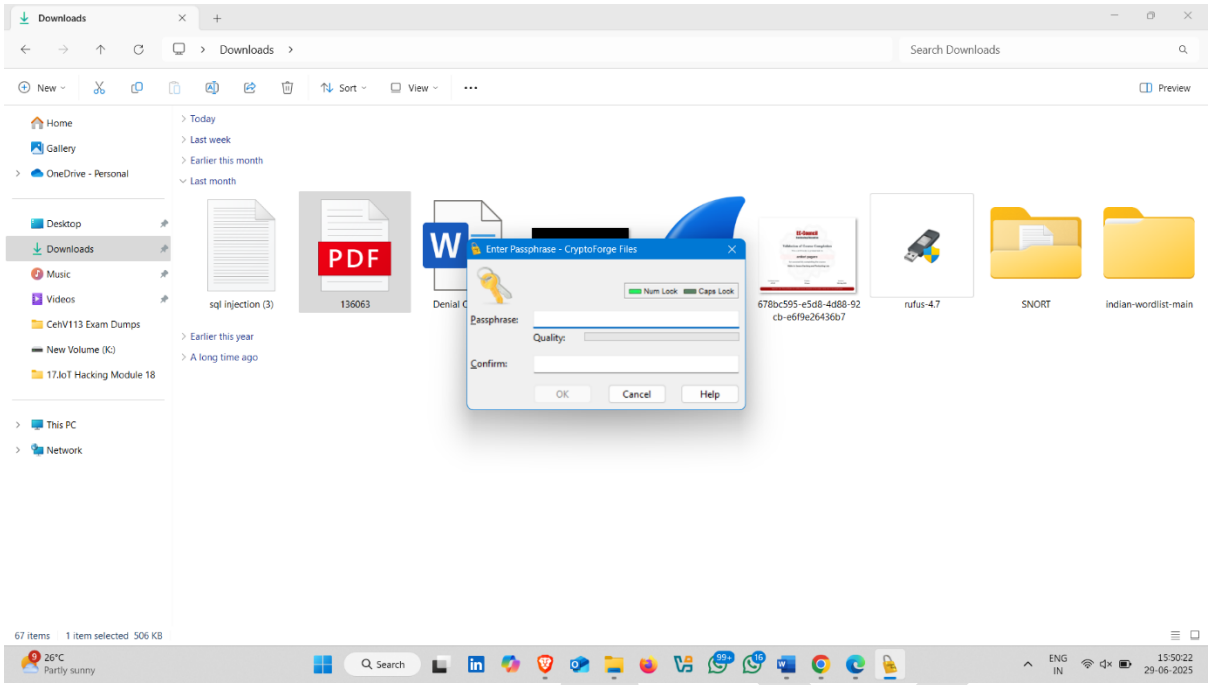
- **After Successfully install crypto-forge**
- **Select a file/folder to encrypt**



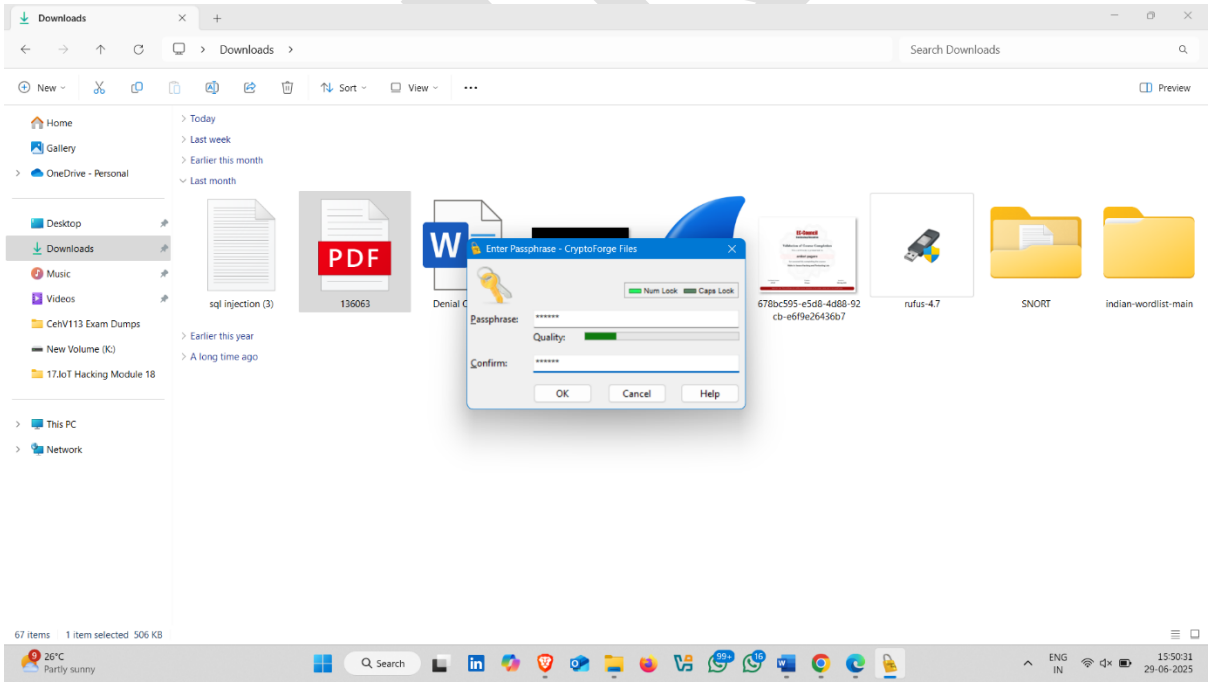
- **Right click on document and then click on encrypt**



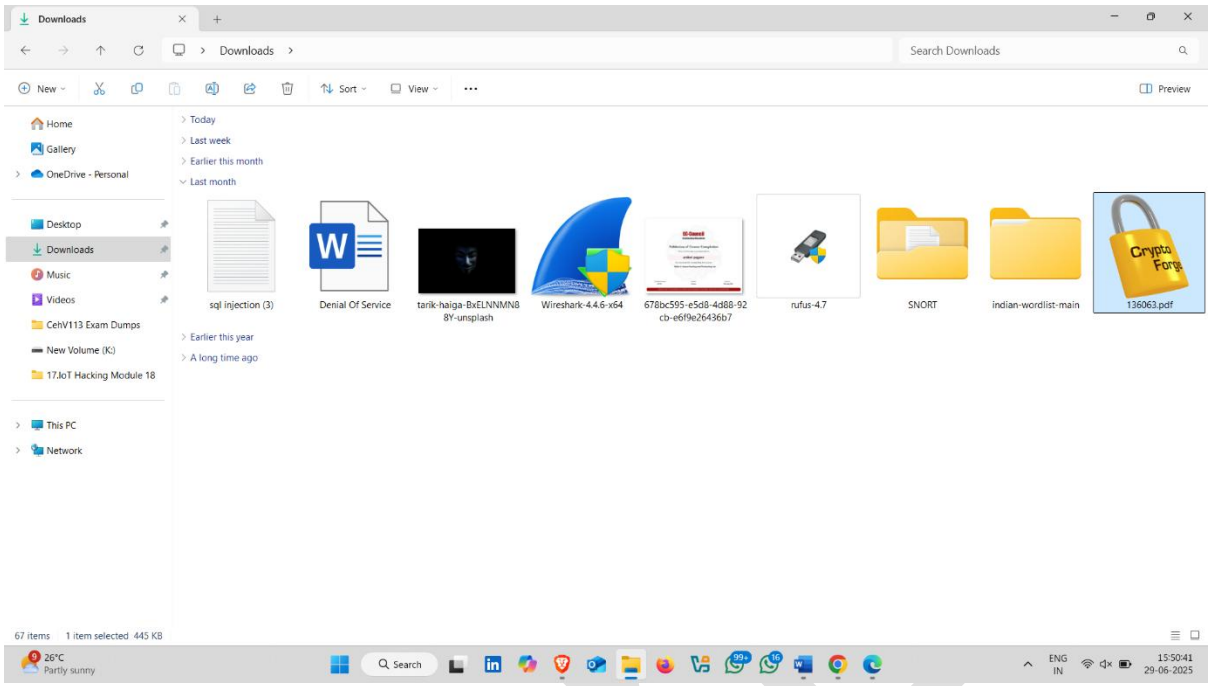
- Enter a strong Passphare/password



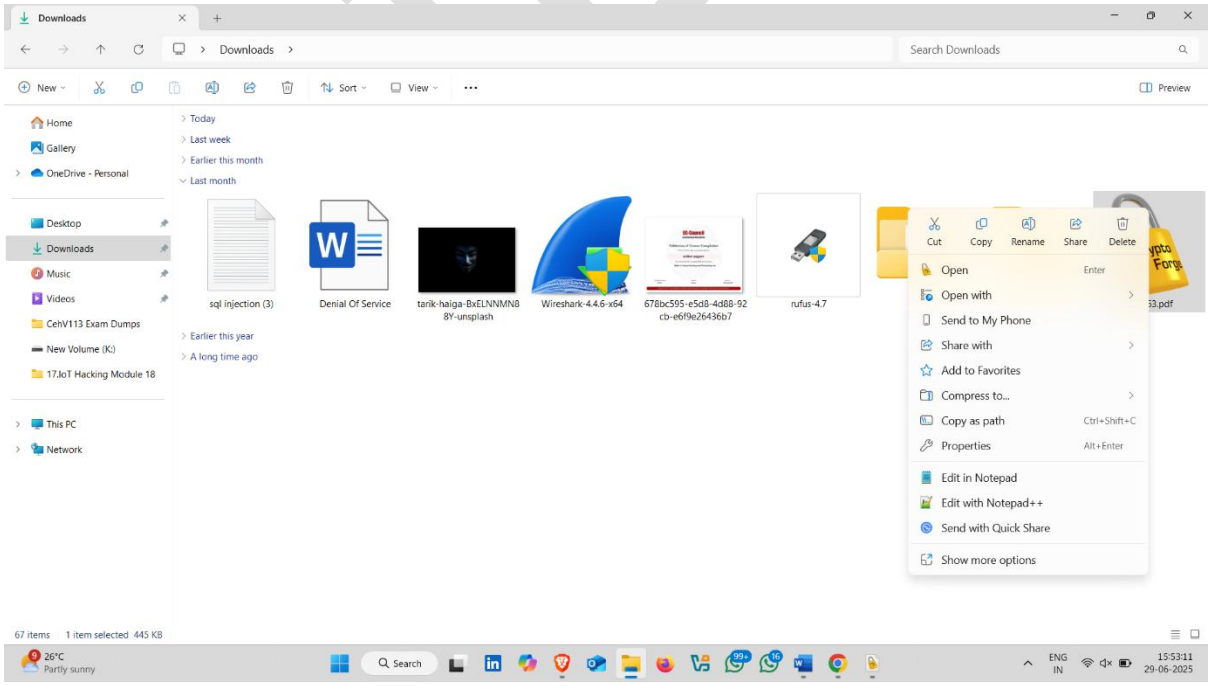
- **Click on ok**



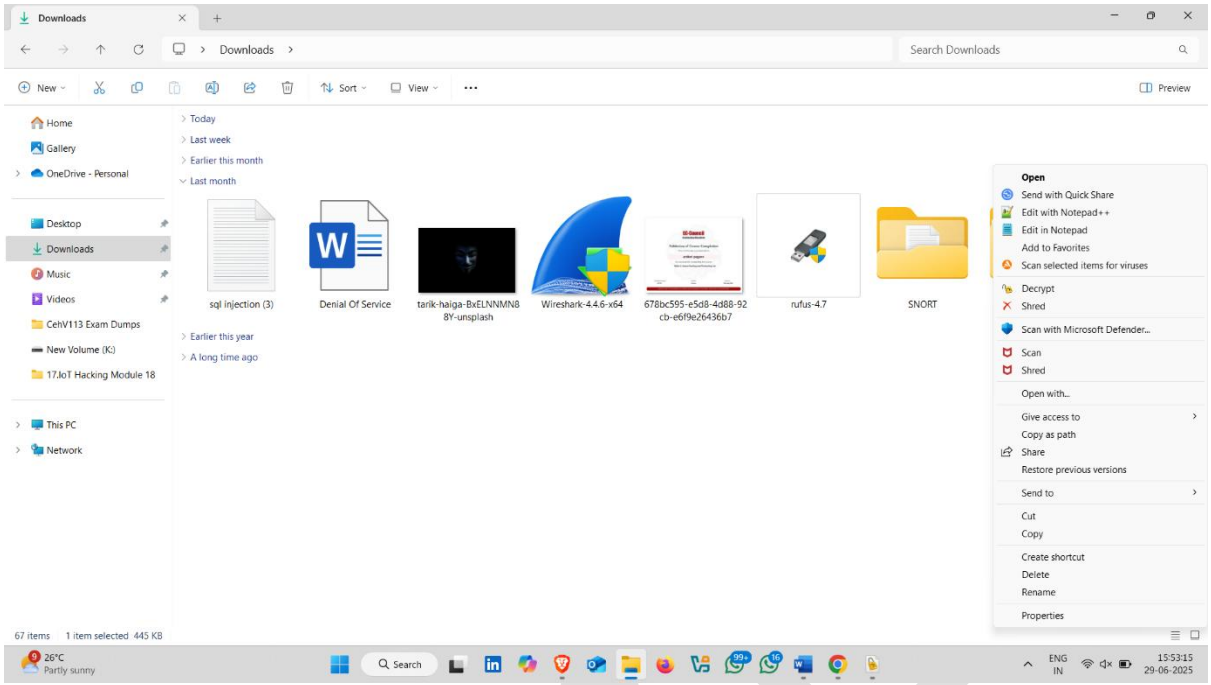
- **Document Encrypted 👇✅**



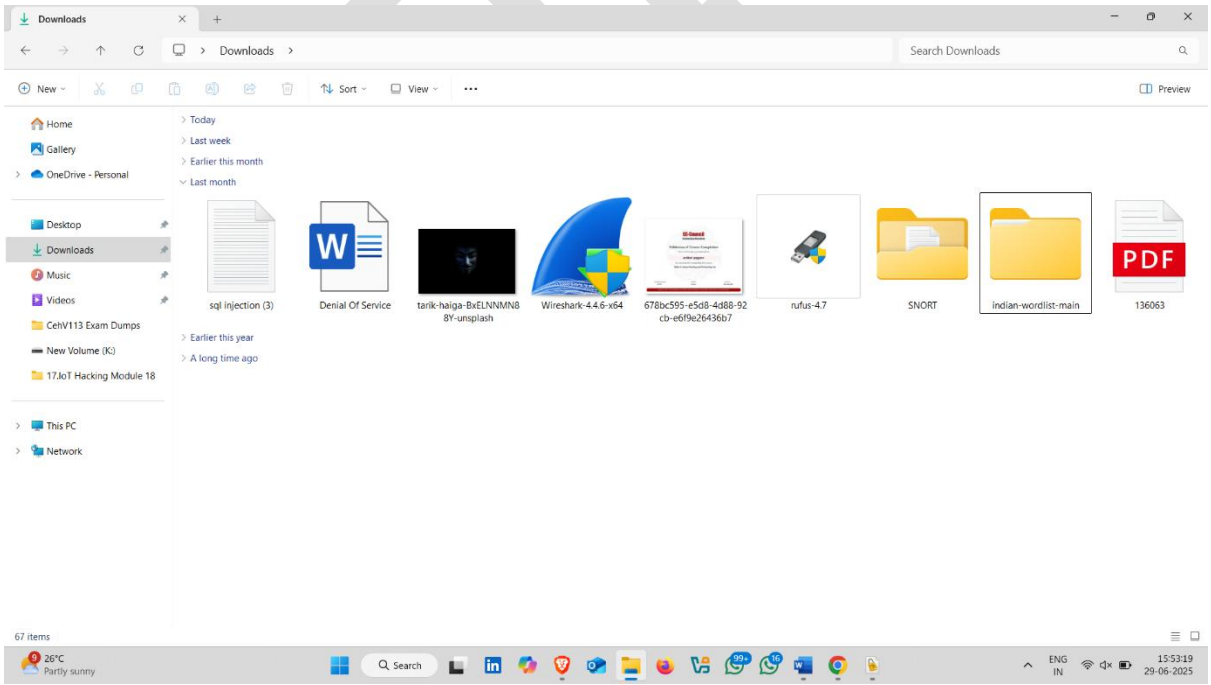
- TO Decrypt it
- Right click on **encrypted** document , click on **show more options**



- Click on **Decrypt**



- **Provide a passphare/password**
- **Document decrypted** ✅ 👇



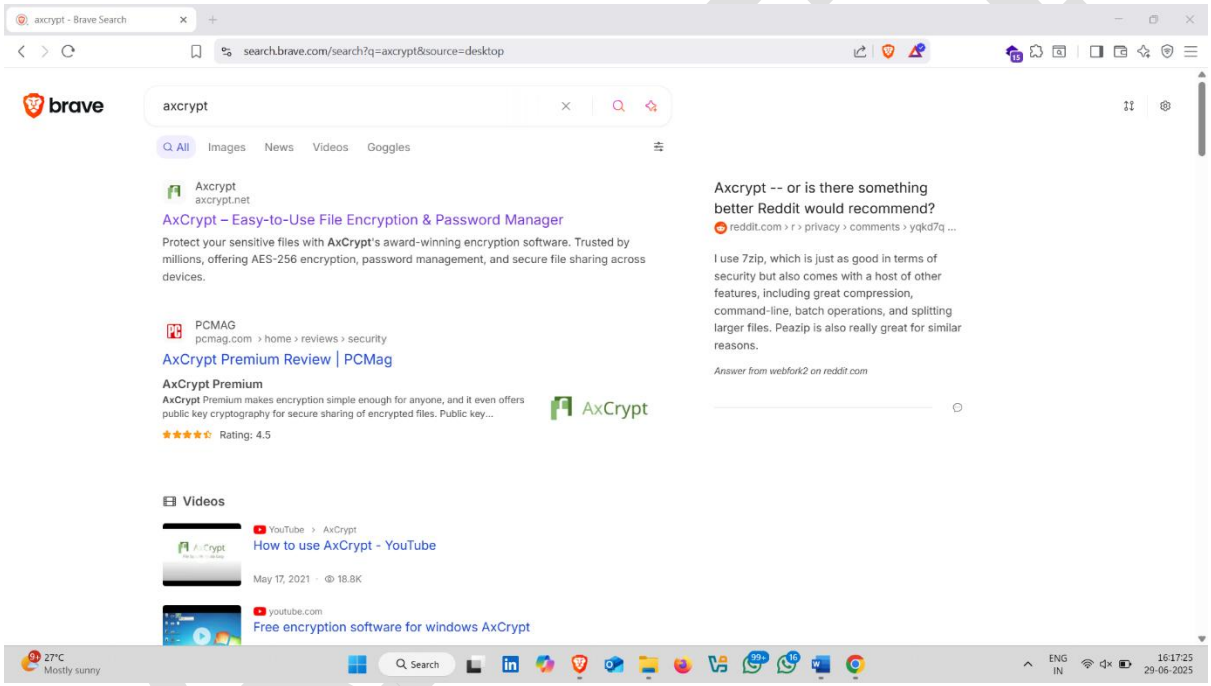
2.Axcrypt

AxCrypt is a **free and easy-to-use file encryption software** designed for **individual users and small teams** to protect sensitive files using **AES encryption**. It allows you to **secure files with a password**, so only you (or someone you trust) can open or access them.

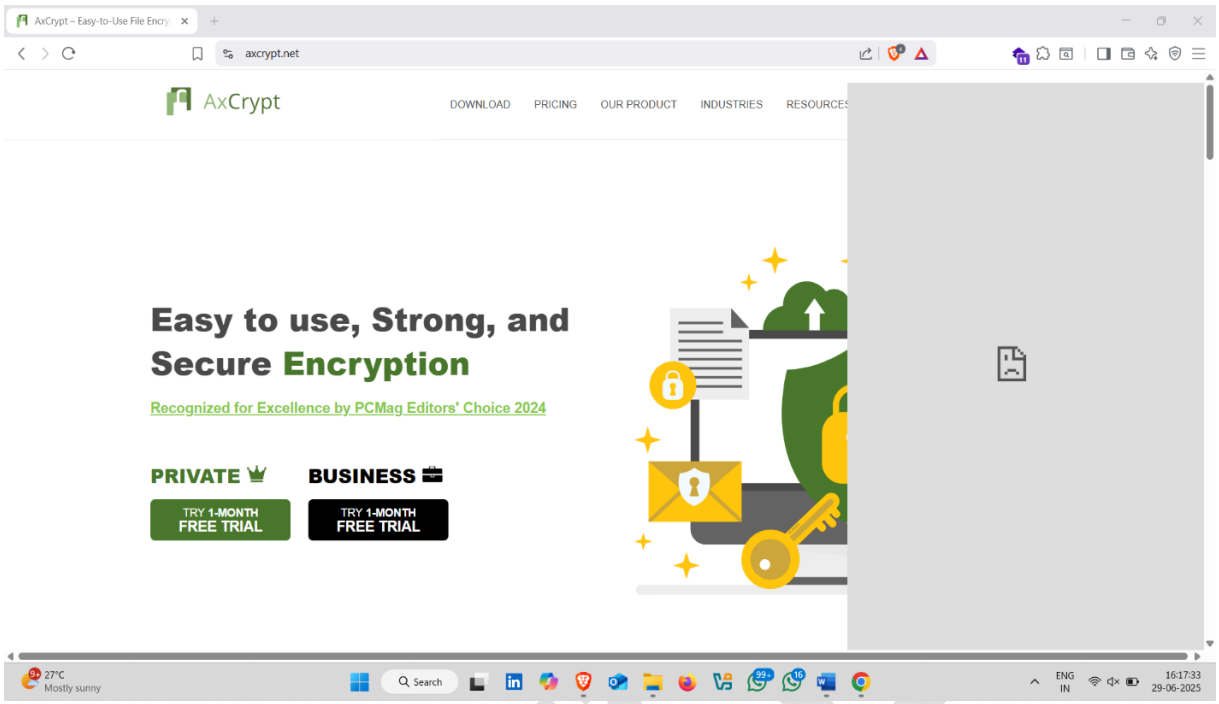
How to download it

- Open Browser and search **Axcrypt**
- **Click on First Website**

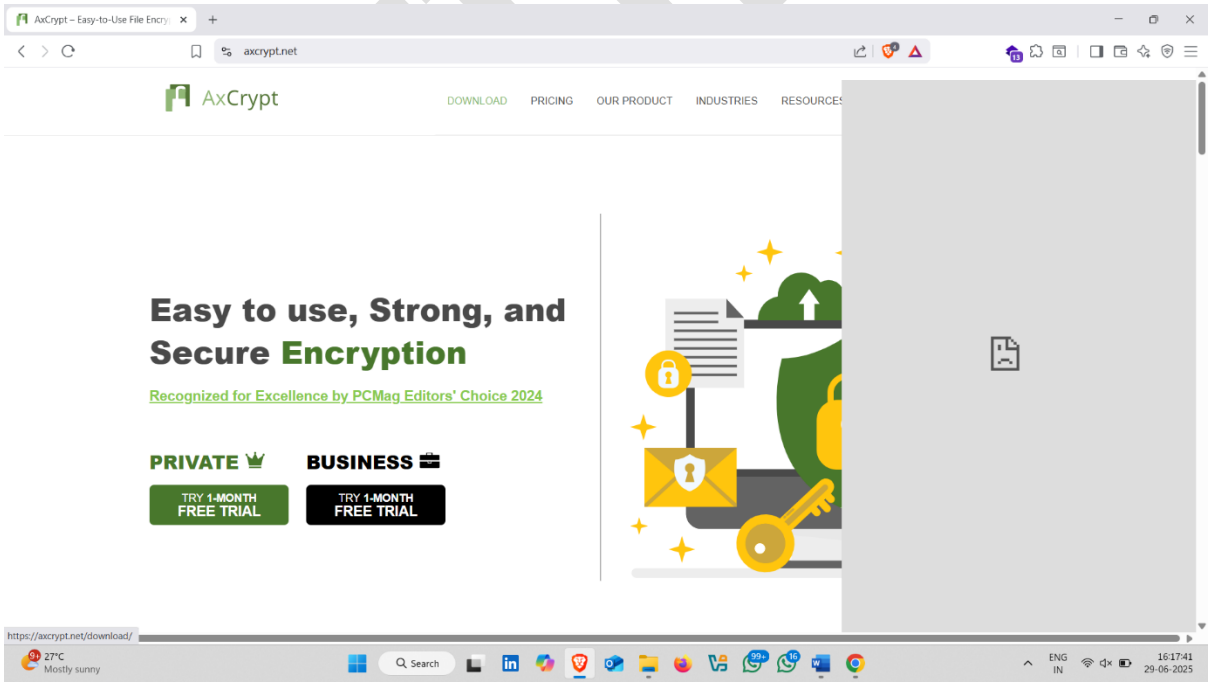
Download Link -: https://axcrypt.net/download/



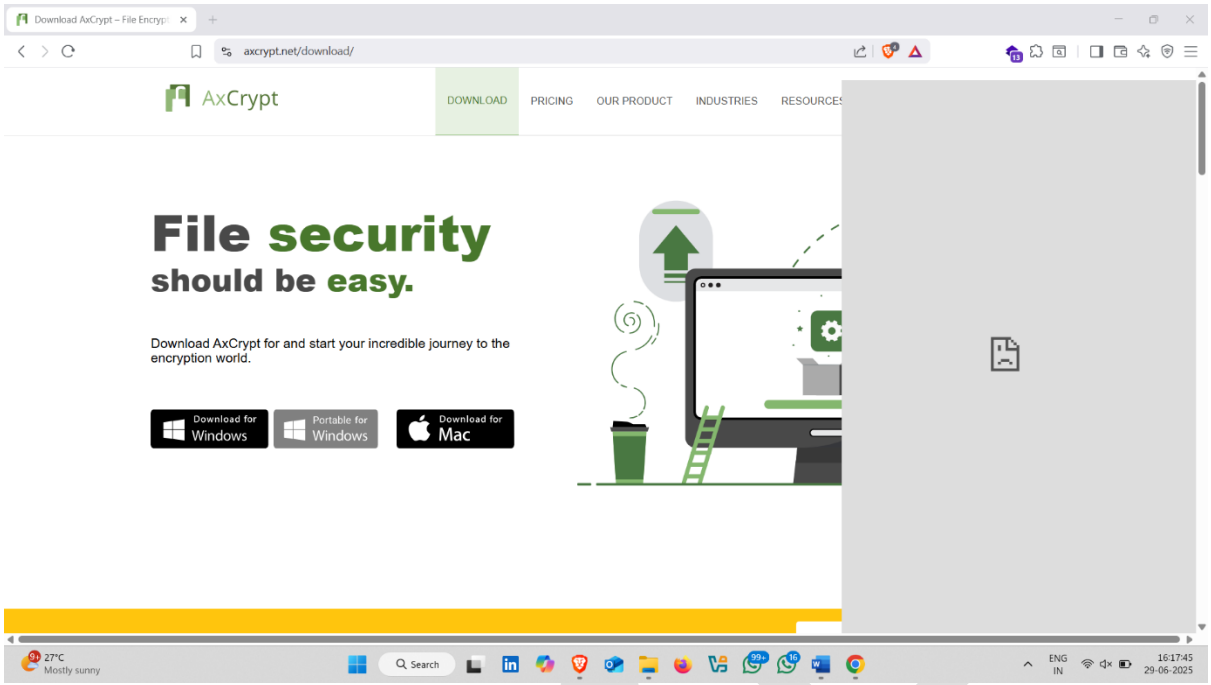
- Website Open 👇



- Click on **Download** Option

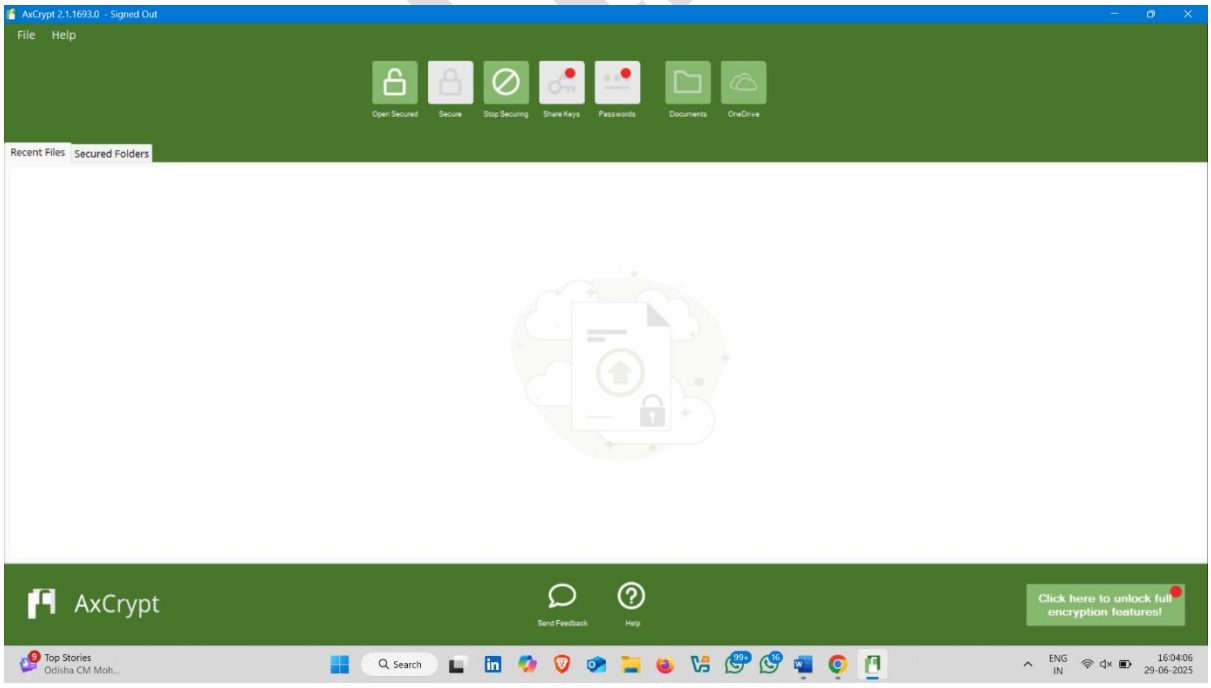


- And then click on **Downloads for Windows**



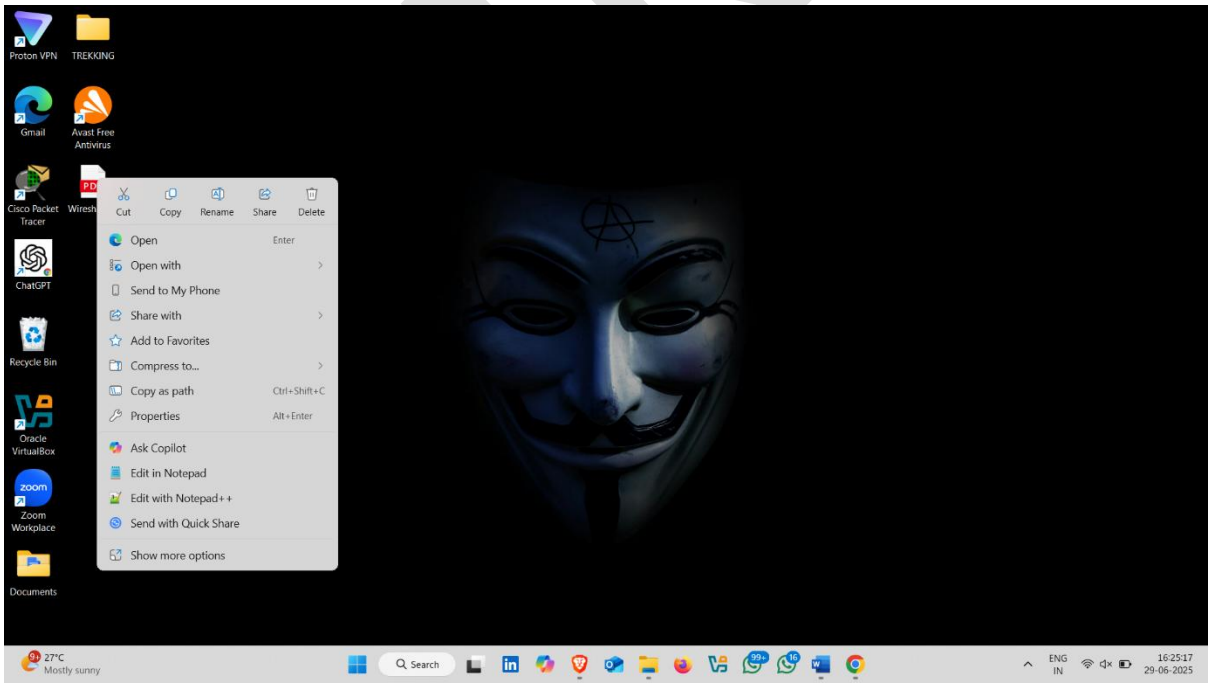
How to use it -:

