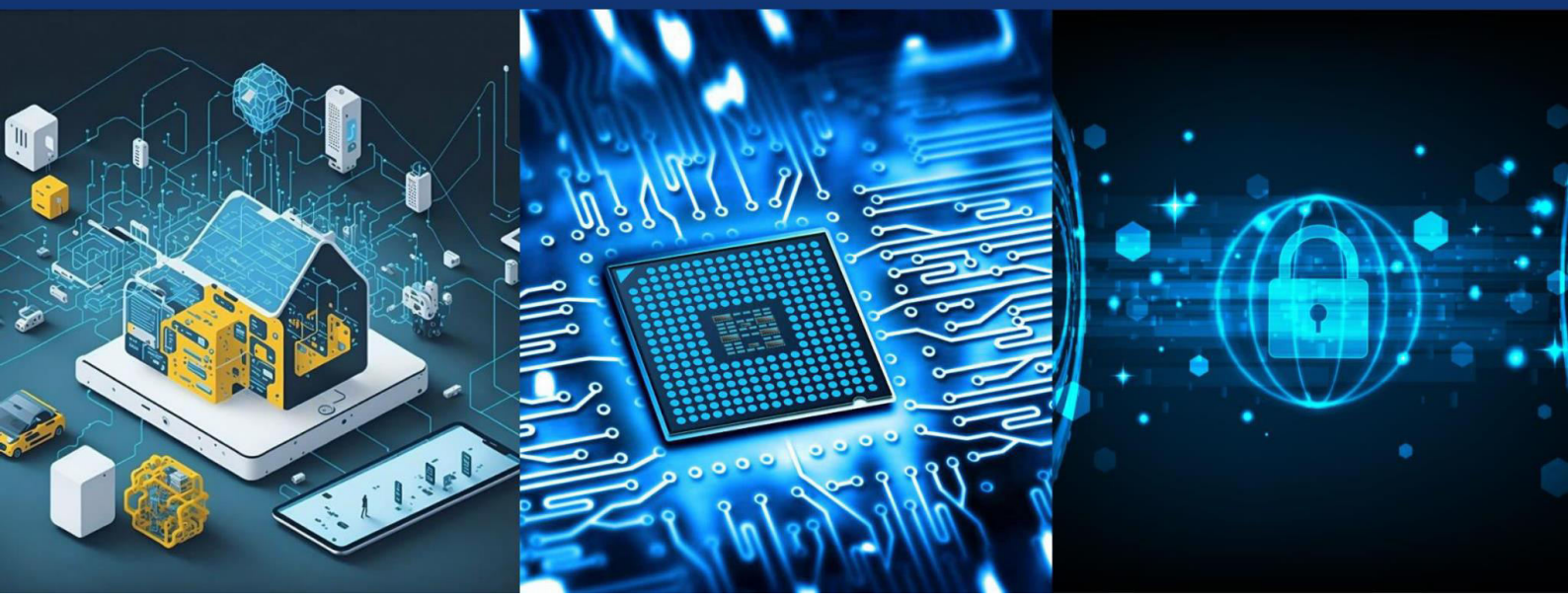
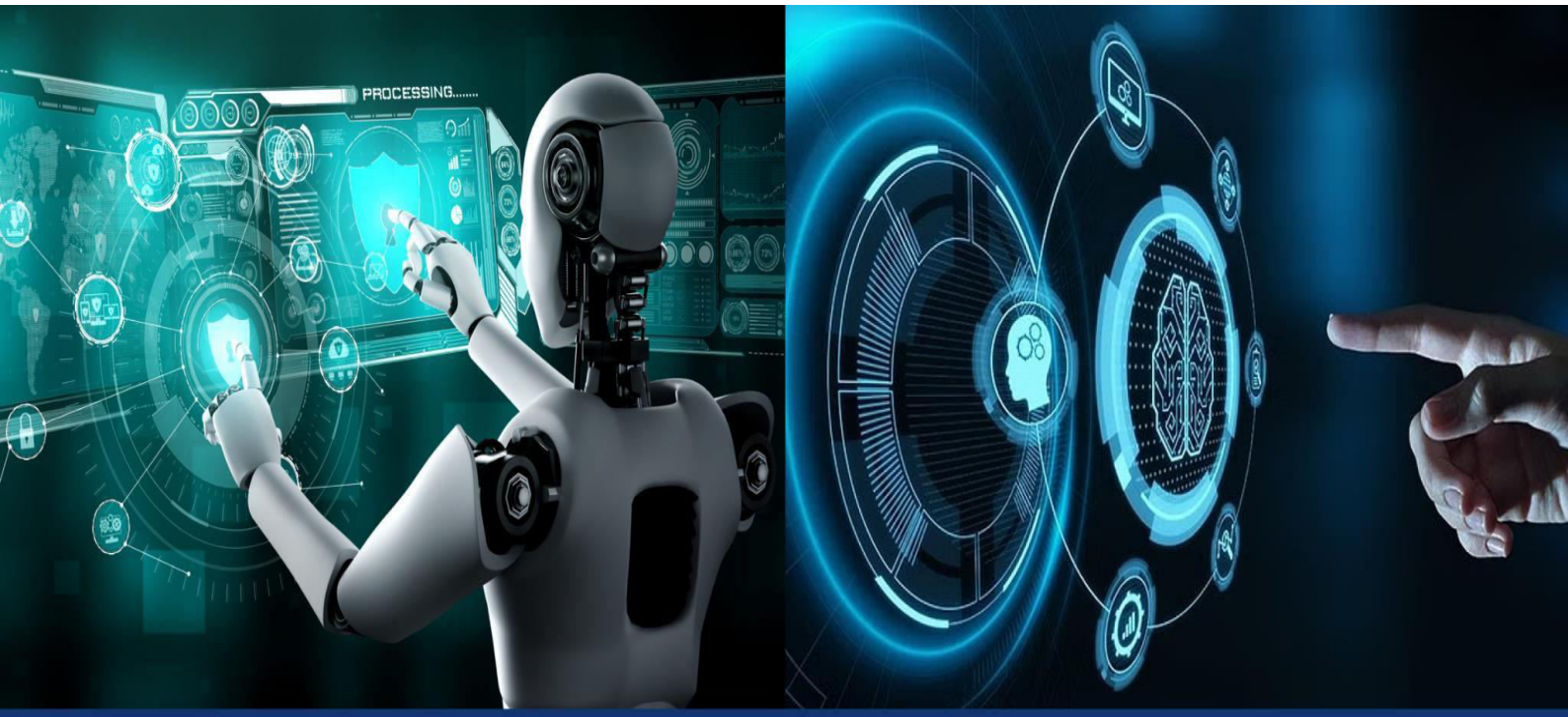
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771 **Volume 13, Issue 6, June 2025**

