INTERNSHIP REPORT

Encryption Techniques and Data Security

---

1. Task Title

Study and Implementation of Encryption Techniques and Secure Communication

---

2. Objective

The objective of this task was to understand fundamental cryptographic concepts and implement encryption techniques to protect data confidentiality, integrity, and authenticity. This task aimed to provide practical exposure to symmetric and asymmetric encryption, hashing, digital signatures, and real-world security applications such as HTTPS and VPNs.

---

3. Introduction

In today's digital world, securing data is a critical requirement. Cryptography plays a vital role in protecting sensitive information from unauthorized access and tampering. This task focused on understanding encryption algorithms, key management, hashing mechanisms, and their practical use in real-world applications. Through theoretical learning and hands-on implementation, core data security principles were explored.

---

4. Concepts Studied

4.1 Symmetric and Asymmetric Encryption

Symmetric encryption uses a single shared secret key for both encryption and decryption, making it fast and efficient. However, secure key distribution remains a challenge. Asymmetric encryption uses a pair of keys—public and private—where data encrypted with one key can only be decrypted using the other. This method is slower but more secure for key exchange and authentication.

4.2 AES File Encryption

Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm known for its speed and strong security. Files encrypted using AES become unreadable without the correct secret key. AES is commonly used for file protection, disk encryption, and secure data storage.

4.3 RSA Key Generation

RSA is an asymmetric encryption algorithm used to generate a public-private key pair. The public key can be shared openly, while the private key must remain confidential. RSA is primarily used for secure communication, digital signatures, and exchanging symmetric keys securely.

4.4 Digital Signatures

Digital signatures are used to verify the authenticity and integrity of data. They ensure that the sender is genuine and that the data has not been modified during transmission. Digital signatures combine hashing and asymmetric encryption and are widely used in secure emails, software updates, and online transactions.

4.5 Hashing and Integrity Verification

Hashing algorithms convert data into a fixed-length hash value. Even a minor change in the original file results in a completely different hash. Hashing is used to verify data integrity and detect tampering. Algorithms such as SHA-256 are commonly used for this purpose.

4.6 Comparison of Encryption Algorithms

Encryption algorithms were compared based on performance, security strength, and real-world usage. Symmetric algorithms like AES are faster and suitable for large data encryption, while asymmetric algorithms like RSA provide secure key exchange but are computationally expensive.

---

5. Real-World Applications

Encryption technologies are widely implemented in daily digital communication.

- **HTTPS** ensures secure communication between web browsers and servers by encrypting data in transit.
- **Virtual Private Networks (VPNs)** encrypt internet traffic to protect user privacy and prevent data interception on public networks.
  These systems use a combination of encryption, hashing, and digital certificates.

---

6. Tools and Techniques Used

- Cryptographic concepts and algorithms
- File encryption and decryption techniques

- Hashing and integrity verification methods
- Key generation and digital signature concepts

