SECURE ID VAULT

Mrs. A. Navya (Guide), Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

Boddupalli Vijjaya Balajii Veerraj, Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

Reddy Seshu, Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

Pulamarasetty Dilip Atchuth Kumar, Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

Shaik Khaja Asif, Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

Pattika Jeevan Pradeep, Computer science & Engineering Department, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

ABSTRACT

Traditional student identity systems rely on direct sharing of personal information through resumes, ID cards, and online platforms, which exposes users to risks such as data leakage, impersonation, and unauthorized access. These systems lack controlled access mechanisms and do not provide a reliable method to verify the authenticity of users or requesting entities.

This paper presents Secure ID Vault, a secure digital identity platform that utilizes a Decentralized Identifier (DID)-based architecture for privacy-preserving identity management. Each student is assigned a unique DID, represented as a QR code within a mobile application. The QR code contains no personal data and acts only as a reference pointer. Upon scanning, the DID is sent to a centralized backend server, which performs validation, verification, and access control before returning only authorized information.

The system incorporates an administrative verification mechanism to authenticate both students and companies, ensuring a trusted ecosystem. Additionally, a secure messaging module is integrated, enabling verified companies to communicate directly with students through controlled backend APIs without exposing personal contact details. A lightweight anomaly detection mechanism is also implemented to identify suspicious activities such as excessive QR scans and invalid DID requests.

The platform is developed using Flutter for the mobile interface, Node.js with Express for backend services, and SQLite for data storage. The proposed system demonstrates improved privacy, controlled data sharing, and secure communication, making it suitable for academic identity management and recruitment environments.

KEYWORDS:

Digital Identity, Decentralized Identifier (DID), QR Code Authentication, Access Control, Identity Verification, Secure Messaging, REST API, Privacy-Preserving Systems

INTRODUCTION

The rapid growth of digital platforms and online recruitment systems has significantly increased the need for secure identity management in academic environments. Students frequently share personal information such as resumes, contact details, and identification documents across multiple platforms, often without control over how this data is accessed, stored, or reused. This uncontrolled data sharing exposes users to

risks including data leakage, impersonation, phishing, and unauthorized access.

Traditional identity systems rely on direct data exchange, where sensitive information is embedded in documents or QR codes and shared with requesting entities. These approaches assume trust in the receiving party, which is not always valid in real-world scenarios. Furthermore, such systems lack mechanisms for dynamic access control, verification of entities, and auditability of data usage, making them vulnerable to misuse.

Modern digital platforms provide partial solutions through authentication and profile-based systems; however, they still require users to expose significant amounts of personal data. Additionally, there is no standardized method to ensure that only verified entities can access identity information. This creates a clear need for a secure, controlled, and privacy-preserving identity management system that minimizes data exposure while maintaining usability.

To address these challenges, this paper presents Secure ID Vault, a secure digital identity platform based on Decentralized Identifiers (DIDs). The system introduces a novel approach where identity is represented using a DID encoded as a QR code, which contains no personal information. Instead of exposing data directly, the QR code acts as a reference pointer, and all identity resolution is handled through a centralized backend server.

The backend performs validation, verification, and access control before returning any information. Both students and companies are verified through an administrative process, ensuring that only trusted entities participate in the system. This architecture eliminates direct data exposure and enforces strict control over what information is shared.

In addition to identity management, the system integrates a secure messaging module that enables communication between verified companies and students without revealing personal contact details. A lightweight anomaly detection mechanism is also incorporated to identify suspicious activities such as excessive QR scans or invalid identifier requests.

This paper presents the design, implementation, and evaluation of the BLACK ID system. The main contributions are:

1. A secure DID-based identity framework that separates identity representation from actual data storage.
2. A backend-controlled access mechanism that ensures privacy-preserving data sharing.
3. An administrative verification model for establishing trust between students and companies.
4. A secure messaging system that enables controlled communication without exposing personal information.
5. A lightweight security monitoring approach for detecting abnormal system usage patterns.

The rest of the paper is organized as follows: Section 2 reviews related work in digital identity and access control systems. Section 3 describes the implementation study, including the existing system and the proposed architecture. Section 4 details the technologies and system components used. Section 5 presents system evaluation and discussion. Section 6 concludes the paper and outlines future enhancements.

LITERATURE SURVEY

1. Traditional Identity Systems and Their Limitations

Traditional identity systems in academic and recruitment environments rely on direct sharing of personal data through resumes, ID cards, and online forms. These systems lack controlled access mechanisms and expose sensitive information unnecessarily. Once shared, users lose control over their data, leading to risks such as data leakage, impersonation, and unauthorized reuse. Additionally, there is no mechanism to verify whether the requesting entity is legitimate, making these systems inherently insecure.

2. QR-Based Identity Systems

QR codes are widely used for identity representation in tickets, student IDs, and event systems. However, most existing QR-based systems embed personal data directly within the QR code. This creates a major security vulnerability, as anyone scanning the QR can access the data without authorization. Furthermore, these systems do not provide backend validation or dynamic access control, making them unsuitable for secure identity management.

3. Access Control Mechanisms

Access control models such as Role-Based Access Control (RBAC) are commonly used to restrict access to sensitive data. While these mechanisms are effective in controlled environments, many identity systems fail to implement them properly. In most cases, data is either fully exposed or completely restricted, with no fine-grained control over what specific information should be shared. This lack of selective disclosure reduces both security and usability.