© 2025 IJIRCCE | Volume 13, Issue 6, June 2025| DOI:10.15680/IJIRCCE.2025.1306209

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TriCipher: A Master-Key-Protected Hybrid Encryption Framework Built from First-Principle Number Theory

Dr Gururaj J.P

Associate Professor, Government First Grade College, Davangere, India

ABSTRACT: In the era of exploding digital communications, file privacy is one of the biggest worries in today's computing environment. Numerous commercial encryption tools typically depend on pre-formed cryptographic libraries that conceal the hidden algebra, are usually quite opaque and commonly utilize a solitary calculation as insurance. In this paper, we describe the design and implementation of a hybrid, number-theory-based file encryption/decryption system that builds all of its cryptographic primitives from first principles — Miller–Rabin primality testing, the Extended Euclidean Algorithm, RSA with Chinese Remainder Theorem (CRT) improvements, ElGamal password exchange, and Diffie–Hellman key exchange — without reliance on third-party cryptographic libraries. It breaks input files into segments and encrypts each segment with a constant rotation of RSA, ElGamal, and Diffie-Hellman-dependent stream cyphers, as well as all per-chunk secret parameters are packed up and sealed with a master RSA key pair. An admin and end-user-oriented role-based interface that exposes the scheme over a Flask-based web application (for registration, authentication, upload, key management, and FAQ administration) backed by MySQL as the persistence layer. The functional tests confirm correct round-trip recovery of original files across workflows including registration, authentication, encryption and decryption. Our results show that, by unifying a collection of number-theoretic primitives, in a rotating hybrid scheme such that no single-number theoretic primitive may be targeted for cryptanalysis, one can achieve both transparency and pedagogical clarity while providing an effective real-world bridge between classical number theory (the uniform generator) and applied cybersecurity engineering (the upper layered architecture).

