

# ByteCrypt

Introduction to Computers and Programming in C (ES202)

Project Presentation

Creator: Aayan Khan

Section: 1CSE-1Y

Enrolment No.: A2305225054

Batch: 2025-29

Teacher: Dr. Ashish Mani

# Key Questions

What?

How?

Why?



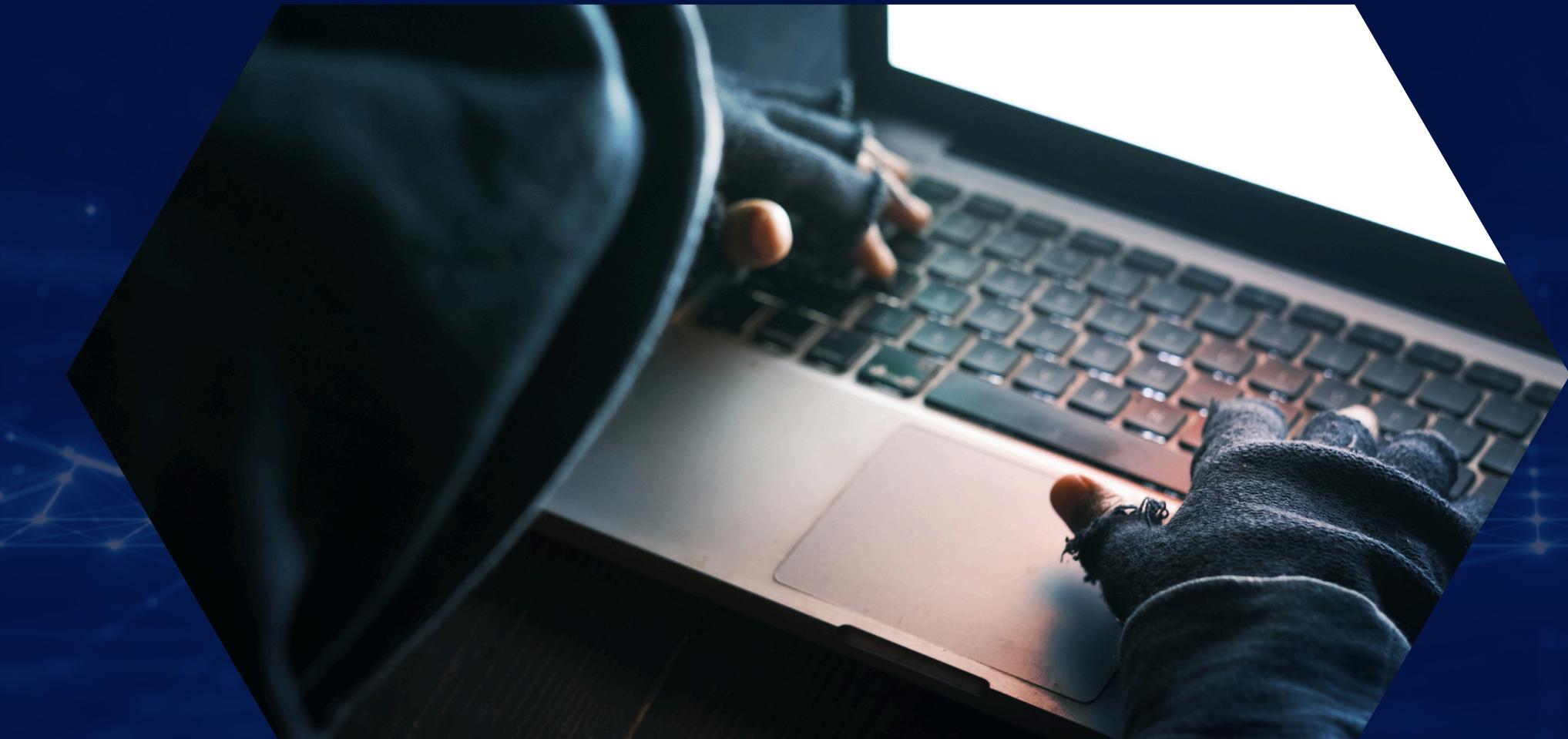
```
3 .
53 ..
915
49:31 boot
15:58 dev
Sep 09:32 etc
Sep 15:52 home
Sep 2015 lib -> usr/lib
. Sep 2015 lib64 -> usr/lib
3. Jul 10:01 lost+found
1. Aug 22:45 mnt
38. Sep 2015 opt
21. Sep 15:52 private -> /home/encr
21. Sep 08:15 proc
12. Aug 15:37 root
21. Sep 15:57 run
0. Sep 2015 sbin -> usr/bin
9. Sep 2015 srv
. Sep 15:51 sys
. Sep 15:45 tmp
. Aug 15:39 usr
Jul 10:25 var
Sep 15:55
```