- After Downloading , installed app and open it
- Interface

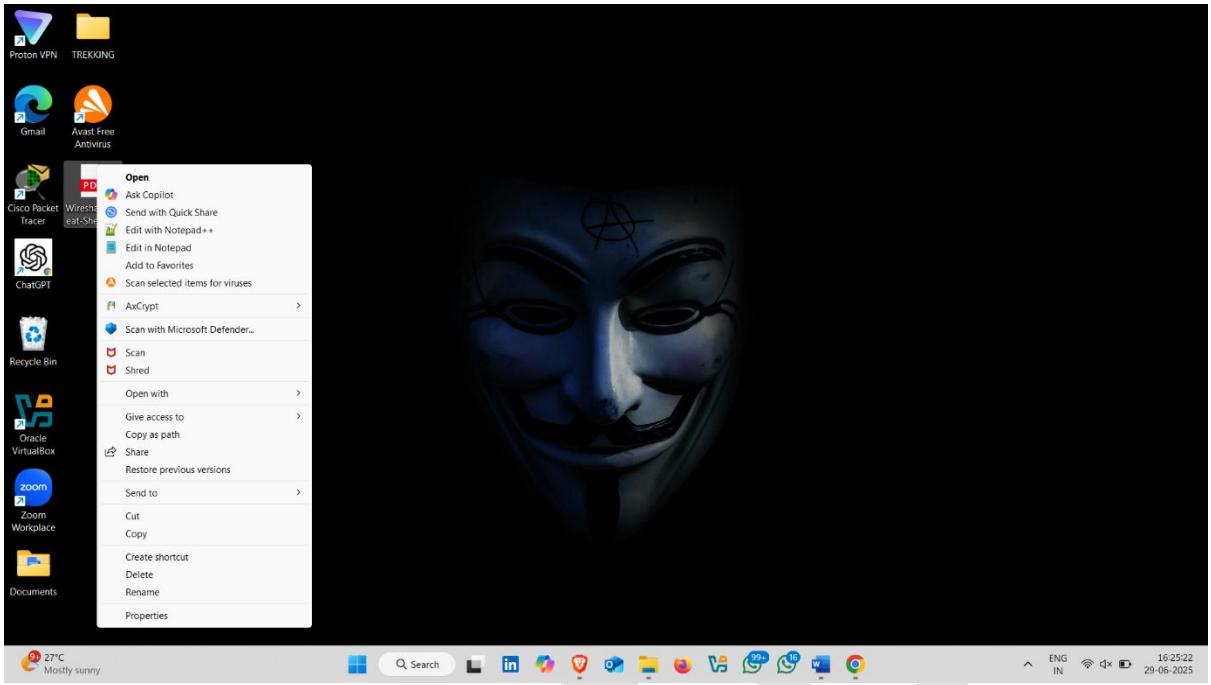


- Select file/folder want to encrypted

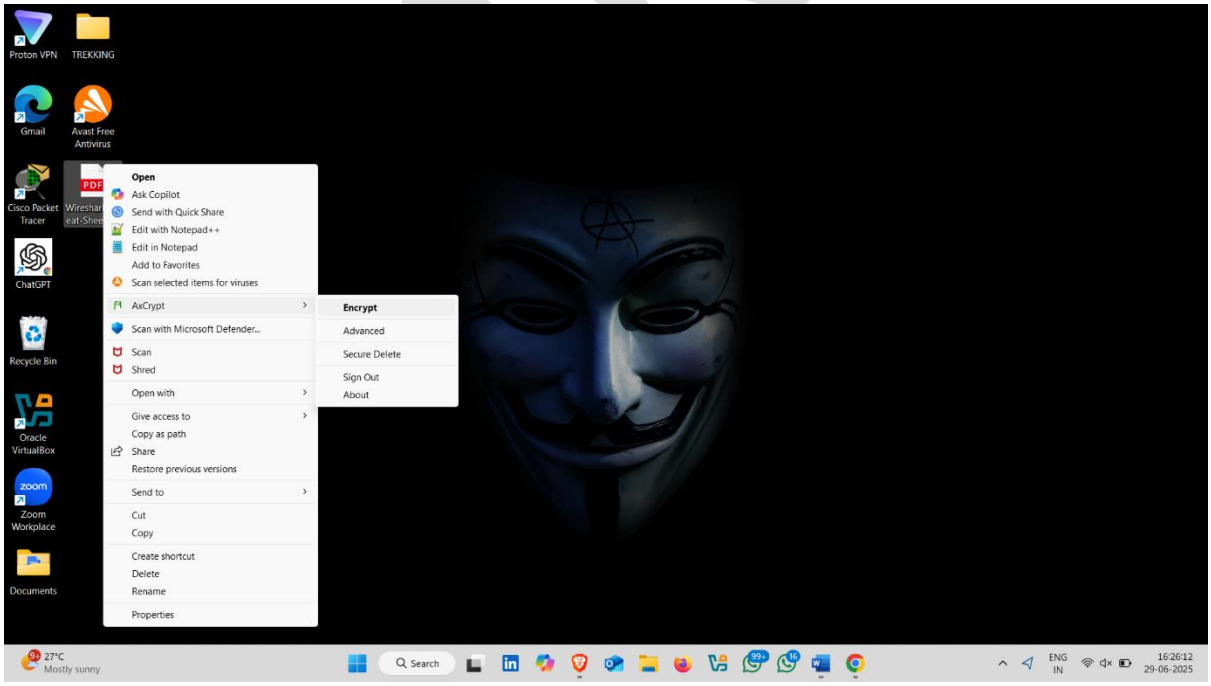
- Right click on folder and select **show more options**



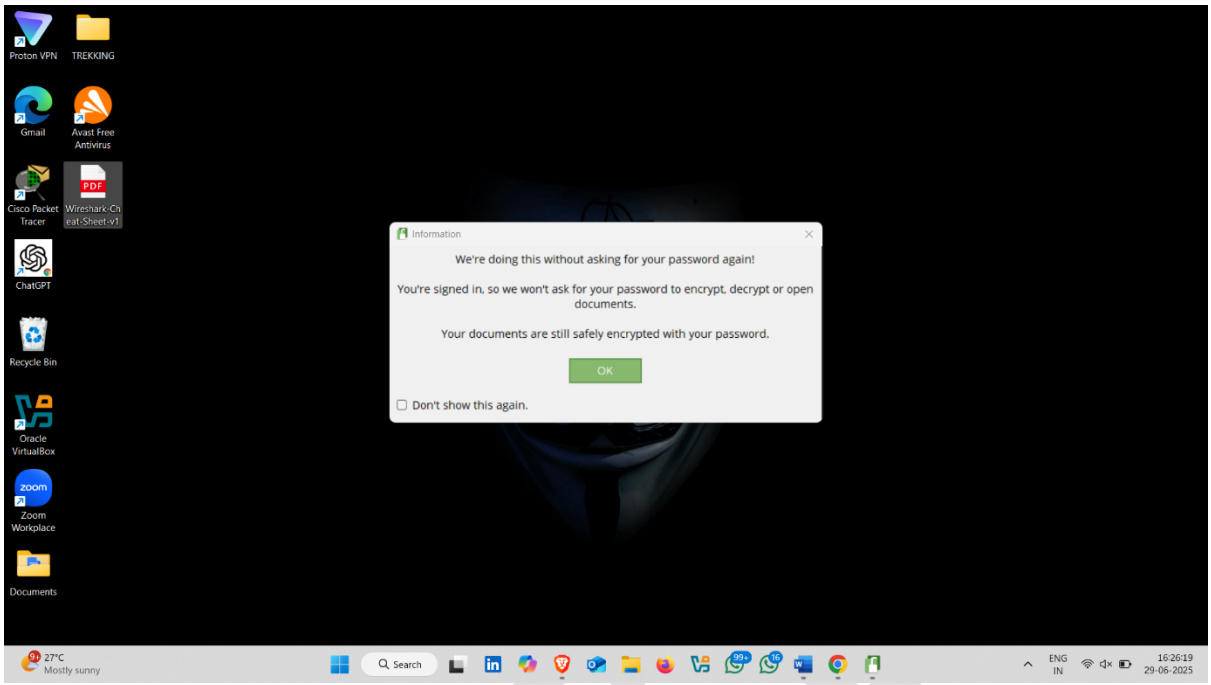
- Now click on **axcrypt**



- **Click on Encrypt**

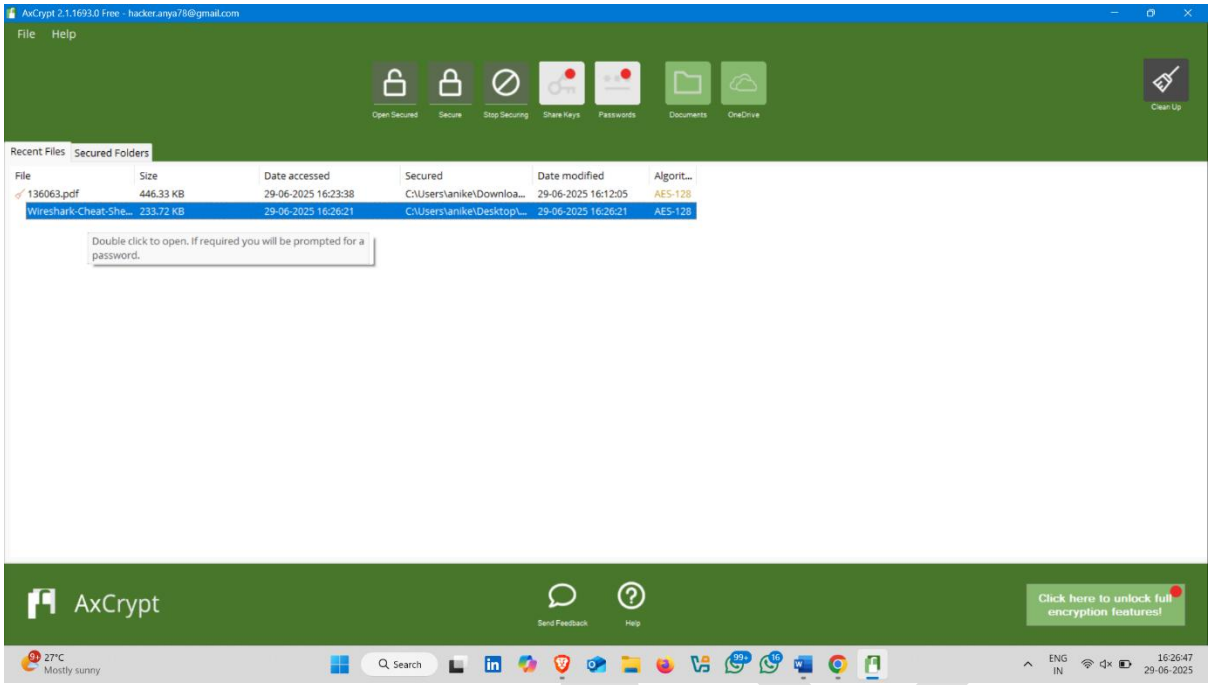


- I’m already sign-in on axcrypt application so he don’t ask me for password

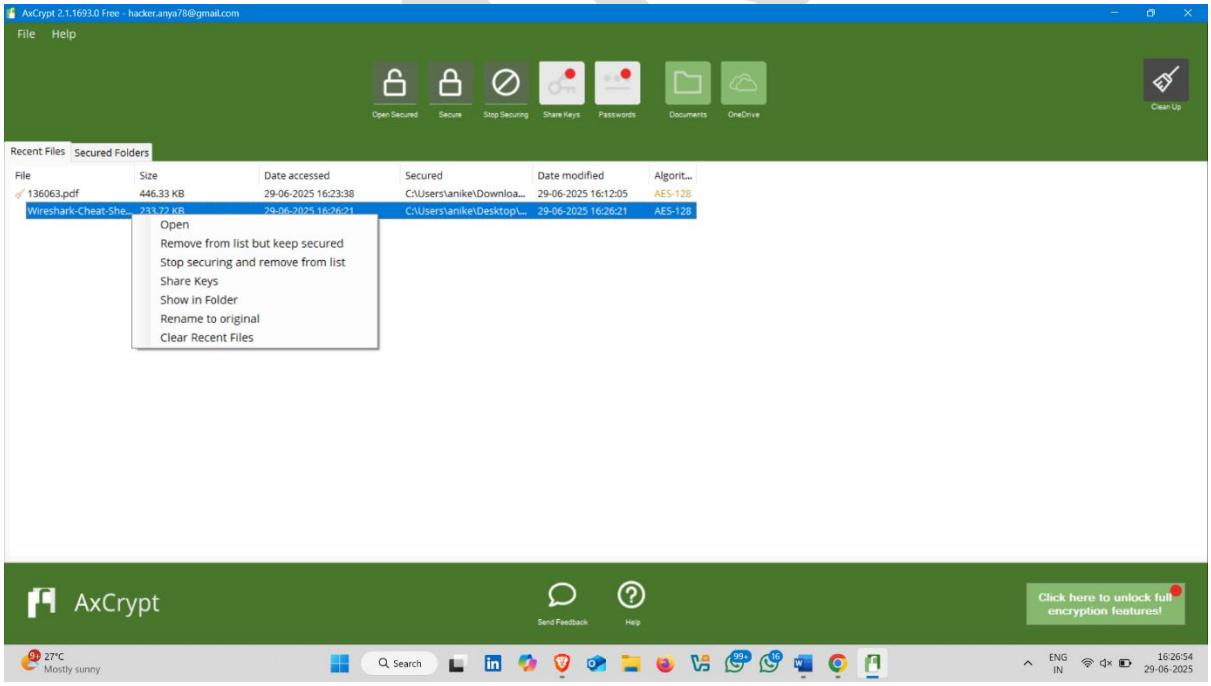


- **Folder Encrypted 👇✅**

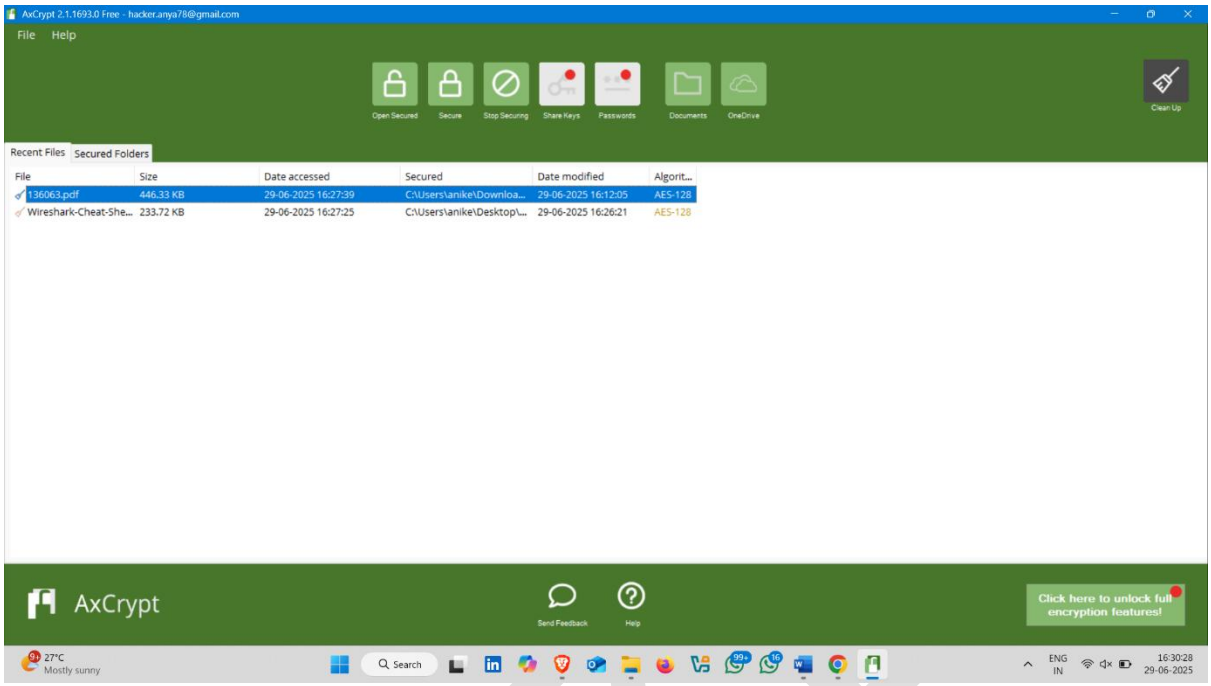
- Open axcrypt application . double click on **encrypt folder**



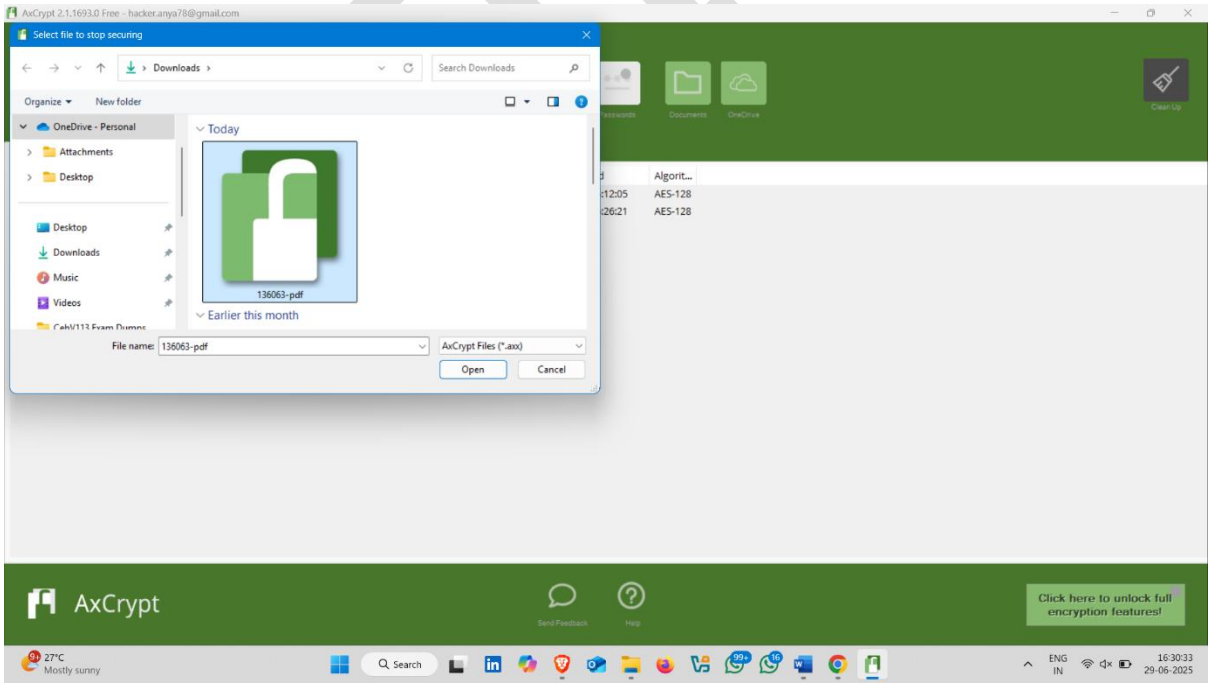
- **There is many options**



- **Now ,** to decrypt folder
- Click on **folder** and then click on **stop securing**



- And select folder , it will be decrypted



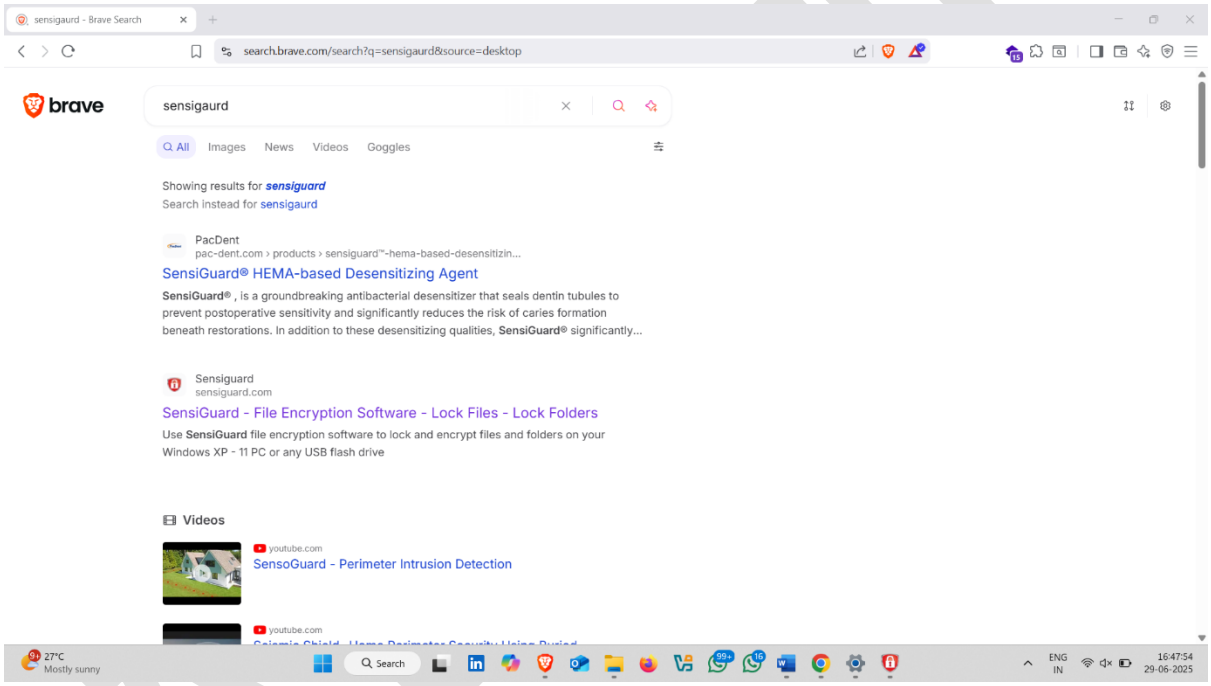
3.SensiGaurd

SensiGuard is a **Windows-based file and folder encryption tool** that helps protect your sensitive data using **strong AES encryption**. It is designed for **personal users** who want to keep their private files safe from prying eyes.

How to download it -:

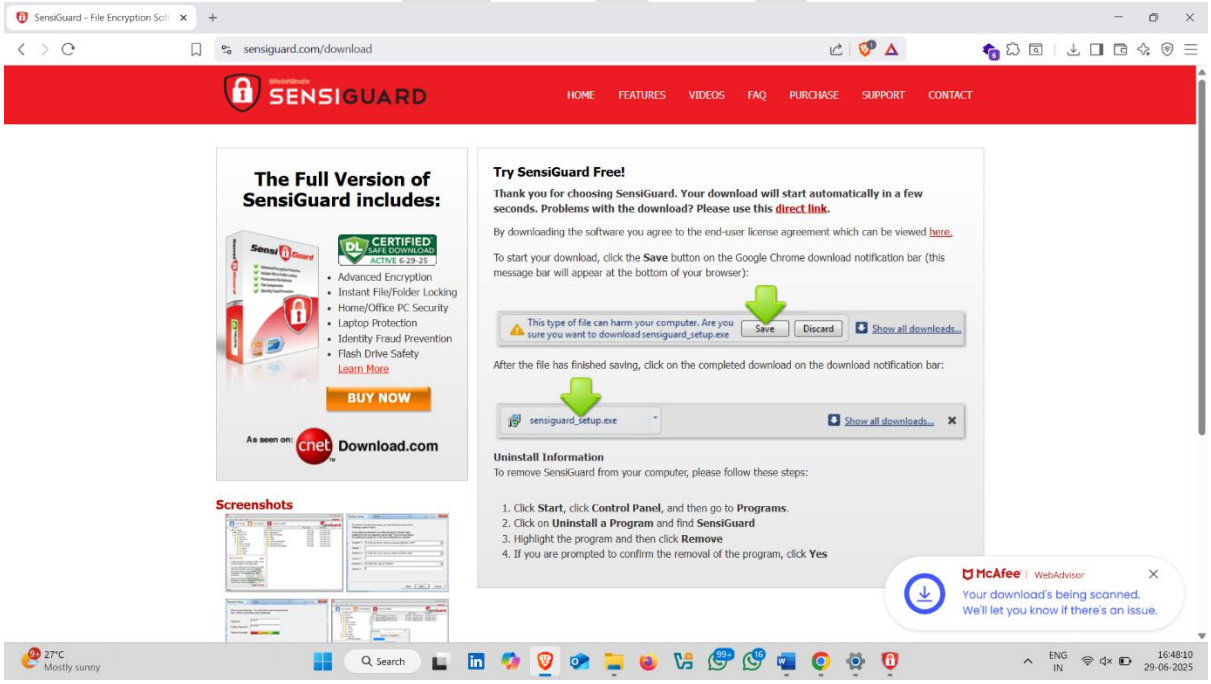
- Open Browser and search **sensiGaurd**
- Click on **second official website**

Download Link-: https://www.sensiguard.com/download



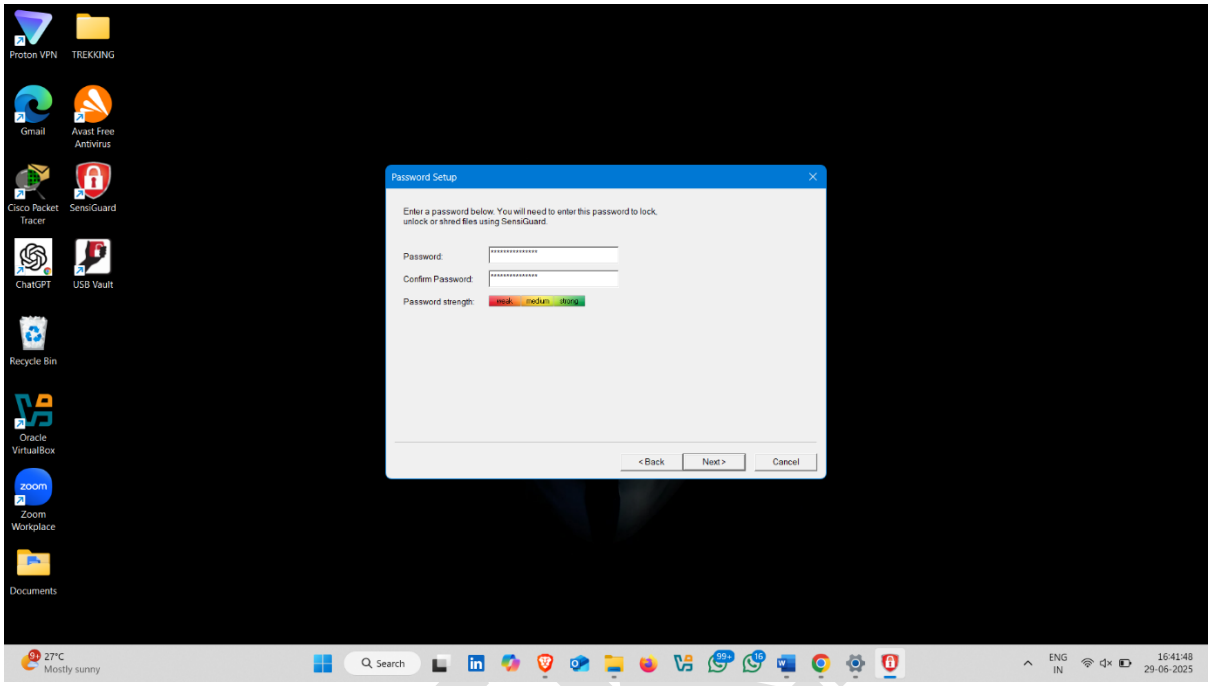
- Click on **Download button**

- Wait , it will started Downloading automatically

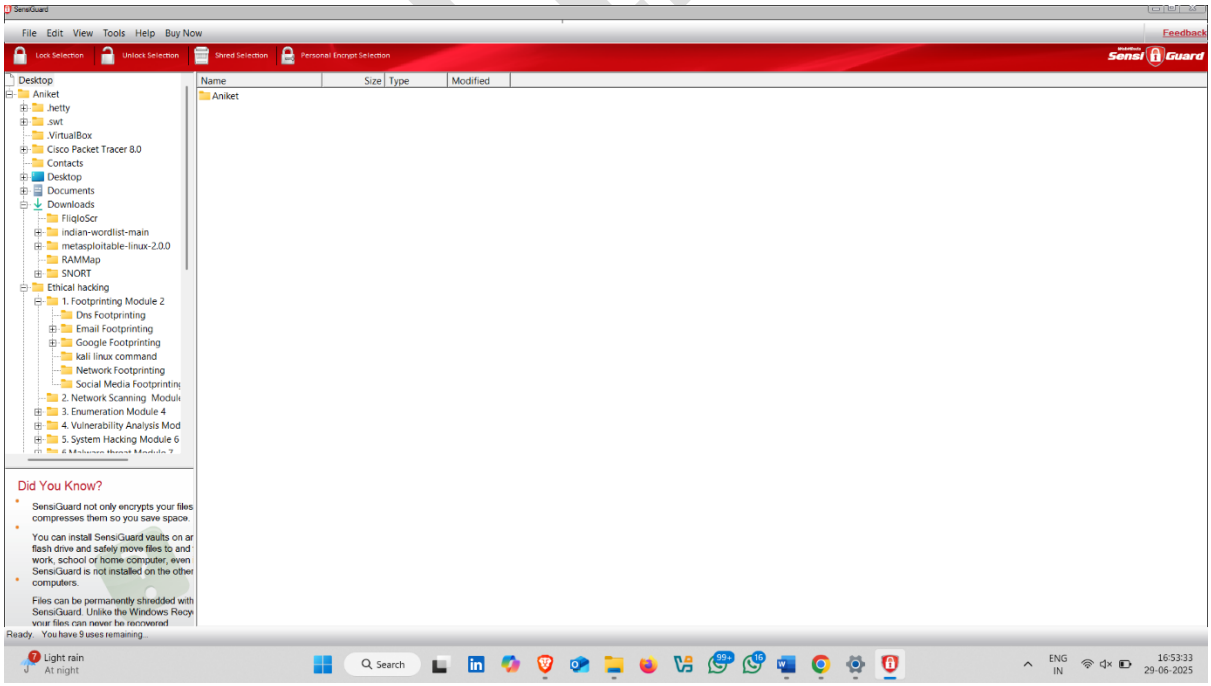


How to use it -:

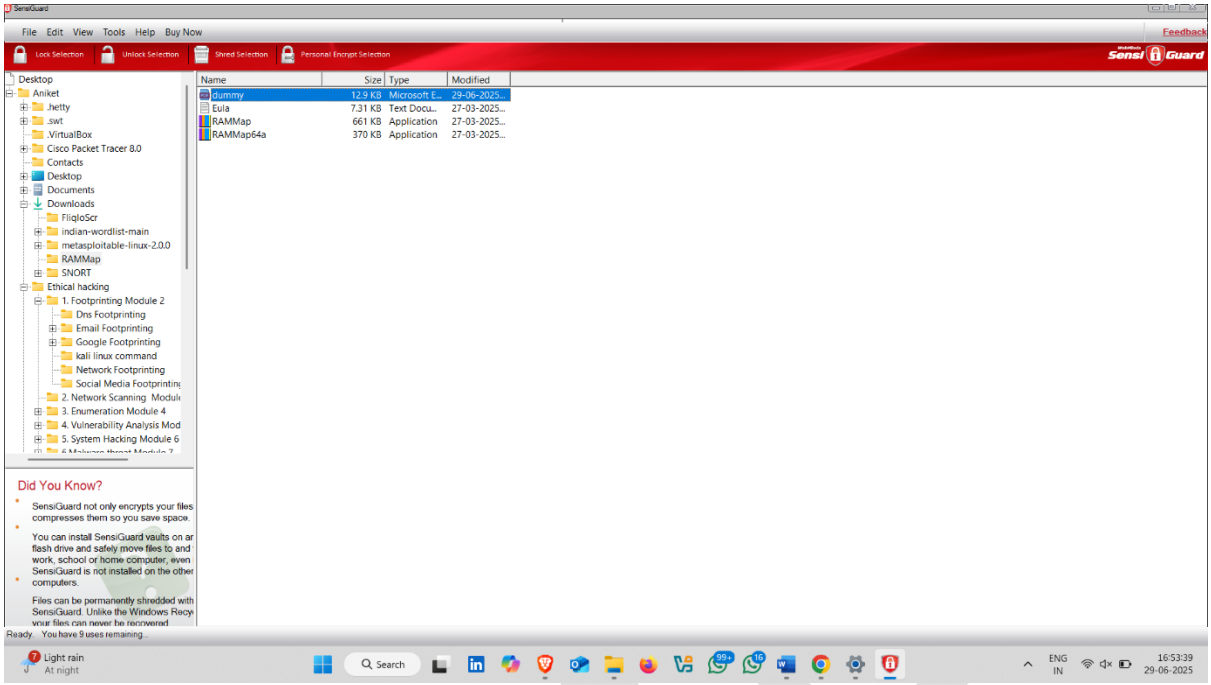
- **After installing , started sensiGaurd Application**
- **During set-up application , set a strong password ✅ 👇**