KEYWORDS: Hybrid cryptography; RSA; ElGamal; Diffie–Hellman; Chinese Remainder Theorem; Miller–Rabin primality test; file encryption; Flask; number theory.

I. INTRODUCTION

How to securely store and transmit digital files has come to epitomize one of the greatest challenges in computing today. The continuous growth in the volume of sensitive information exchanged across untrusted networks between organization, individuals, or devices exposes those interactions to interception and unauthorized access that lead to data breaches [1]. Practical implementations of encryption, driven by number theory and mathematics [1]–[3], are still the primary defence against these threats: public-key algorithms such as RSA [4], ElGamal [5] and Diffie–Hellman [6] provide fundamental security for confidentiality, key exchange and data integrity (authentication) in virtually any modern security system.

The simple way that commercial and open-source security tools achieve this protection is to merely call into well-developed, existing crypto libraries. Although these libraries are well-audited, it operates like a black box: the modular arithmetic, primality testing and discrete-logarithm operations that lend security to the algorithms are unknown to the developer, limiting transparency as well as educational value. Also, many file-protection tools avoid heavy multi-practice cryptographic mechanism by relying on one simple technique putting risk on the flaws of a single algorithm instead combining several complementary schemes which mitigates that risk.

We close these gaps with a hybrid number-theory-based file encryption and decryption system that implements RSA, ElGamal, and Diffie–Hellman entirely in terms of first principles, including Miller–Rabin primality testing [4], [5], extended Euclidean for modular inverse computations, and CRT-RSA decryption — all combined using a rotating chunk-level hybrid architecture. The secret parameters emitted during the encryption of each chunk are all packaged together and secured behind a single master RSA key which allows this entire file to be recovered only with possession of the

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

original private key. The system is app as a web application in FLASK, having role-based access for admin and user backed to MySQL database.

The rest of this paper is structured as: The related work is summarized in Section II as it relates to existing hybrid cryptosystems and where the current work falls with respect to these systems. The proposed system and its mathematical underpinnings are described in Section III. Section IV discusses system design, from architecture to modules and database schema. Section V discusses implementation details. Section VI describes functional testing and results. The last section of the paper, Section VII, is a conclusion and future work.

II. RELATED WORK

The field of public-key cryptography began when Diffie and Hellman introduced a method for two parties to agree on a shared secret over an insecure channel without having shared any secret beforehand, relying on the conjectured intractability of the discrete logarithm problem [1]. RSA is the first practical public-key cryptosystem, proposed by Rivest, Shamir and Adleman [2], its security relies on the factorization of a product of two large primes. ElGamal then took the Diffie–Hellman concept and worked it out into a complete public-key encryption and signature system that was directly based on the discrete logarithm problem [3]. Combined, these three schemes provide the number-theoretic backbone for most public-key (including elliptic-curve versions of it) systems that came after.

A few studies have also explored the idea of using such hybrids by composing these primitives in a way that encapsulates their individual strengths. Flores-Carapia et al. proposed a dynamic hybrid cryptosystem, which associates chaotic maps and the Diffie–Hellman protocol applied to image encryption [6], demonstrating that combining a classical number-theoretic exchange mechanism with auxiliary transformation enhances resistance to statistical attacks. This performance analysis on various combinations of hybrid cryptographic techniques for secure communication shows that the combination of asymmetric key-exchange algorithms and symmetric cyphers leads to a good trade-off between security and overhead [7]. Rivera et al. proposed a hybrid cryptosystem that applies RSA, DSA, ElGamal, and AES encryption schemes in order to defend the data being transmitted, and they showed that the rotation or stacking of different algorithms increases the complexity of successful cryptanalysis compared with a single scheme based on what [8]. More recently, Xie provided a comparative study between RSA, ElGamal and elliptic-curve cryptography (ECC), which showed each algorithm trades off relatively smaller key sizes for computational costs and the underlying hard problem relates [10].

