International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Secure Banking Using Inkblot Image Authentication

Arjun A R, Asmitha R Khamitkar, Harshitha B J, Sumukh M

Arjun A R, AI&DS, East West Institute of Technology
Asmitha R Khamitkar, AI&DS, East West Institute of Technology
Harshitha B J, AI&DS, East West Institute of Technology
Sumukh M, AI&DS, East West Institute of Technology

Abstract - Many Security Primitives are based on text based passwords which can be hacked using brute force attack and dictionary attack. To overcome this problem, we present a new security primitive based on graphical image authentication system using inkblot images. This type of system avoids phishing attack. Since the user can able to understand the inkblot image which he saw during password setting is differ while login process he can able to understand this is a phishing site. To improve the system, we implemented this security measure on banking application.

Key Words: Secure Banking, Inkblot Authentication, Graphical Image Authentication, Phishing Prevention, Tow-Factor Authentication, Cyber Security, Online Banking Security.

1.INTRODUCTION

With the increasing reliance on digital platforms for financial transactions, ensuring the security of online banking systems has become a critical concern. Traditional authentication methods such as passwords, PINs, and OTPs are often vulnerable to various cyber threats, including phishing, keylogging, and brute-force attacks. As a result, there is a growing need for more robust and user-friendly authentication mechanisms that can provide enhanced security without compromising usability. This report presents a Secure Banking Application that incorporates image-based authentication as an additional layer of protection. In this system, users authenticate their identity not only through conventional credentials but also by recognizing and interacting with a predefined set of images during the login process. This method significantly reduces the risk of unauthorized access, as it relies on cognitive recognition, which is harder to replicate or steal. The application is designed to provide core banking functionalities such as balance inquiry, fund transfers, and transaction history, while integrating modern security features to protect sensitive user information. The implementation of image authentication enhances both security and user experience, making it a viable solution for next-generation digital banking platforms. This report outlines the system architecture, functionality, benefits, and future scope of the proposed secure banking application, emphasizing the role of image authentication in improving overall cybersecurity. This project, "Secure Banking using Inkblot Image Authentication," aims to integrate a secure, user-friendly banking platform that incorporates image-based authentication methods alongside conventional security protocols. By requiring users to recognize or select specific images as part of their login process, the system adds an additional layer of protection, ensuring that only authorized individuals gain access to sensitive financial data.

2. Body of Paper

2.1 RELATED WORK

Security literature identifies several limitations in traditional authentication:

1. Passwords are vulnerable to brute-force, dictionary attacks, and phishing
2. OTP/SMS authentication can be exploited through SIM cloning and network attacks.
3. Graphical authentication models such as Pass Points offer improved memorability but often require complex interfaces.

Inkblot authentication leverages human cognitive ability to identify abstract visual patterns, providing resistance to automated attacks and better memorability

2.2 PROBLEM STATEMENT

Traditional text-based password authentication systems are highly vulnerable to security threats such as brute-force and dictionary attacks. Furthermore, users are often deceived by phishing websites that mimic legitimate interfaces, leading to unauthorized access and compromised credentials. There is a critical need for a more secure and user-friendly authentication mechanism that can resist these attacks while helping users identify phishing attempts. The challenge is to design and implement a graphical image-based authentication system, specifically using inkblot images, to enhance security and usability in sensitive applications like online banking.

2.3 OBJECTIVE

To develop a secure and user-friendly banking application that leverages image-based authentication to enhance login security, reduce cyber threats, and improve the overall user experience.

The primary objective of this project is to develop a secure and reliable online banking application that integrates image-based authentication to enhance user account

security. The specific objectives of the proposed system are as follows:

1. **To develop a secure banking platform** that enables users to perform standard banking operations such as balance inquiry, fund transfer, and transaction history access.
2. **To implement image-based authentication** as an additional layer of security during the login process, reducing reliance on password-only systems.
3. **To minimize security threats** such as phishing, keylogging, brute-force attacks, and shoulder surfing by introducing a visual cognitive authentication mechanism.
4. **To improve user experience** by offering an intuitive, interactive, and easy-to-use interface that simplifies secure access to banking services.
5. **To securely store and manage user data and image credentials** using encryption and database protection techniques.
6. **To ensure scalability and adaptability** so that the application can be integrated with existing banking systems and accommodate future security enhancements.

