# Encryption and Decryption Tool using Python

Student Name: Mayar Yousef Al-saeed
Student ID: 461210731
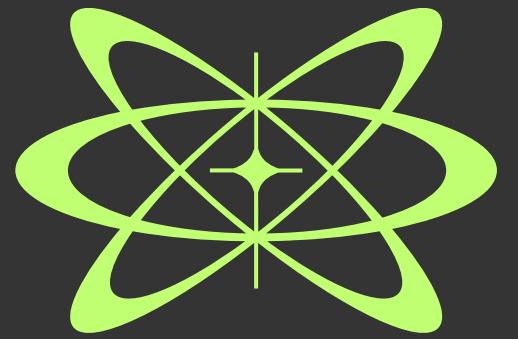
What is Encryption?
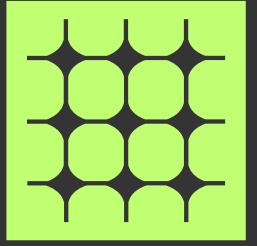
- ***Encryption***: Converting readable information-plaintext-into an unreadable information -Ciphertext-
- ***Decryption*** : The inverse process—converting Ciphertext back to Plaintext using the correct Secret Key.
- ***Importance***: protects personal and sensitive information
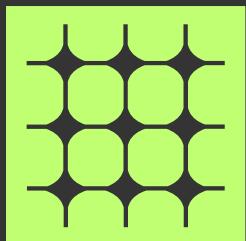
The Role in Cybersecurity
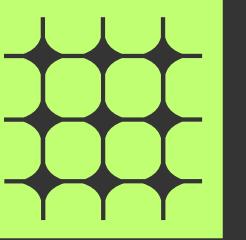
- **The Goal of Cybersecurity**: Protecting digital systems, networks, and data
- **Encryption achieves three core security objectives:**
1. **Confidentiality** : Data remains secret.
2. **Integrity** : Data has not been altered or tampered with.
3. **Authentication** : Verifying the identity of the sender or data source

Python Cryptography
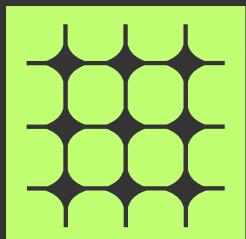
- *Python provides powerful libraries, like cryptography, to easily apply encryption.*
- *Cryptography is the science of securing information using mathematical methods.*
- *It ensures confidentiality, integrity, and authenticity of data.*
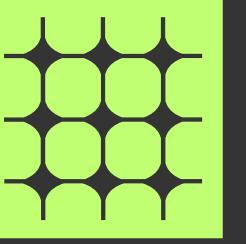
History

- *Ancient ciphers: Caesar cipher*
- *WWII: Enigma machine.*
- *Modern encryption: AES and RSA.*
- *Cryptography protects all digital communication today.*
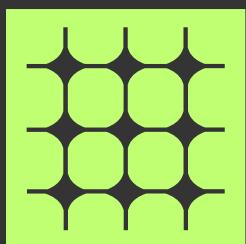
Algorithms
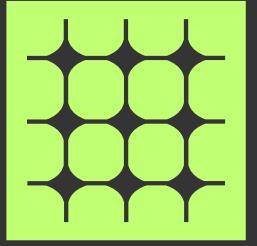
- **Symmetric Encryption:**
1. *One shared key (fast but needs secure exchange).*
2. *Example: AES, Fernet.*
- **Asymmetric Encryption:**
3. *Public and private keys (more secure).*
4. *Example: RSA.*
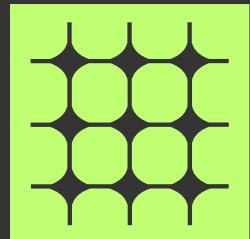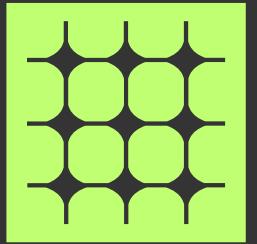
Selected Algorithm: Fernet
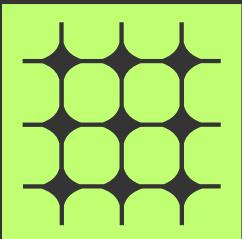
- *Uses AES + HMAC for encryption and verification.*
- *Secure and easy to use.*
- *Great for learning and projects.*
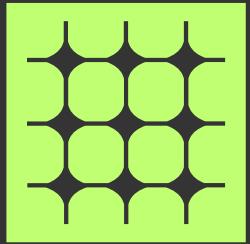
Program

```python
#First, we need to import (bring in) the 'Fernet' tool from the cryptography library to use it.
from cryptography.fernet import Fernet
# This line creates a unique Secret Key that will be used for both locking and unlocking the message.
# Simply put, this is our private password for the encryption process.
key = Fernet.generate_key()
# Now, we create the actual 'cipher' object (the lock/unlock tool) and give it the 'key' it needs to operate.
cipher = Fernet(key)
#--- Encryption (Locking the Message) ---

# This is the original, readable message (Plaintext) that we want to protect.
message = "Hello, this is a secret message."
# We must convert the normal text (string) into 'bytes' because computers encrypt data in byte format Think of it as preparing the message for the l
encoded_message = message.encode()
# This is the main action: we use the 'cipher' tool to lock (encrypt) the message bytes.
encrypted = cipher.encrypt(encoded_message)
# We print the result. The 'b' prefix shows it's a byte string (the unreadable Ciphertext).
print("Encrypted message:", encrypted)

# --- Decryption (Unlocking the Message) ---

# We use the same 'cipher' tool to unlock (decrypt) the ciphertext.
# This only works because the 'cipher' object already holds the correct 'key'.
decrypted_bytes = cipher.decrypt(encrypted)

# Since the result is still in bytes, we must convert it back to readable text (string) using .decode().
decrypted = decrypted_bytes.decode()

# The final step: printing the original, unlocked message!
print("Decrypted message:", decrypted)
```
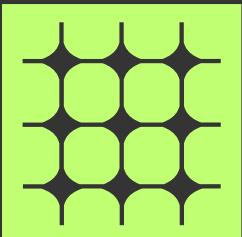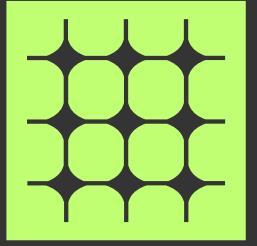
Results

Encrypted message: b'gAAAAABpA36OZodLqKJltX5j2vmm54_veJf2n-VKchMXAipqLu3x3LeQEKTNiS1aM5YcqteznOZg7t_ky4AjfdUu3hTF4GHm7qhSkIQ9dOojUQSc-HwtCLU0Ga6_VJpbh5naKiPzMAZh
'
Decrypted message: Hello, this is a secret message.
PS C:\Users\myals\OneDrive\Desktop\PythonVScode>
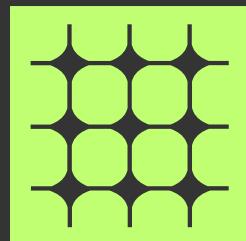
- **Encrypted Message:** *This output, a long string of unreadable characters, visually confirms confidentiality. It demonstrates that without the correct key, the data is protected from any unauthorized party.*

- **Decrypted Message:** *The successful restoration of the original plaintext confirms the integrity and reliability of the cryptographic process, showing that the encryption is fully reversible only when the correct secret key is applied*

Conclusion

- *The tool shows the importance of encryption in cybersecurity.*
- *Fernet ensures data confidentiality and integrity.*
- *Future work: add RSA and key management.*

THANK YOU