The main take-away from those works is two well-established facts that motivate the present system. Hybrid designs that mix independent number-theoretic hard problems (or, more generally, an algorithm based on single things like RSA to integer factorization and ElGamal and Diffie–Hellman to discrete logarithms) are, in practice, considerably stronger against attacks than any design based on a single assumption. Secondly, most of the hybrid implementations that currently exist rely on externally-written cryptographic libraries for their primitive building blocks, with even a basic number-theoretic component (such as prime generation or modular inversion / modular exponentiation) left untested and opaque. In contrast, our proposed system implements each primitive from first principles and rotates among three separate schemes at the per-chunk granular level but consolidates all resulting secrets under a single master RSA key such that they can be decrypted only in controlled, single-point fashion.

III. PROPOSED SYSTEM

3.1 Mathematical Foundations

The system embeds five fundamental number-theoretic building blocks without using third-party cryptographic libraries during implementation.

- **Miller–Rabin Primality Test:** Our system to certify primality is based on the probabilistic Miller–Rabin test, first presented as a conditional deterministic test by Miller [4], and then extended into a highly efficient randomized algorithm of Rabin [5] offering a bounded probability of error over large independent rounds.
- **Extended Euclidean Algorithm:** The extended Euclid algorithm computes the GCD of two integers and (simultaneously) finds the Bézout coefficients that express it as a linear combination: modular multiplicative inverses are computed with this method for use in RSA private exponent computation and ElGamal signature components.
- **RSA + CRT:** In this method, the two primes p and q Miller–Rabin-certified large random numbers are used to generate RSA key pairs. It is optimized by using Chinese Remainder Theorem (CRT), which allows the expensive

modular exponentiation modulo n = pq to be replaced by two smaller ones modulo p and q, one then recombining them later it helps with decryption latency for large or repeated operations [2].

- **ElGamal Encryption:** For chunks mapped to the ElGamal scheme, encryption is done in a cyclic group using an ephemeral (assuming) random exponent whose security depends on the discrete logarithm problem formalized by ElGamal [3].
- **Diffie–Hellman Key Exchange:** For the chunks assigned to the Diffie–Hellman-based scheme we derive a shared secret from the classical [1] Diffie–Hellman exchange and then hash it with SHA-256 into a keystream that is XORed with the chunk data, thereby acting as a light-weight stream cipher based on number-theoretic primitives (as opposed to standard symmetric ciphers).

3.2 Hybrid, Chunk-Rotating Architecture

Instead of encrypting the entire file at once using one algorithm, it breaks each uploaded file into fixed size blocks, and then encrypts successive blocks according to a rotating sequence using RSA, ElGamal or the Diffie–Hellman-derived stream cipher. This rotation agrees with the design pattern demonstrated in hybrid structures [6]–[8]: because RSA and ElGamal/Diffie–Hellman are built on different hard problems — integer factorization for RSA vs. the discrete logarithm problem for ElGamal and Diffie–Hellman — an attacker at least partially breaking one scheme's assumptions does not gain a straightforward attack on pieces encrypted with the other schemes jointly used. The first principal of Secret is that all algorithm-specific secret parameters generated in the process (e.g., per-chunk keys, ephemeral exponents and derived seeds) are both serialized and then encrypted using a single master RSA key pair, so a user only requires one private key artifact — the downloaded key file — to completely reverse an individual chunk-rotation stage back to the original file.