---

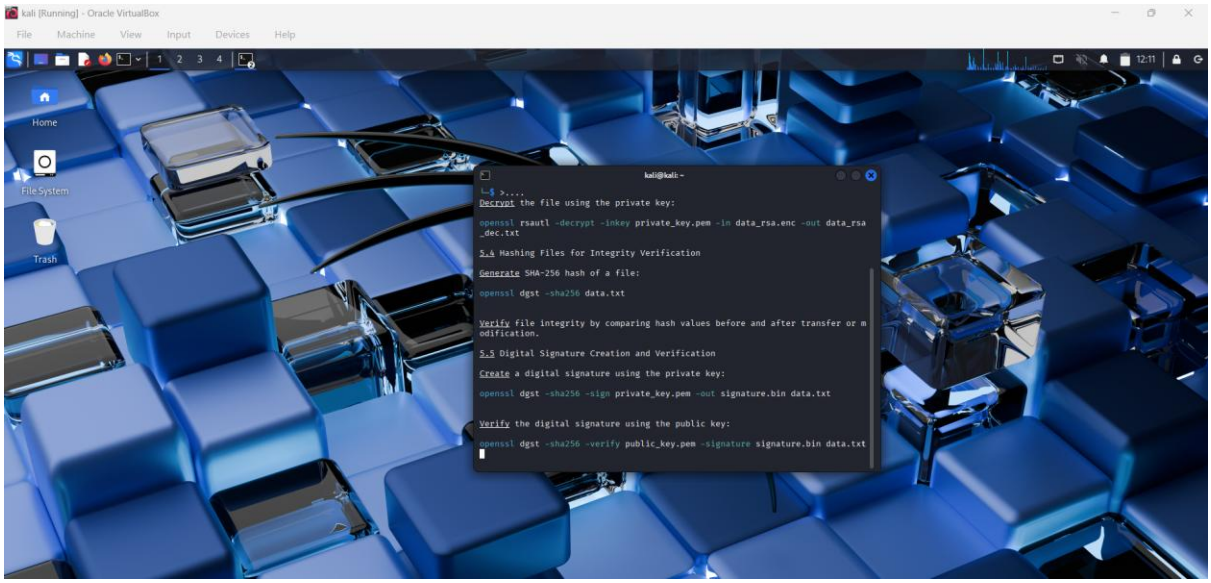
7. Outcome and Learning Experience

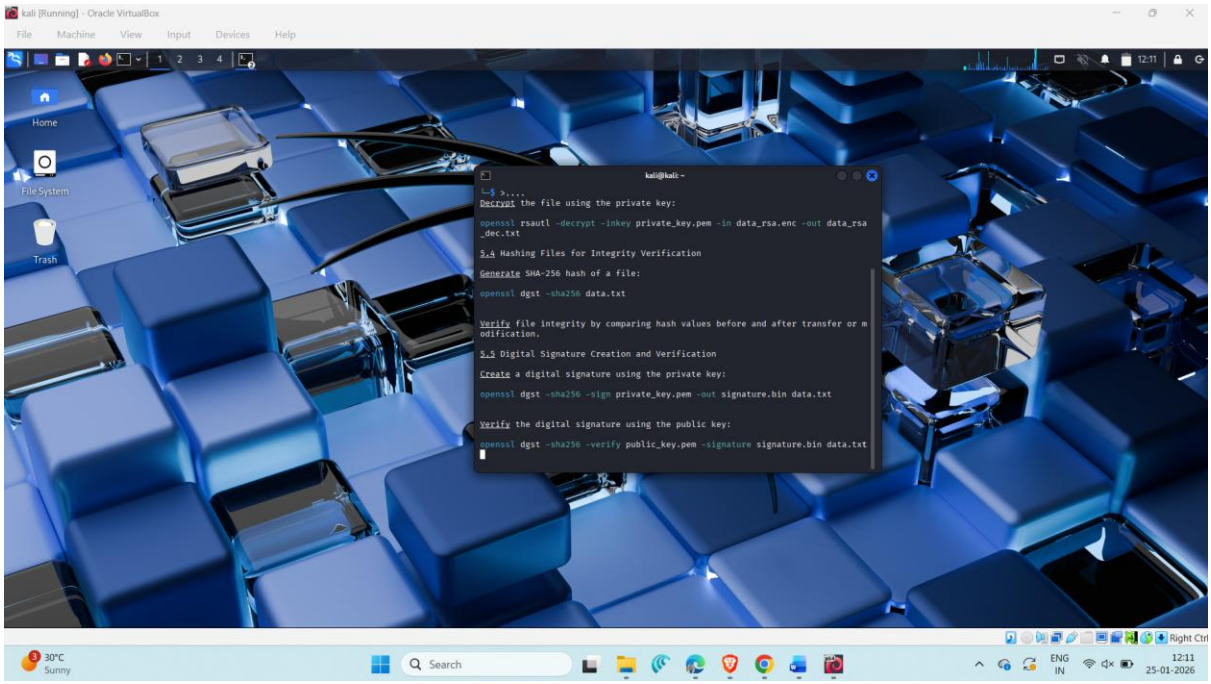
Through this task, a strong understanding of encryption fundamentals and secure communication mechanisms was gained. Practical knowledge of AES encryption, RSA key generation, hashing, and digital signatures enhanced awareness of how data security is implemented in real-world systems.

