

# **BABU BANARASI DAS UNIVERSITY**



## **SCHOOL OF Computer Application**

**Department of Cybersecurity and forensics**

**Identity and Access Management.**

**Session 2025-26**

**(BCACSN13102)**

## **PRACTICAL LAB FILE**

**SUBMITTED BY: -**

NAME – ADARSH GUPTA

Roll No - 1240264

SECTION– BCA CS 22

**SUBMITTED TO: -**

Mr. Anand Kumar Gupta

# INDEX

S.No.	Name of Experiments	Sign/ Remark
1.	RSA ENCRYPTION AND DECRYPTION	

# RSA Encryption & Decryption — Practical Exercise Report

**Title:** RSA Encryption & Decryption Practical — Report

**Author:** [ADARSH GUPTA]

**Team / Course:** [BCA (CS&F) / IBM faculty practice]

**Date:** [2025-10-10]

**Objective:**

- Demonstrate RSA key generation, encryption, and decryption in a controlled lab environment.
  - Explain security considerations: key sizes, padding schemes (PKCS#1 v1.5 vs OAEP), and common pitfalls.
- 

## 1. Executive Summary

Briefly summaries the practical: what was implemented, tested, and the main conclusions (1–3 paragraphs). Example:

This exercise demonstrates generation of RSA key pairs, the encryption of plaintext using public keys, and decryption using private keys. Implementations were tested .The exercise highlights the importance of proper padding (OAEP recommended) and appropriate key sizes (2048+ bits) for modern security.

---

## 2. Scope & Environment

- **Scope:** Local lab only — no public key servers, no production data. All keys and test vectors are lab-generated.
- **Environment:**
  - OS: [WINDOWS]
  - Tools: CHROME :- BROWSER , DEVGLAN :- WEBSTIE FOR ENC AND DEC.

## 3. Authorization & Safe Practices

- Confirm instructor approval for the lab work.
  - Use only sample data; do not use or publish private keys for production systems.
  - Keep private keys locally and encrypted if stored.
-

## 4. Background (non-actionable overview)

- RSA is an asymmetric cryptosystem used for confidentiality and key exchange. Security depends on large prime generation, padding scheme, and key length. Modern guidance recommends RSA 2048 bits minimum; consider migrating to elliptic-curve algorithms for high-security needs.

## 5. Methodology / Steps (High-level)

This section explains what was done without including overly prescriptive attack instructions.

Step1) open chrome → in the search box type RSA encryption and decryption → click search

Step2) click on the first website named “devglan”



Step3) scroll down you will see “**Generate RSA Key Pair Online**”

Below is an online tool to perform RSA encryption and decryption as an RSA calculator.

### Generate RSA Key Pair Online

Select RSA Key Size

2048 bit

**Generate RSA Key Pair**

**Public Key(X.509 Format)**

Download Public Key

**Private Key(PKCS8 Format)**

Download Private Key

### RSA Encryption and Decryption

Below is the form to take the inputs for the encryption and decryption. By default, the public/private keys are prepopulated with the keys generated above. You can also supply your own public/private key pairs.

#### RSA Encryption

Enter Plain Text to Encrypt

Enter Plain Text to Encrypt

#### RSA Decryption

Enter Encrypted Text to Decrypt (Base64)

Enter Encrypted Text to Decrypt

[Digital signature software](#)

> Encrypt Image Online

> Online File Encrypt Decrypt

> Online Text Encrypt Decrypt

> Online Bcrypt Hashing

**Accurate Reference, Accurate results**

Explore Our Extensive Portfolio of Reference Materials Across Diverse Industries!

**Learn More**

**Supelco**

Step4) select the size of the key for e.g.: {2048 bit} → click on generate RSA key pair → both the public key and the private key will be generated.

Select RSA Key Size

2048 bit

Generate RSA Key Pair

**Public Key(X.509 Format)**

-----BEGIN PUBLIC KEY-----  
MIIBJANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAnkwFXmruIG/U9ThQ/CQe  
gEWYUibQ7YrAWEV67k12HMD+I3KOWXtqE1ubvpOtJS003XDu3IGU716qM5cd  
WXTqeJoujtz+bZyfxEQ1ubvpOtJS003XDu3IGU716qM5cd  
YnYWK2tG8CqbCNmEAq//FqGZ6Ca1GBLfKirz6d94Wgf8yu4  
D+frVID2YmfhKsPMxNqVLg1r4OCmcwE7iwSTTk0f05/Oa7

Download Public Key

**Private Key(PKCS8 Format)**

-----BEGIN RSA PRIVATE KEY-----  
MIIEVQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSJAQEAoIBAGQeTA9eau4qb9T1  
OF0BJBgS1Zh8JdVJksBYRXnuTXVco34IXco5Zx2p4m6N63P5nJ+P6R0RW5u+  
k60I47dc07eZTVXqotlx1dthYra0bwKps2YOC/8W0ZnoJUYEt8gKvPp33haB/zK  
7gP5+IWUPZiZ+Eqw8ZE2q6uDWq4KZZATULBJNOTR87n85sg7/Ub6rmD9neFH  
K0eWV5N7Npql+nuJ7yOz0pJ15Up9xt28w0V1UgM+ddGiz/tipZKUowcj8DAZWwD

Download Private Key

Online Text Encrypt Decrypt

Online Bcrypt Hashing

Merck

Accurate Reference,  
Accurate results

Explore Our Extensive  
Portfolio of Reference  
Materials Across  
Diverse Industries!