3.3 Web Application and Role-based Access

A Flask web application exposes this cryptographic core to end users. Flask was chosen due to its minimal overhead and flexibility in integrating our functionality — such as this example, where we integrate the encryption/ chunking/ key-restoration logic into custom modules instead of relying on a full-stack framework. The application defines two roles. Separately, administrators authenticate and are able to view / remove registered user accounts as well as manage the FAQ knowledge base. Users register with basic account info, Authenticate, upload a file to be encrypted, download the master key generated by system and later upload an encrypted file along with the corresponding key so as to retrieve it in original format. Session state is utilized to restrict access to the upload, download and encryption/decryption routes therefore only authenticated users can invoke cryptography operations. MySQL is the relational backing store for admin credentials, user accounts and FAQ entries.

IV. SYSTEM DESIGN

The structure for the system consists of an Admin module and a User module. Admin module: it authenticates the administrator and there are account-management and FAQ-management operations. It contains the User module which takes care of registration, logging in, file encryption, file decryption and viewing FAQs. The data flow in the system initiates at the web interface followed by passing through the Flask application server where validation and session handling occurs, then either coloaded to cryptographic engine (for all encrypt/decrypt requests) or MySQL database for authentication and content. The relational schema consists of three main tables: one for administrator credentials, one for user sign up/credentials (name, phone number, username and password) and an FAQ table with question–answer pair. This separation of concerns — which deals with presentation, application logic, cryptographic engine and persistence layers — isolates the number-theoretic core to ensure that it can be independently tested from those components that face the web.

© 2025 IJIRCCE | Volume 13, Issue 6, June 2025| DOI:10.15680/IJIRCCE.2025.1306209

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

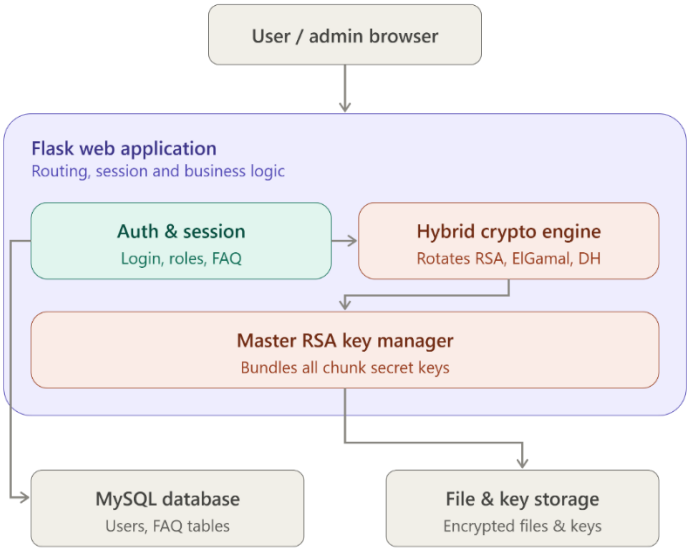


Figure 4: Architecture of the Hybrid Number-Theory-Based File Encryption System

The block diagram the fig 4 shows that the architecture of the system for file encryption based on hybrid number theory is shown in. At the top, the user or admin browser represents entry point here, as both users types interact with same web interface, but enabled for lower and one admin access. This browser passes requests down to the Flask web app, which is the most dominant part of our system and is illustrated as a large purple container containing all of the backend logic.

In the Flask app, the module for authorization & session will have the functionality of logging in and how admins and users keep their session both running on this. When a user is login in and uploads the file then control comes to the hybrid crypto engine which is core of encryption. Instead of employing a single algorithm on the file as a whole, it divides the file into chunks and rotates through RSA, ElGamal or Diffie–Hellman when encrypting each chunk (so different regions of the file may be protected with different cryptographic schemes).

Once every single chunk is encrypted and has produced its own unique Key material, all of these keys are given to the master RSA key manager which packs them together and wraps them with one global master RSA key pair. This allows the user to later decrypt the whole file just with one single key — you only download a single key, while the original encryption used several algorithms.

Finishing off the rest of the diagram are two external systems on top of the Flask container. The auth and session module connects to a MySQL database which stores user credentials and FAQ entries in this instance because it relates only account and content related data, not any crypto. The primary RSA secret manager is accessed separately to storage, where the encrypted chunks of files and the exportable. pem key country file ultimately placed for the destination user to download it. This demarcation of the account-management path and security-critical encryption path matches closely how their actual codebase is structured; authentication and database logic remain loosely coupled to edge logic dealing with image encryption, chunking, key-bundling etc.