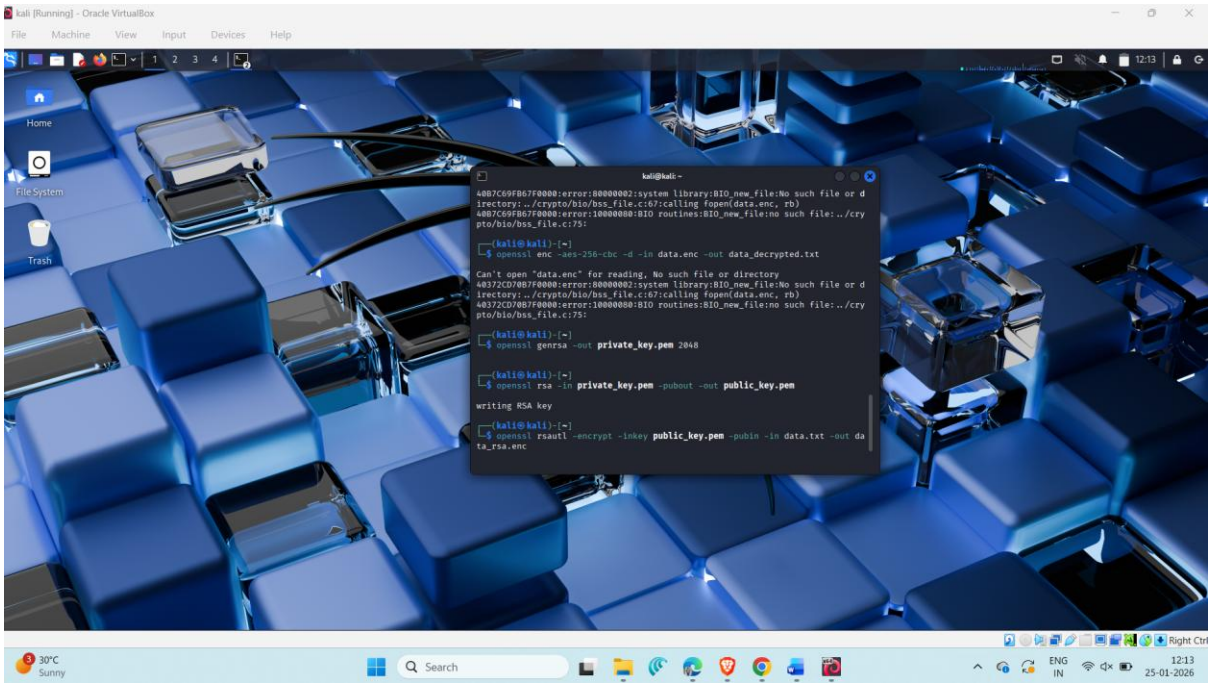
---

8. Conclusion

This task provided valuable insight into cryptography and data security practices. Understanding encryption algorithms and their real-world applications is essential for cybersecurity professionals. The knowledge gained through this task forms a strong foundation for advanced security concepts and secure system design.


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Home
File System
Trash
kali@kali: ~
└─$ >....
Decrypt the file using the private key:
openssl rsautl -decrypt -inkey private_key.pem -in data_rsa.enc -out data_rsa_dec.txt
5.4 Hashing Files for Integrity Verification
Generate SHA-256 hash of a file:
openssl dgst -sha256 data.txt
Verify file integrity by comparing hash values before and after transfer or modification.
5.5 Digital Signature Creation and Verification
Create a digital signature using the private key:
openssl dgst -sha256 -sign private_key.pem -out signature.bin data.txt
Verify the digital signature using the public key:
openssl dgst -sha256 -verify public_key.pem -signature signature.bin data.txt


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Home
File System
Trash
kali@kali: ~
└─$ >....
Decrypt the file using the private key:
openssl rsautl -decrypt -inkey private_key.pem -in data_rsa.enc -out data_rsa_dec.txt
5.4 Hashing Files for Integrity Verification
Generate SHA-256 hash of a file:
openssl dgst -sha256 data.txt
Verify file integrity by comparing hash values before and after transfer or modification.
5.5 Digital Signature Creation and Verification
Create a digital signature using the private key:
openssl dgst -sha256 -sign private_key.pem -out signature.bin data.txt
Verify the digital signature using the public key:
openssl dgst -sha256 -verify public_key.pem -signature signature.bin data.txt
Right Ctrl
30°C
Sunny
Search
ENG
IN
12:11
25-01-2026


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Home
File System
Trash
kali@kali: ~
40B7C69FB67F0000:error:80000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss_file.c:67:calling fopen(data.enc, rb)
40B7C69FB67F0000:error:10000080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:75:
(kali@kali)-[~]
$ openssl enc -aes-256-cbc -d -in data.enc -out data_decrypted.txt
Can't open "data.enc" for reading, No such file or directory
40372CD70B7F0000:error:80000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss_file.c:67:calling fopen(data.enc, rb)
40372CD70B7F0000:error:10000080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:75:
(kali@kali)-[~]
$ openssl genrsa -out private_key.pem 2048
(kali@kali)-[~]
$ openssl rsa -in private_key.pem -pubout -out public_key.pem
writing RSA key
(kali@kali)-[~]
$ openssl rsautl -encrypt -inkey public_key.pem -pubin -in data.txt -out data_rsa.enc
Right Ctrl
30°C
Sunny
Search
ENG
IN
12:13
25-01-2026