Learn More

Step 5) scroll down a little bit you will see RSA encryption and decryption window this is for encryption of text.

Step 6) write any plain text you want to encrypt in the encryption window → click on encrypt.

**RSA Encryption**

Enter Plain Text to Encrypt

this is a practice assignment of ibm

Enter Public/Private key

-----BEGIN PUBLIC KEY-----  
MIIBJANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAnkwFXmruIG/U9ThQ/CQe  
gEWYUibQ7YrAWEV67k12HMD+I3KOWXtqE1ubvpOtJS003XDu3IGU716qM5cd  
WXTqeJoujtz+bZyfxEQ1ubvpOtJS003XDu3IGU716qM5cd  
YnYWK2tG8CqbCNmEAq//FqGZ6Ca1GBLfKirz6d94Wgf8yu4  
D+frVID2YmfhKsPMxNqVLg1r4OCmcwE7iwSTTk0f05/Oa7

RSA Key Type: ☒ Public key ☐ Private Key

Select Encryption Algorithm

RSA/ECB/PKCS1Padding

Encrypt

Step7) encrypted output will be generated in the window → copy it and then past it in decryption window → click on decrypt then you will see the decrypted text.

## RSA Decryption

Enter Encrypted Text to Decrypt (Base64) ⓘ

```
hyrjVPCGHo/XfRqlyEDzEVAblR1UhW2hGK4nohNtNBb1VzsafBq/j66f  
vZVF5eLXwttuc+943AQ8q9K//LHksdlTC6DGkBLTOfMRD2vfw5egvKU  
S83aqpJmSa+sbl9WM58ek4a8EUivkGrGbcluPEz30j8tZFzSdgoxhz3Y  
wg5TLMQ==
```

Enter Public/Private key ⓘ

```
-----BEGIN RSA PRIVATE KEY-----  
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCeT  
A9eau4gb9T1OFD8JB6gS1ZhSJtDt/KsBYRXruTXYcx34iXco5Ze2p4mi  
6N63P5tnJ+PGORDW5u+k60II47dcO7elZTvXqozlx1dhYra0bwKpsI2Y  
QCr/8WoZnoJrUYEt8qKvPp33haB/zK7gP5+tWUPZIZ+Eqw8zE2q8uD  
Wvg4KZzATuLBJNOTR87n85rsG7/Ub6rmO9neFHK0eWV5N7NpqL+n
```

RSA Key Type: ⓘ ☐ Public key ☒ Private Key

Select Decryption Algorithm ⓘ

RSA/ECB/PKCS1Padding ▼

**Decrypt**

Decrypted Output:

```
this is a practice assignment of ibm
```

---

## 6. Observations & Key Takeaways

- RSA cannot encrypt messages larger than modulus minus padding overhead — hybrid encryption (RSA encrypts symmetric key) is recommended for larger data.
- Always use OAEP (or a modern standardized padding) for RSA encryption to avoid chosen-ciphertext vulnerabilities.
- Use at least 2048-bit keys; prefer 3072-bit or stronger for long-term security depending on the threat model.
- Do not implement your own crypto primitives; use well-reviewed libraries and follow best practices.

---

## 7. Security Considerations & Common Pitfalls

- **Padding oracle vulnerabilities:** Avoid using vulnerable padding modes without authenticated encryption.
- **Deterministic encryption:** Avoid raw RSA encryption without padding (deterministic and insecure).

- **Key storage:** Never commit private keys to public repos. If you must include keys for demonstration, mark them clearly as test/demo keys and consider encrypting them or using ephemeral keys.
  - **Hybrid crypto:** Use RSA to encrypt symmetric keys (AES-GCM) for large data.
- 

## 8. Recommendations

- Use OAEP padding for encryption and PSS for signatures.
  - Implement hybrid encryption for files/large payloads.
  - Enforce secure key sizes and rotate keys periodically.
  - Add comments in code showing how to move from demo to secure production practices.
- 

## 9. Appendix — Files and Code

- `rsa_examples/OpenSSL/` — commands used and sample PEM files (demo keys only).
- `rsa_examples/python/` — `generate_keys.py`, `encrypt.py`, `decrypt.py` with usage examples.
- `test vectors/` — `plaintext.txt`, `cyphertext`, sample outputs.
- Screenshots and console outputs: `screenshots/`

**Important:** If including sample private keys in the repo for demonstration, store them under `demo keys/` and mark clearly. Best practice: do not publish private keys in public repos—use. `.gitignore` to exclude them or make the repo private.

---

*End of RSA practical report template.*