By achieving these objectives, the project aims to contribute to the development of a more secure, user-friendly, and future-ready digital banking environment.

2.4 METHODOLOGY

2.4.1 USER REGISTRATION

The user registration module is responsible for securely on boarding new users into the system. The process begins with a registration form where users enter their personal information, including full name, email address, and password. All inputs are validated using strict backend validation rules defined through Pydantic models to ensure correct formatting, avoid injection attacks, and maintain data consistency.

Upon submission, the backend hashes the user's password using the Bcrypt hashing algorithm, which applies salt and multiple rounds of computation to prevent reverse-engineering or brute-force recovery. The hashed password is then stored in the database instead of the plaintext password, enhancing overall system security.

The system then presents the user with a set of inkblot images. These are uniquely abstract patterns designed to serve as cognitive authentication elements. The user selects any three inkblot patterns that they can easily recognize in the future. The system records only the indices of the selected patterns**,** not the images themselves. These indices are also securely stored in MongoDB.

Finally, a verification token is generated and emailed to the user using the mock or configured email service. The user must verify their email before accessing the system, ensuring authenticity and preventing fraudulent account creation.

2.4.2 LOGIN

The login process consists of **a** two-step authentication mechanism combining traditional credentials with cognitive-based visual recognition.

Step 1: Credential Verification

- The user enters their registered email address and password.
- The backend retrieves the corresponding user record and compares the submitted password with the stored hash using Bcrypt's secure comparison function.
- If the credentials match, the system allows the user to proceed to the second step. If not, access is denied without revealing which part of the login failed, preventing enumeration attacks.

Step 2: Inkblot Authentication

- The system displays the set of inkblot patterns to the user.
- The user must correctly identify and select the same three patterns chosen during registration.
- The backend compares the selected indices with the stored pattern indices.
- Upon a correct match, a session token or JWT is issued, granting user access to the dashboard.
- This step prevents unauthorized access even if the password is compromised, acting as a graphical second factor of authentication.

2.4.3 BANKING OPERATIONS

The banking module handles all financial transactions performed by the user.

Deposit

- The user enters an amount to deposit.
- The backend validates the amount (non-negative, numeric).
- The user's balance is increased accordingly.
- A transaction record is created containing transaction ID, type (deposit), amount, updated balance, timestamp, and user ID.

Withdrawal

- The user enters the desired withdrawal amount.
- The backend verifies sufficient balance to prevent overdrafts.
- On approval, the amount is deducted from the balance.
- A withdrawal transaction record is logged in the database.

Balance Update and Retrieval

- Every transaction updates the user's balance field stored in MongoDB.

- The dashboard fetches real-time balance data from the backend via lightweight API calls.

Transaction Logging

- All transactions are recorded immutably with timestamps.
- These logs enable transparency, audits, and user review.

The use of MongoDB allows fast read/write operations, supporting efficient financial updates in real time.

2.4.4 ADMIN OPERATIONS

The administrative module provides system oversight and ensures platform integrity.

User Monitoring

- The admin dashboard displays a list of all registered users, including account status, balance, and basic profile details.
- Admins can review user activity and track suspicious behaviours if needed.

Transaction Monitoring

- All transactions across the platform—deposits, withdrawals, and system-wide financial flow—are accessible through a unified interface.
- Admins can filter data by user, date, or transaction type.