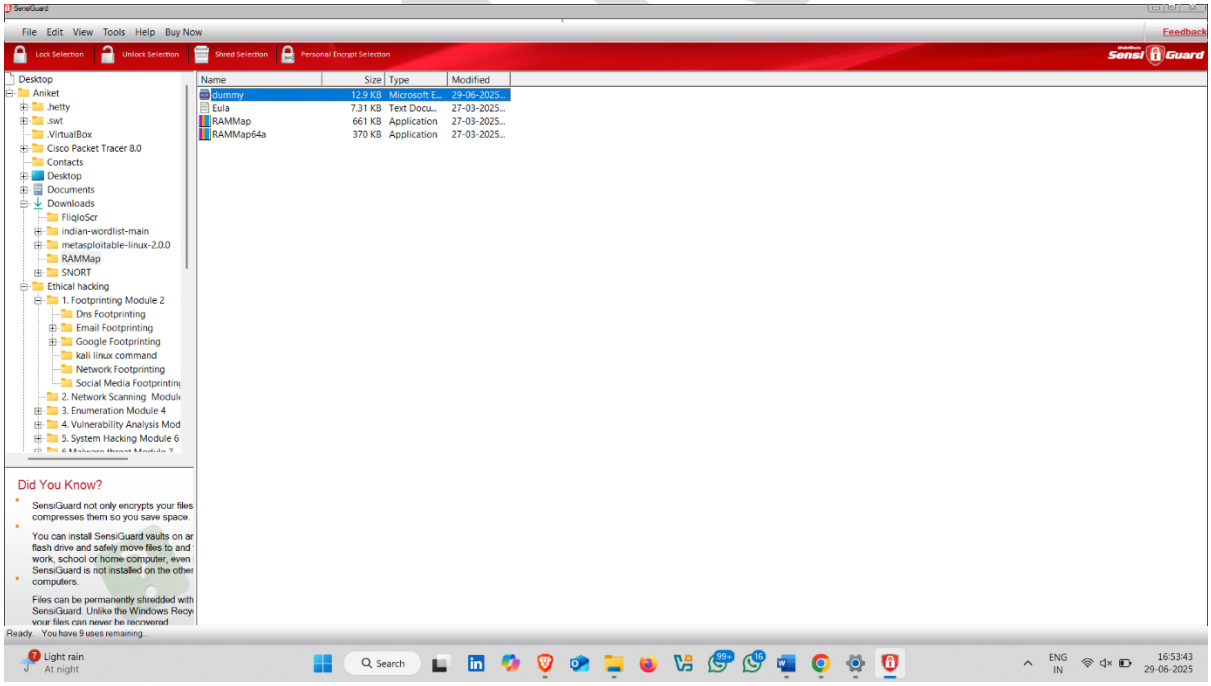
Now open the app



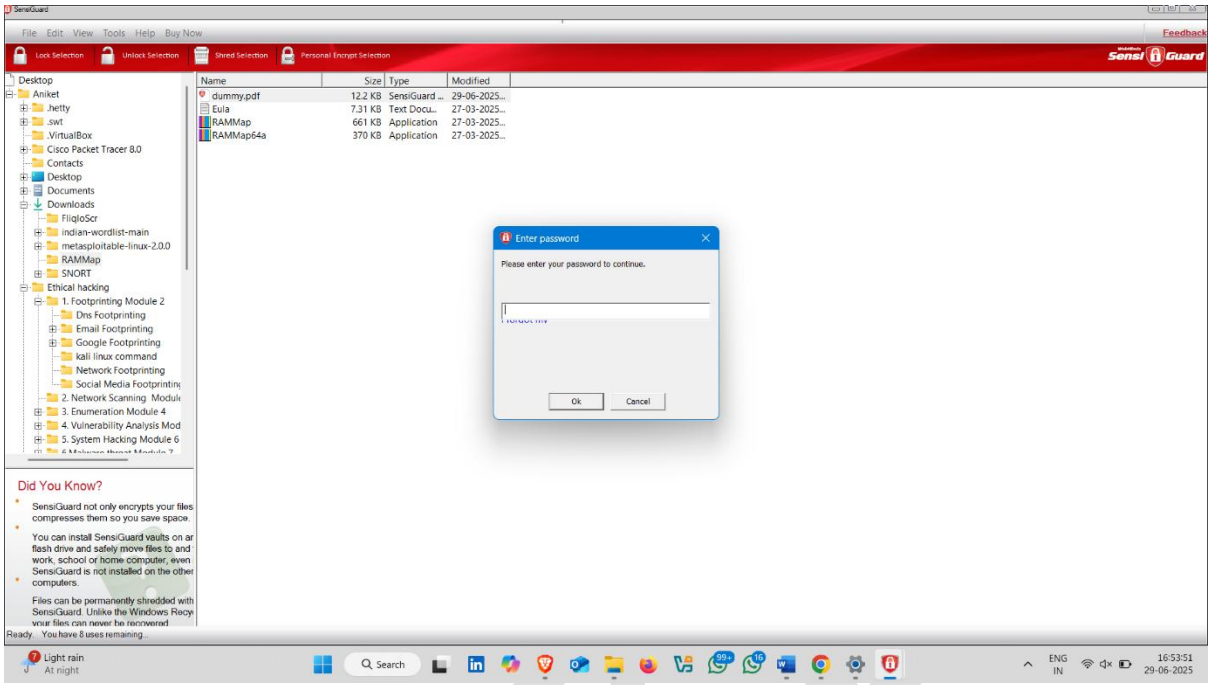
- Select File or Folder for encryption



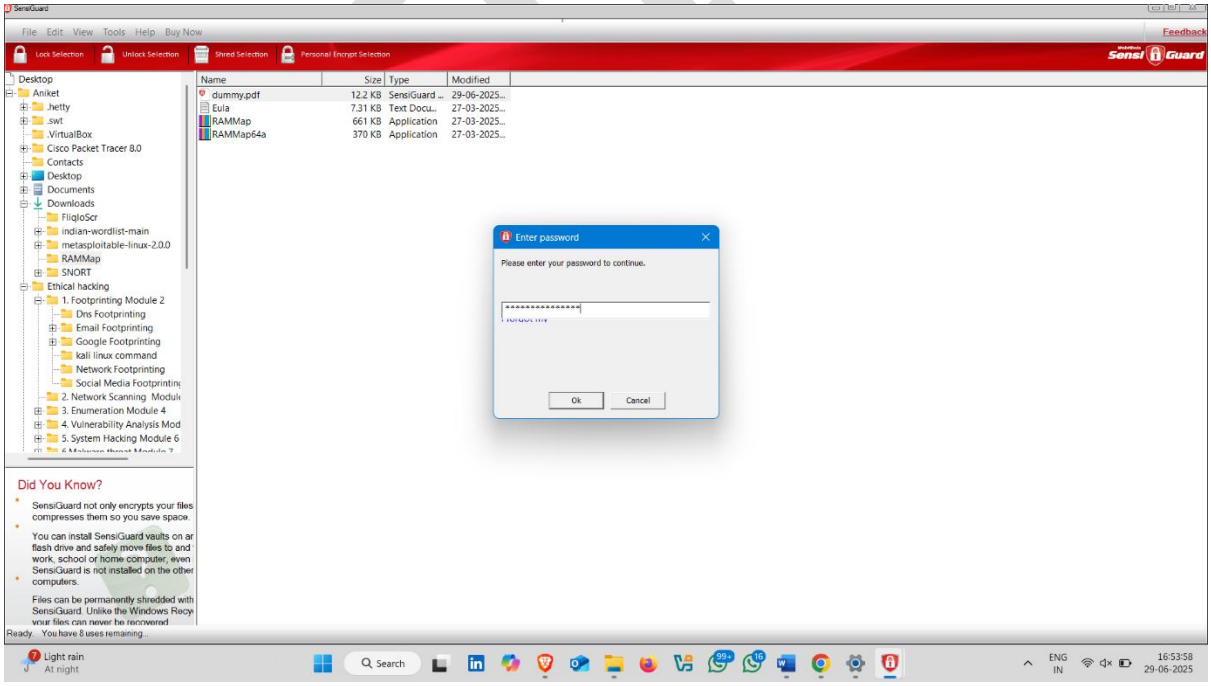
- After selection click on , **lock section** and it will be encrypted



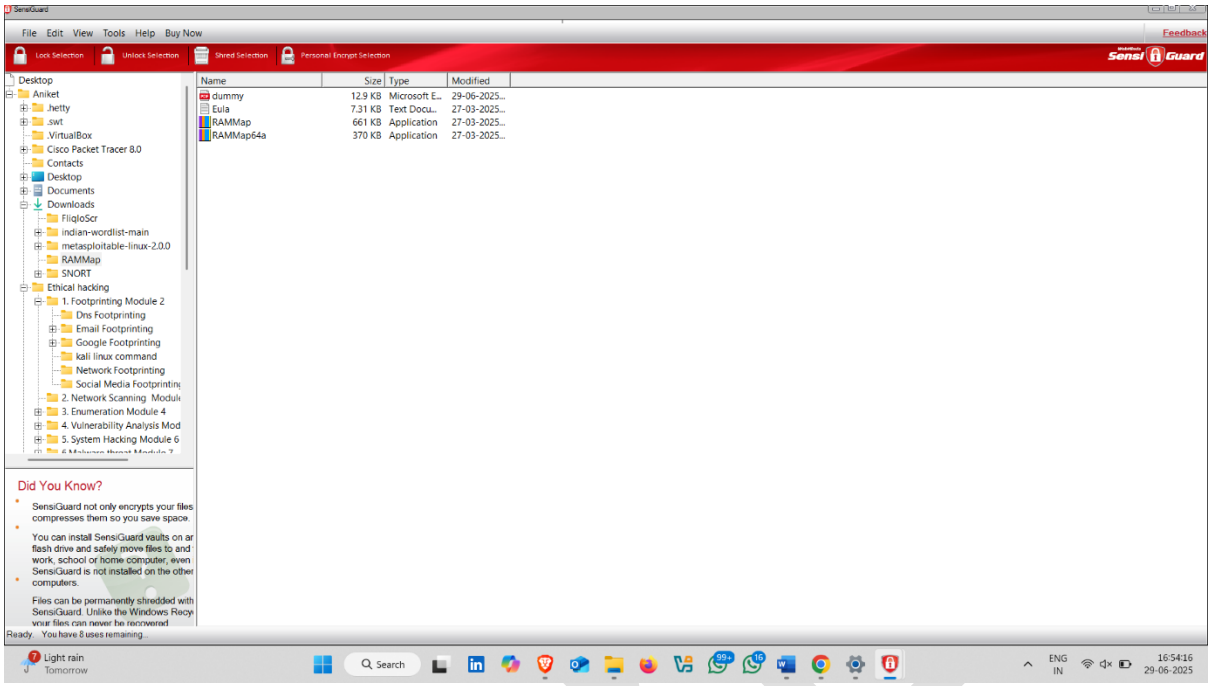
- To Decrypt It , click on Folder and enter password that you enter during installation



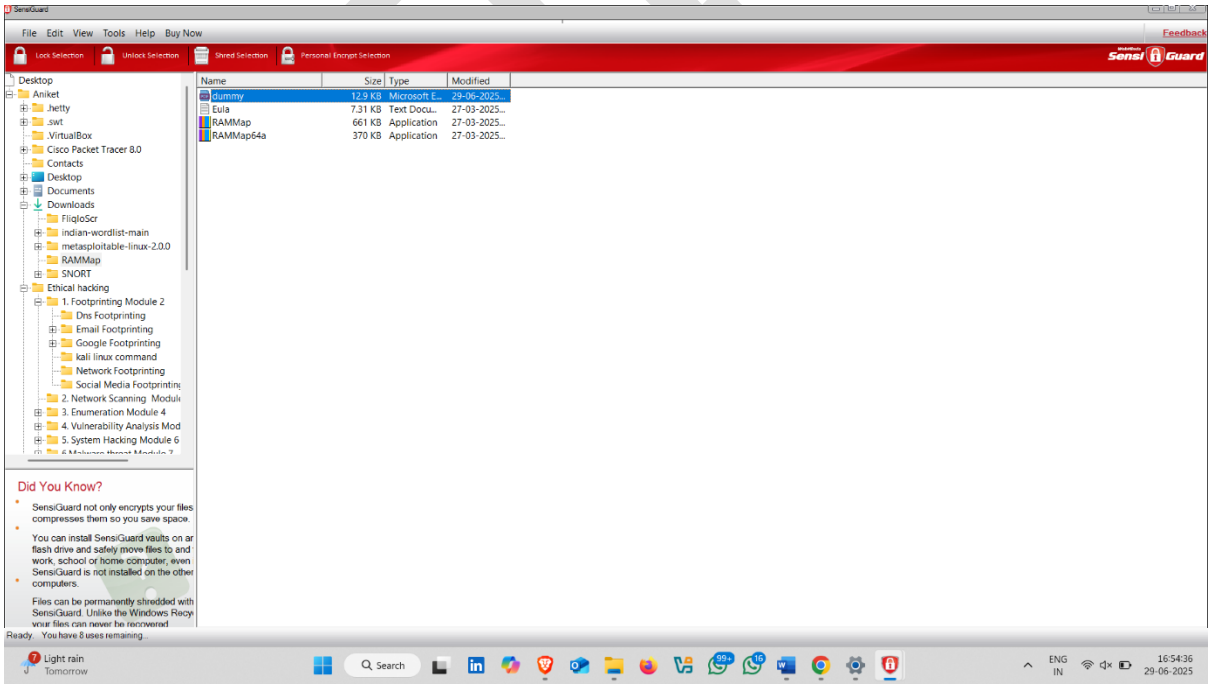
- Click on **ok**



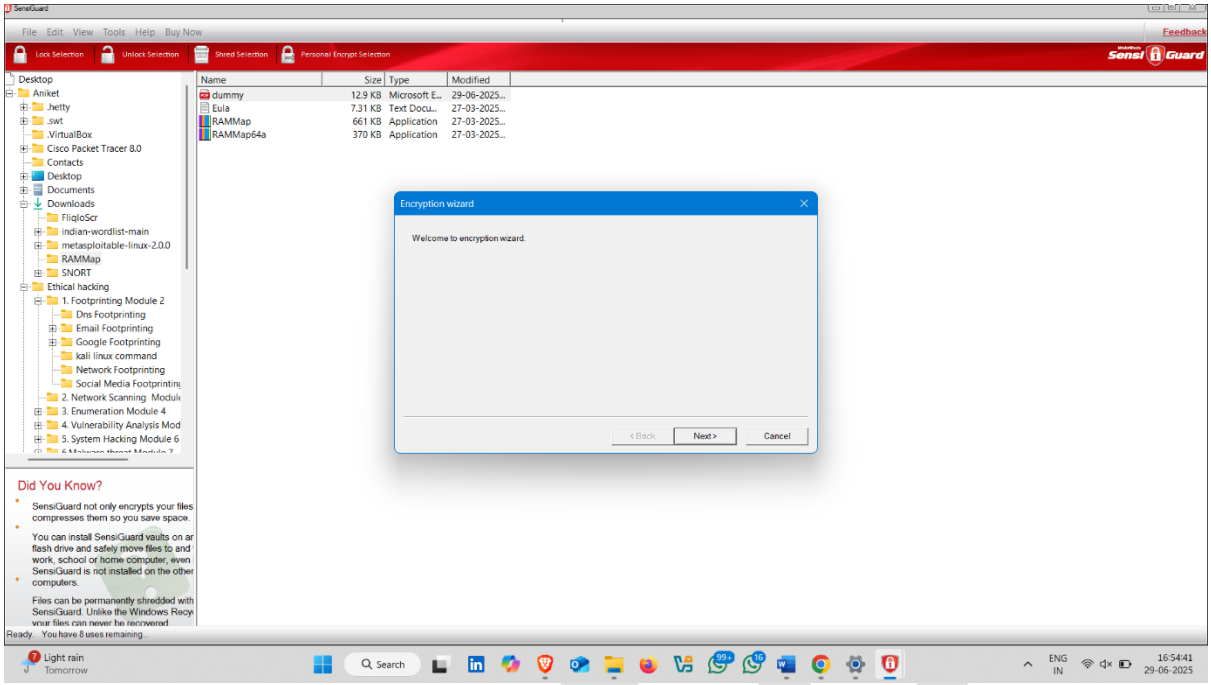
- Here , it unlocked 👇✅



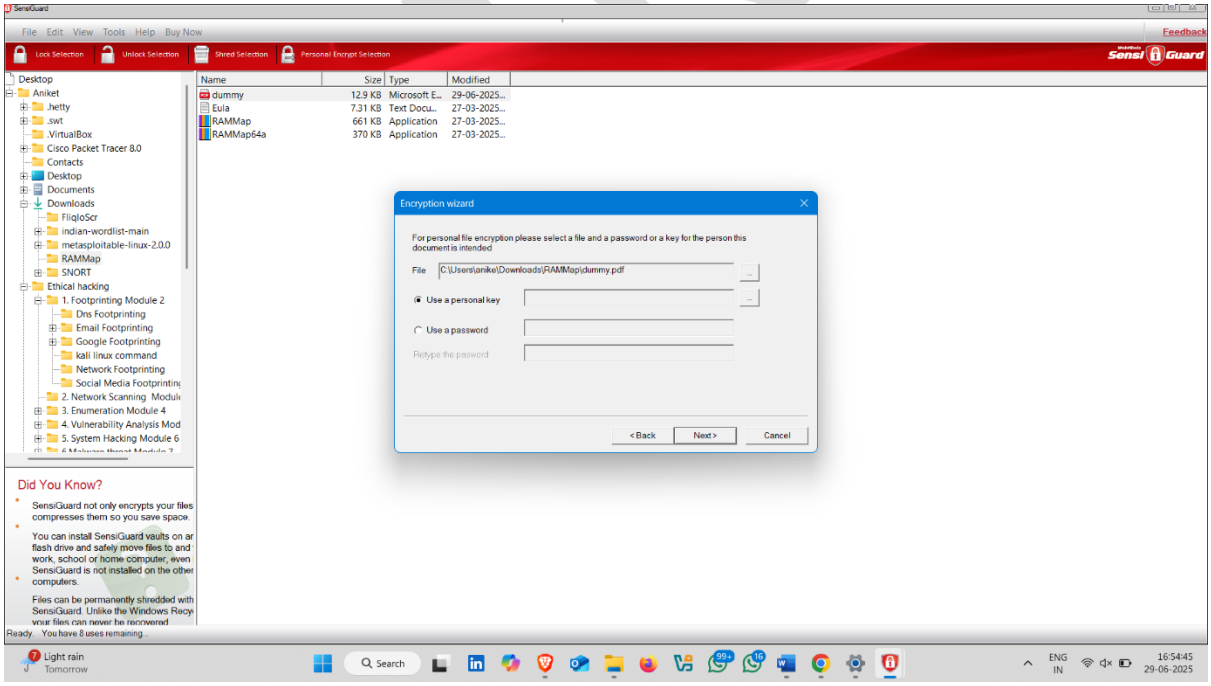
- You can also set different password
- Click on file , and then click on **personal Encryption Selection**



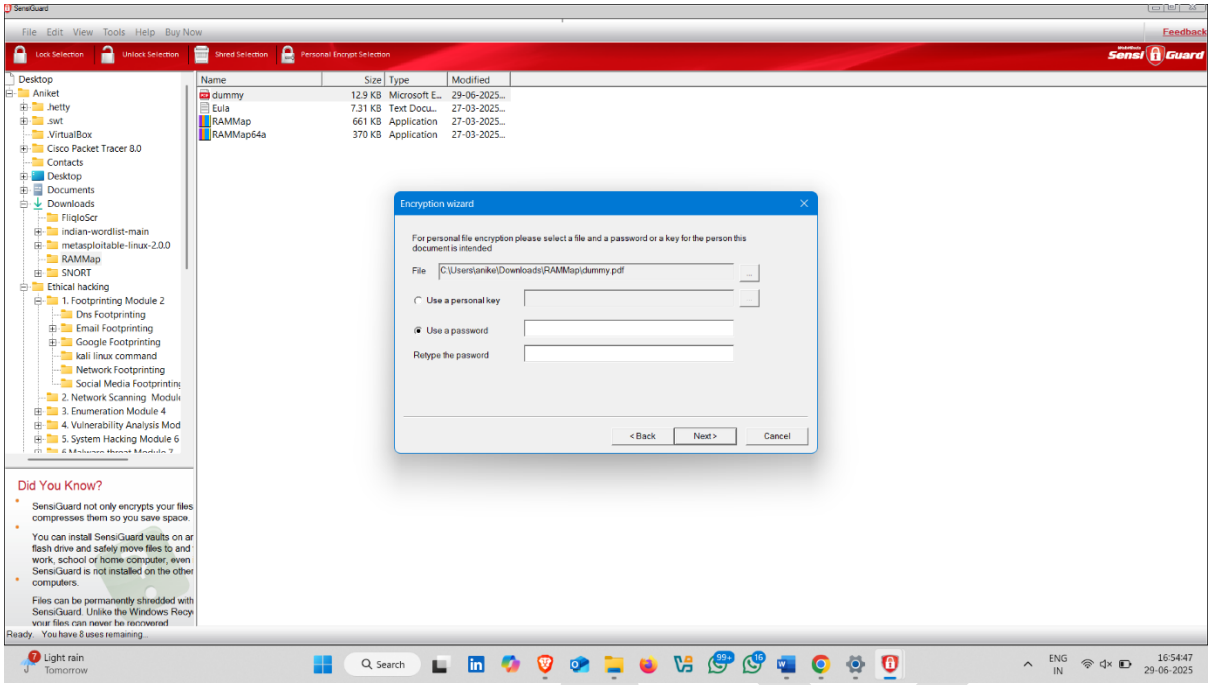
- Click on **next**



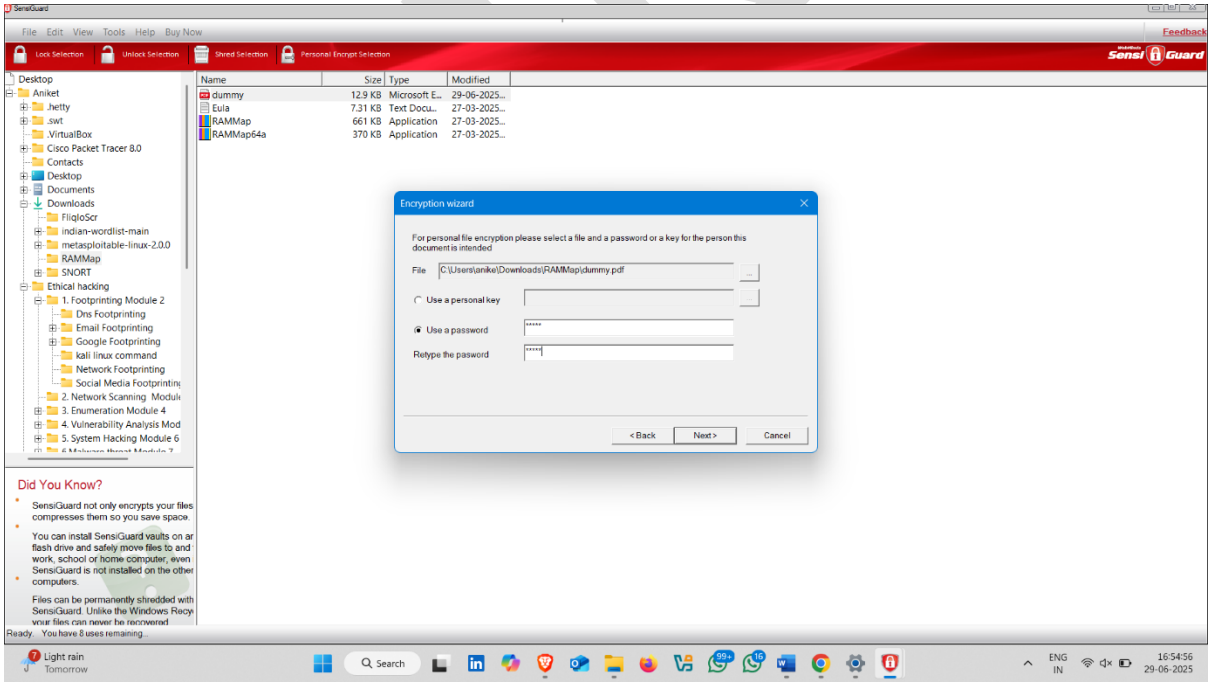
- Select **personal key** or **password**



