RNS Trust ®

RNS Institute of Technology

Autonomous Institute of Technology Affiliated to VTU
Accredited with NAAC A+ Grade

AI enhanced dual server public key encryption for secure cloud

TEAM

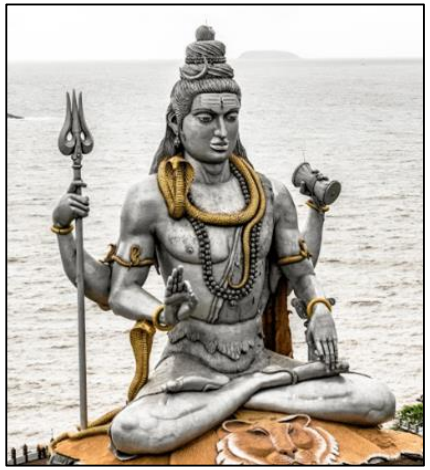
SRUSHTI S	1RN22CY039
BK PRAMILA	1RN23CY402

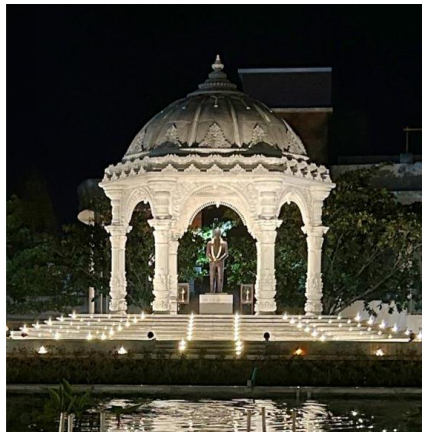
Guide

Mr. Dhanraj

Designation

Assistant Professor





23
Years of
Academic Excellence
RNS INSTITUTE
OF TECHNOLOGY

Tribute to Our Founder

Dr. R N Shetty
Founder
1928 - forever

Agenda

- Introduction & Problem Statement
- Proposed System: Dual Server Model
- Architecture Diagram & Workflow
- Phase 1 Code Implementation Overview
- Hands-on Demonstration (Web UI + Backend)
- Current Output & Results
- Future Scope: AI-Driven Search
- Conclusion

Introduction

With the rise of cloud storage, users upload sensitive data such as personal files, business documents, and medical records to third-party servers.

To protect privacy, data is encrypted before uploading — but once encrypted, searching files using keywords becomes difficult.

Existing solutions like Public Key Encryption with Keyword Search (PEKS) allow users to search over encrypted data without revealing the data itself.

However, traditional **PEKS** systems rely on a single server, which introduces serious security risks.

Our project focuses on a Dual-Server Encryption system to increase security and plans to use AI to enhance keyword search usability.

