**Hide.py**

import argparse

from hashlib import sha256

def embed\_message(doc\_file, secret\_msg, output\_file):

with open(doc\_file, 'r') as file:

content = file.read()

# Generate a hash of the secret message

hash\_digest = sha256(secret\_msg.encode()).hexdigest()

# Embed the hash and the secret message into the document content

embedded\_msg = f"{content}\n###{hash\_digest}:{secret\_msg}###"

with open(output\_file, 'w') as file:

file.write(embedded\_msg)

print("Secret message embedded successfully.")

def main():

parser = argparse.ArgumentParser()

parser.add\_argument('-f', help='Document File', dest='docfile', required=True)

parser.add\_argument('-m', help='Enter your Secret Message', dest='secretmsg', required=True)

parser.add\_argument('-o', help='Output File Path and Name', dest='outputfile', required=True)

args = parser.parse\_args()

doc\_file = args.docfile

secret\_msg = args.secretmsg

output\_file = args.outputfile

try:

embed\_message(doc\_file, secret\_msg, output\_file)

except FileNotFoundError:

print("File not found. Please provide a valid file path.")

except Exception as e:

print("An error occurred:", e)

if \_name\_ == "\_main\_":

main()

**extract.py**

import argparse

from hashlib import sha256

def extract\_message(doc\_file):

with open(doc\_file, 'r') as file:

content = file.read()

start\_index = content.rfind("###")

end\_index = content.rfind("###", 0, start\_index)

if start\_index != -1 and end\_index != -1:

embedded\_msg = content[end\_index + 3:start\_index]

# Split the embedded message into hash and message

hash\_digest, secret\_msg = embedded\_msg.split(':', 1)

# Re-calculate the hash of the message

recalculated\_hash = sha256(secret\_msg.encode()).hexdigest()

# Compare the recalculated hash with the original hash

if recalculated\_hash == hash\_digest:

return secret\_msg

else:

return None

else:

return None

def main():

parser = argparse.ArgumentParser()

parser.add\_argument('-f', help='Document File', dest='docfile', required=True)

args = parser.parse\_args()

doc\_file = args.docfile

extracted\_msg = extract\_message(doc\_file)

if extracted\_msg:

print("Extracted secret message:", extracted\_msg)

else:

print("No valid hidden message found.")

if \_name\_ == "\_main\_":

main()