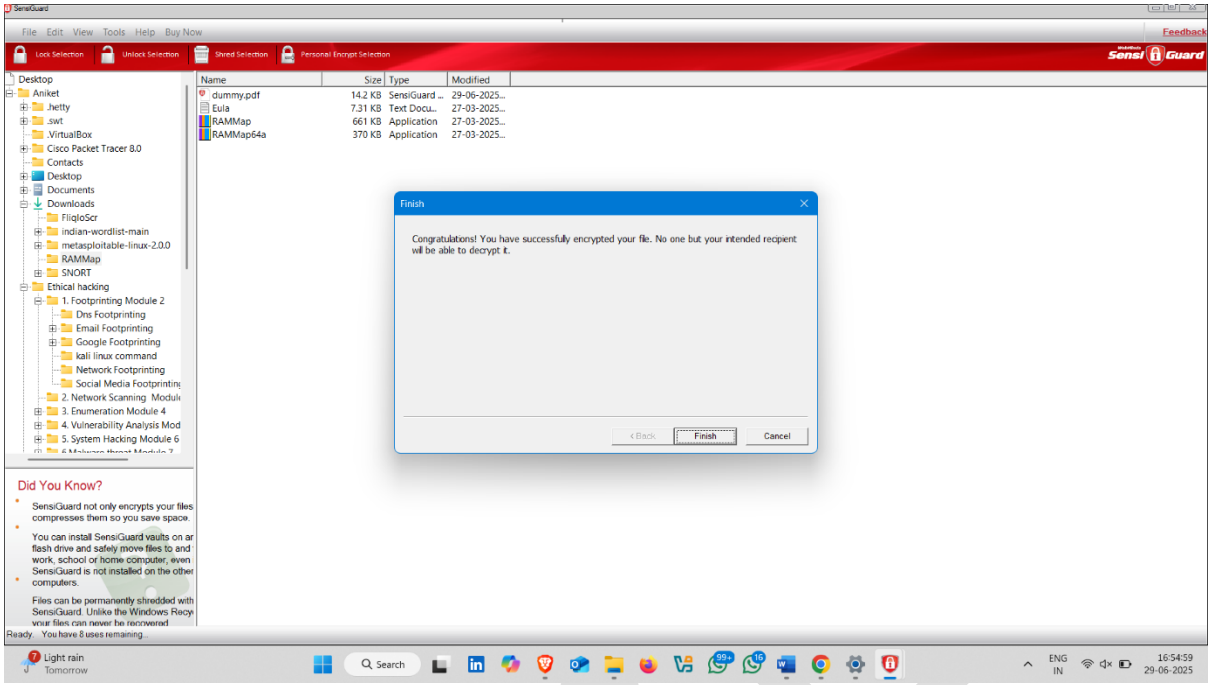
- Set password



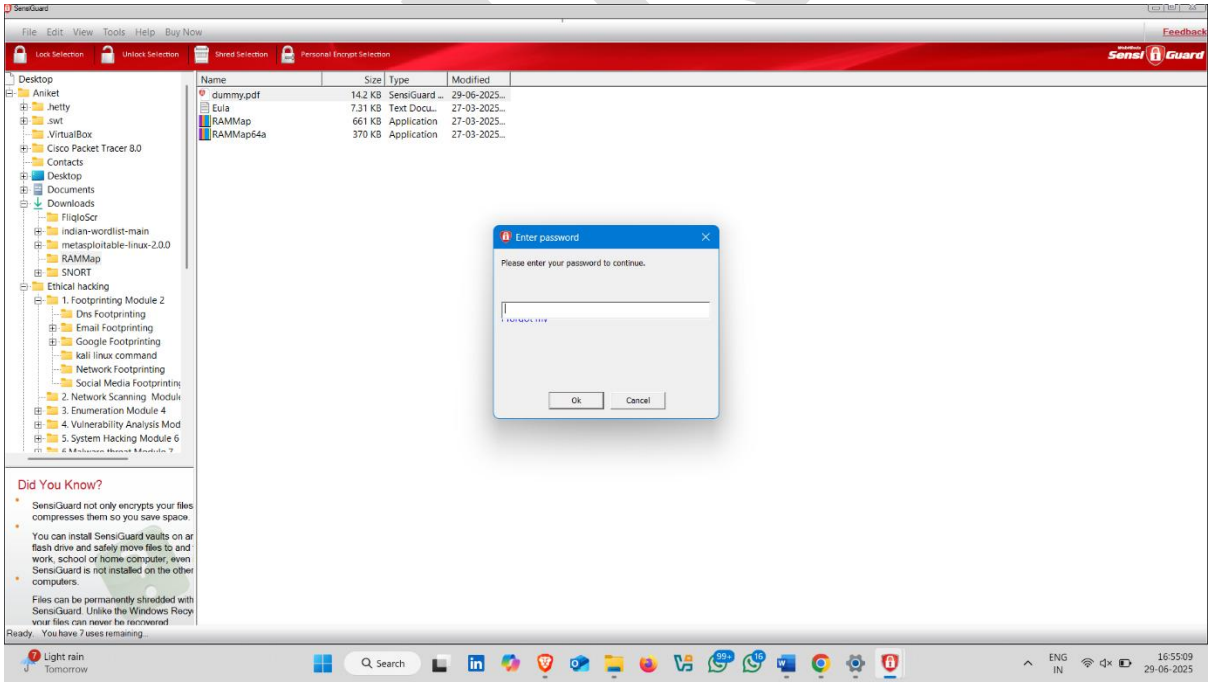
- Click on next



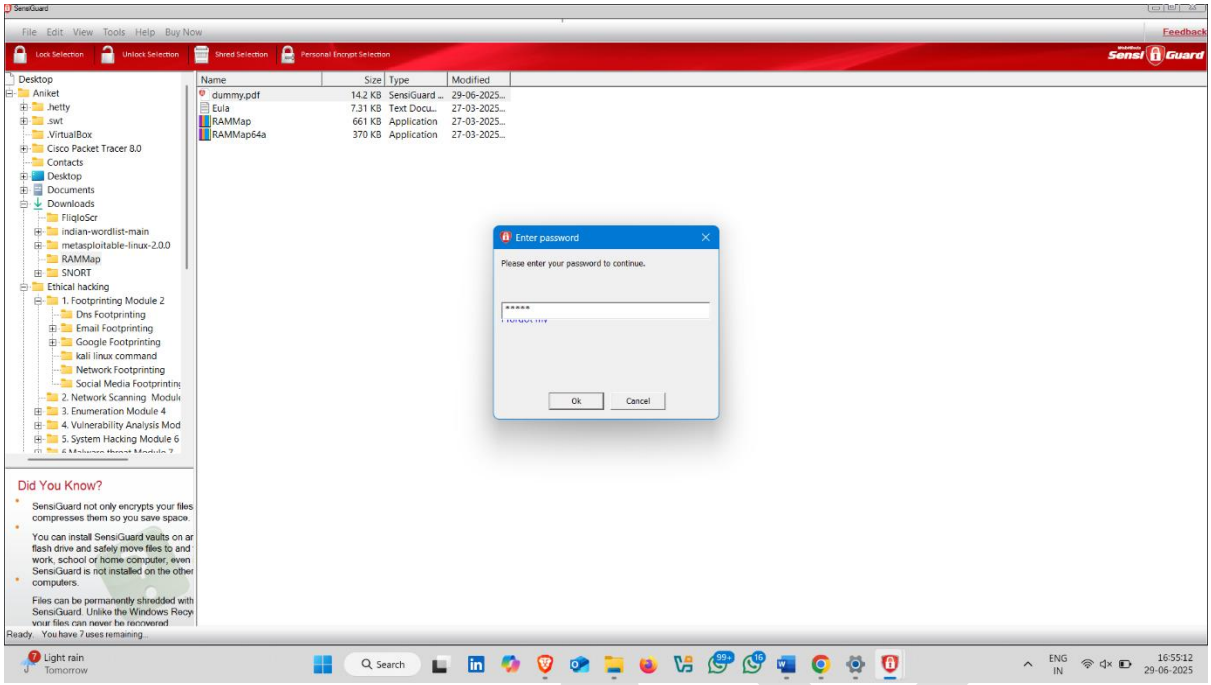
- Click on **finish**

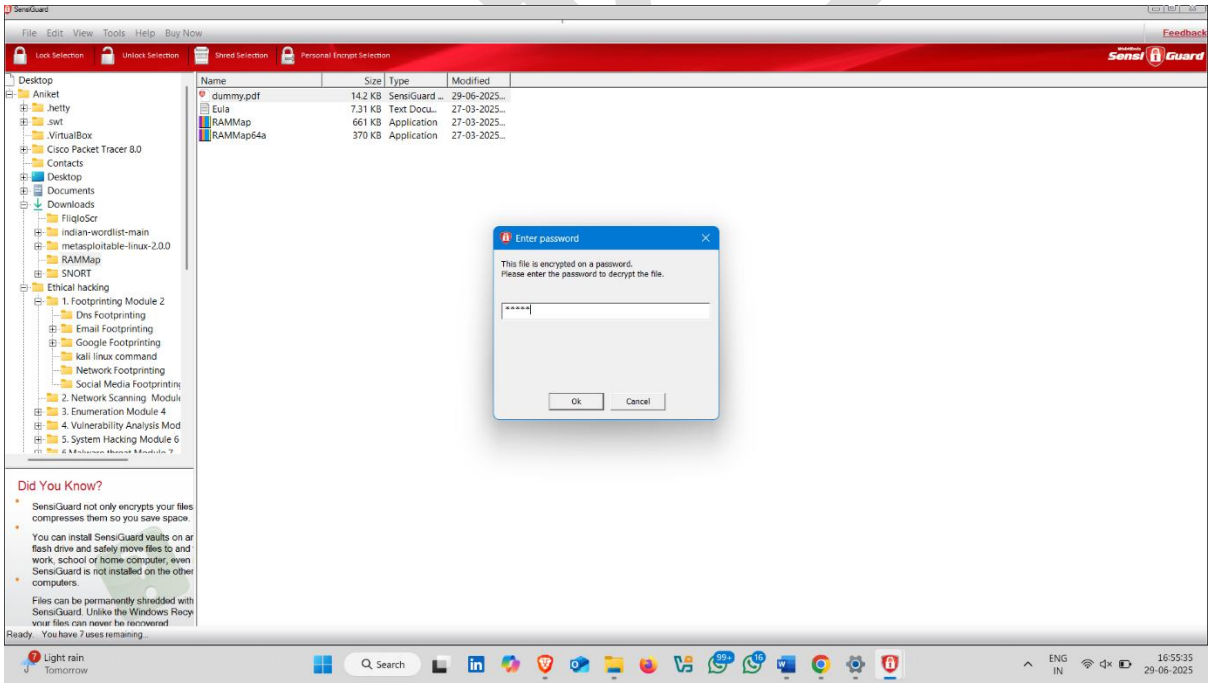


- Now , click on file and enter password

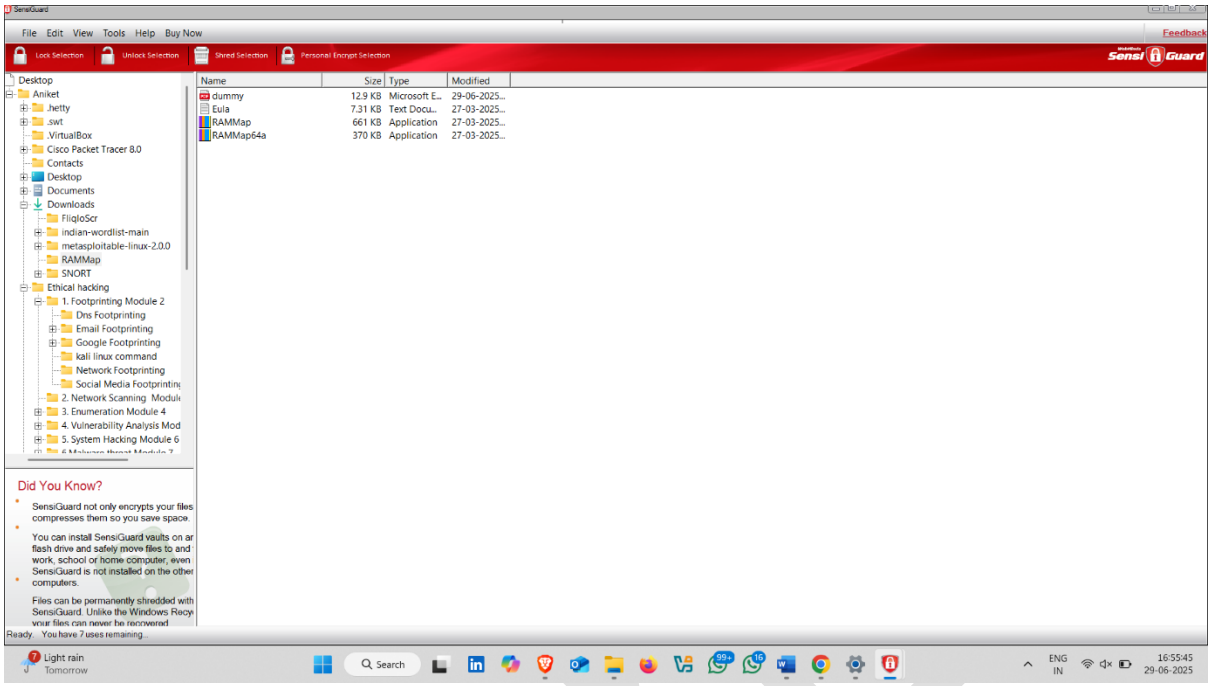


- Click on **ok**


Enter password
Please enter your password to continue.
*****
Ok
Cancel


Enter password
This file is encrypted on a password.
Please enter the password to decrypt the file.
*****
Ok
Cancel

- **File unlocked ✅ 👇**

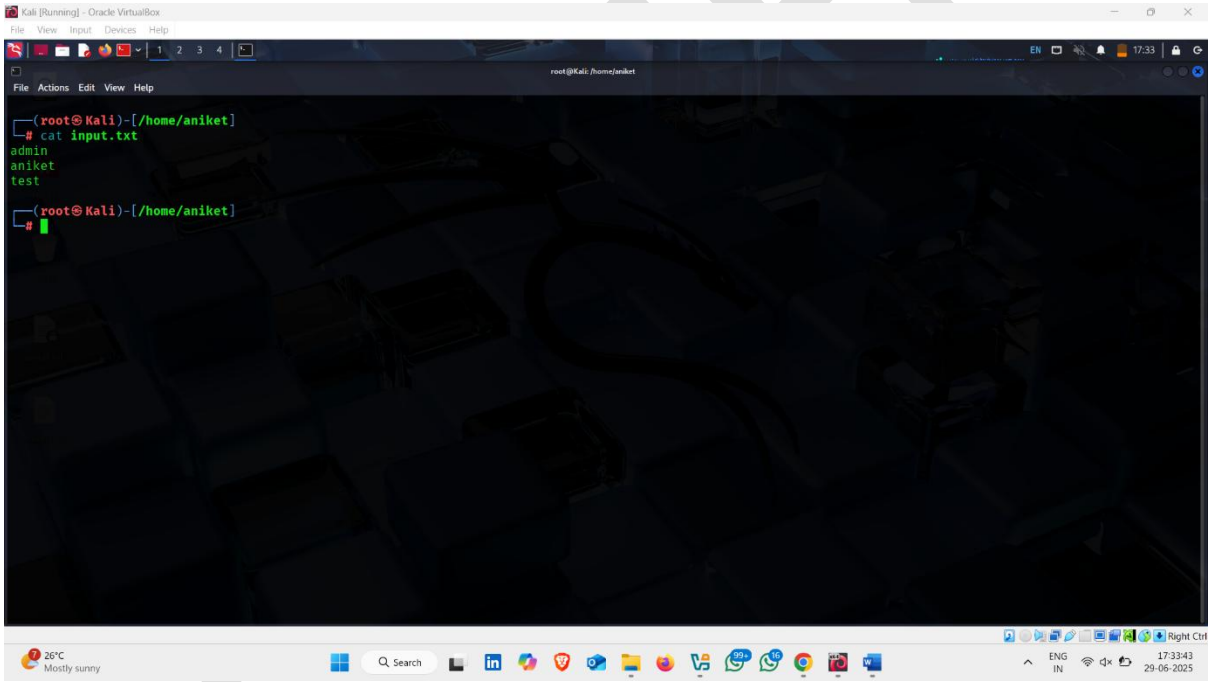


4.Openssl

OpenSSL is a powerful, open-source **cryptographic toolkit and library** used for implementing **SSL (Secure Sockets Layer)** and **TLS (Transport Layer Security)** protocols, as well as performing a wide range of **cryptographic operations** like encryption, decryption, certificate generation, key management, and secure communication.

How to use it -:

- Open kali linux/ParrotOs terminal and put some words in txt file
- **Input.txt file** 👇✅



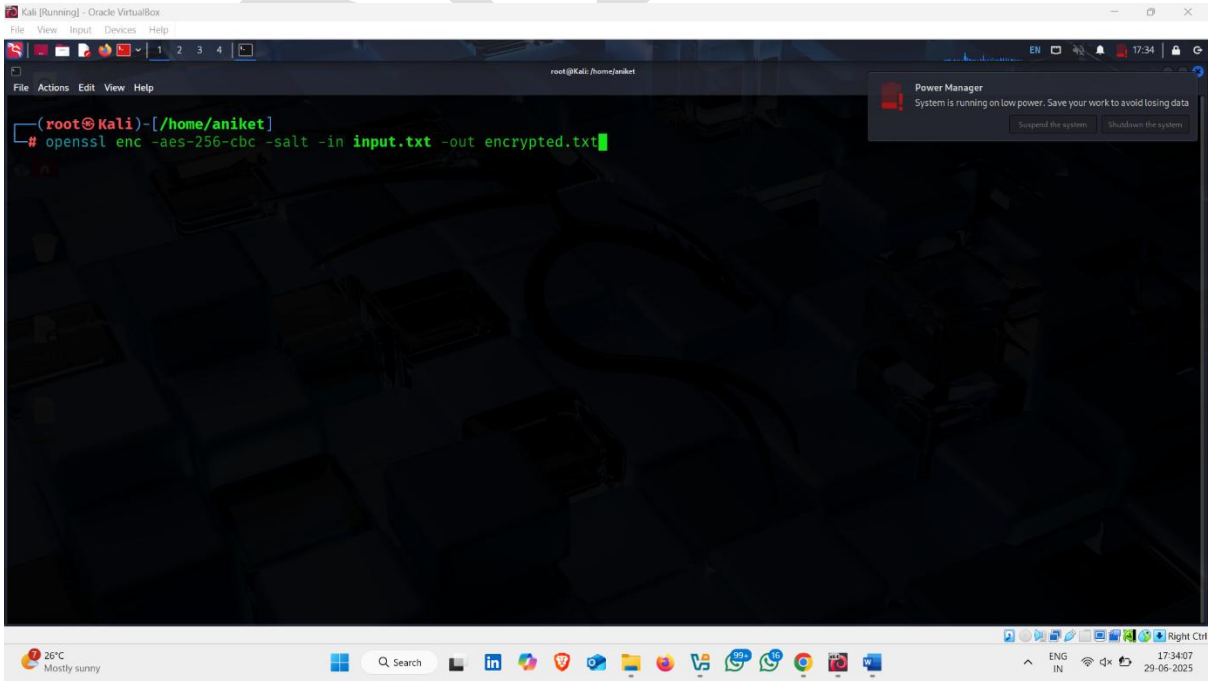
- **Type following command**

Command -: openssl enc -aes-256-cbc -salt -in plain.txt -out encrypted.txt

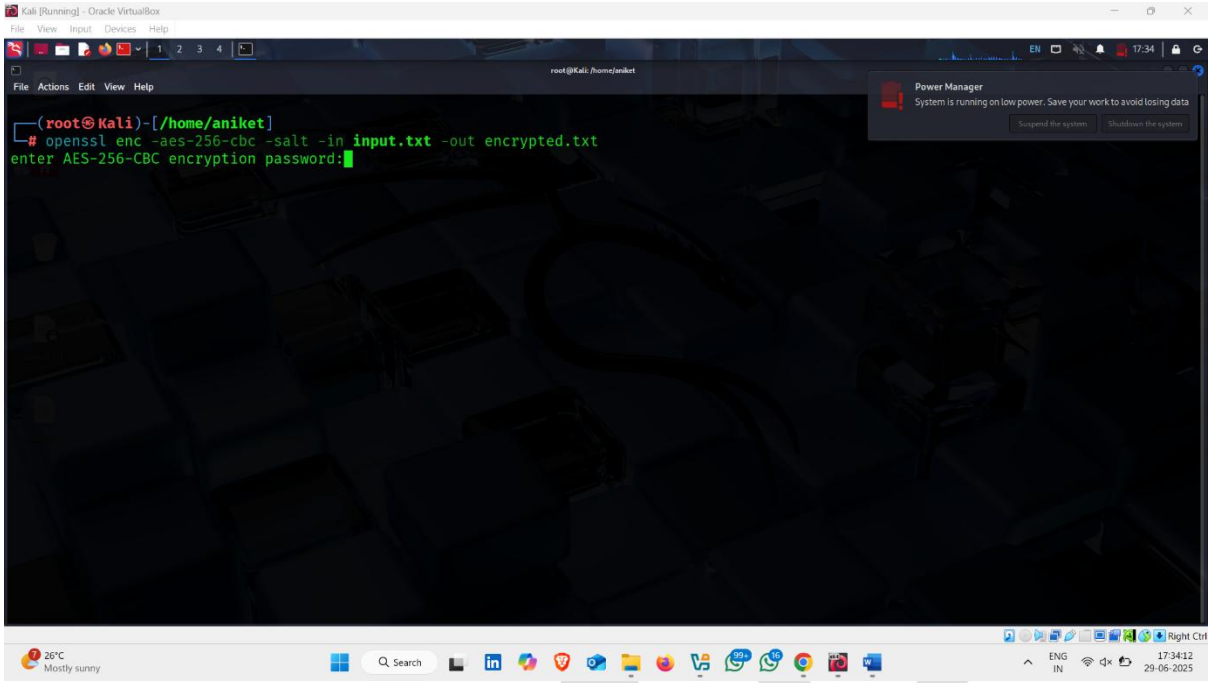
Explanation -: This command encrypts a file named plain.txt using the **AES-256-CBC (Cipher Block Chaining)** symmetric encryption algorithm**.**

🧩 **Breakdown of Options:**

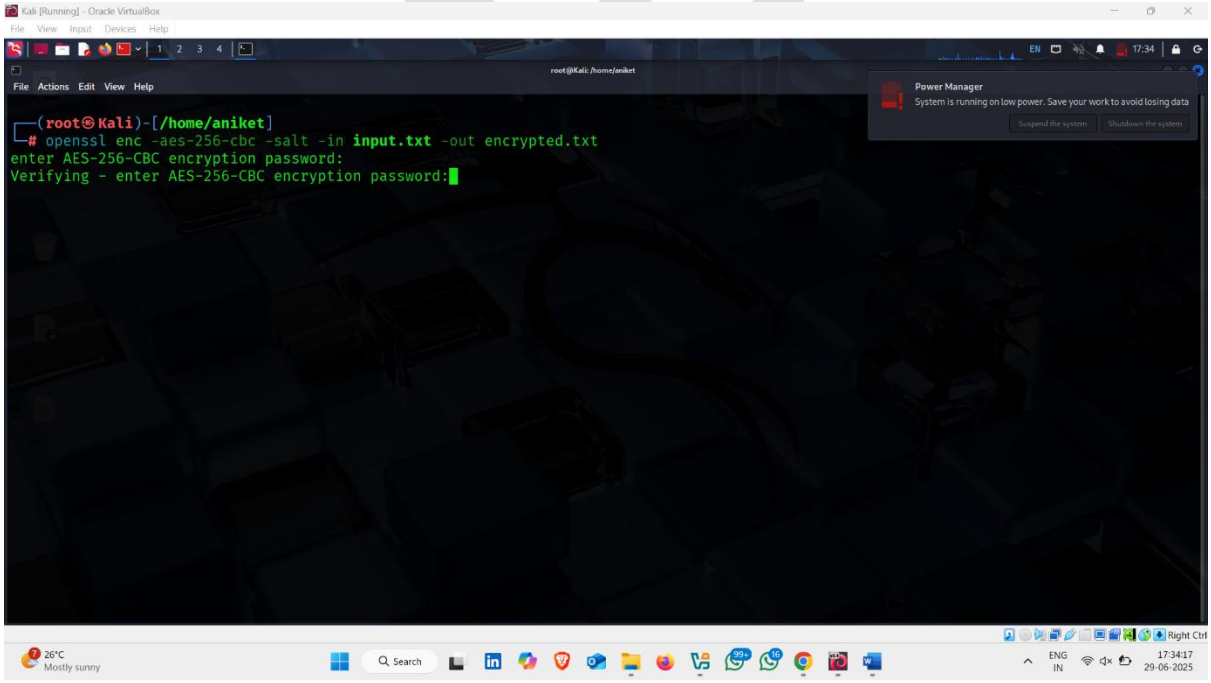
Option	Description
openssl enc	Uses the OpenSSL encryption tool.
-aes-256-cbc	Algorithm: AES with 256-bit key in CBC mode.
-salt	Adds a cryptographic salt to protect against rainbow table attacks (recommended).
-in plain.txt	Input file to encrypt.
-out encrypted.txt	Output file for the encrypted data.



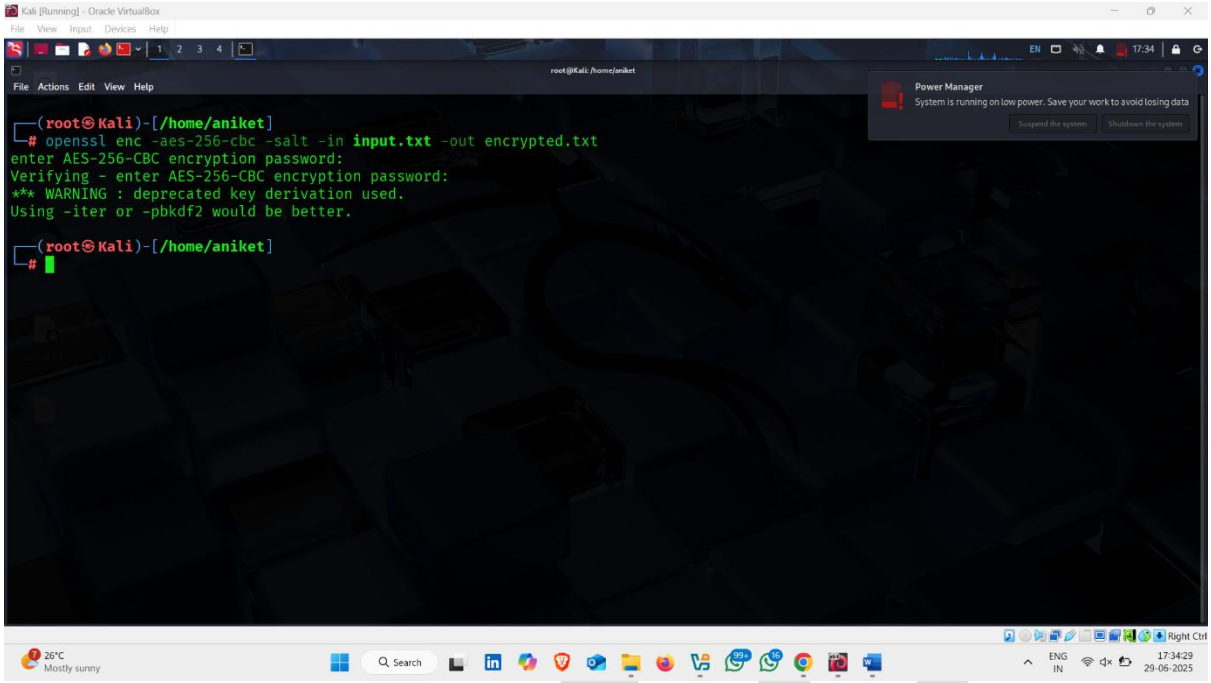
- Set a password



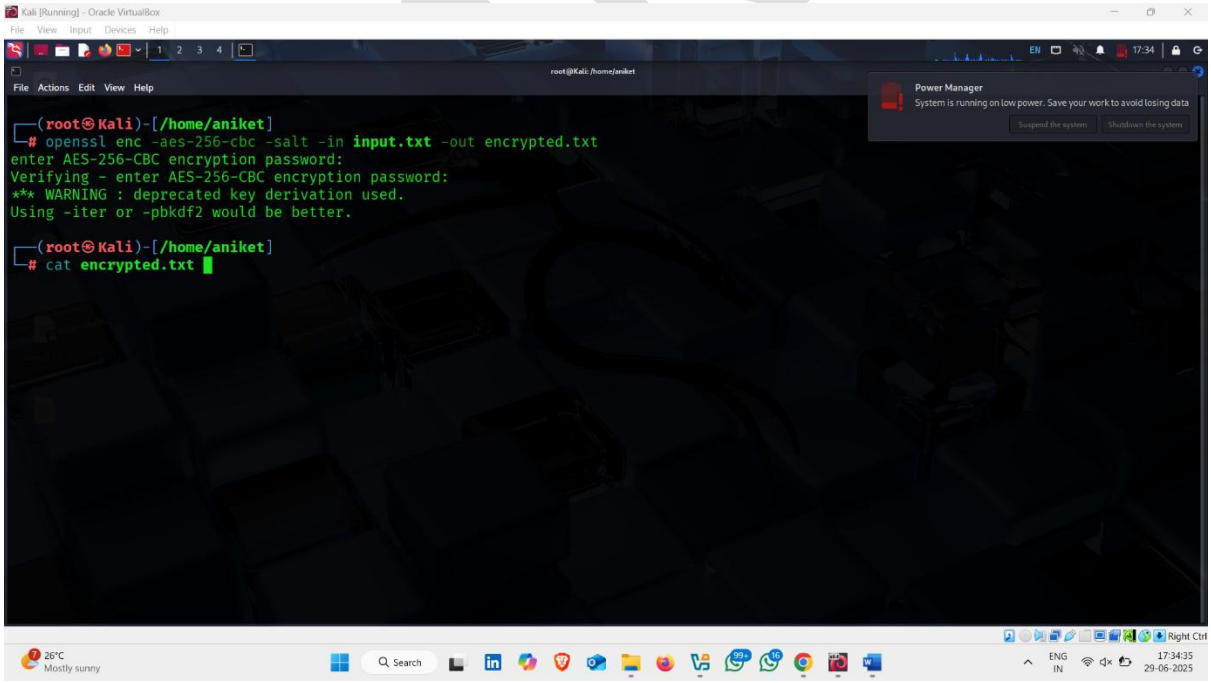
- Re-enter password ✅



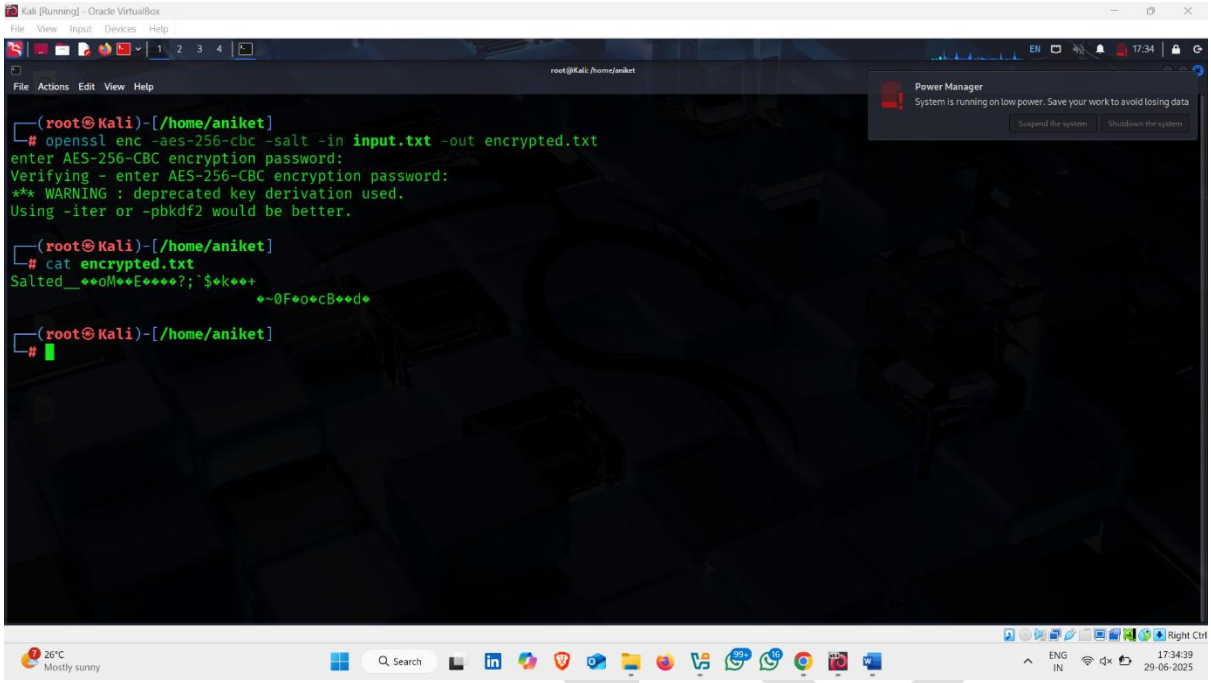
- Here ,it encrypted the input.txt file content



- Now , using **cat** command check it



- **Encryption ✅ 👇**



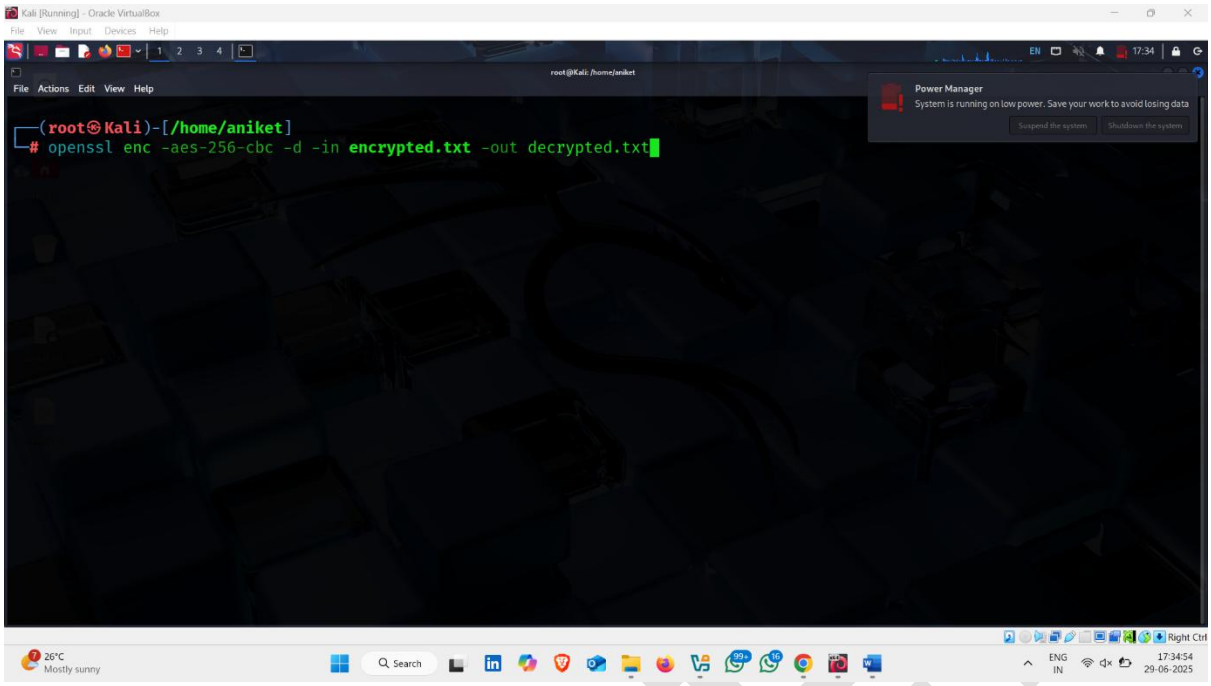
- Now , try to decrypt the hash
- type following command

Command -: openssl enc -aes-256-cbc -d -in encrypted.txt -out decrypted.txt

Explanation -: This command decrypts the previously encrypted file **encrypted.txt** and writes the original content back into **decrypted.txt**.

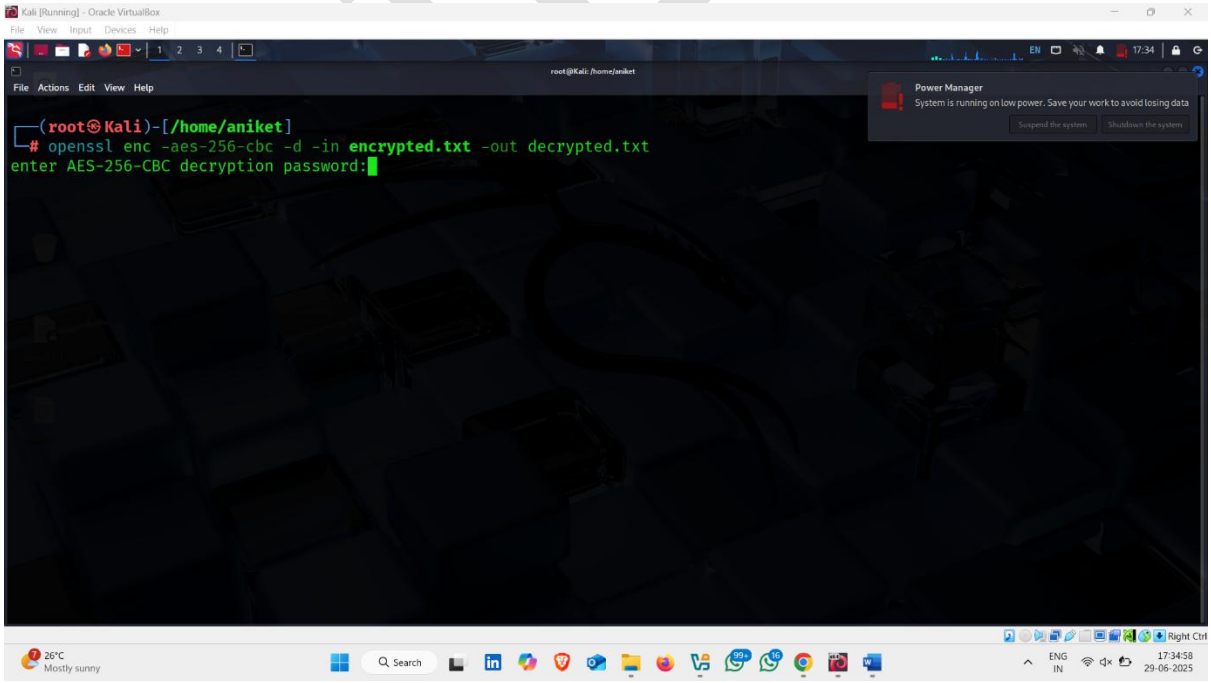
🧩 **Breakdown of Options:**

Option	Description
-aes-256-cbc	Uses the same AES-256-CBC cipher for decryption.
-d	Tells OpenSSL to decrypt (instead of encrypt).
-in encrypted.txt	File that contains the encrypted data.
-out decrypted.txt	Where the decrypted result will be stored.

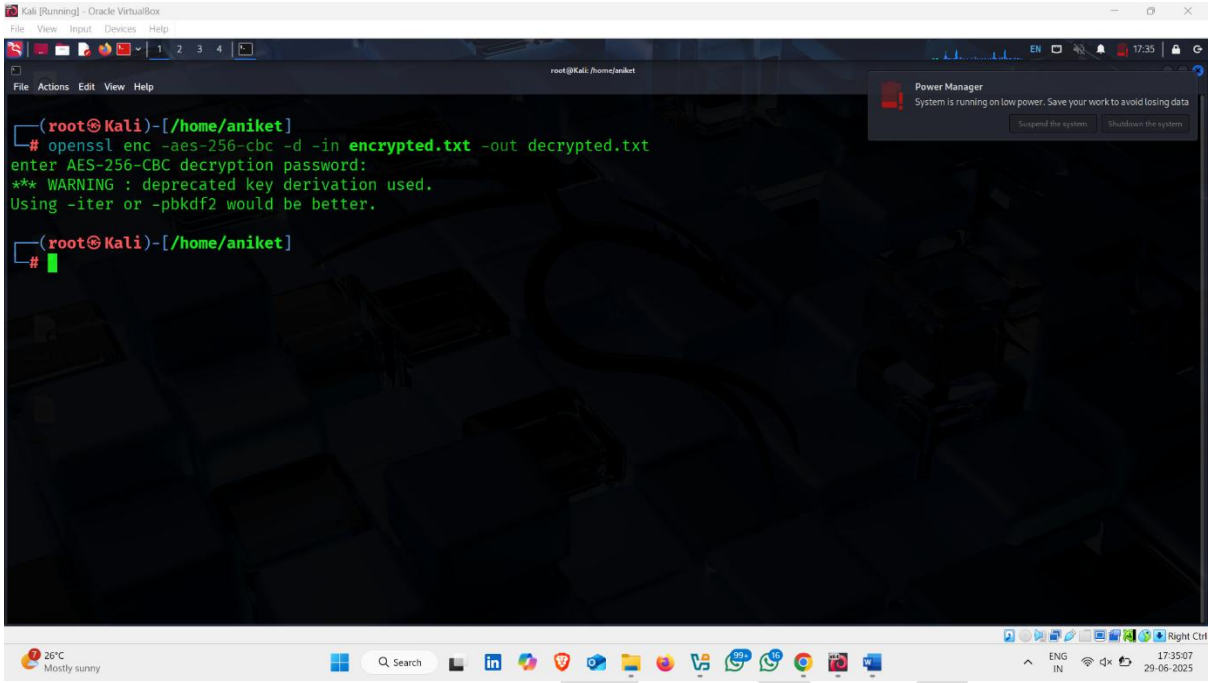


🔑 You must enter the same password used during encryption. If the password or cipher doesn't match, decryption will fail or produce junk output.