# What is ByteCrypt?

## Description

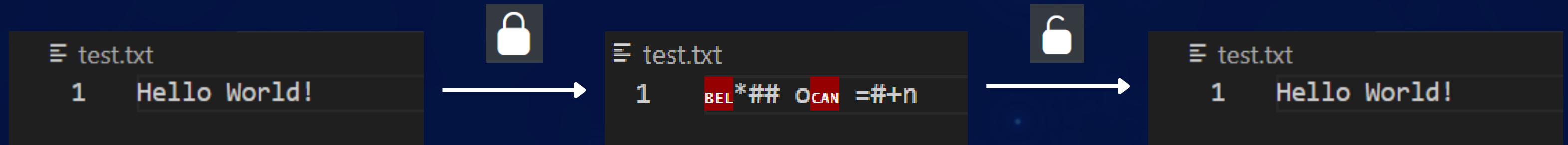
- A C-based project capable of encrypting and decrypting files by performing direct bit-level manipulation of their data.
- The process works by having the user provide a password to secure the document, which generates a corresponding key for use.
- The files can essentially be of any data type, such as audio, plain text, video, or even PDFs.



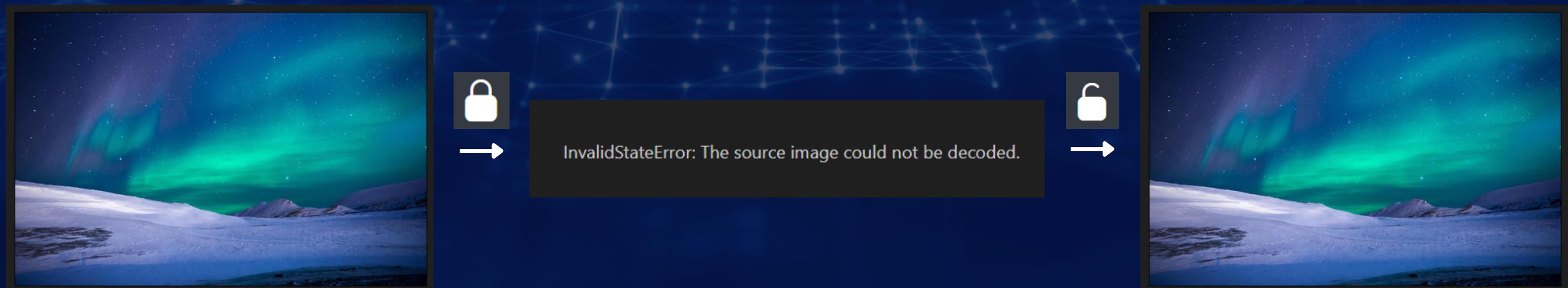
# Demonstration



## # Plain Text File

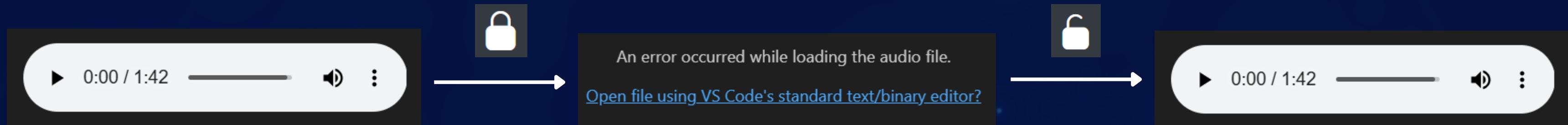


## # Image File

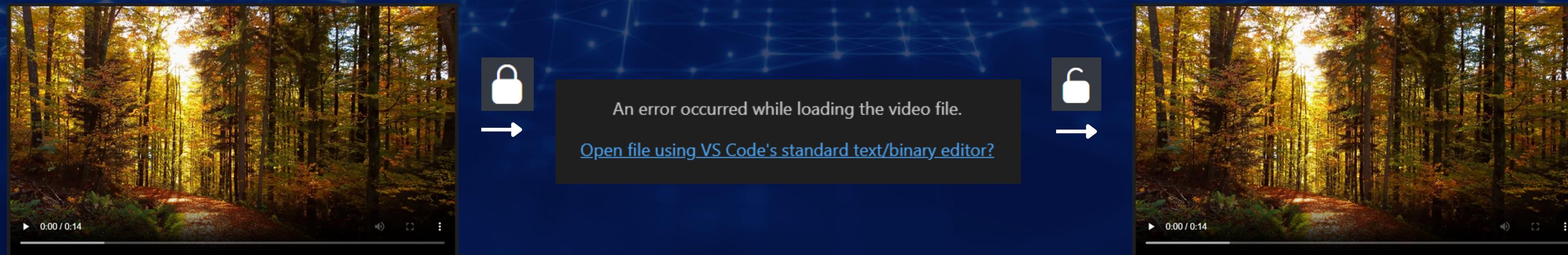


# Demonstration

## # Audio File



## # Video File



# Demonstration

## # PDF File

### Sample PDF

*This is a simple PDF file. Fun fun fun.*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus facilisis odio sed mi. Curabitur suscipit. Nullam vel nisi. Etiam semper ipsum ut lectus. Proin aliquam, erat eget pharetra commodo, eros mi condimentum quam, sed commodo justo quam ut velit. Integer a erat. Cras laoreet ligula cursus enim. Aenean scelerisque velit et tellus. Vestibulum dictum aliquet sem. Nulla facilisi. Vestibulum accumsan ante vitae elit. Nulla erat dolor, blandit in, rutrum quis, semper pulvinar, enim. Nullam varius congue risus. Vivamus sollicitudin, metus ut interdum eleifend, nisi tellus pellentesque elit, tristique accumsan eros quam et risus. Suspendisse libero odio, mattis sit amet, aliquet eget,



Invalid or corrupted PDF file.