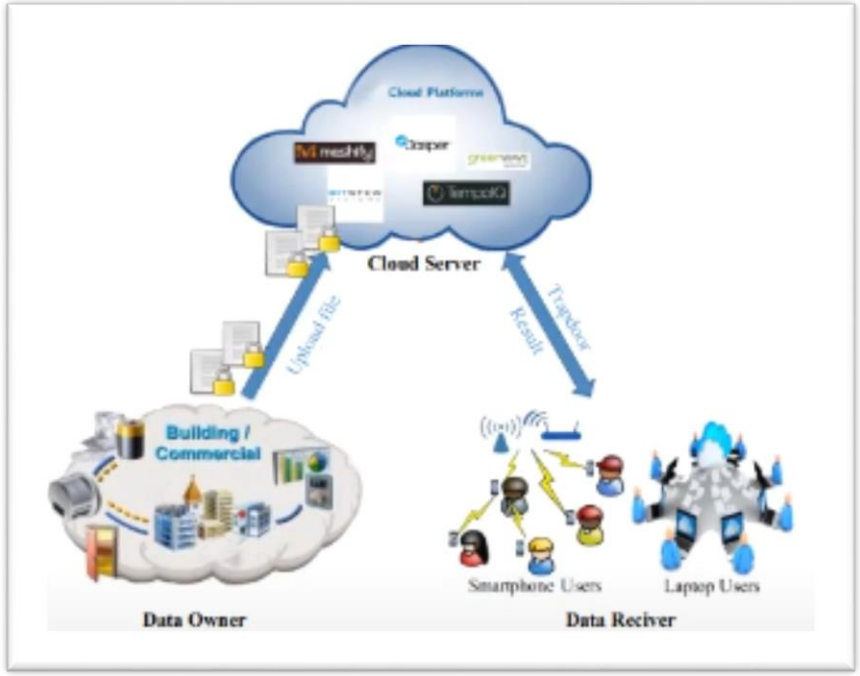
Problem Statement

- In traditional PEKS systems, a single server handles encrypted data storage and keyword-based search.
- This design is vulnerable to Inside Keyword Guessing Attacks (**IKGA**) — where a malicious insider tries to guess keywords and gain unauthorized access.

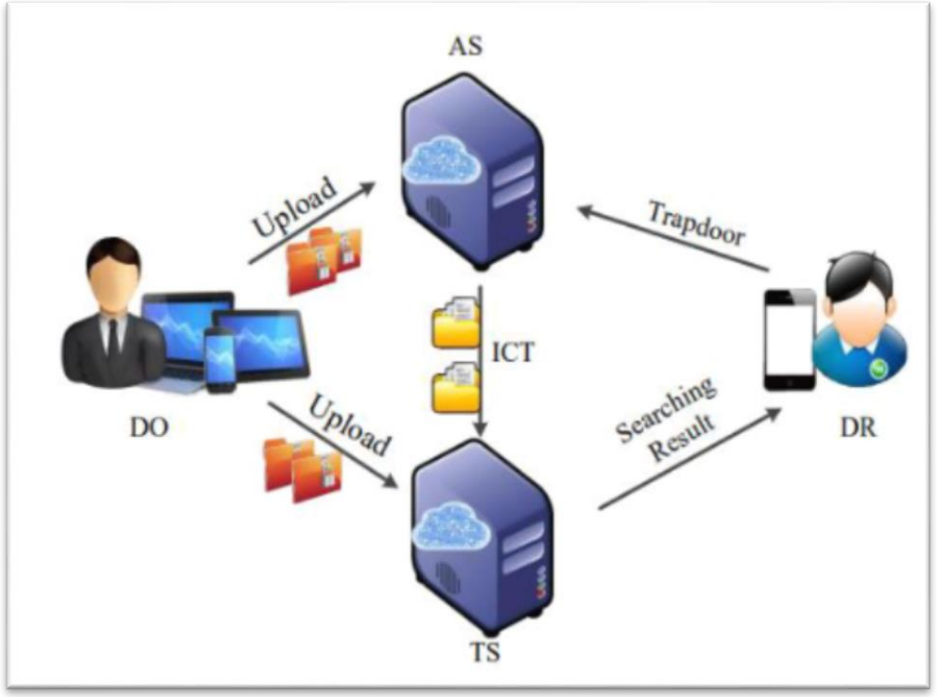
The system requires exact keyword matches, which:

- Reduces usability
- Fails if users make typos or use different terms
- Lacks intelligent or semantic search

Compared:



Traditional Single-Server PEKS model



Dual Server Model

Contd..

Real-World Examples of Related Attacks:

- Apple iCloud Leak (2014):Attackers gained unauthorized access and inferred personal content by guessing login credentials and exploiting metadata.

- Dropbox Metadata Leak (2012): Insiders or attackers accessed file names and search-related data, leading to partial exposure of sensitive content despite encryption.

Proposed System: Dual Server Model

To address the vulnerabilities of traditional single-server PEKS, we propose a Dual Server Architecture:

1. Assistant Server (AS)Receives encrypted keyword/trapdoor from the user. Generates Intermediate Cipher text (ICT).Approves legitimate users and assists in secure file access.

2. Test Server (TS)Verifies the trapdoor keyword against stored keyword hash.Confirms keyword match or mismatch. Only if both AS and TS agree, access is granted to the file.