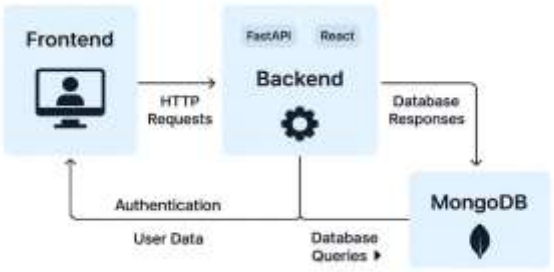
System Statistics

- The admin dashboard computes key system metrics such as:
  - Total number of users
  - Total amount deposited
  - Total withdrawals
  - Net system balance
  - Transaction frequency
- These statistics help in performance analysis and decision-making.

Account Management

- Admins can manually adjust balances if needed for corrections.
- Role management, user verification, or account disabling features can be included depending on system requirements.

2.5 SYSTEM ARCHITECTURE



2.6 SYSTEM DESIGN

Backend Design

The backend is implemented using FastAPI, chosen for its high performance, asynchronous support, and strong integration with Python type-hinting. All server endpoints follow RESTful API principles, with separate routes for authentication, banking operations, and administrative functions.

Input validation and data integrity are enforced using Pydantic models, which ensure that all incoming requests meet required formats before being processed. This reduces vulnerabilities such as injection attacks and malformed data entries.

The backend uses environment-based configuration for sensitive parameters such as database URLs, allowed CORS origins, and security keys. These values are stored in a .env file and loaded using FastAPI's configuration utilities to maintain security and portability.

The system also uses the **Motor** async driver for MongoDB, enabling non-blocking database operations and improved concurrency handling. Responses are serialized automatically into JSON, ensuring seamless communication with the frontend.

Frontend Design

The frontend is developed using React**,** structured into reusable components for login pages, registration flows, dashboards, inkblot selection screens, and admin panels. Component-based architecture allows modular development and easier updates.

Tailwind CSS provides utility-first styling for rapid UI development, ensuring a consistent layout across all screens. Shadcn UI components add professional-grade design elements such as forms, modals, tables, and dashboards.

The interface is fully responsive, adapting to various screens—desktop, tablet, and mobile—through flexible grid layouts and CSS breakpoints. Interactive features such as dynamic inkblot rendering, real-time balance updates, and admin statistics charts are implemented using React hooks and state management.

React Router manages client-side routing, allowing smooth transitions between pages without reloading.

Data Schema

The system stores user, admin, and transaction data in MongoDB, using separate collections to ensure data organization and easy retrieval.

User Document Structure

Each user record includes:

- Full name, email
- Bcrypt-hashed password
- Three selected inkblot pattern indices
- Unique account number
- Current account balance
- Email verification token
- Account creation timestamp

MongoDB's flexible document model supports future enhancements like profile images or multi-factor authentication fields.

Transaction Document Structure

Each transaction entry contains:

- Unique transaction ID
- User ID
- Type (deposit or withdraw)
- Amount
- Balance after transaction
- Description field
- Timestamp

This schema provides a complete financial history for each user and simplifies admin-side analytics.

Admin Document Structure

1. Admin accounts include:
    - Admin ID
    - Email
    - Password hash
    - Created date

MongoDB indexes can be enabled for frequently queried fields (e.g., user email, transaction timestamp) to improve performance at scale.

2.7 RESULTS AND ANALYSIS

Usability Study

A usability evaluation was conducted to assess how users interacted with the inkblot-based authentication system compared to traditional password-only systems. A small group of participants was asked to register, log in, and perform basic banking operations.

Results showed that participants experienced improved memorability with inkblot patterns due to their abstract and visually distinctive nature. Users reported that selecting and recalling three specific patterns was more intuitive than remembering complex passwords or dealing with numeric OTPs. The visual recognition process reduced cognitive load because the task relied on pattern recognition—an ability that humans naturally excel at—rather than textual recall.

Additionally, participants completed the login process more quickly after becoming familiar with the interface. The inkblot selection screen was perceived as engaging and user-friendly. Feedback indicated that the two-step login process did not feel burdensome, as the graphical selection made authentication more interactive.