© 2025 IJIRCCE | Volume 13, Issue 6, June 2025| DOI:10.15680/IJIRCCE.2025.1306209

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. RESULT

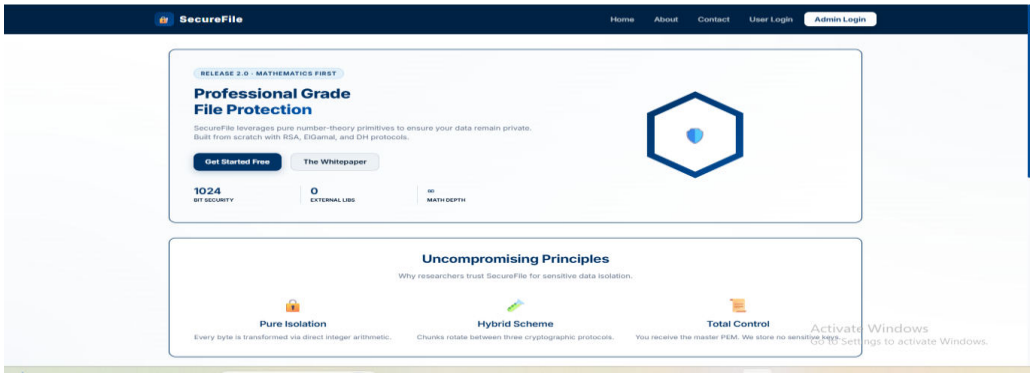


Figure 5.1: Home page

The home page (fig 5.1) shows the first place users will go to enter the system; it gives them an overview of how the file encryption platform works and highlights some of its important functions. The users are easy to navigate login, registration & other important areas like the FAQ for smooth access to services.

Figure 5.2: Registration page

User Registration (Fig. 5.2): Whenever a new user needs to create an account, the system gives this page where the user will enter valid personal details including name, phone number as well as username and password. This provides a secured entry mechanism that stores user credentials in the database for searching and verification at login to access features of the system.

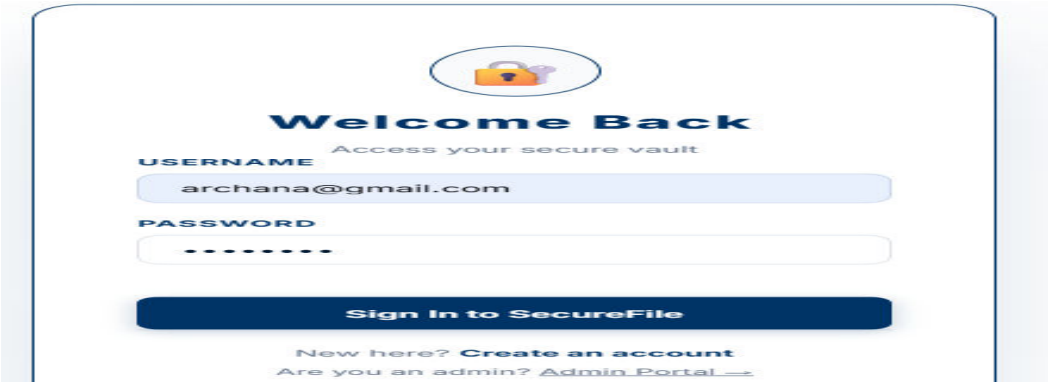


Figure 5.3: User Login page

© 2025 IJIRCCE | Volume 13, Issue 6, June 2025| DOI:10.15680/IJIRCCE.2025.1306209

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The User Login page (5.3) allows registered users to login the system using their valid username and password. It authenticates the user credentials and allows access to various features such as file encryption, decryption, and FAQ.