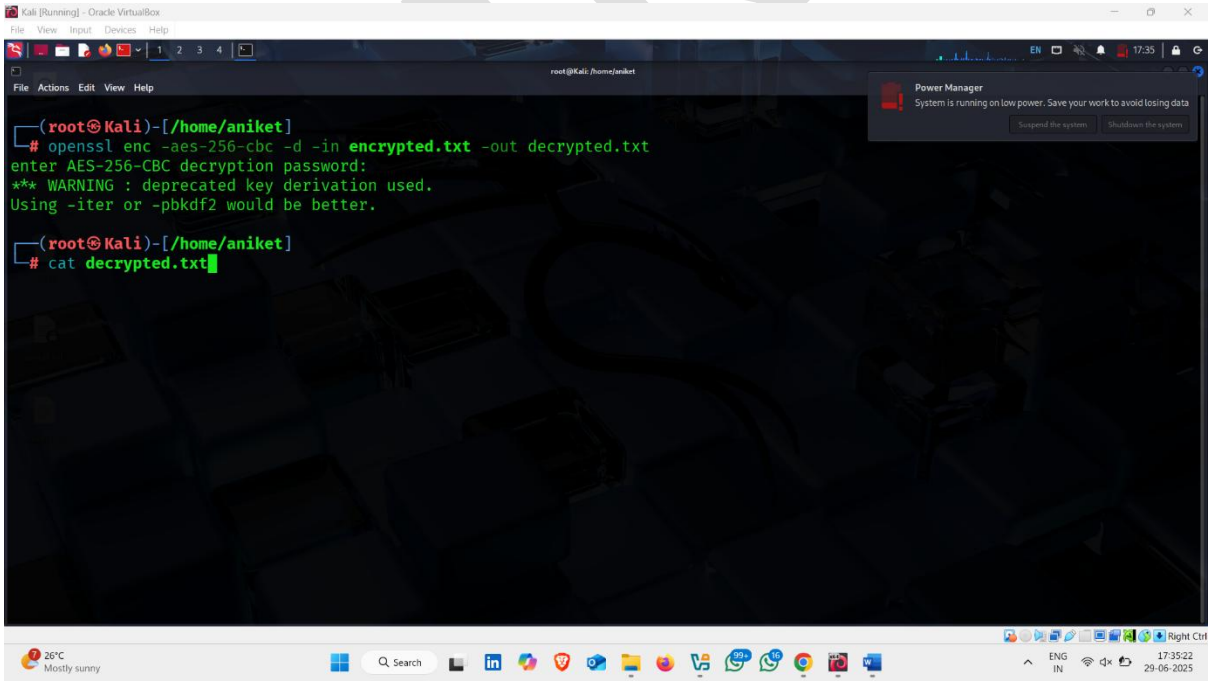
- Enter password



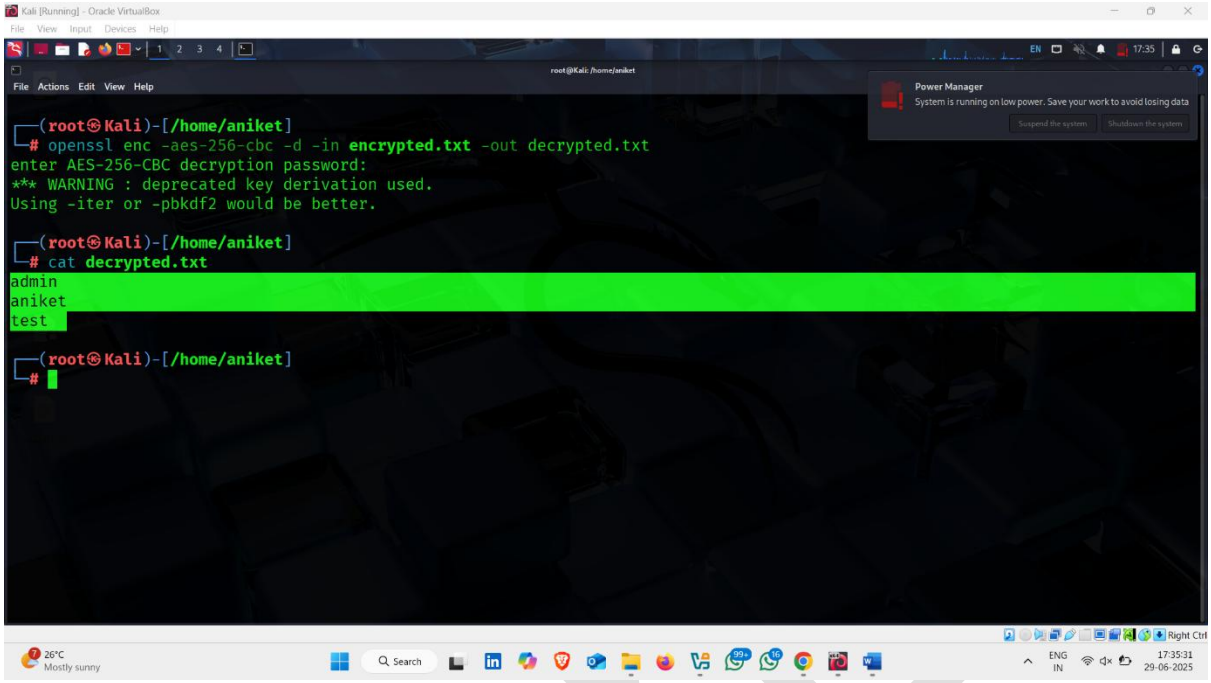
- Here , it decrypted



- Now , check it using **cat** command



- Decrypted 👇 ✅



5.CrypTool

CrypTool 2 is designed with educational and practical cryptography in mind. Here's a complete breakdown of its **main objectives**:

✅ Main Objectives of CrypTool 2

Objective	Description
🧠 **Cryptography Education**	Help students, enthusiasts, and professionals learn **how cryptographic algorithms work** visually and interactively.
🔍 **Understand Encryption/Decryption**	Demonstrate **symmetric and asymmetric encryption**, including how keys are used to secure and unlock information.

Objective	Description
🔐 **Practice Real Cryptographic Algorithms**	Let users experiment with **real algorithms** like AES, DES, RSA, ElGamal, and hash functions like SHA-256.
🧪 **Hands-On Cryptanalysis**	Teach how attacks like brute-force, frequency analysis, and known-plaintext work by letting users test them on ciphers.
📜 **Explore Digital Signatures & Hashing**	Understand the **integrity and authenticity** features of digital signatures and cryptographic hashes.
🛠️ **Build Visual Workflows**	Allow users to **construct custom workflows** for testing cryptography concepts using a drag-and-drop interface.
🧰 **Learn by Doing (Visual Simulations)**	Make abstract cryptographic concepts **concrete and intuitive** through simulations and step-by-step execution.
🎓 **Support Academic Learning**	Serve as a learning and demonstration tool for **cybersecurity courses, CTFs, and university-level projects**.

🧩 Example Learning Goals You Can Achieve with CrypTool 2

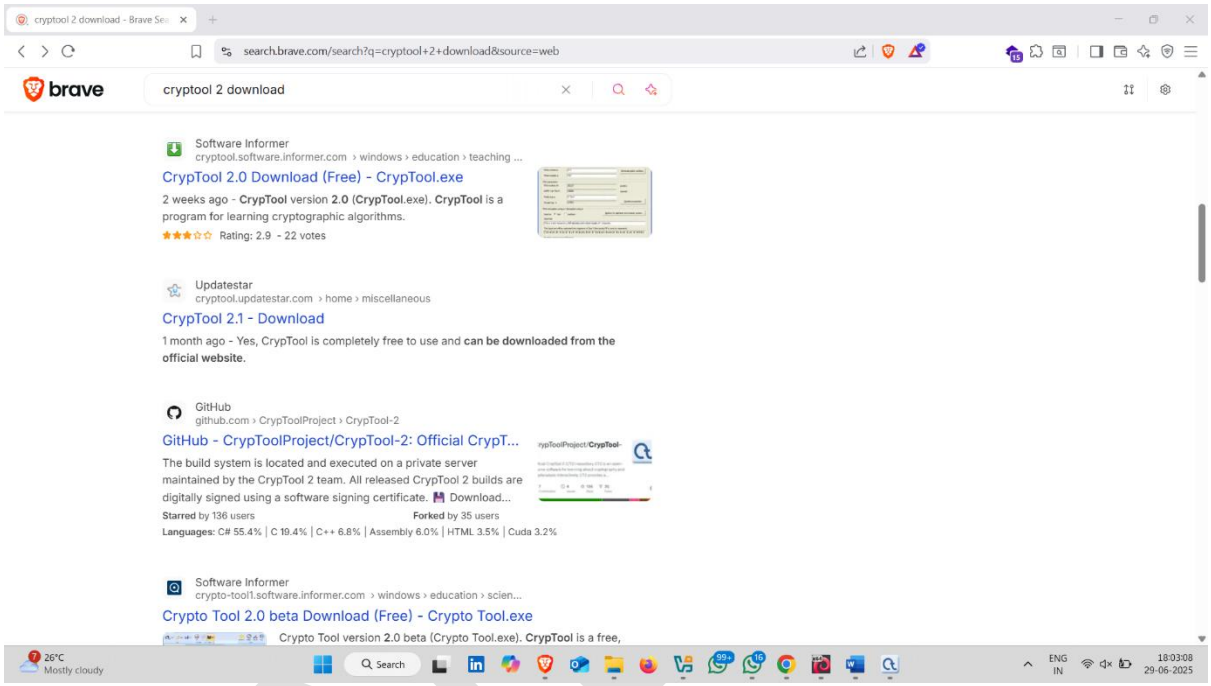
- Encrypt/decrypt data with **Caesar, Vigenère, AES, RSA, etc.**
- Analyze ciphertext with **frequency analysis**
- Understand how **public and private keys** work in asymmetric encryption
- Simulate **digital signatures** and verify them
- Learn how **hash collisions** can break integrity

- Explore **modular arithmetic**, **prime numbers**, and other math behind crypto

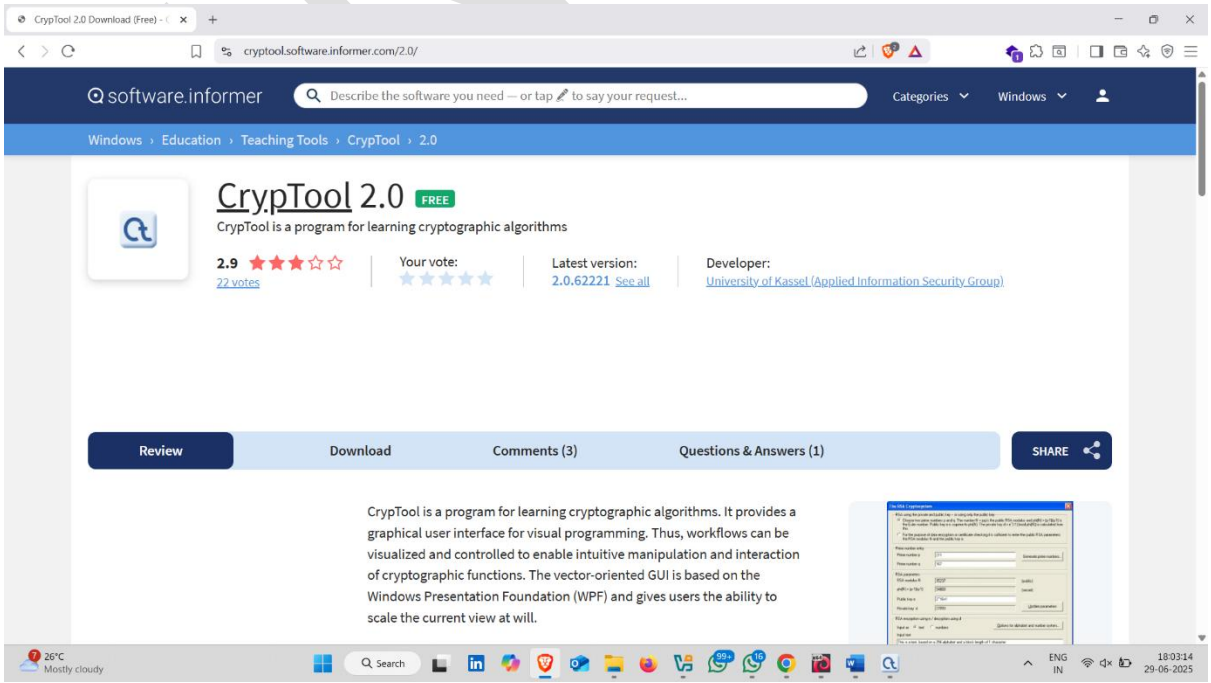
---

How to Download it -:

- Open Browser and search CrypTool 2 Download
- Click on **Software Informer** website

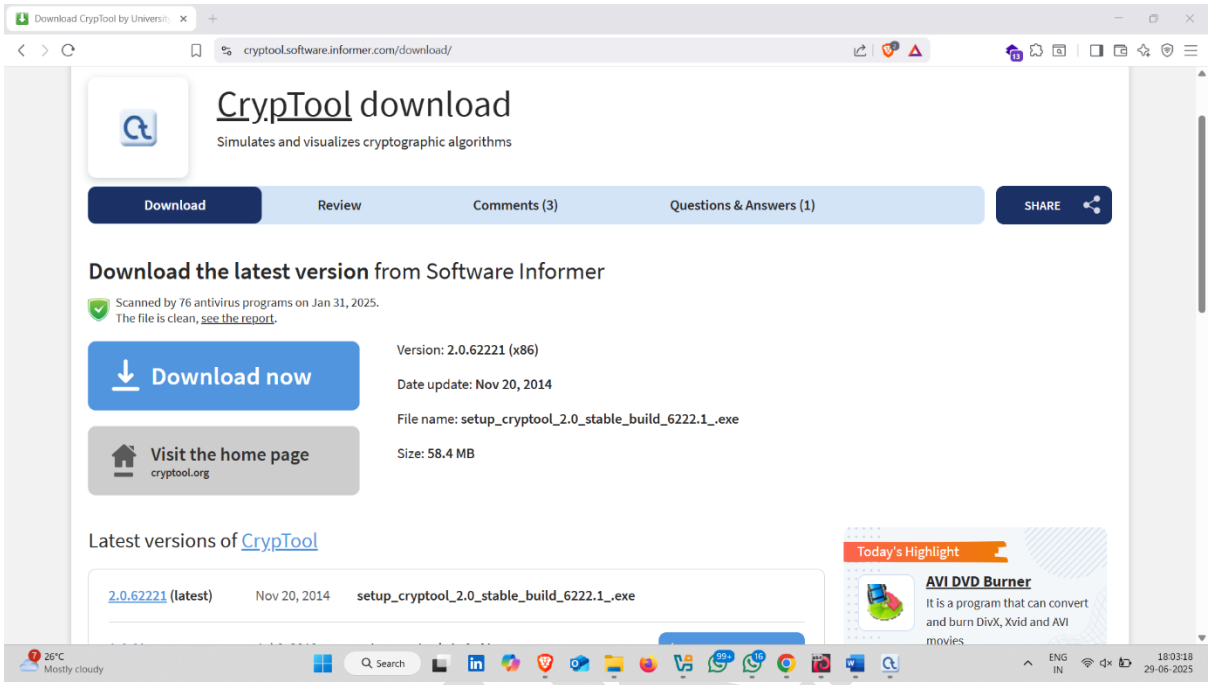


- Click on Download Section



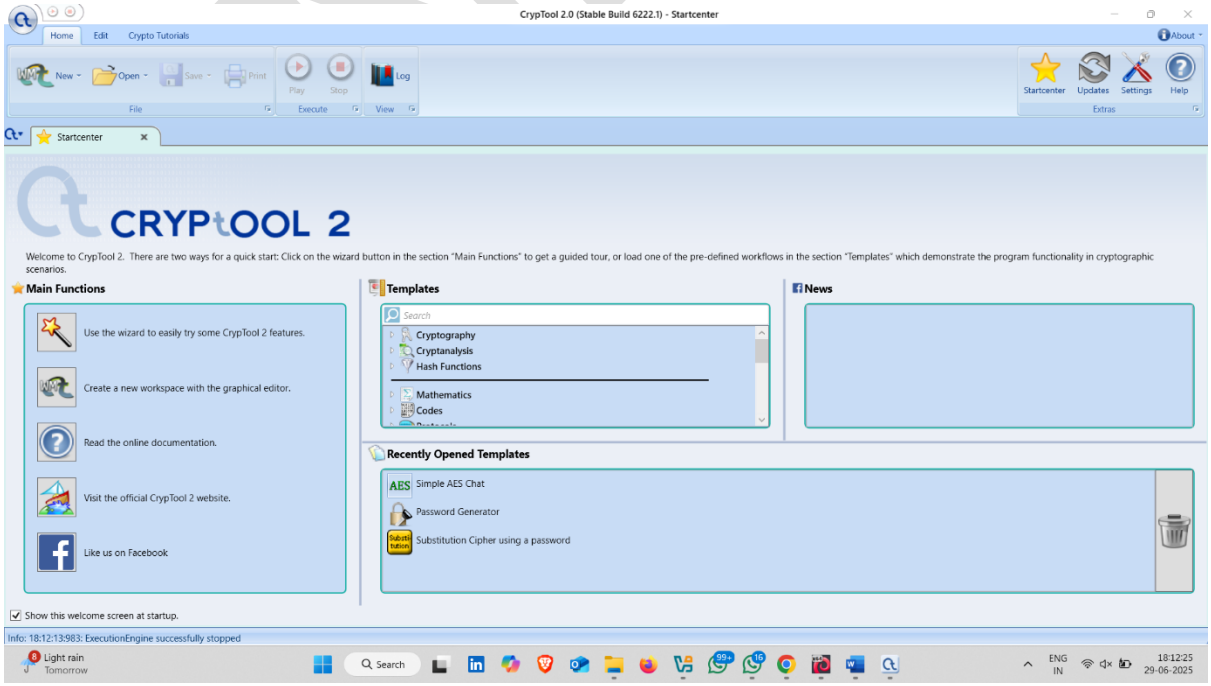
- Click on **Download Now**

Download Link-
:https://cryptool.software.informer.com/download/

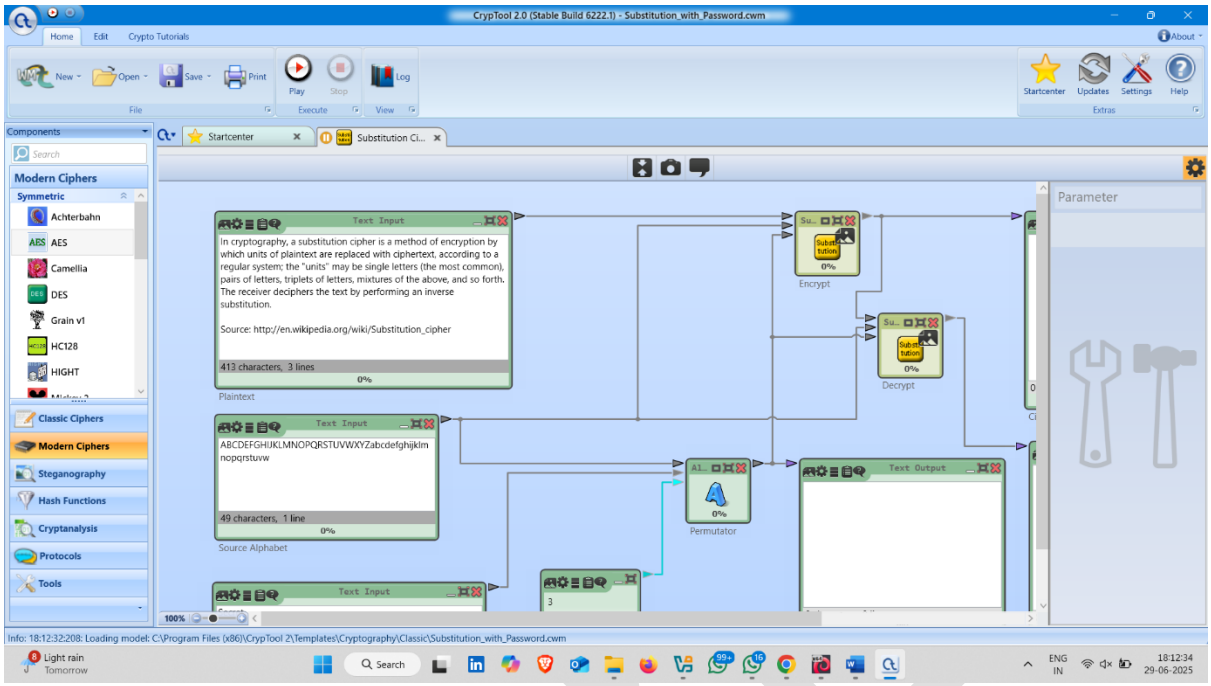


How to use it -:

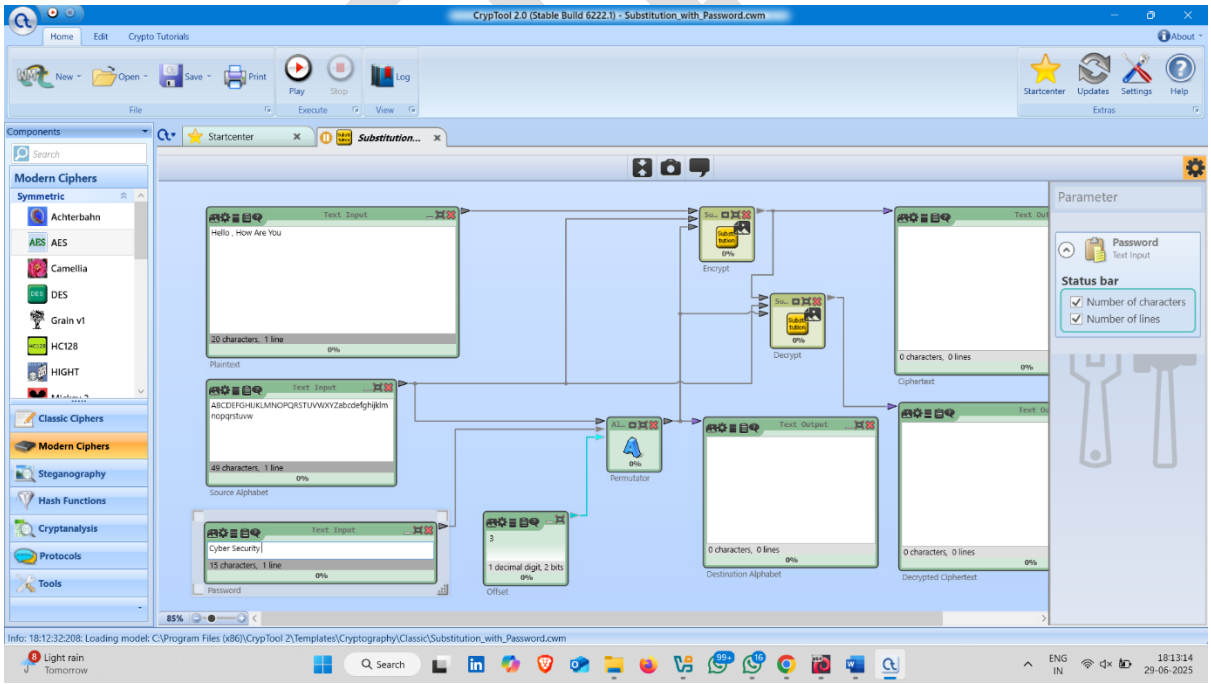
- Click on **substitution cipher using a password**

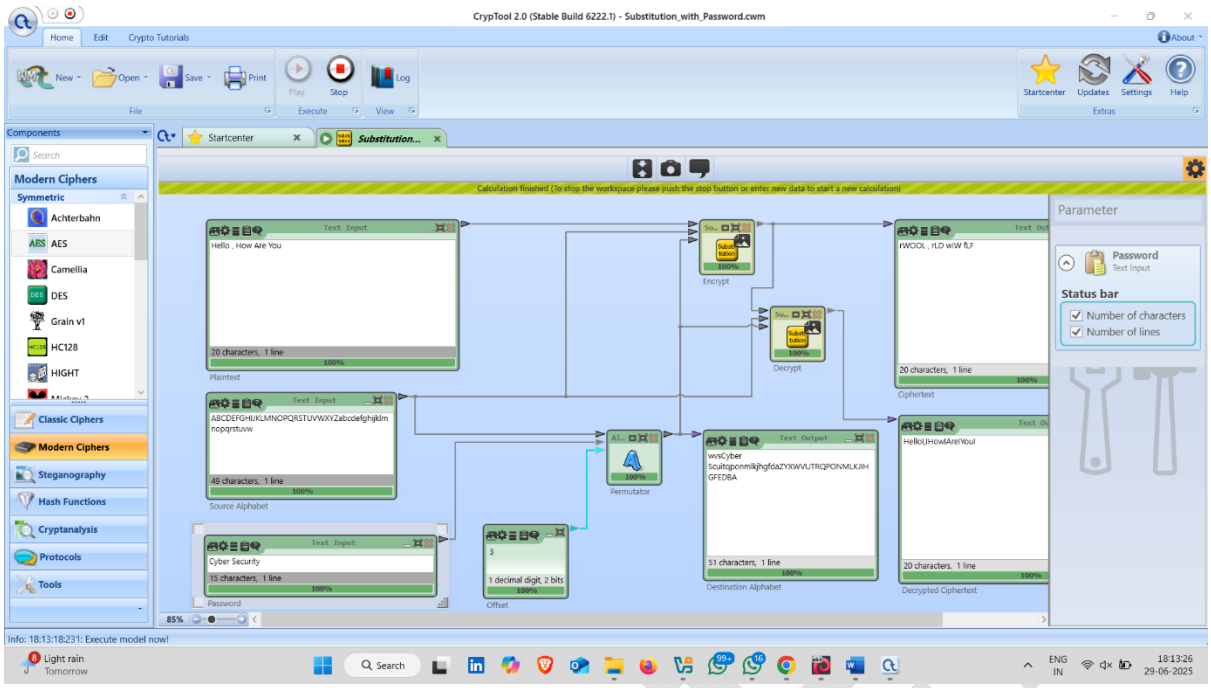


- **Window open – how AES Encryption Work ✅ 👇**



- Ente your text to show how AES Encryption Work
- After enter text in input field , click on **play** button





Explanation 👇

🎯 Objective of This CrypTool Workflow

The objective of this CrypTool 2 workflow is to demonstrate how to:

👉 **Encrypt and Decrypt text using a password-based substitution cipher.**

It visually shows how a **classical substitution cipher** works, but instead of a fixed substitution alphabet, it **generates a new one based on a password** (adding an extra layer of security and variation).

🧠 Step-by-Step Explanation of Each Block

Let's break it down based on what we see in your image:

🟩 1. Plaintext Input Block

- **Text:** Hello , How Are You
- This is the **original message** you want to encrypt.

🟦 2. Password Input

- **Text:** Cyber Security
- This password is used to **generate a shuffled alphabet** for encryption (via permutation logic).

🟩 3. Source Alphabet

- A-Z and a-z letters.
- The **standard alphabet** used as the input reference for substitution.

🟪 4. Permutation Logic (Based on Password + Offset)

- The **Password + Offset = new order** of characters.
- This generates the **Destination Alphabet** using the "Permutator" block.
- Output seen:
  wvxCyberScutapomnklighfdaZYXWVUTRQPNMLKJHGFEDBA

🟧 5. Encrypt Block (Substitution)

- Encrypts plaintext using the **source alphabet → destination alphabet mapping**.
- Output (Ciphertext):
  rWOOL , rLD wIW FLF
  → This is the encrypted version of your input text.

🟥 6. Decrypt Block (Substitution)

- Takes the ciphertext and **reverses the substitution** using the same password-derived destination alphabet.

- Output (Decrypted Ciphertext):
  Hello,|How|Are|You! (with minor formatting changes, still correct).

🔽 7. Text Output Blocks

- Display the result of encryption and decryption.
- **Three outputs are shown**:
  - Ciphertext (encrypted message)
  - Destination Alphabet (used for mapping)
  - Decrypted Plaintext (should match original)

✅ Summary of Data Flow

[Plaintext] + [Password] + [Offset] + [Source Alphabet]

↓

Generate Permutated Alphabet

↓

Substitution → Ciphertext

↓

Reverse Substitution → Decrypted Ciphertext

🎯 Educational Objective of This Lab

Learning Goal	Explanation
🔑 Understand substitution ciphers	Learn how simple character replacement ciphers work.
🔐 Add password-based permutation	Learn how to dynamically generate cipher alphabets using passwords.

Learning Goal	Explanation
🔁 See real-time encryption/decryption flow	Instantly visualize how text is encrypted and decrypted.
⚠️ Appreciate the weaknesses of classical ciphers	Useful in comparing classical vs modern symmetric ciphers.

---

6. AES Crypt

AES Crypt is a free, open-source application that allows you to **encrypt and decrypt files** using the **AES-256** (Advanced Encryption Standard) algorithm. It provides both **graphical (GUI)** and **command-line** interfaces, making it easy for both beginners and advanced users to secure sensitive data.

📜 Quick Overview

Feature	Details
🔒 **Encryption**	Uses **AES-256**, a military-grade symmetric encryption algorithm.
💻 **Platforms**	Available on **Windows, macOS, Linux, Android, and iOS**.
🆓 **License**	Free and Open Source
🔐 **Encryption Type**	**Symmetric** — uses a single password for both encryption and decryption.

🎯 Main Objective of AES Crypt

To provide a **simple and secure** way for users to:

- **Encrypt confidential files** (documents, photos, financial data, etc.)
- **Protect files from unauthorized access**
- **Easily decrypt them** later with a password

🖥️ How AES Crypt Works

1. 🔑 You choose a **password** (which acts as the key).
2. 📁 AES Crypt **encrypts the file** using AES-256 and appends .aes to the filename.
3. 🔓 To decrypt, you **use the same password** with AES Crypt.
4. ✅ The original file is restored if the password is correct.

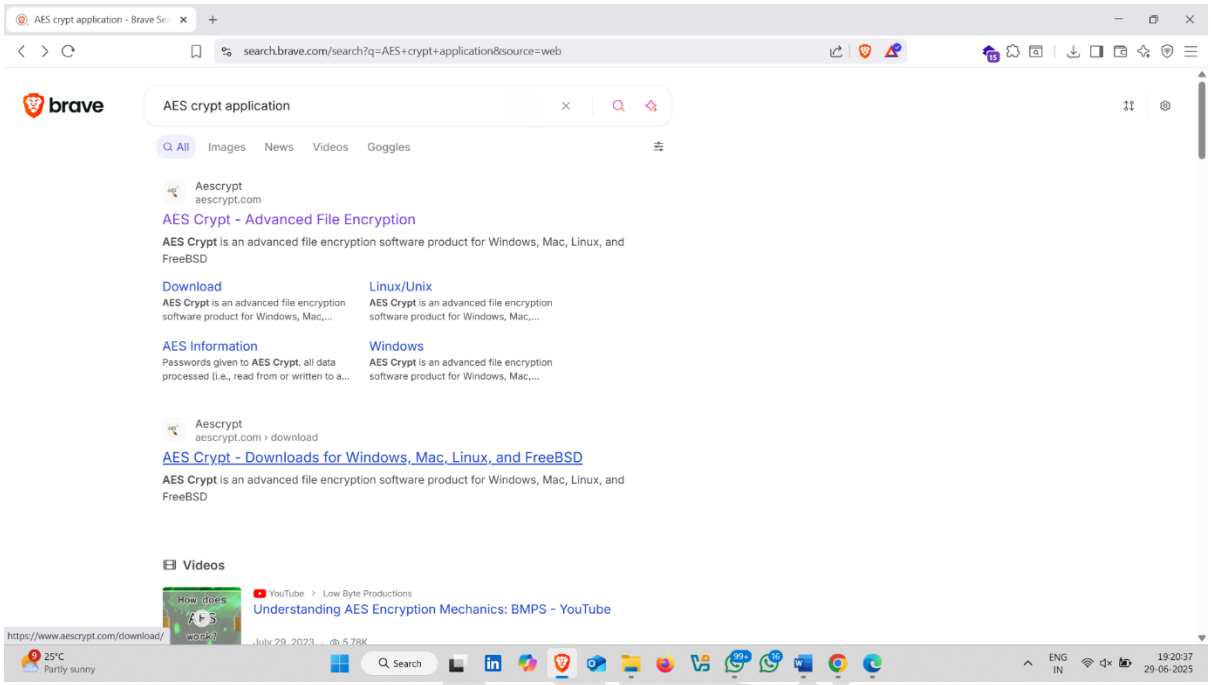
AES-256 is currently one of the **most secure symmetric encryption standards**, trusted by governments and security professionals worldwide.