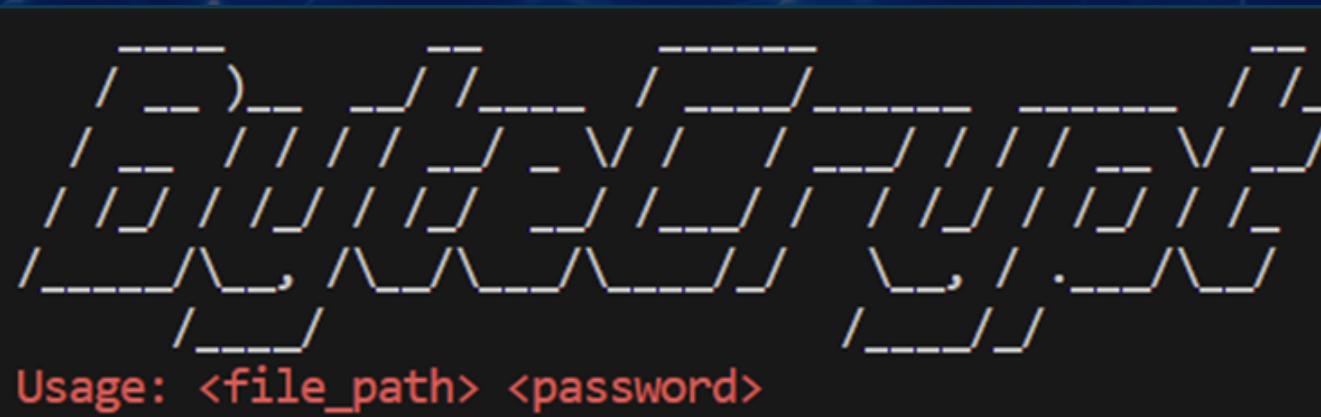


### Sample PDF

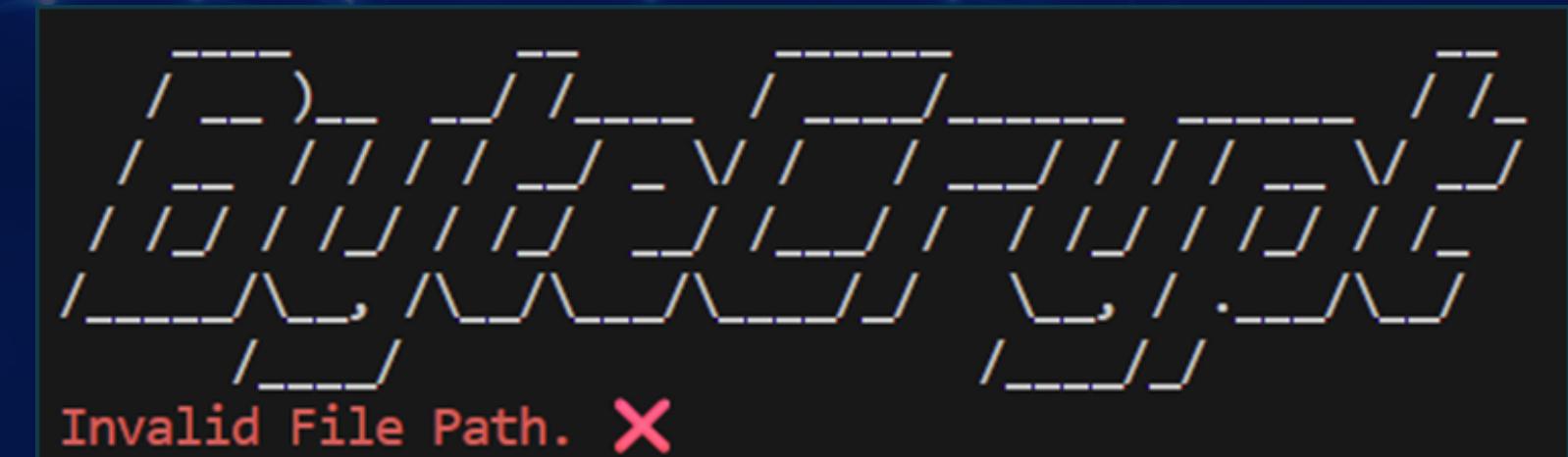
*This is a simple PDF file. Fun fun fun.*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus facilisis odio sed mi. Curabitur suscipit. Nullam vel nisi. Etiam semper ipsum ut lectus. Proin aliquam, erat eget pharetra commodo, eros mi condimentum quam, sed commodo justo quam ut velit. Integer a erat. Cras laoreet ligula cursus enim. Aenean scelerisque velit et tellus. Vestibulum dictum aliquet sem. Nulla facilisi. Vestibulum accumsan ante vitae elit. Nulla erat dolor, blandit in, rutrum quis, semper pulvinar, enim. Nullam varius congue risus. Vivamus sollicitudin, metus ut interdum eleifend, nisi tellus pellentesque elit, tristique accumsan eros quam et risus. Suspendisse libero odio, mattis sit amet, aliquet eget,

## #Invalid Usage



## #Invalid File Path Provided



# How does it work?

## Generating the Key

- The user enters a password along with the file path through the CLI.
- The Daniel Bernstein Rolling Algorithm is used to generate the key.
- The algorithm entails:
  - A base key value of 5381
  - A multiplier variable (33)
  - For each char in password
    - $\text{key} = \text{key} * \text{multiplier} + (\text{int}) \text{char};$
  - Return  $\text{key \% 256}$



# How does it work?

## Securing the File

- The file is opened in binary read-write mode.
- An infinite loop is started:
  - A chunk buffer of 4096 bytes is created.
  - Data is read from the file, with the number of bytes read stored in a variable n.
  - If n is 0, EOF → break the loop.
  - For each byte in the chunk
    - $\text{chunk}[\text{byte}] = \text{chunk}[\text{byte}] \wedge \text{key};$
  - Pointer moves backwards by n bytes.
  - Modified chunk is written back.
  - Changes are flushed.
- The file is closed.



# Why?



- The primary motivation was to create a simple, fast, and accessible method for securing files on local systems.
- This thereby reduced reliance on heavy external tools or complex cryptographic libraries.
- By utilizing XOR-based encryption with a password-derived key, this project provides an easy-to-understand and platform-independent approach to file protection.

---

Thank You