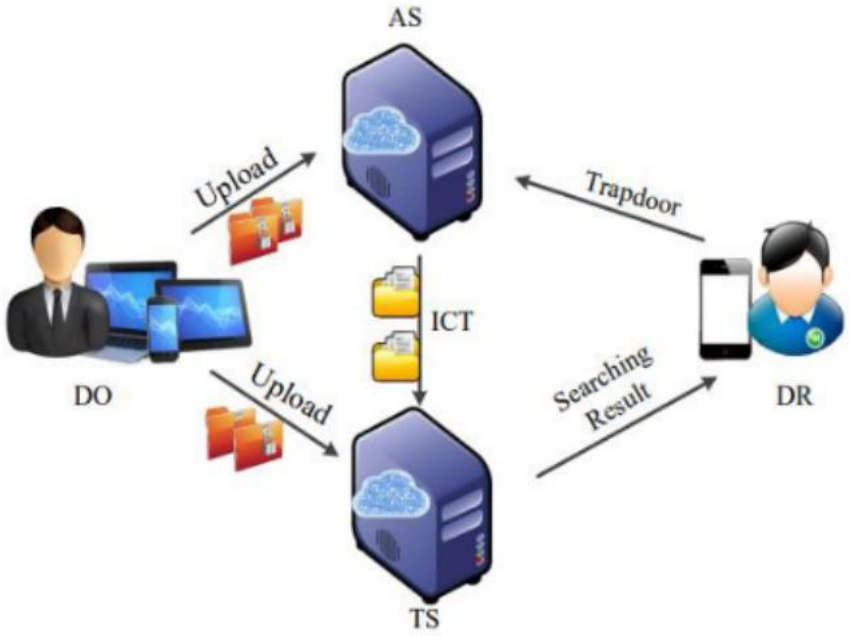
This dual-server model ensures: No single point of trust or failure Protection against IKGA (Inside Keyword Guessing Attack) Better authentication and verification flow

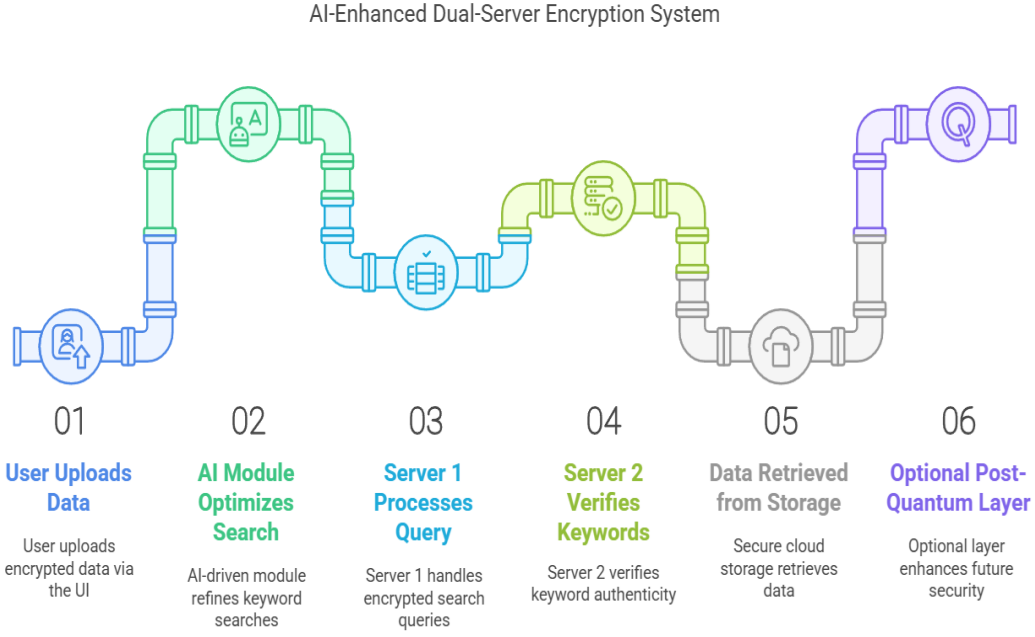
Contd..

- AI-Driven Keyword Search Traditional systems require exact keyword matching. In the next phase, we will integrate AI/NLP techniques to:
- Suggest relevant or corrected keywords
- Enable semantic search (e.g., “secure”=“protected”)
- Improve usability without compromising security

Architecture Diagram & Workflow

Dual server mode





Contd..