🔧 Key Features

Feature	Description
📂 File-based encryption	Encrypts individual files (not folders or full drives)
🔐 Strong encryption	Uses AES-256 with random salt and IV
🧩 Cross-platform	Available on major operating systems
🖱️ Easy GUI	Just right-click to encrypt or decrypt
🧰 CLI available	Use in scripts or automation
🔐 No key file needed	Just use a password to encrypt/decrypt

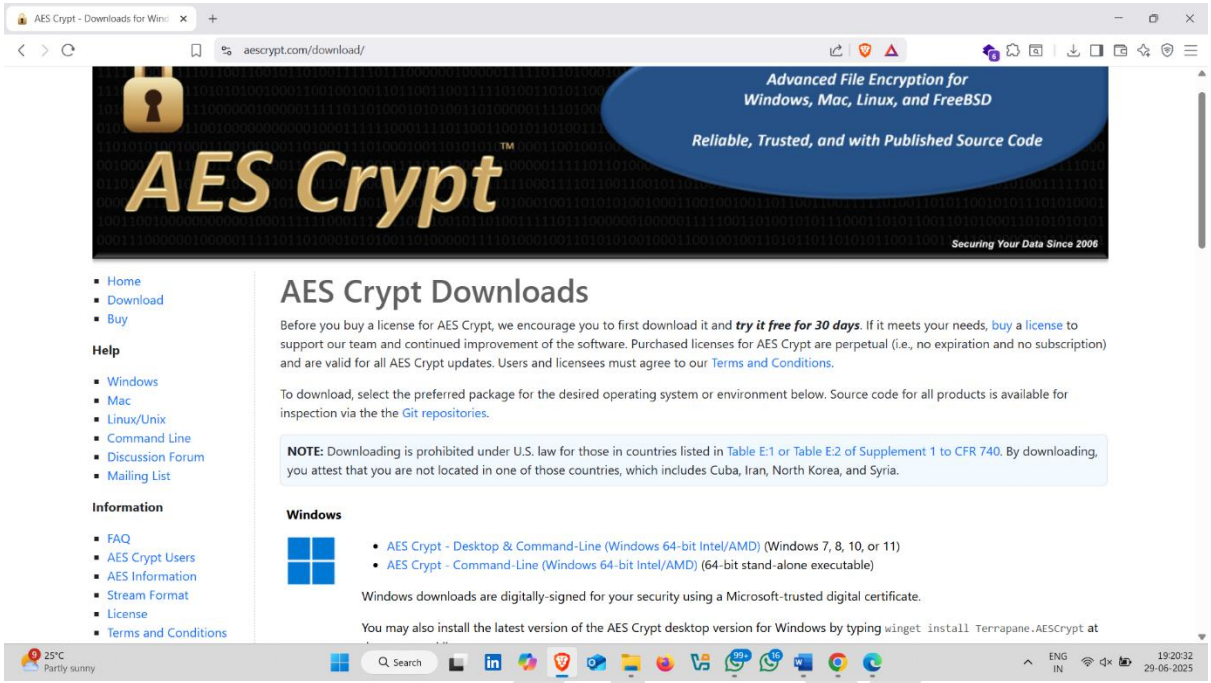
How to Download It -:

- Open Browser and search **AES Crypt Application**
- Click on **first official website**



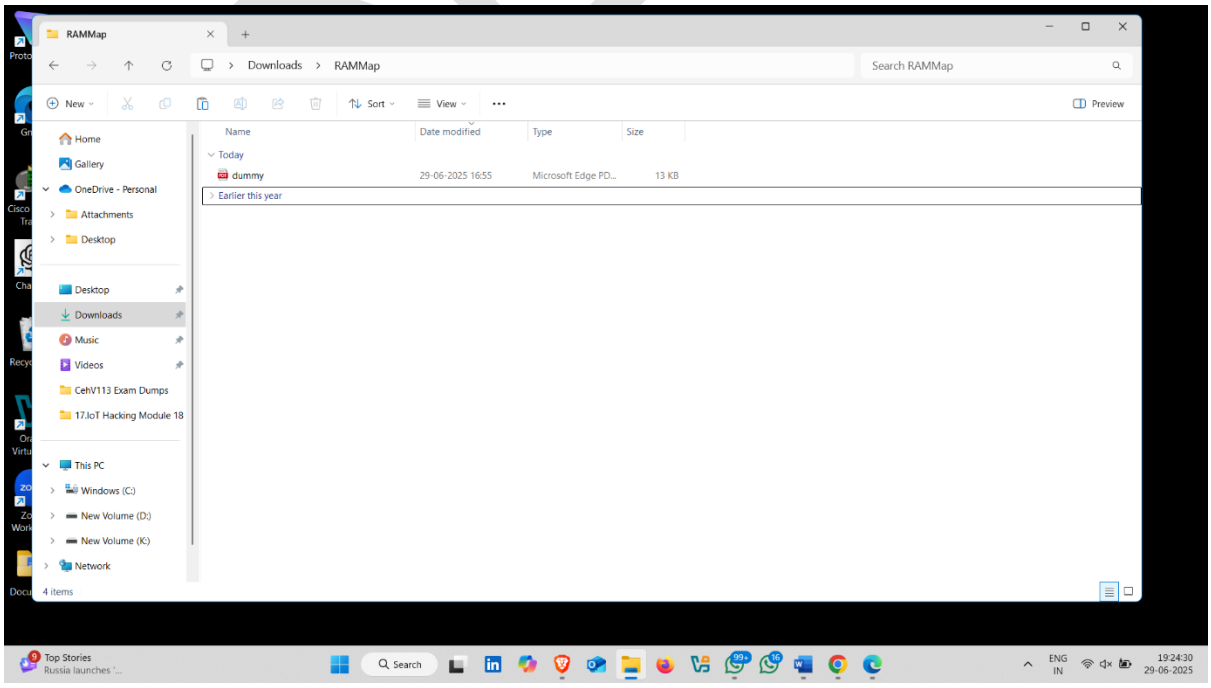
- Click on **Download Section**

- Now, click on first **Desktop and command line** option , and download will started automatically

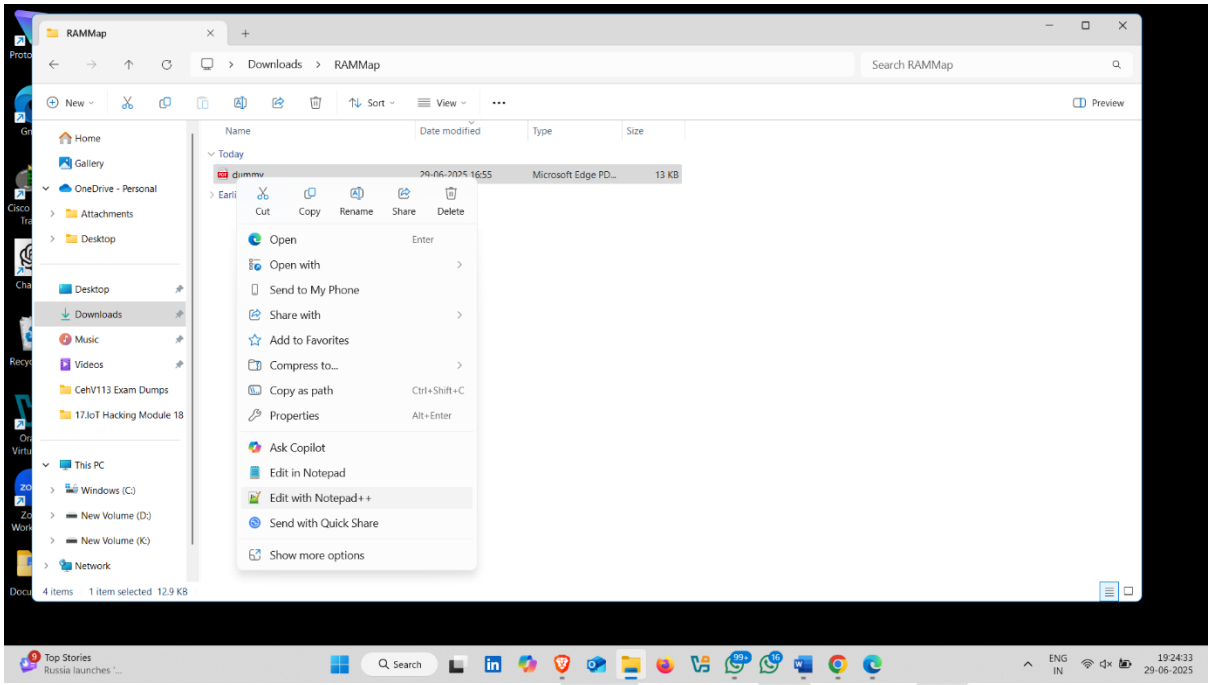


How to use it -:

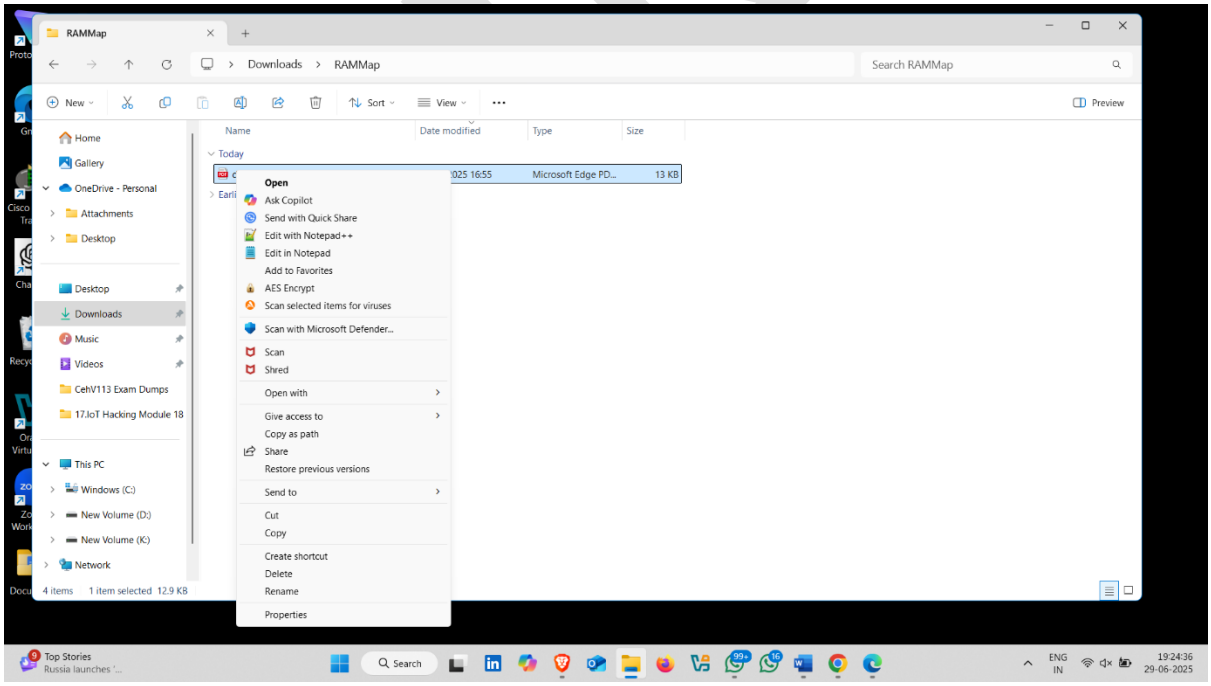
- **After downloading application , installed it .**
- Now select the file or folder that you want to encrypt



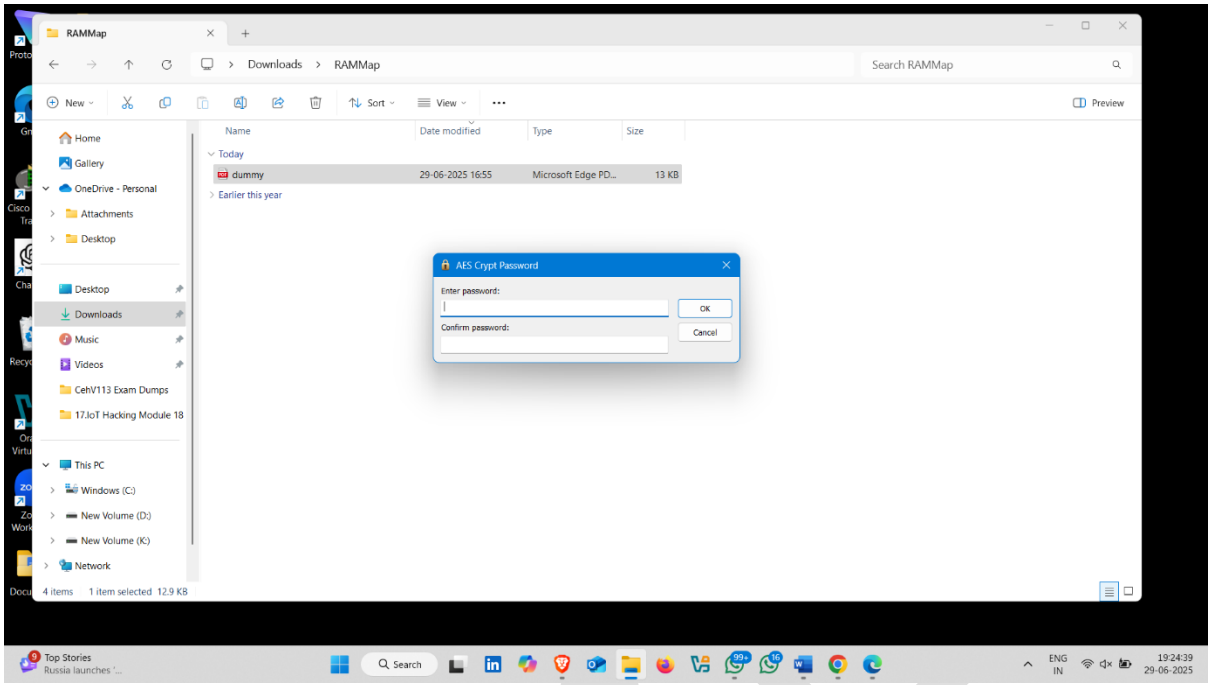
- Right click on file and select **show more options**



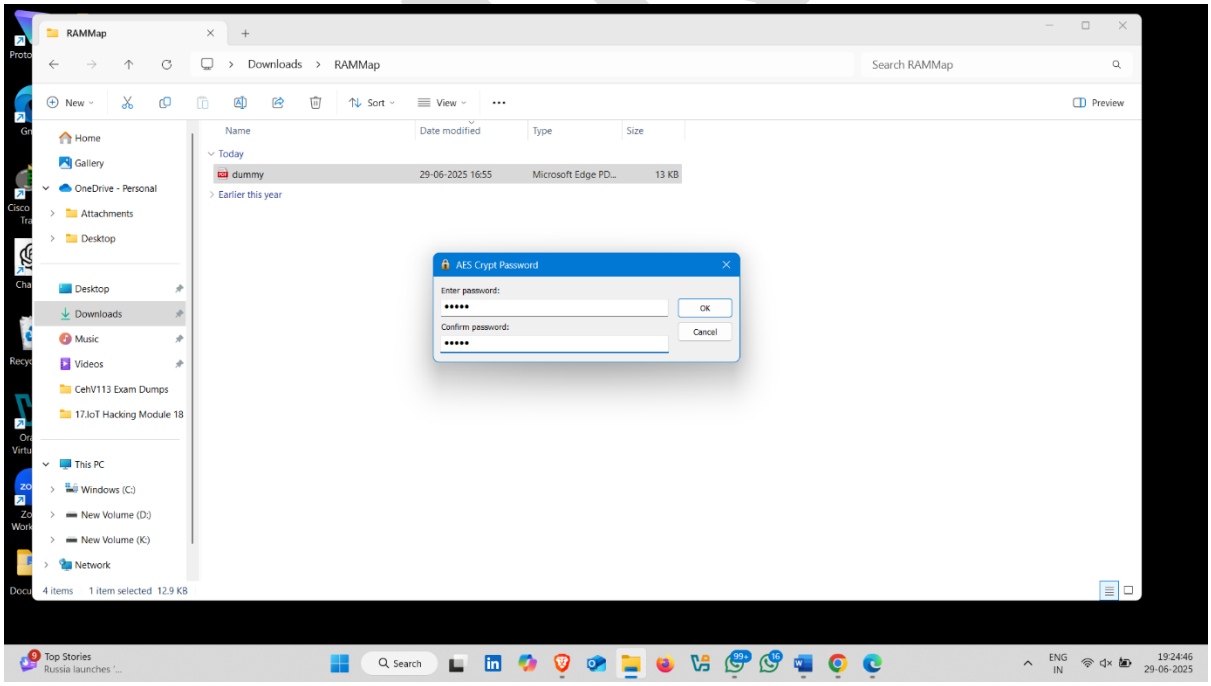
- Now , select **AES Encrypt**



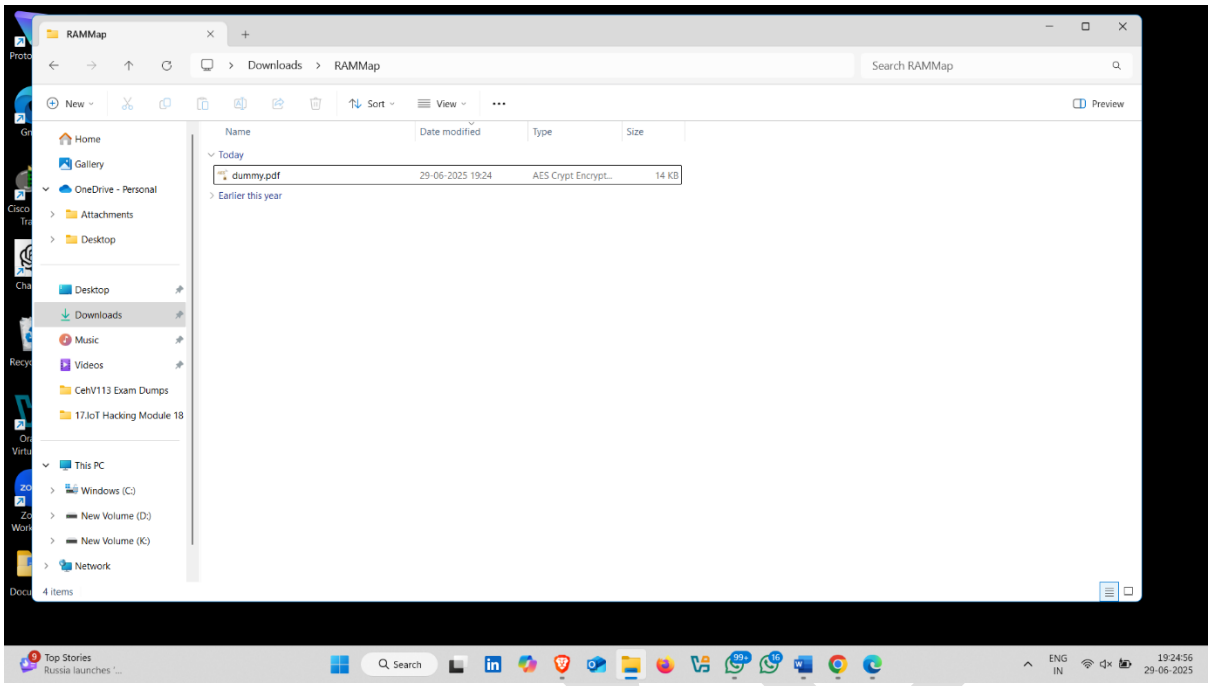
- Enter a password



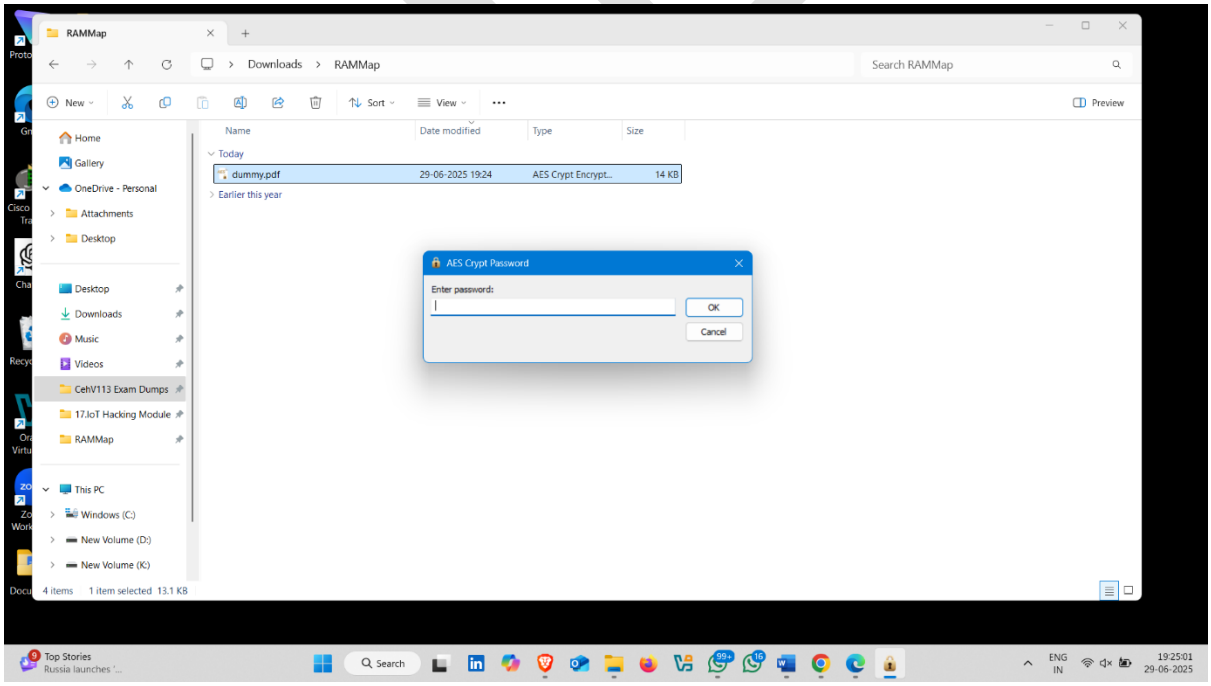
- Click on **ok**



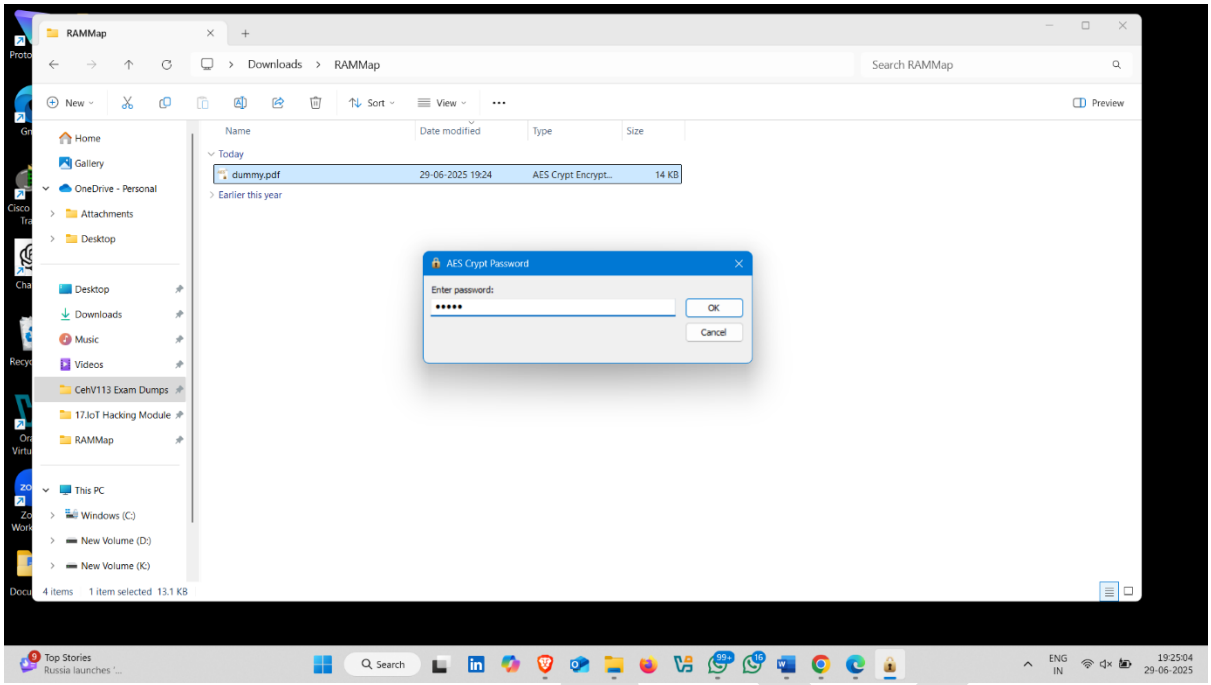
- Here , file is encrypted 👇 ✅



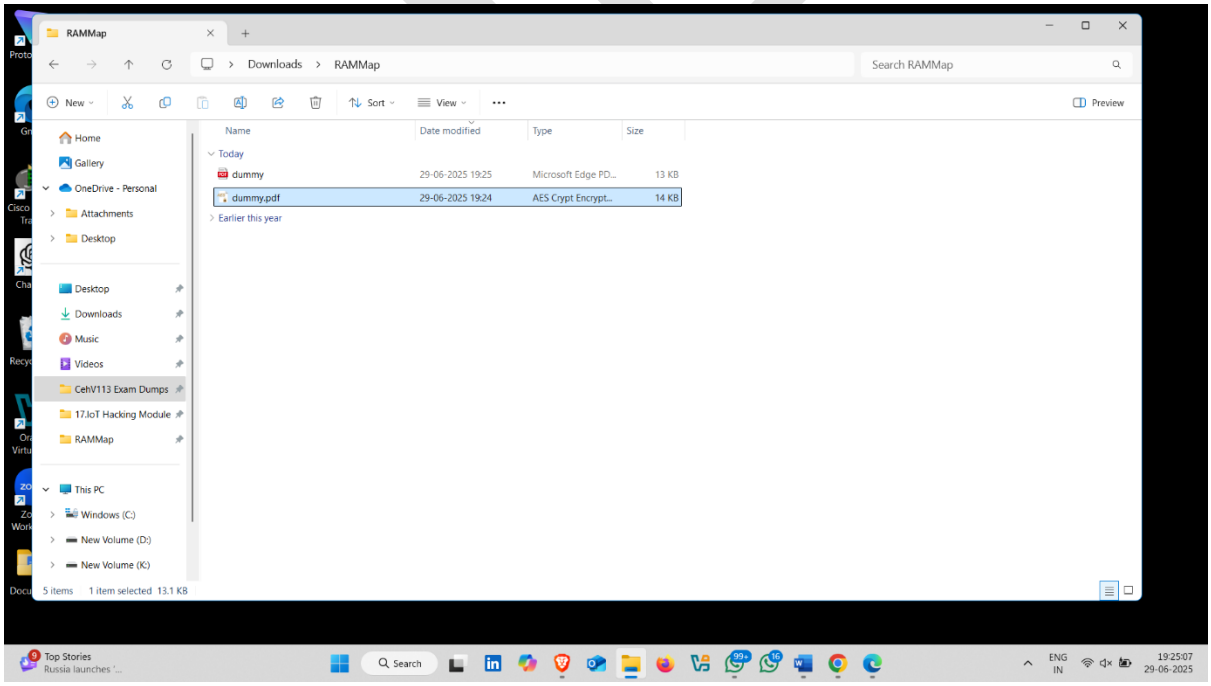
- To Decrypt this file , click on it and provide password



- Click on **ok**



- **Here , file is decrypted** ✅ 👇



7.HashCalc

HashCalc is a free Windows-based tool used to compute cryptographic hashes, checksums, and HMACs (Hashed Message Authentication Codes) for files, text, and strings.

It is developed by SlavaSoft and widely used for quickly verifying file integrity and performing hash-related tasks.