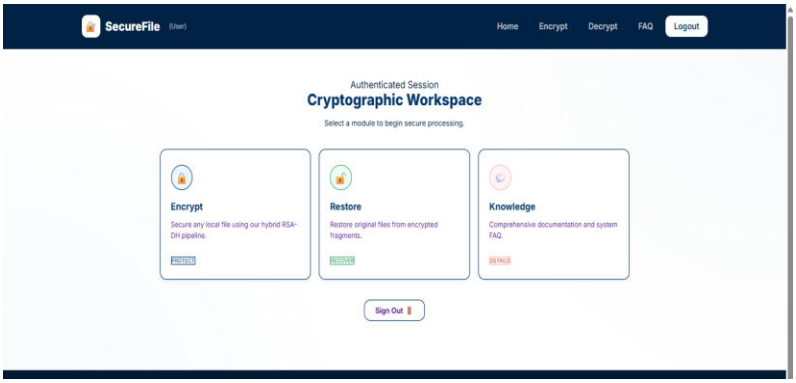


Figure 5.4: User Home page

After successful Login, the User Home Page (fig 5.4) is shown, which essentially serves as a Dashboard for all system features to be consumed by the user. It has these three buttons for encrypting files, decrypting files and FAQs.

Figure 5.5: Encrypt page

The Encrypt Page (fig 5.5) offers users two options to upload files and encrypt them securely by using a hybrid cryptographic system. It analyses the file with several encryption algorithms and creates an exceptionally elaborate secure key necessary for decrypting at a later time.

Figure 5.6 Decrypt Page

© 2025 IJIRCCE | Volume 13, Issue 6, June 2025| DOI:10.15680/IJIRCCE.2025.1306209

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This page (fig 5.6) will allow you to upload an encrypted file, along with the key that is needed to decrypt it and restore the data. It checks the key, applies the right decryption algorithms and recovers the file, making it safe.

Figure 5.7: Admin Login page

Admin Page: The Admin Login page (fig 5.7) provides a mechanism for administrators to log into the system by providing valid credentials. So, on successful authentication, the admin is redirected to access the dashboard to handle users and FAQ. Info.

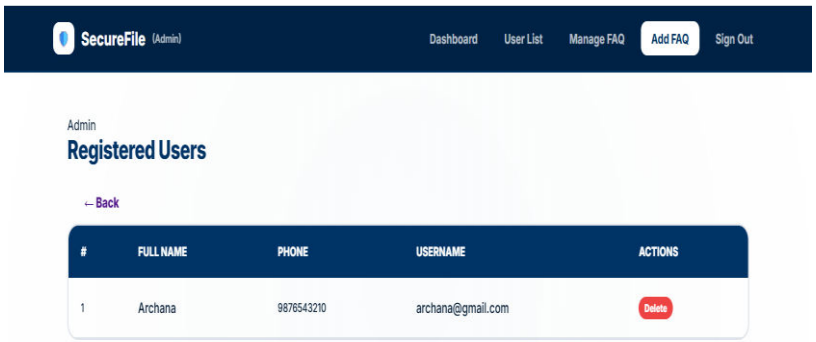


Figure 5.8: Admin users List

This page (Fig. 5.8) shows all available users in the system with their basic details. It enables an administrator go through user accounts and remove them when needed to keep the system secure and manageable.