Workflow Steps:

- Data Owner: Inputs data and a keyword Data is encrypted Keyword is hashed and stored securely Trapdoor is generated from the keyword
- Assistant Server (AS):Receives trapdoor Generates Intermediate Cipher text (ICT)
- Test Server (TS):Receives ICT and compares trapdoor against stored keyword hash Confirms match or mismatch
- If both AS and TS agree → Encrypted data is retrieved/decrypted for the Data Receiver

Contd..

Data Receiver enters an imprecise or related keyword (e.g., "confidential").

AI/NLP Engine analyzes and suggests the best-matching keyword (e.g., "secure").

Suggested keyword is: Converted to trapdoor Sent to AS and TS for verification.

Normal dual-server encryption workflow continues.

Phase 1 Code Implementation Overview

- Technologies Used
- Programming Language: Python
- Web Framework: Flask
- Frontend: HTML, CSS, Bootstrap (via templates)
- Tools: Visual Studio Code (VS Code)

Contd..

app.py – Main Web Application (Flask):

- Manages routes (/, /upload, /search)
- Handles user input, keyword hashing, and encryption
- Displays result (match/mismatch) via index.html

encryption.py – Encryption Module

Contains logic to:

- Encrypt user data
- Hash the keyword
- Generate trapdoor for secure keyword matching

3

Contd..

assistant.server.py – Assistant Server:

- Accepts the trapdoor
- Generates Intermediate Cipher text (ICT)
- Assists in verifying user request validation

test.server.py – Test Server:

- Verifies keyword trapdoor against the stored hash
- Confirms whether the keyword matches
- Grants or denies access based on match result

Algorithms Used:

SHA-256 for Keyword Hashing

- Industry-standard cryptographic hash function Produces a fixed 256-bit hash Secure, irreversible, and collision-resistant

Custom XOR/Basic Symmetric Encryption

- Lightweight and fast for prototype/demo Easy to implement in educational projects

- **Hands-on Demonstration (Web UI +Backend)**

Current Output & Results:

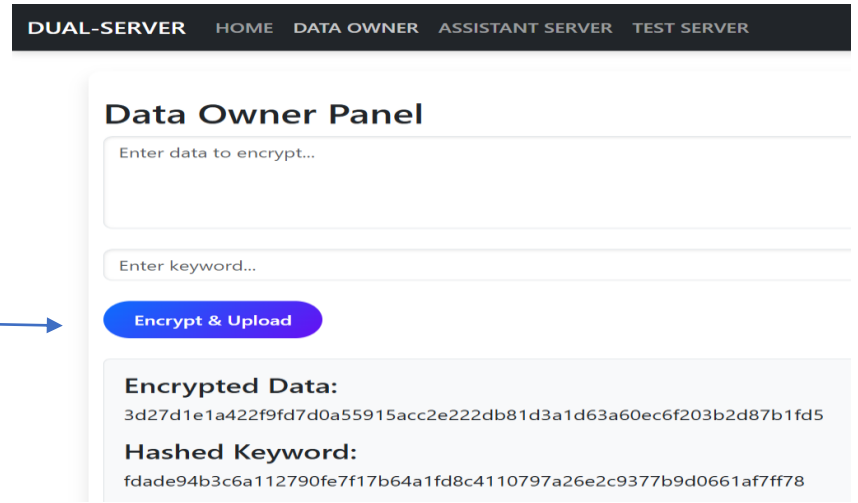


Figure 1: Data uploaded with encrypted keyword

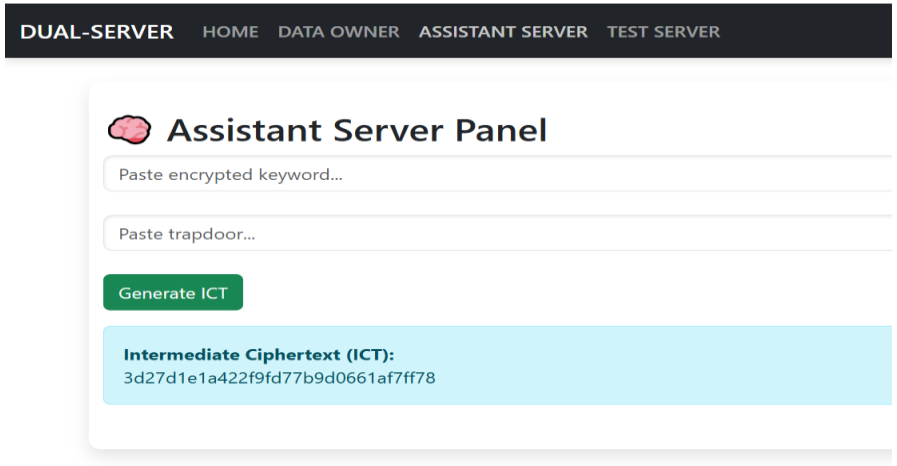


Figure 2: Intermediate cipher text generation

Contd..

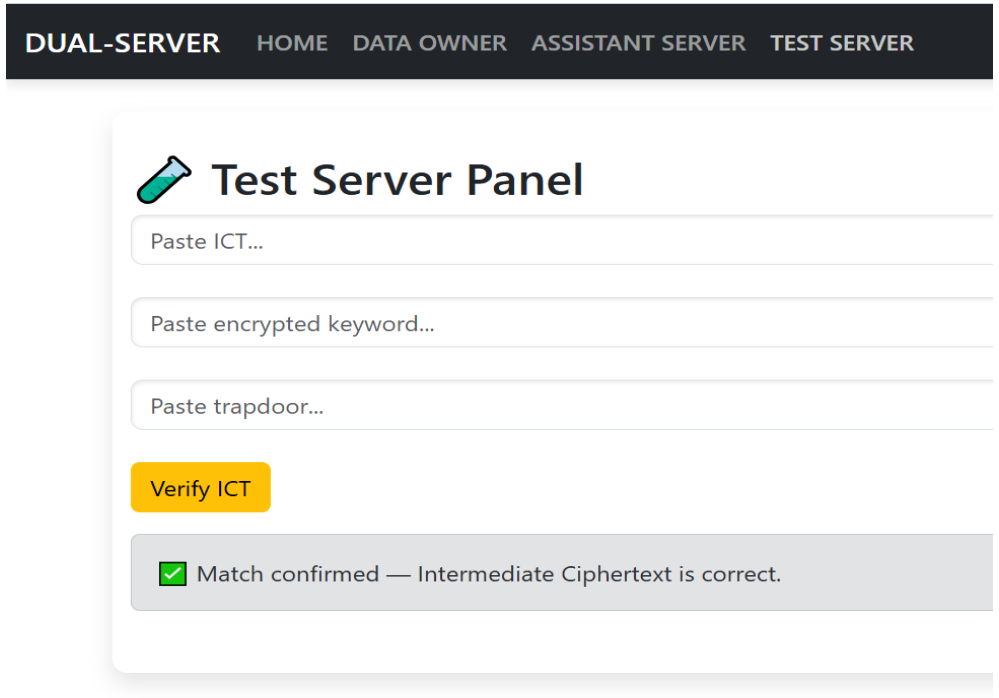


Figure 3: Match scenario shown on UI

Future Scope – AI-Driven Keyword Search

- Implement an AI module to enhance the keyword search experience.
- Use Natural Language Processing (NLP) techniques to suggest the closest matching keyword.
- Improves accuracy, flexibility, and usability without compromising security. Planned tools: spaCy, TF-IDF,or BERT for semantic similarity and keyword correction.

Conclusion

Successfully implemented the part of the project, focusing on:

Secure data encryption Trapdoor-based keyword handling Dual server verification logic Ensures strong protection against insider keyword guessing attacks (IKGA).Demonstrated backend functionality using manual flow for clarity. In Phase 2, we will automate user search and integrate AI-based smart keyword recognition to enhance usability.

ESTD : 2001
An Institute with a Difference

“Thank you for your attention.”

ESTD : 2001
An Institute with a Difference

"We are now open to questions."