Cryptography Experiment Report

1. Aim

To study and implement basic cryptographic techniques including symmetric a nd asymmetric encryption, hashing, and digital signatures, and to understand th eir real-world applications in securing data and communications.

---

2. Objectives

* Understand symmetric and asymmetric encryption
* Encrypt files using AES
* Generate RSA public–private key pairs
* Learn the concept of digital signatures
* Hash files and verify data integrity
* Compare encryption algorithms
* Study real-world cryptography applications
* Document observations and findings

---

3. Tools and Technologies Used

* OpenSSL (for encryption, hashing, and key generation)
* Linux command line
* Text files (sample data)
* Cryptographic algorithms (AES, RSA, SHA-256)

---

4. Theory

4.1 Symmetric Encryption

Uses a single secret key for both encryption and decryption. It is fast and effici ent but requires secure key distribution.

Example: AES (Advanced Encryption Standard)

4.2 Asymmetric Encryption

Uses a key pair consisting of a public key and a private key. It solves the key di stribution problem but is slower than symmetric encryption.

Example: RSA

4.3 Hashing

A one-way process that converts data into a fixed-length hash value. It ensures data integrity but does not provide confidentiality.

Example: SHA-256

4.4 Digital Signatures

Used to verify authenticity, integrity, and non-repudiation of data using hashing and asymmetric encryption.

---

5. Experimental Procedure

5.1 File Encryption Using AES

* A plaintext file was encrypted using AES symmetric encryption.
* A password-based key was used for encryption.

Result:
The encrypted file was unreadable without the correct key.

5.2 RSA Key Generation

* A public and private key pair was generated using RSA.
* The public key was used for encryption.
* The private key was used for decryption.

Result:
Only the private key holder could decrypt the encrypted data.

5.3 Digital Signature Creation

* The file was hashed.
* The hash was encrypted using the sender’s private key.
* The receiver verified it using the sender’s public key.

Result:
File authenticity and integrity were successfully verified.

5.4 Hashing and Integrity Verification

* The file was hashed before and after modification.
* Hash values were compared.

Result:
Any change in the file produced a different hash value.

---

6. Comparison of Encryption Algorithms

Algorithm	Type	Speed	Security	Usage
AES	Symmetric	Very Fast	Very High	File & disk encryption
RSA	Asymmetric	Slow	High	Key exchange
ECC	Asymmetric	Faster than RSA	Very High	Mobile & IoT
DES	Symmetric	Fast	Low	Obsolete

---

7. Real-World Applications

* **HTTPS:** Secure web communication using TLS
* **VPN:** Encrypts network traffic over public networks
* **Disk Encryption:** Protects stored data
* **Secure Email:** Ensures confidentiality and authenticity

---

8. Observations

* Symmetric encryption is efficient for large data
* Asymmetric encryption is essential for secure key exchange
* Hashing effectively detects data tampering
* Digital signatures ensure trust and accountability
* Modern systems use a hybrid cryptographic approach