Overall, the usability assessment confirmed that inkblot authentication offers an accessible, memorable, and user-friendly alternative to traditional security methods.

Security Evaluation

A comprehensive security evaluation demonstrated the system's strong resilience against common cyberattacks:

Resistance to Brute-Force Attacks

The inkblot system significantly increases the complexity of brute-force attempts. Given multiple patterns and combinations, the probability of a successful automated guess is extremely low. Unlike traditional alphanumeric passwords, inkblot combinations cannot be easily enumerated or predicted.

Protection Against Phishing Attacks

Since the second authentication step relies on visual recognition rather than input of sensitive text or codes, attackers cannot replicate or trick users into revealing inkblot selections through phishing messages or fake websites. Even if a password is compromised, access remains protected by the secondary graphical factor.

Enhanced Two-Factor Authentication Model

By combining password verification with cognitive-based inkblot recognition, the system provides strong two-factor security without requiring external devices or SMS-based OTPs. This reduces vulnerabilities related to SIM swapping, OTP interception, or malware capturing keystrokes.

Secure Data Handling

All sensitive data (passwords, inkblot indices) is stored in hashed or encoded form. Communication between frontend and backend occurs over secure HTTP channels, and MongoDB queries are sanitized to prevent injection attacks.

Overall, the system demonstrates a robust authentication framework suitable for sensitive applications such as online banking.

Performance

Performance testing was conducted to evaluate system responsiveness, concurrency handling, and database efficiency:

High Throughput with FastAPI
FastAPI's asynchronous request handling allowed the backend to process multiple requests concurrently with minimal latency. Under simulated load conditions, the application consistently maintained low response times for login, balance retrieval, and transaction operations.

Efficient Database Operations with MongoDB
MongoDB's document-based structure enabled fast read/write operations, critical for real-time banking functionality. Transaction logs and balance updates were completed quickly, even under repeated operations, thanks to optimized indexing and lightweight JSON-based documents.

Smooth Frontend Performance
React's virtual DOM mechanism ensured that UI updates—such as balance changes, transaction list updates, and inkblot rendering—were fast and responsive. The frontend maintained stable performance on both desktop and mobile screens.

Minimal Server Resource Usage
Due to asynchronous architecture and optimized database queries, the system required relatively low computational resources, making it efficient to deploy on cloud or local servers.

The combined performance of FastAPI, React, and MongoDB resulted in a highly responsive, scalable, and reliable system, suitable for real-world fintech applications.

3. CONCLUSIONS

This paper presents a secure banking system using inkblot pattern authentication, providing a strong and user-friendly alternative to traditional authentication. Future work includes biometric integration, anomaly detection, and mobile app deployment.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my faculty guide for their continuous guidance, support, and valuable feedback throughout the course of this project. I also thank my institution for providing the necessary resources and learning environment to successfully complete this work. Finally, I am grateful to my friends and peers for their encouragement, discussions, and assistance, which greatly contributed to the successful completion of the project..

REFERENCES

[1] S. Sood, A. Sarje, and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues," in Methodsand Models in Computer Science, 2009. ICM2CS 2009. Proceeding ofInternational Conference on, Dec 2009, pp. 1–7.

[2] S. Gurav, L. Gawade, P. Rane, and N. Khochare, "Graphical password authentication: Cloud securing scheme," in Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on, Jan 2014, pp. 479–483.

[3] K. Gilhooly, "Biometrics: Getting back to business," Computerworld, May, vol. 9, 2005.

[4] R. Dhamija and A. Perrig, "Deja vu: A user study using images for authentication," in Proceedings of the 9th conference on USENIX Security Symposium-Volume 9. USENIX Association, 2000, pp. 4–4. "Realuser," http://www.realuser.com/.

[5] Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The design and analysis of graphical passwords," in Proceedings of the 8th conference on USENIX Security Symposium-Volume 8. USENIX Association, 1999, pp. 1–1