📜 Key Features of HashCalc

Feature	Description
🧮 Hash Functions	Supports MD5, SHA-1, SHA-256, SHA-384, SHA-512, RIPEMD160, etc.
📁 Input Support	Can hash files, text, hex strings, or binary data
🔐 HMAC Calculation	Supports keyed hashes (HMAC)
📋 Multiple Output Formats	Outputs hash in hex, base64, etc.
⚡ Fast and Lightweight	Small, portable tool for quick hashing

---

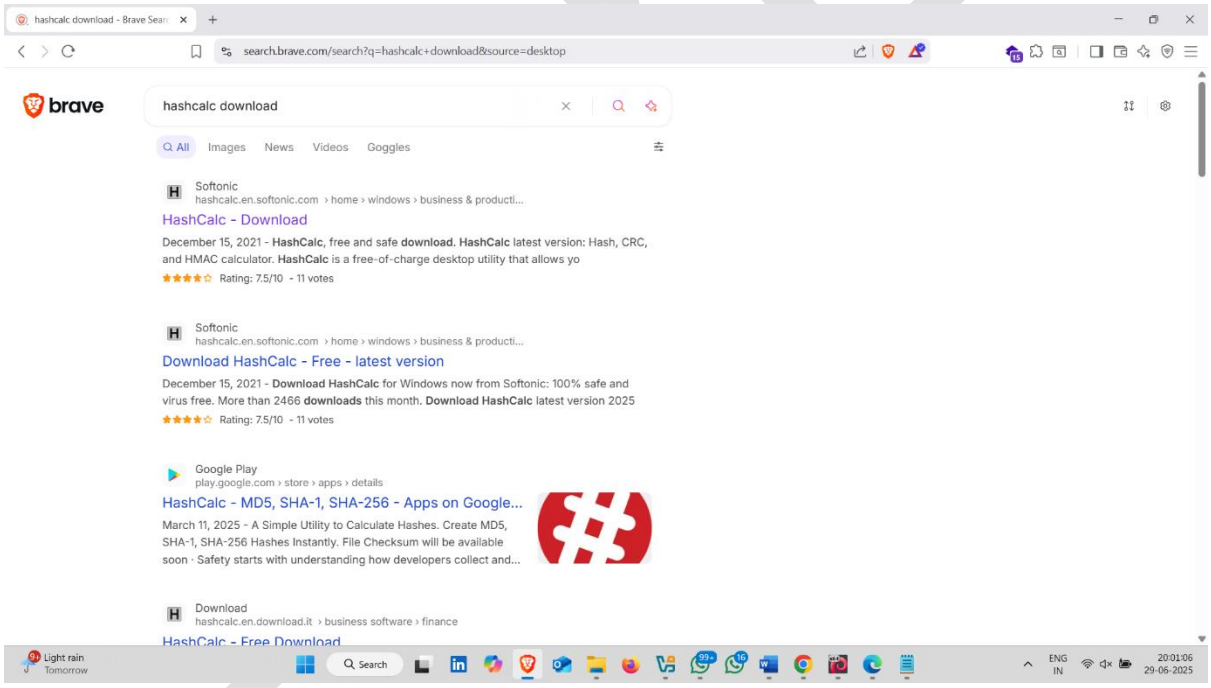
🎯 Primary Use Cases

Use Case	Example
✅ File integrity checking	Compare a file's hash with original (e.g., ISO downloads)
🔑 Password hashing practice	Hash test passwords (text input)

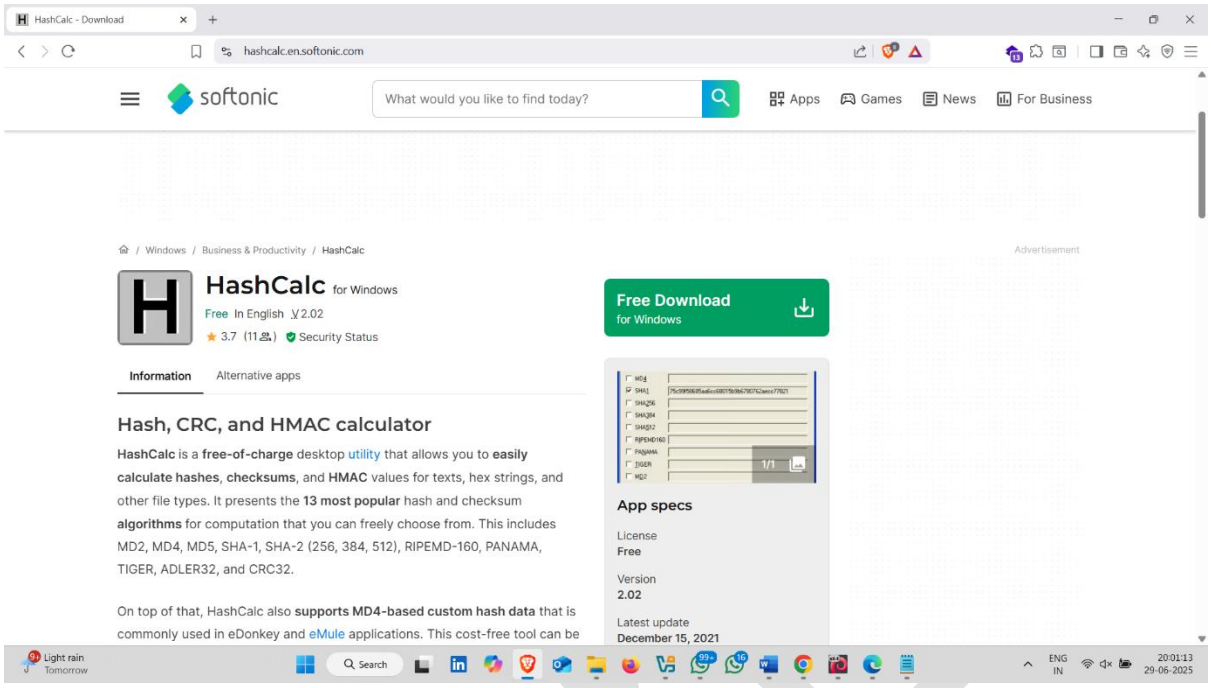
Use Case	Example
HMAC generation for APIs	Secure message authentication
Malware analysis	Calculate file hashes (MD5/SHA-1) for malware samples
Digital forensics	Hash evidence files for integrity

How to download it -:

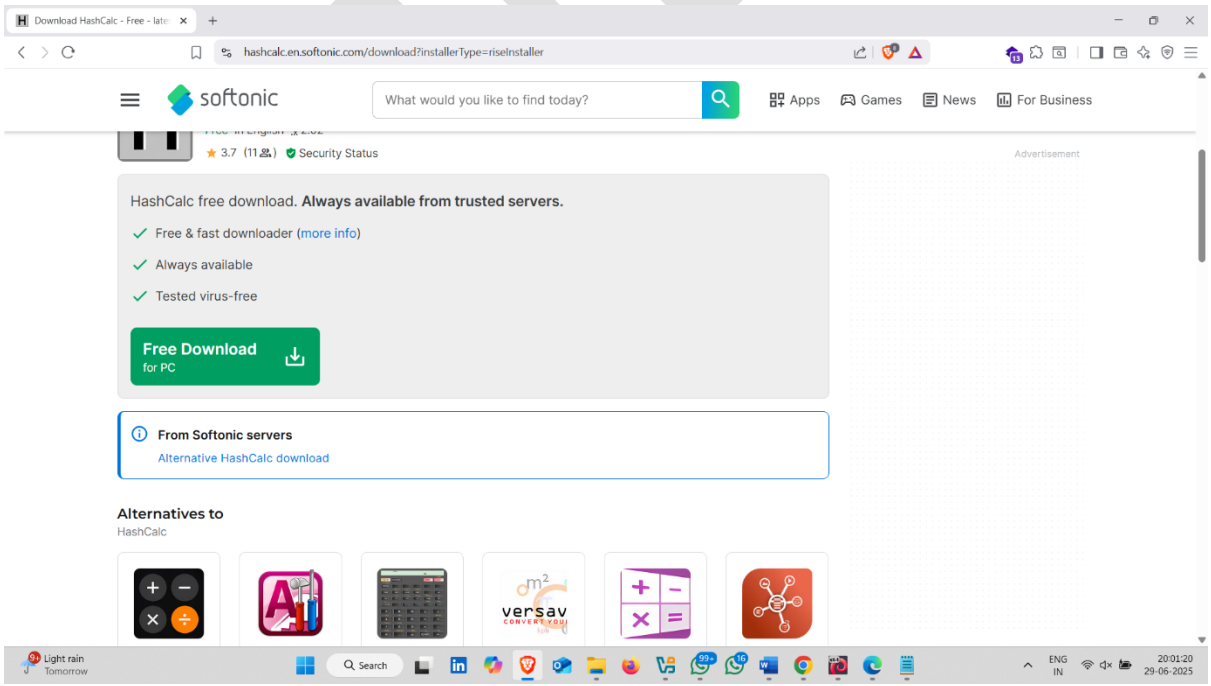
- Open browser and search **Hashcalc**
- click on first official **softonic** website



- click on **free download**

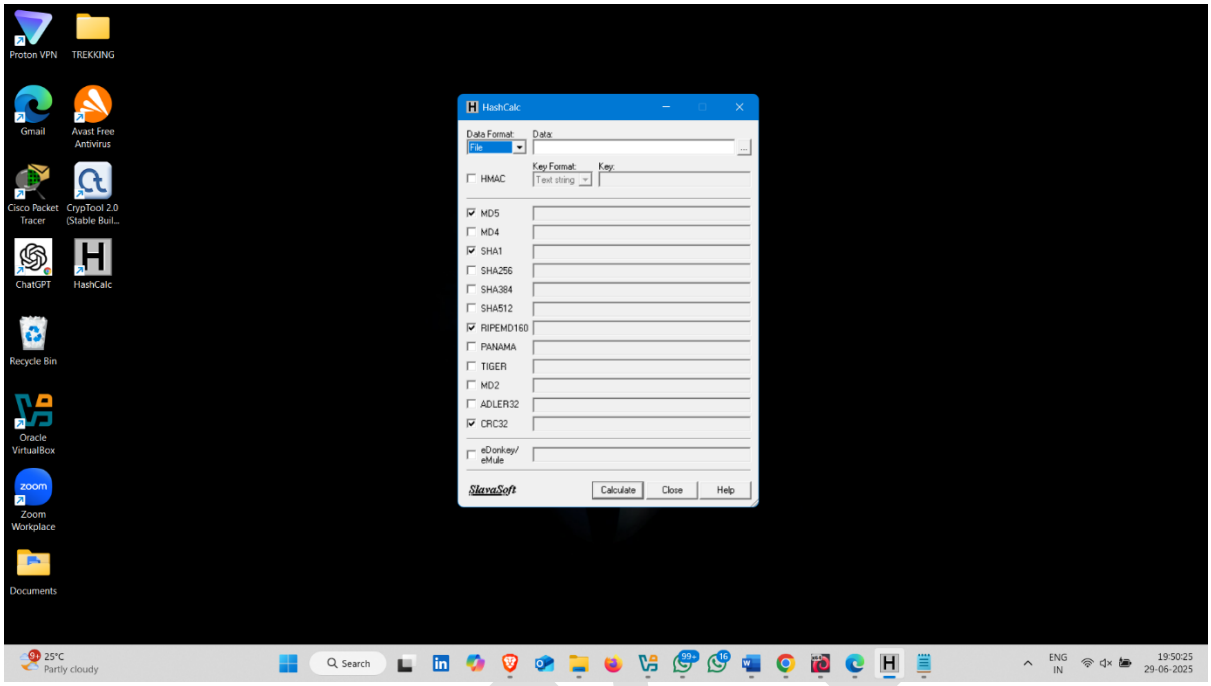


- once again click on **Free download**

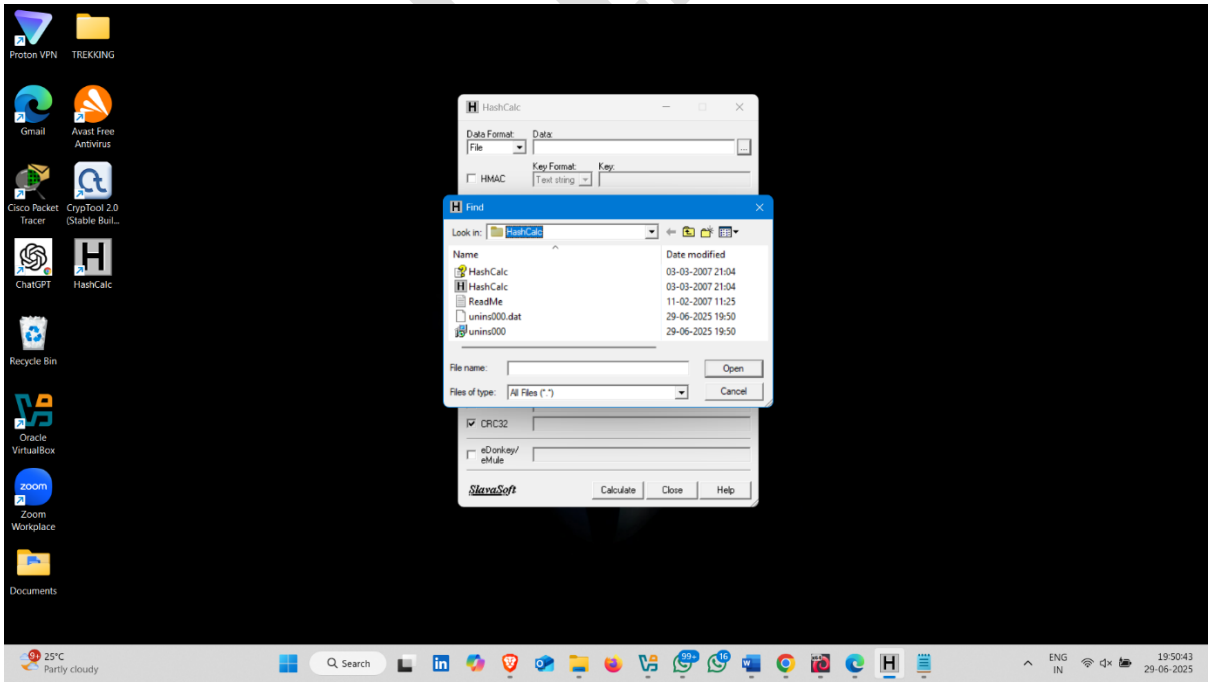


How to use it -:

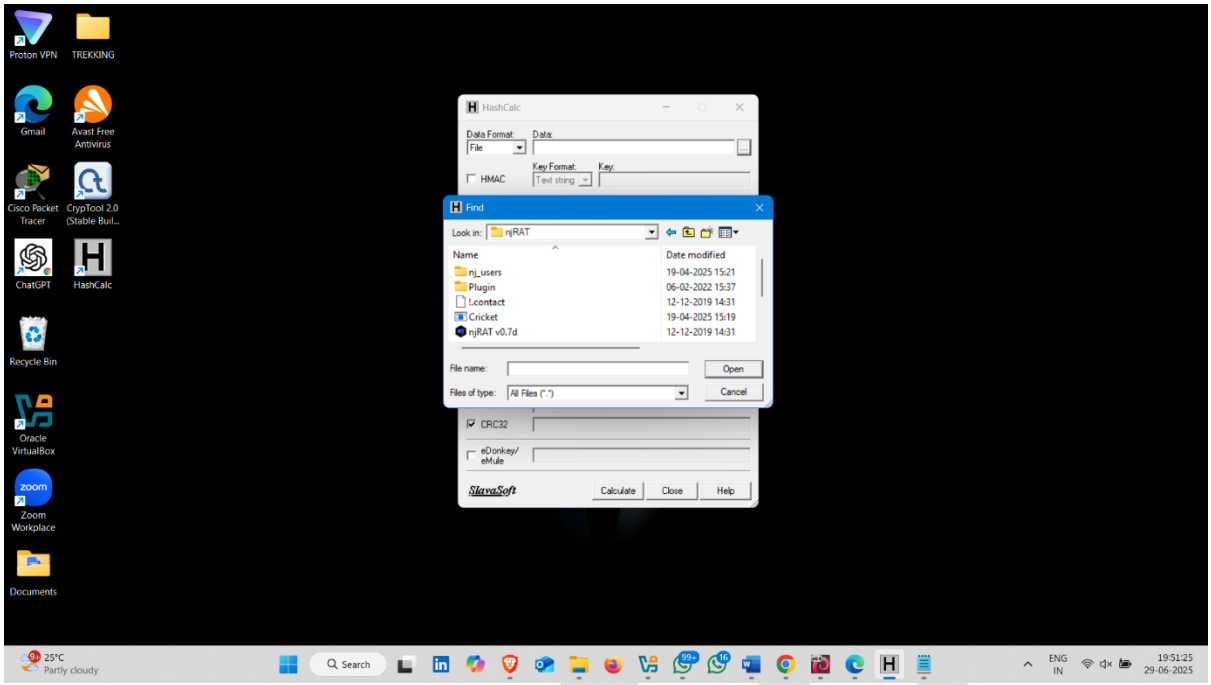
- **After installed application , open it .**



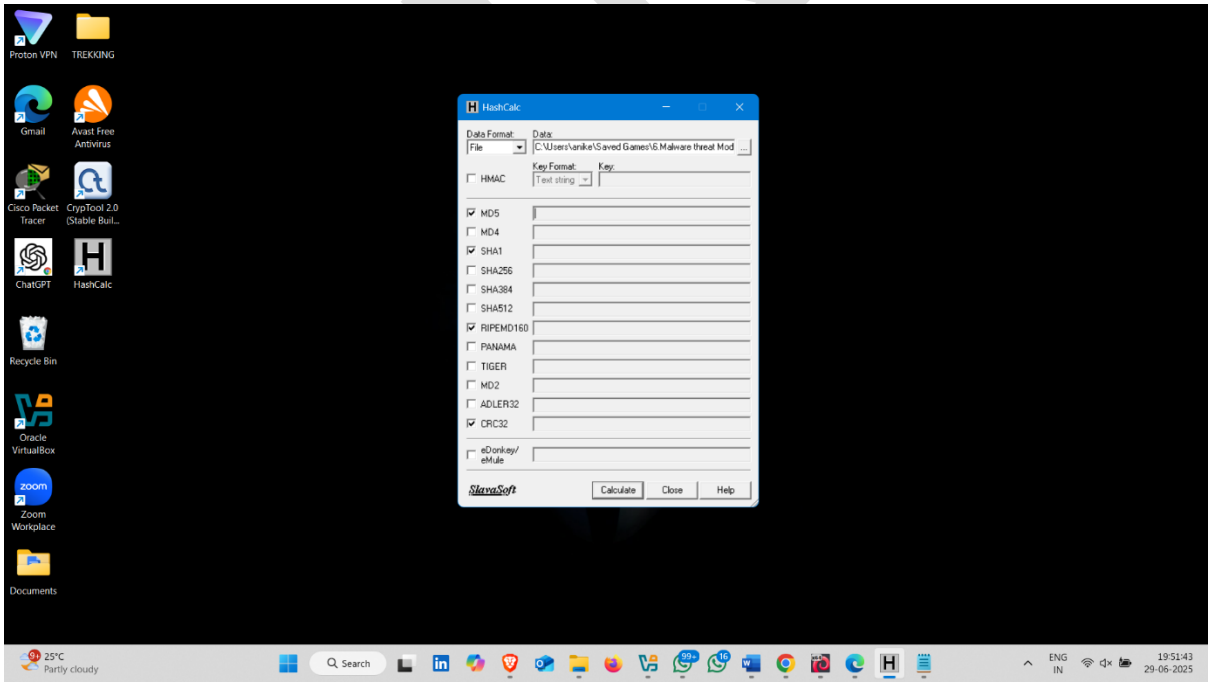
- Now click on **down arrow** 🔽 , select file that you want to generate a hash



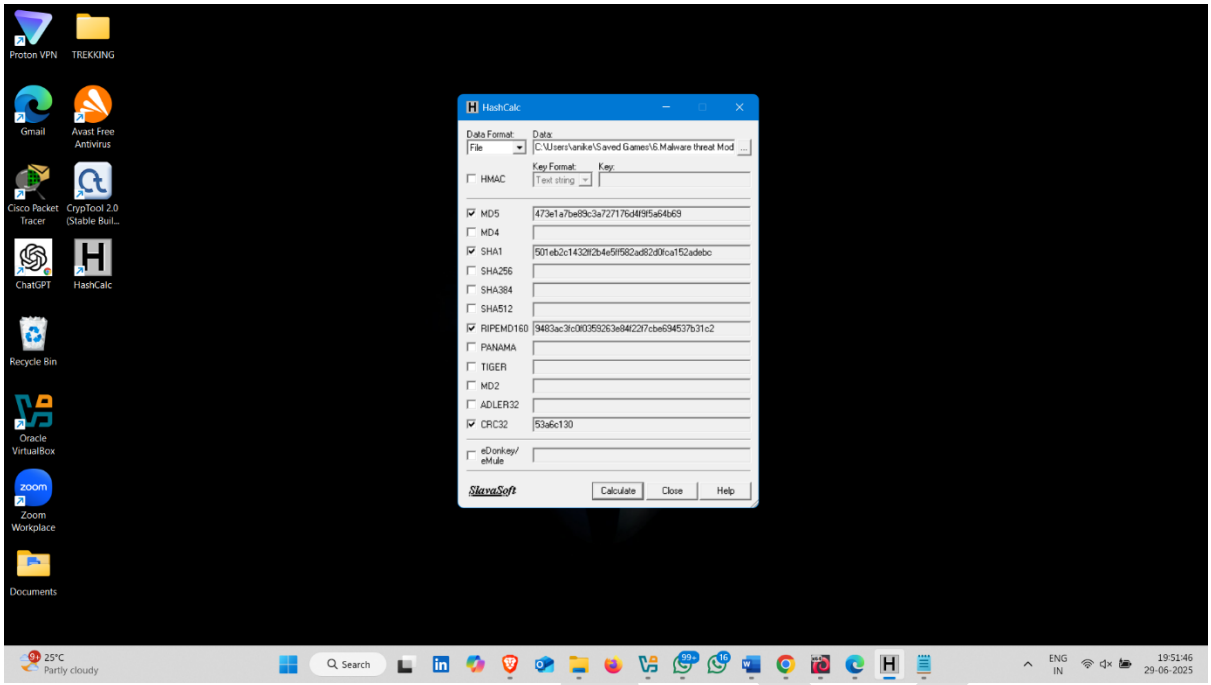
- Select file and click on open



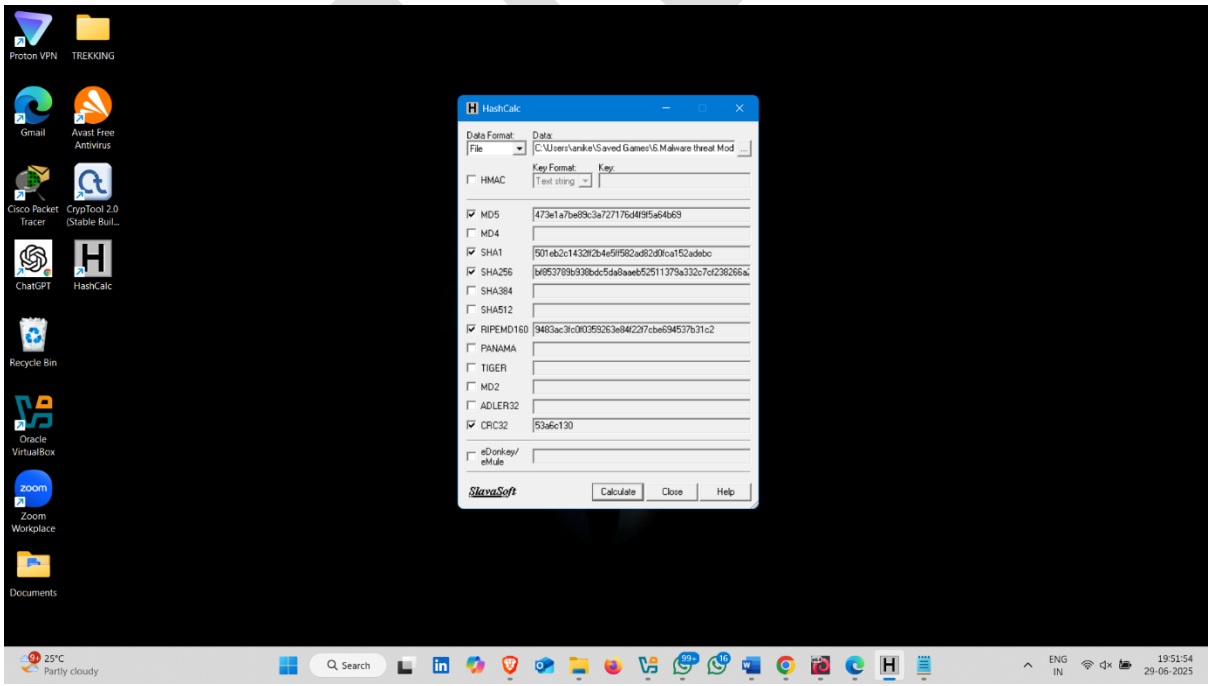
- After select the file, click on **calculate** button



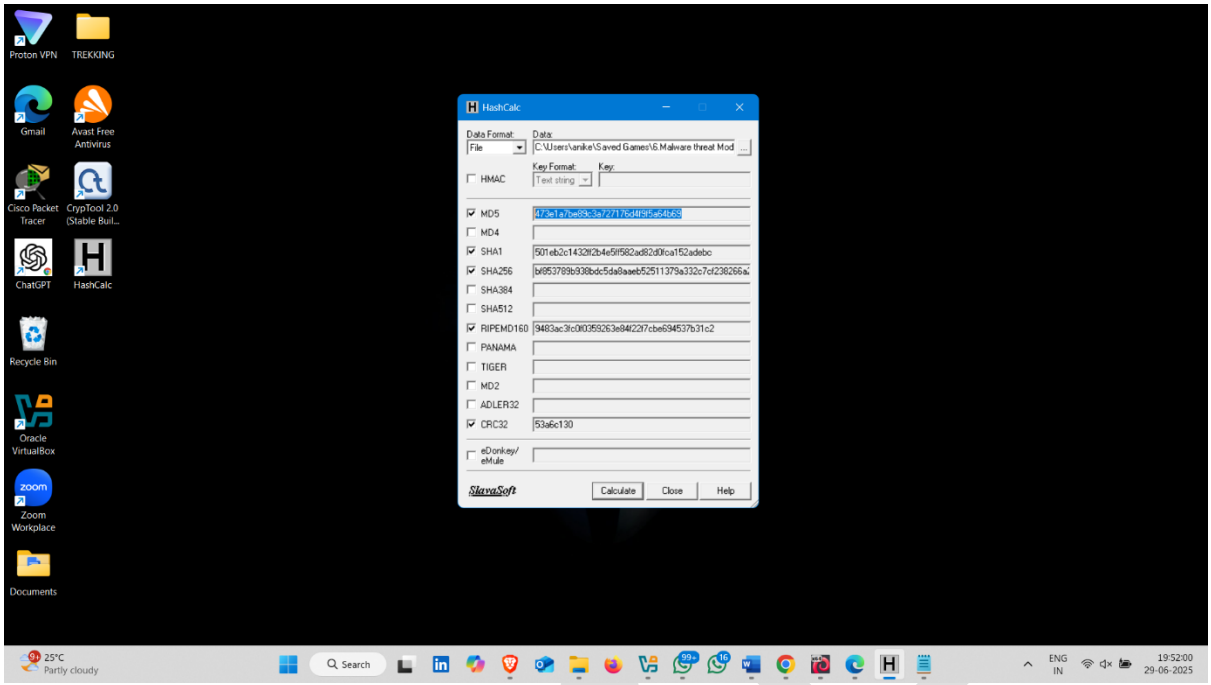
- Here it generate all hashes that you selected



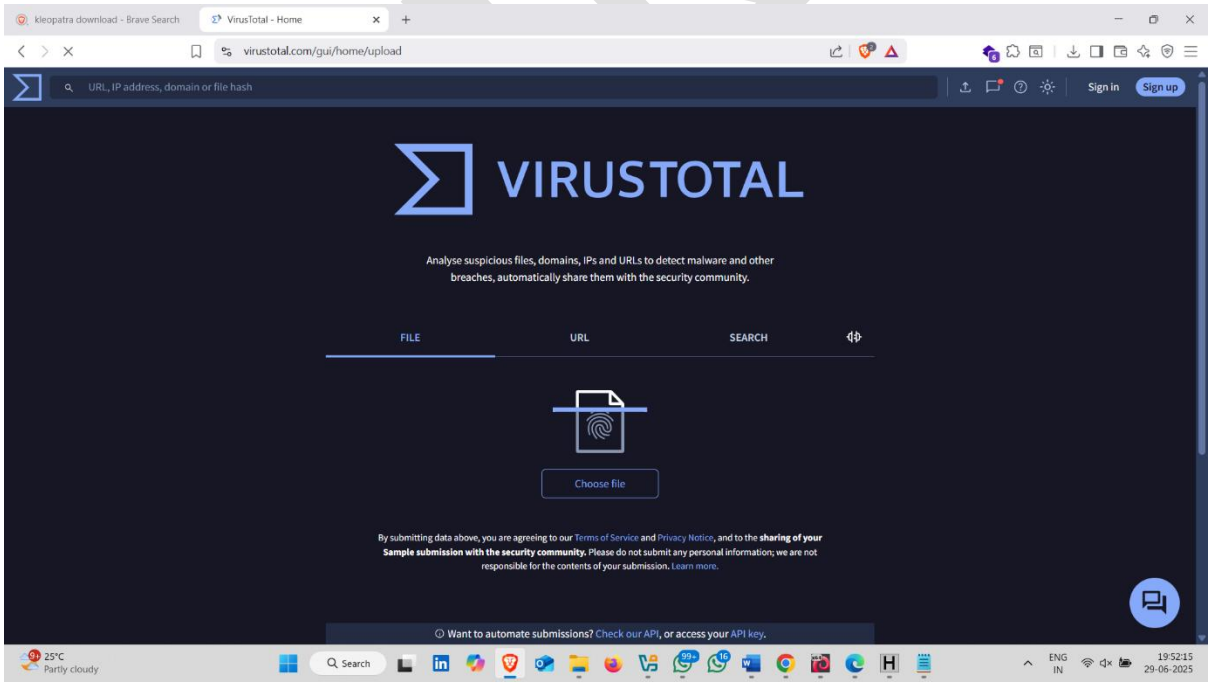
- Now you can also check file are malicious or not



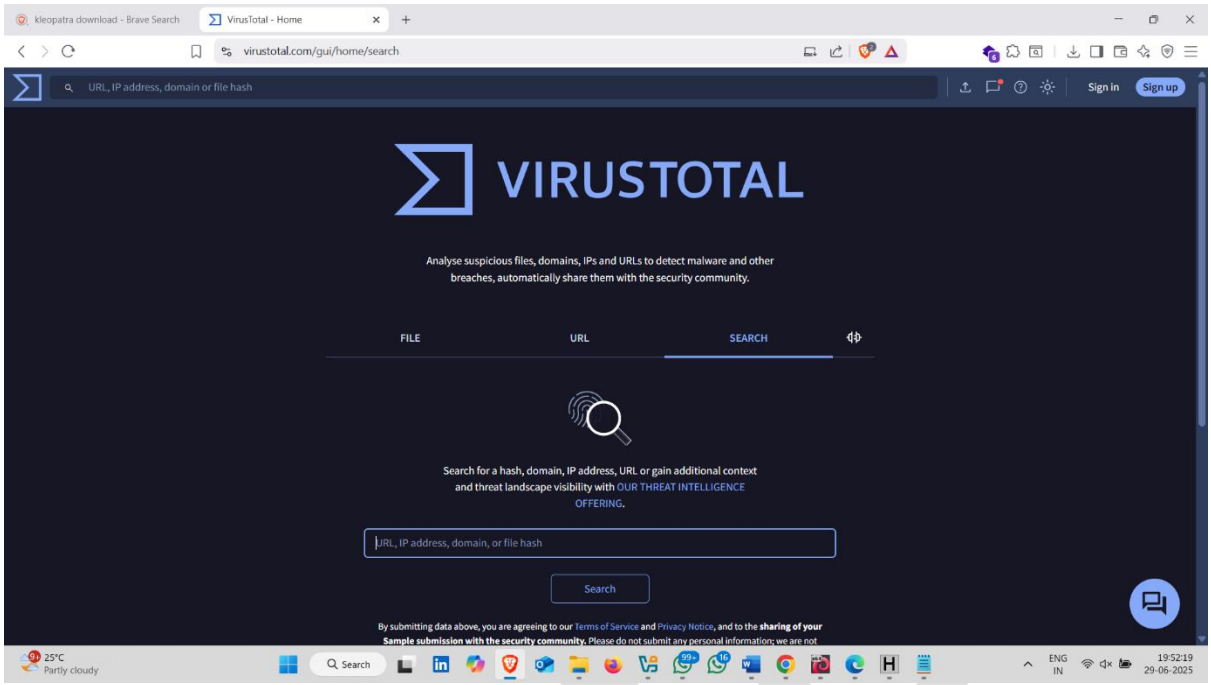
- just simply copy hash



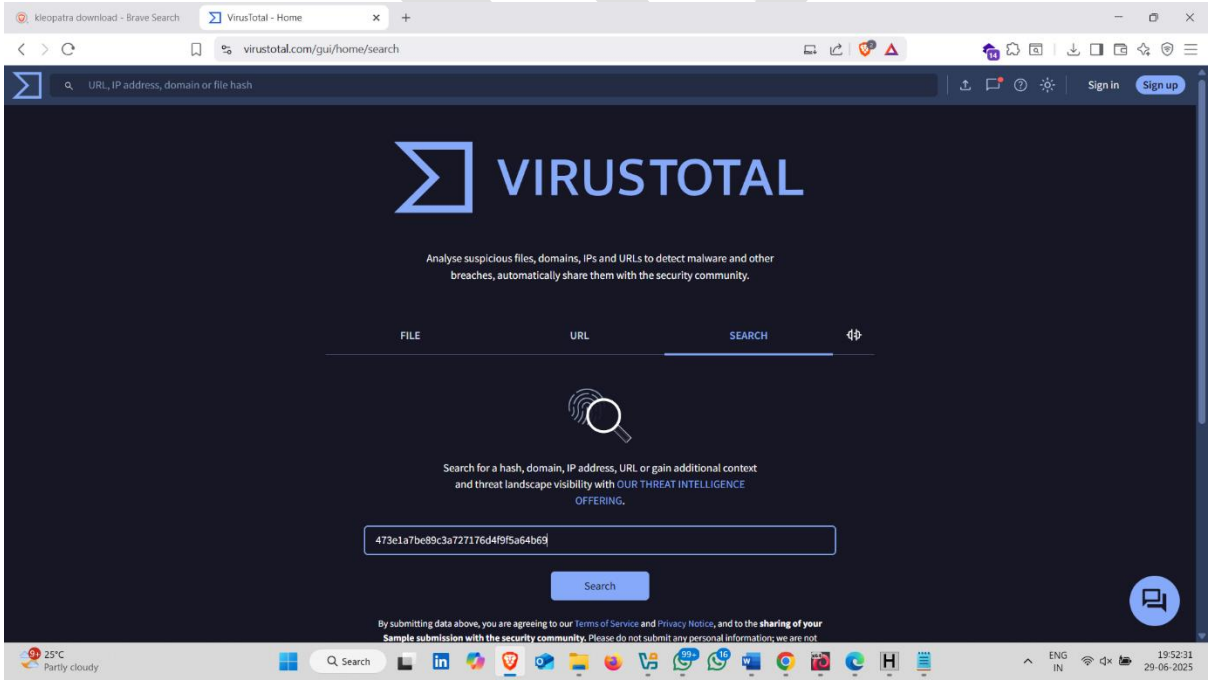
- open browser and search virus total website



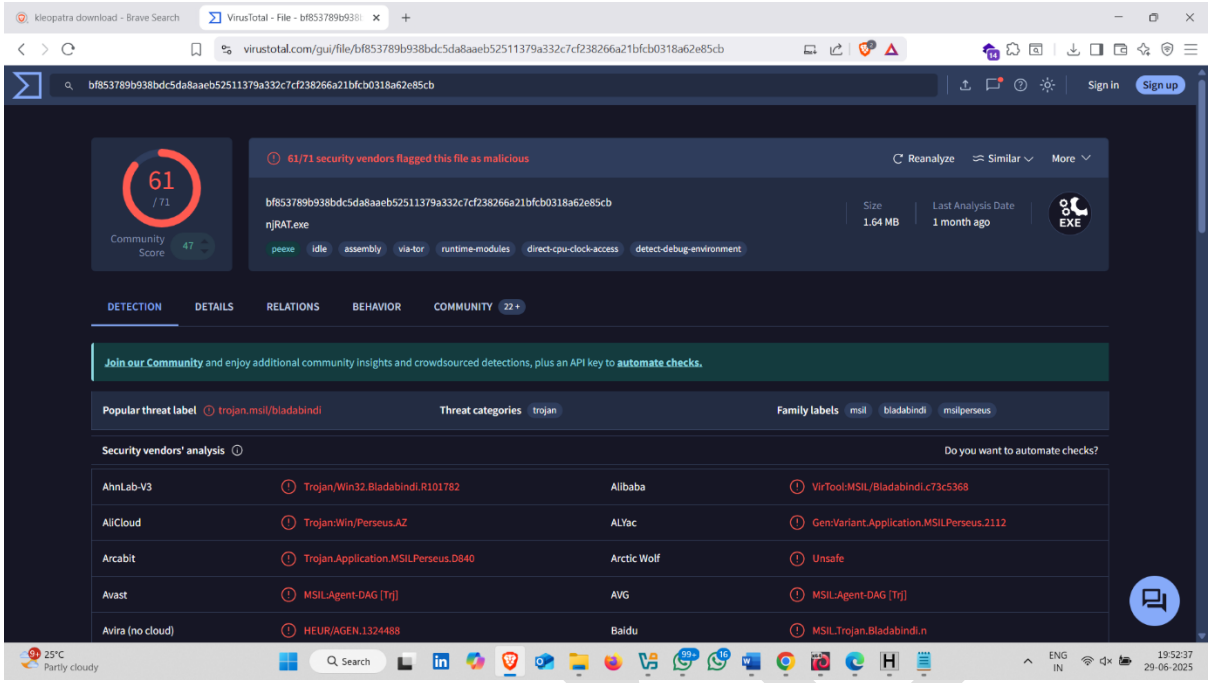
- in search section paste hash



- click on **search**



- here , it shows it's a malicious program hash 👇 ✅



8.CyberChef

CyberChef (aka "The Cyber Swiss Army Knife") is a powerful, free, web-based tool developed by **GCHQ (UK's intelligence agency)**. It allows you to **analyze, decode, encode, convert, and encrypt data** in a modular, drag-and-drop way — all right in your browser.