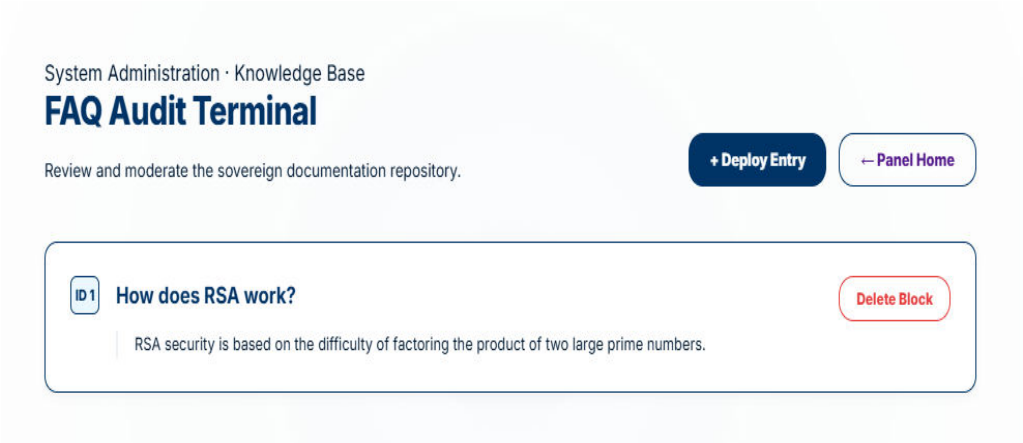


Figure 5.9: Admin FAQ List page

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Admin FAQ List: This page shows (fig 5.9) all the FAQs with their answers present in the system. It allows the administrator to add more new FAQs or delete old ones.

VI. CONCLUSION

This paper introduced a new hybrid, number-theory-based system for the encryption and decryption of files, thereby constructing RSA, ElGamal and Diffie–Hellman in their entirety from first principles — including supporting primitives such as Miller–Rabin primality testing, the Extended Euclidean algorithm (GCD), and CRT-optimized RSA decryption — rather than using libraries. It uses three number-theoretically independent schemes, rotates them at the file-chunk level and collects all the ten million+ resulting secrets under a single master RSA key to increase cryptanalysis cost/effort over a single-algorithm system component basis while also being transparent enough for educational use. The system, delivered as a role-based Flask web application backed by MySQL supports registration, user authentication/authorization, encryption/decryption, key generation/key distribution and functional testing, demonstrating the correct lossless recovery of encrypted files. The work shows that classical number theory is still a useful and pedagogically sound basis for real-world cybersecurity engineering, although it should be used directly rather than as a black box.

6.1 Future Work

Further extensions will add features to support even more or post-quantum-resilient encryption, switching storage to cloud environments (to enable scalable secure file hosting), multi-factor authentication (e.g., one-time passwords or biometrically powered access), an activity log to identify suspicious activities, encrypted file sharing between registered users, a companion mobile client, and automated background data backup and recovery of encrypted files.

REFERENCES

[1] New Directions in Cryptography[1. IT-22, no. 6, pp. 644–654, Nov. 1976.
[2] DOI: 10.1145/1394391.1394400] [2] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and the public-key cryptosystems." Communications of the ACM, vol Feb.1978, 21(2) :120–126.
[3] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, vol. J. IT–31, no. 4, pp. 469–472, Jul. 1985
[4] G. L. Miller, Riemann's Hypothesis and tests for Primality, Journal of Computer and System Sciences 15:287-301, 1977 13, no. 3, pp. 300–317, 1976.
[5] M. O. Rabin, "Probabilistic Algorithm for Primality Testing," pp. 12, no. 1, pp. 128–138, 1980.
[6] R. Flores-Carapia, V. M. Silva-García, and M. A. Cardona-López, "A dynamic hybrid cryptosystem using chaos and Diffie–Hellman protocol: An image encryption application," Appl 13, no. 12, art. 7168, 2023, doi: 10.3390/app13127168.
[7] Z Zhang, Q., & Hu, M. (2020). Research of materials storing system based on RFID technology and mobile terminal. International Journal of Manufacturing Science and Technology 6(4):33–40 doi:10.5815/ijmsc 2020. 04. 04.
[8]. Rucio, "Hybrid cryptosystem Based on RSA, DSA, ElGamal and AES" International Journal of Scientific &Technology Research vol. 8, no. 10, 2019.
[9] Abusukhon, A., Anwar, M. N., Mohammad, Z. & Alghannam B. (2019) A hybrid architecture based on Diffie Hellman and Text to Image Encryption Algorithm 65 – 81, Journal of Discrete Mathematical Sciences and Cryptography, 22(1). https://doi.org/10.1080/09720529.2019.1569821
[10] Y. Xie, A Comparative Study of RSA vs. ElGamal vs. ECCENCRYPTION ALGORITHM (2003-2014) Theoretical and Natural Science v., vol. pp. 40–46. 2023, doi:10.54254/2753-8818/14/20240886.

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details