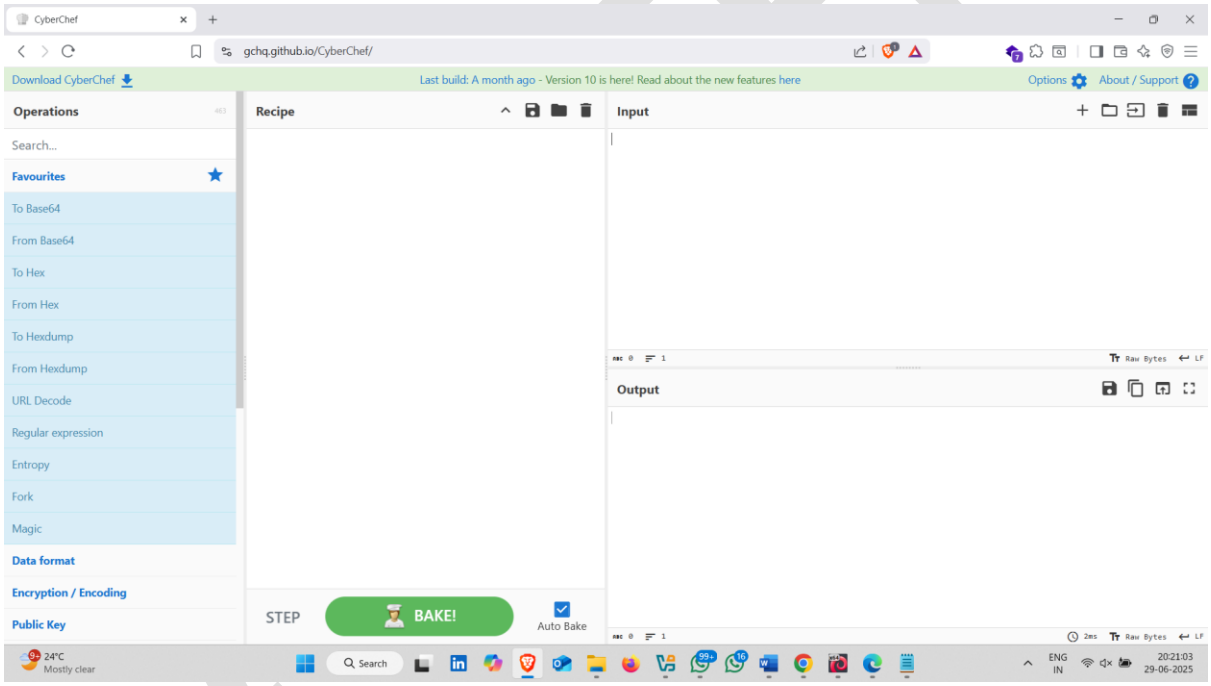
📜 Key Features of CyberChef

Feature	Description
🔐 **Encryption & Decryption**	AES, DES, XOR, RSA, etc.
🧬 **Encoding/Decoding**	Base64, URL, HTML, Hex, etc.
🧠 **Data Analysis**	Entropy, magic bytes, hexdump, etc.
🧱 **Hashing Functions**	MD5, SHA-1, SHA-256, SHA-512, etc.
🔍 **Regex & Text Tools**	Extract emails, IPs, strings

Feature	Description
🛠️ **Forensics Utilities**	Steganography, file carving, JWT inspection
📂 **Drag-and-Drop Interface**	Stack operations visually and edit live
🧩 **No Installation Needed**	Runs entirely in-browser (offline too!)

How to use it -:

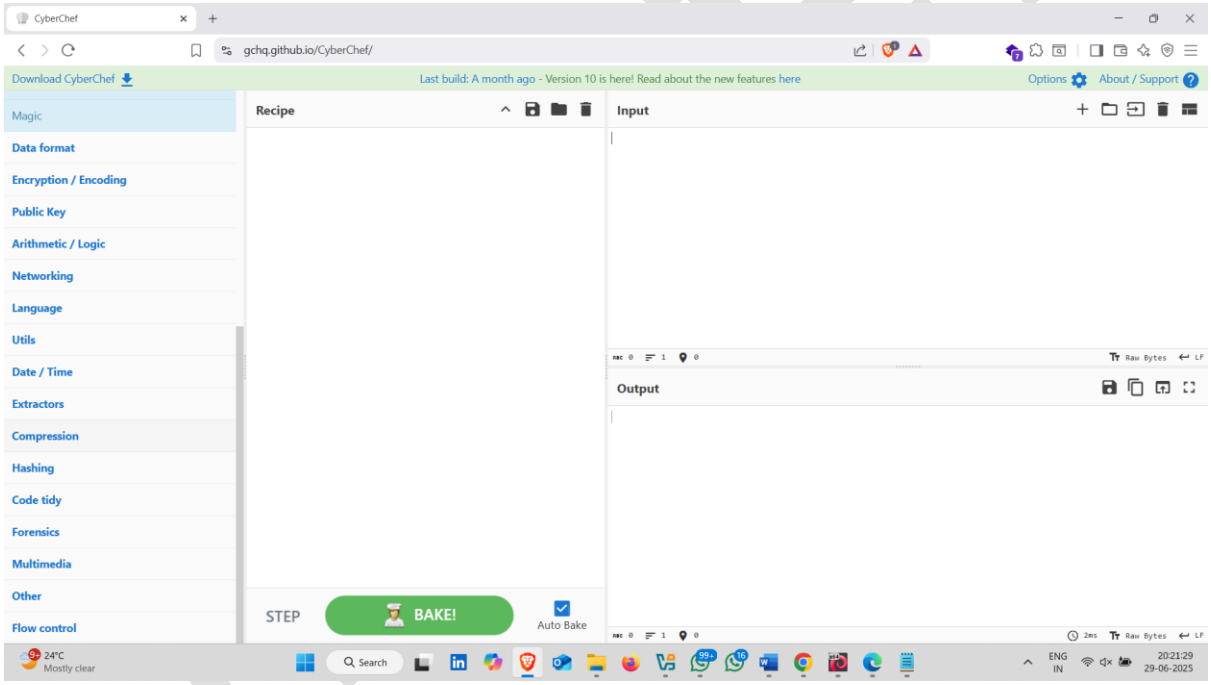
- cyberchef interface ✅ 👇



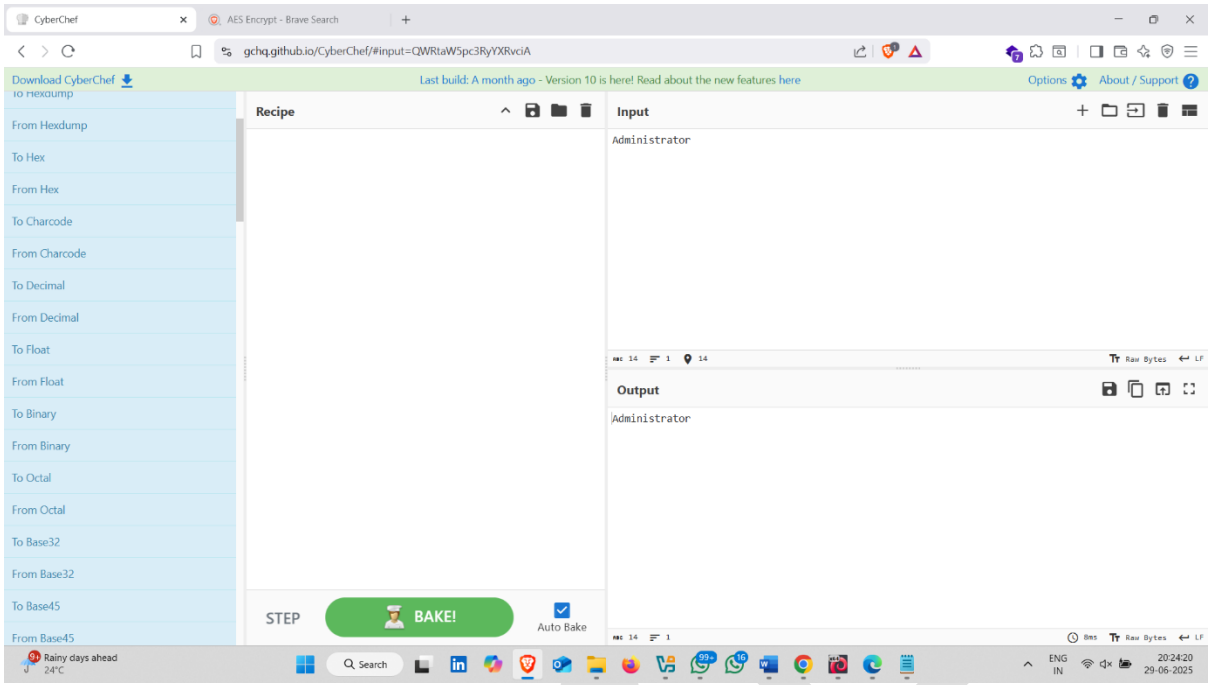
📋 CyberChef Tool Categories – Summary Table

Category	Definition
Magic	Automatically detects and decodes unknown or encoded data.
Data Format	Converts between formats like Base64, Hex, Binary, UTF-8, JSON, etc.
Encryption / Encoding	Encrypts, decrypts, encodes, or decodes data using various algorithms.
Public Key	Works with public-key cryptography like RSA for encryption/decryption.
Arithmetic / Logic	Performs math and logic operations like addition, XOR, shifts, AND/OR.
Networking	Handles IP addresses, ports, MACs, URLs, WHOIS, and network data.
Language	Changes character encodings, translates text, and modifies case or format.
Utils	Provides utility tools like trim, sort, deduplicate, split, clean data.
Date / Time	Converts and calculates between different time and date formats.
Extractors	Extracts patterns like emails, IPs, URLs, and strings from raw data.
Compression	Compresses or decompresses files (gzip, zlib, bzip2, etc.).
Hashing	Generates hash values like MD5, SHA1, SHA256 for integrity checks.
Code Tidy	Beautifies and formats code (HTML, XML, JSON) for easier reading.

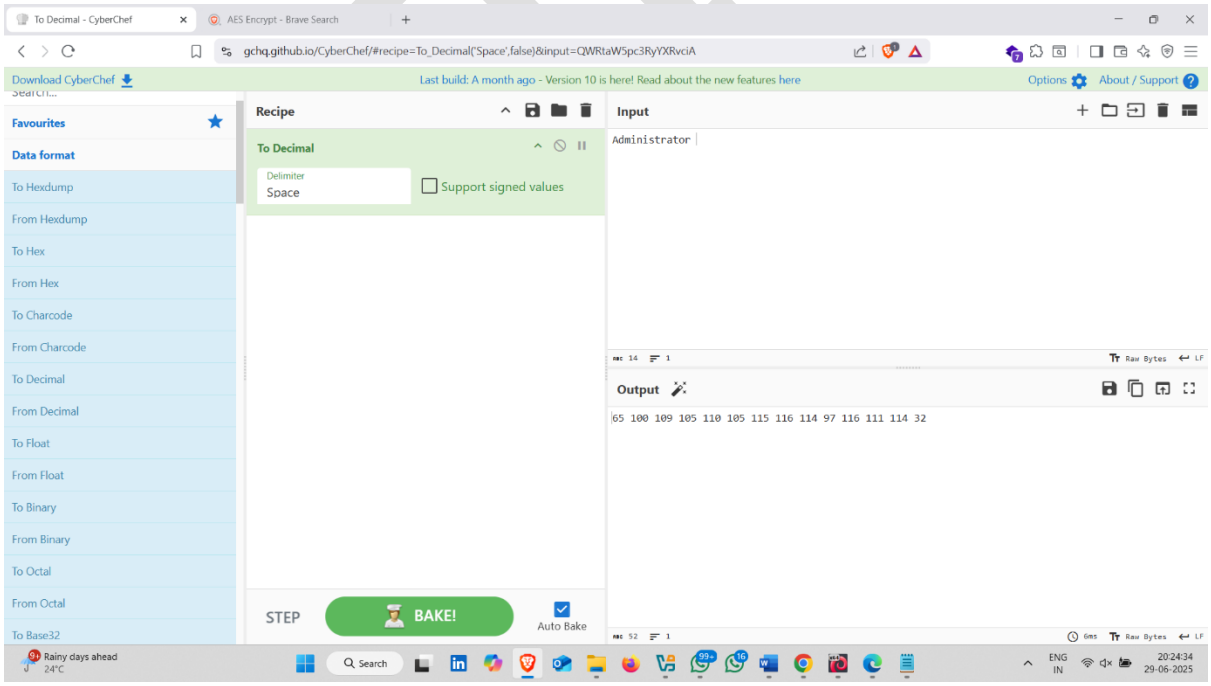
Category	Definition
Forensics	Analyzes files, checks entropy, extracts strings, and investigates data.
Multimedia	Processes media files like images and audio, including metadata extraction.
Other	Miscellaneous tools not categorized elsewhere.
Flow Control	Adds logic (loops, conditions) to control execution flow in recipes.



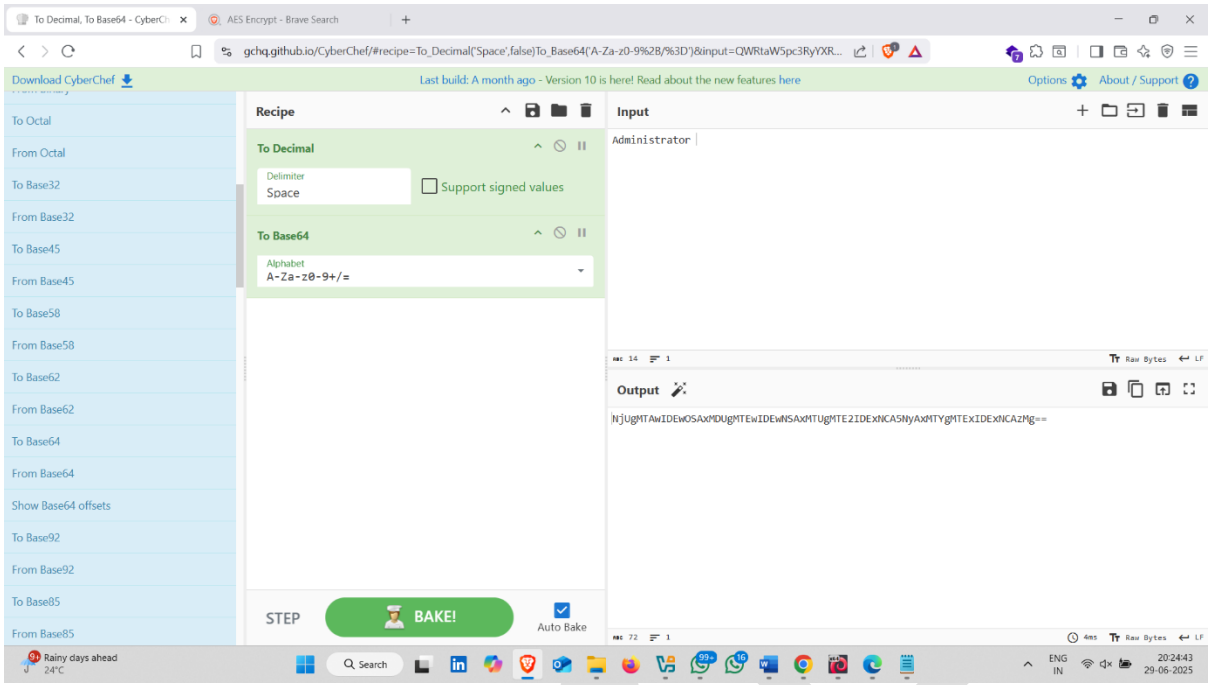
- now enter input



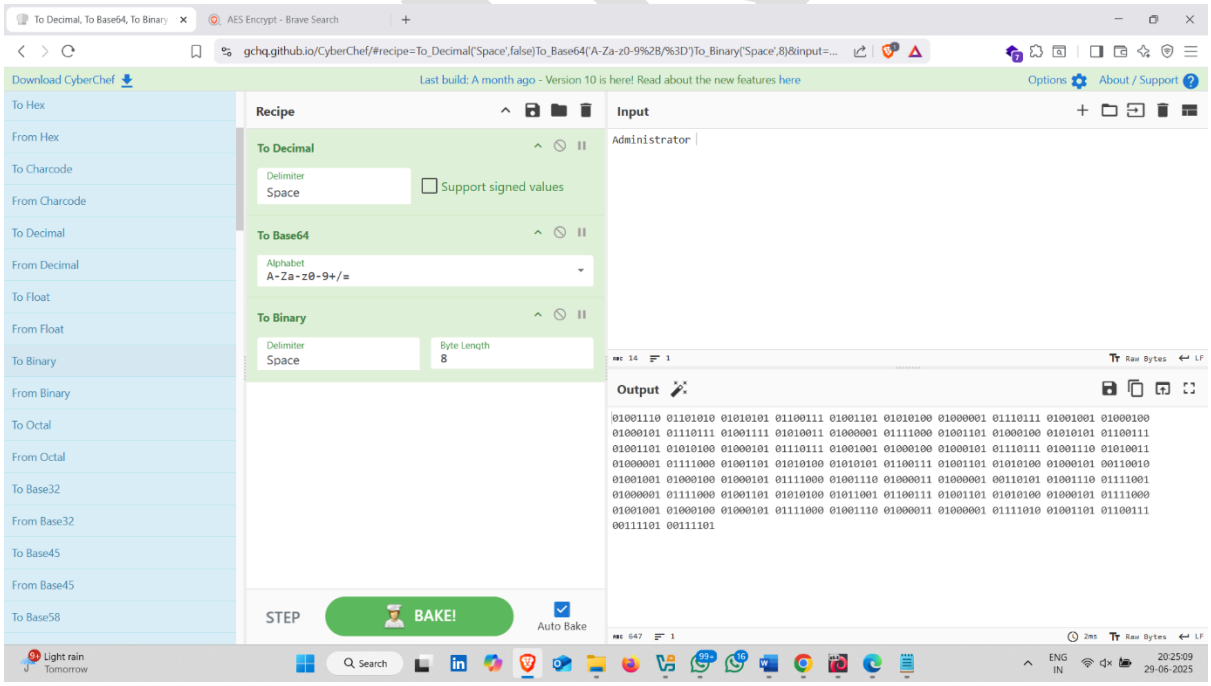
- and select **data format** option and click on **decimal** – it shows your input as a decimal value



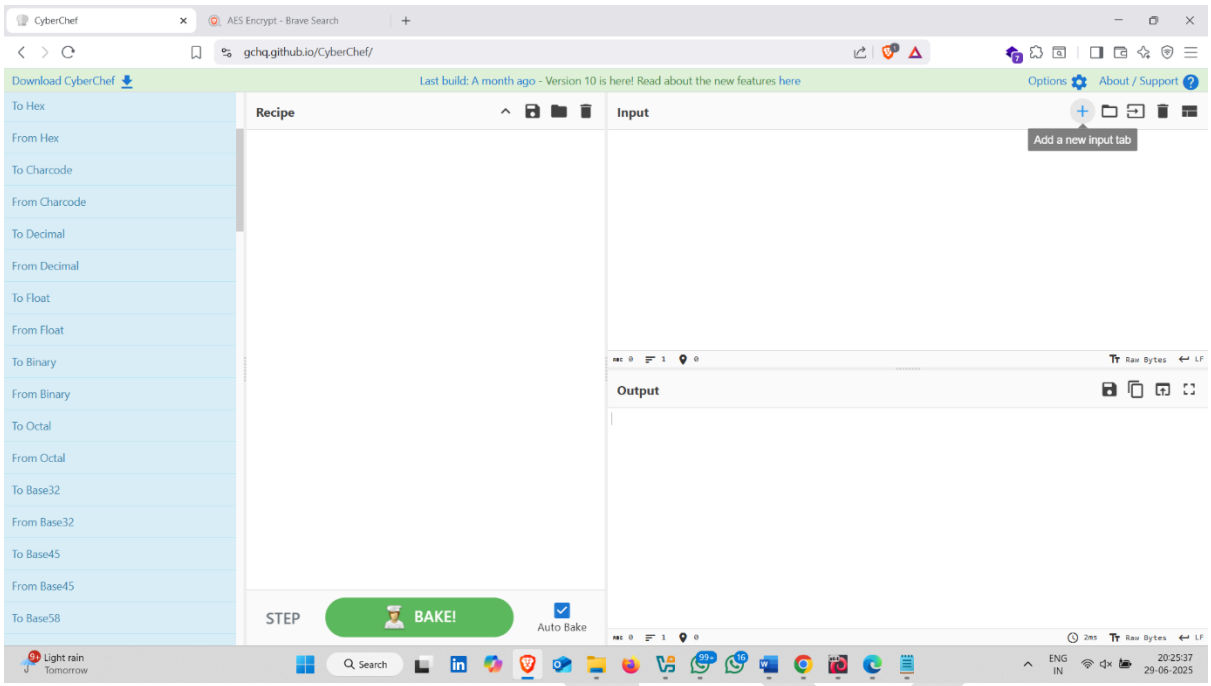
- now select **base64** -- it convert your input in **base64**



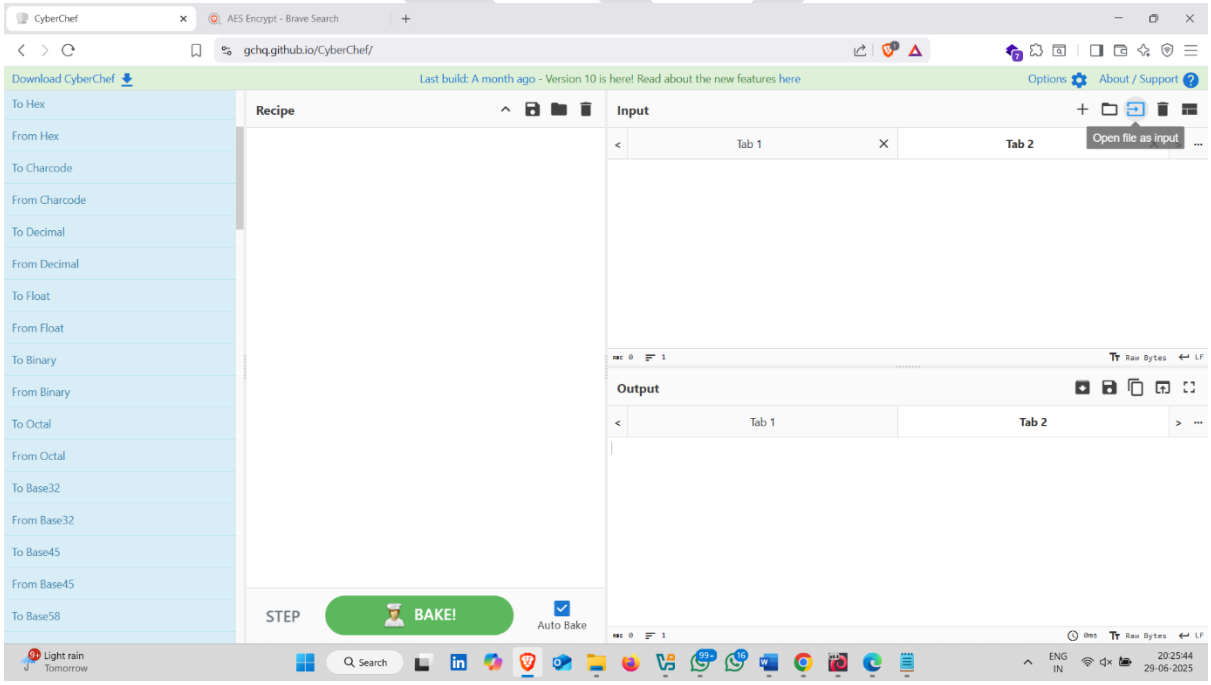
- it also convert in binary



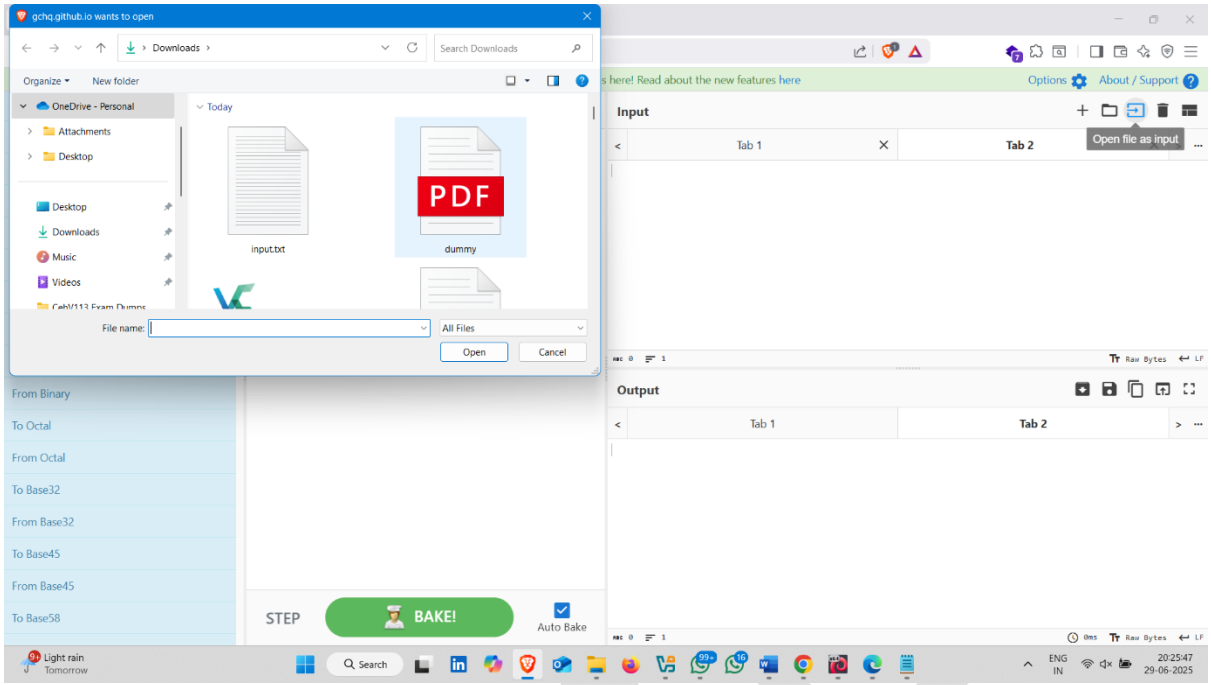
- Now click on “ ➕ ” icon



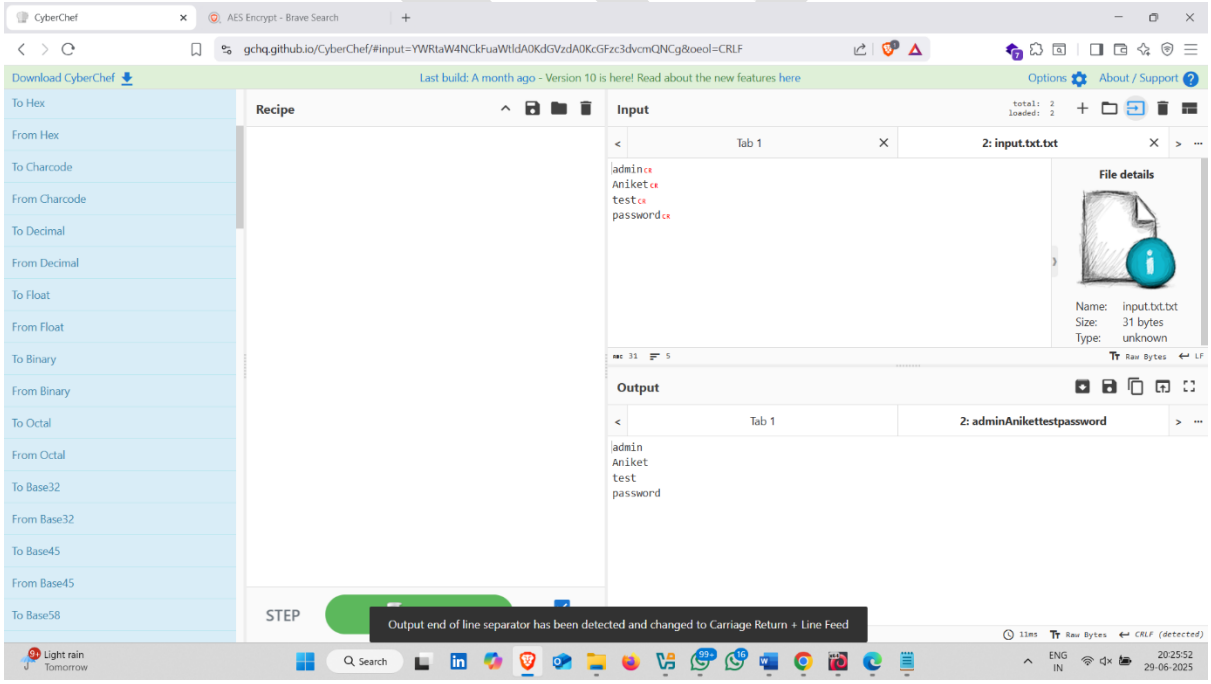
- And click on **open file as input**



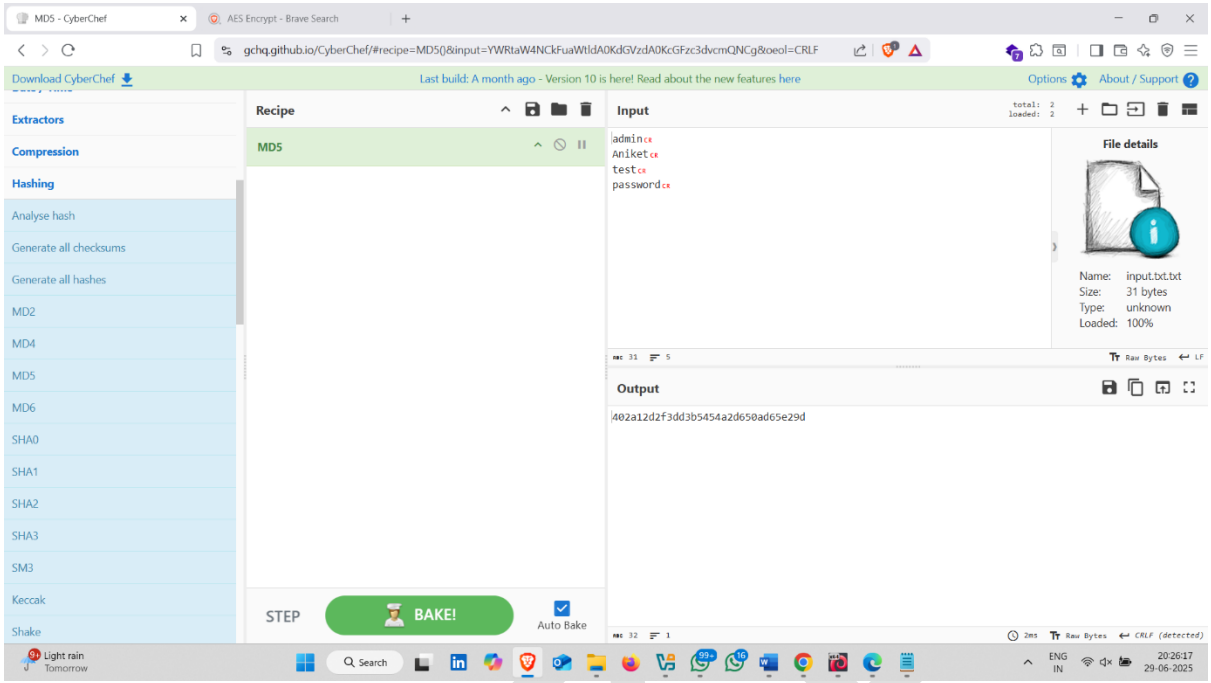
- Select file and click on open



- file as a input



- click on hash and choose **MD5 – it convert your file in MD5 hash**



- now select **SHA3**