4. Backend-Centric Identity Systems

Modern systems are shifting towards backend-controlled architectures where data is not directly exposed to users. Instead, identity information is stored securely on servers and accessed through APIs. This approach allows for validation, auditing, and controlled data sharing. However, many existing implementations still lack proper verification mechanisms and rely heavily on trust assumptions rather than enforced security policies.

5. Secure Digital Identity and DID Concepts

Decentralized Identifiers (DIDs) have emerged as a promising approach for identity management, where identity is represented by a unique identifier rather than raw data. DIDs enable separation between identity representation and data storage, improving privacy and security. However, most DID-based systems are complex, blockchain-dependent, or not optimized for practical use in academic environments.

6. Secure Communication in Identity Systems

Communication between entities in identity systems is typically handled through email or external messaging platforms, which exposes personal contact information. These approaches lack privacy and control, as users cannot restrict who can contact them. Secure, system-controlled messaging mechanisms are required to ensure communication without exposing sensitive details.

7. Security Monitoring and Anomaly Detection

Recent systems incorporate basic security monitoring techniques to detect abnormal activities such as unauthorized access or excessive requests. While advanced systems use machine learning, lightweight rule-based anomaly detection is often more practical for structured systems. Monitoring patterns such as repeated invalid requests or unusual activity can significantly enhance system security without requiring complex models.

8. Gap Analysis

Existing identity systems suffer from several limitations, including lack of data privacy, absence of verification mechanisms, and inability to control data access dynamically. QR-based systems expose sensitive information, and traditional platforms do not ensure trust between users and organizations.

The Secure ID Vault system addresses these gaps by introducing:

- DID-based identity abstraction (no direct data exposure)
- Backend-controlled access and validation
- Admin-based verification for trust establishment
- Secure messaging without revealing personal contact details
- Lightweight anomaly detection for enhanced security

This combination provides a practical, secure, and scalable solution for modern academic identity management.

IMPLEMENTATION STUDY

EXISTING SYSTEM

Traditional student identity management systems rely on direct sharing of personal information through resumes, ID cards, and online profiles. In academic and recruitment environments, students are required to repeatedly submit sensitive data such as contact details, academic records, and identification documents to multiple platforms. This approach leads to excessive data

exposure and increases the risk of data leakage, impersonation, and unauthorized access.

Existing QR-based identity systems are also widely used; however, they often embed personal information directly within the QR code. As a result, anyone who scans the QR code can access sensitive data without any form of authentication or authorization. These systems lack backend validation mechanisms and do not provide control over who can access the information, making them inherently insecure.

Furthermore, most current systems do not include a proper verification process for users or organizations. Students cannot verify whether a company requesting their information is legitimate, and companies cannot reliably confirm the authenticity of student profiles. This lack of mutual trust leads to issues such as fake recruiters, fraudulent profiles, and misuse of personal data.

Another major limitation is the absence of controlled access mechanisms. Once data is shared, users lose control over its usage, and there is no way to restrict or monitor how the information is accessed or distributed. Additionally, communication between students and companies typically occurs through external channels such as email or messaging platforms, which exposes personal contact details and increases the risk of phishing attacks.

Overall, existing systems follow a static and uncontrolled data-sharing model, where sensitive information is directly exposed without verification, access control, or monitoring. This reactive and unstructured approach makes them unsuitable for secure and scalable identity management in modern academic environments.

PROPOSED SYSTEM

The BLACK ID system addresses the limitations of traditional identity systems by introducing a secure, controlled, and privacy-preserving digital identity architecture. The system follows a centralized backend model with a mobile application interface, administrative control, and secure communication mechanisms.

The system operates as follows:

---

1. Student Registration and DID Generation

Each student registers through the mobile application by providing basic details. Upon successful registration, the backend generates a unique **Decentralized Identifier (DID)** for the student.

- The DID is generated using a UUID-based approach
- The DID is stored in the database and linked to the student profile
- The mobile application displays the DID as a QR code

This ensures that no personal data is stored within the QR code.

---

2. QR-Based Identity Representation

The generated DID is encoded into a QR code using the mobile application. The QR code acts only as an identity reference and does not contain any sensitive information.

- QR can be scanned by companies
- QR contains only DID (not personal data)
- Safe even if shared or copied

This design eliminates the risk of direct data exposure.

---

3. DID Resolution and Backend Validation

When a company scans the QR code, the extracted DID is sent to the backend server for processing.

The backend performs:

- DID existence check
- Student verification status check
- Company verification status check

Only after successful validation, the backend resolves the DID and fetches the corresponding student data.

---

4. Access Control and Data Filtering

The system enforces strict access control policies to ensure that only authorized data is shared.

- Verified companies can access permitted student data

- Unverified entities are denied access or given restricted data
- Data is filtered dynamically based on authorization rules

This prevents unauthorized data access and ensures privacy.

---

5. Admin Verification System

An administrative module is used to verify both students and companies.

- Students are verified based on submitted identity details
- Companies are verified based on registration and authenticity checks
- Verification status is stored in the database

This establishes trust within the system.

---

6. Secure Messaging System

The system includes a built-in messaging module that allows communication between verified companies and students.

- Companies send messages via backend APIs
- Messages are stored securely in the database
- Students receive messages through the mobile application

This eliminates the need to share personal contact information.

---

7. Security Monitoring and Anomaly Detection

A lightweight monitoring mechanism is implemented to detect suspicious activities.

- Tracks excessive QR scan requests
- Detects repeated invalid DID attempts
- Flags abnormal usage patterns

Suspicious activities can be logged or restricted to enhance system security.

---

8. Backend and Data Management

The backend server acts as the central authority of the system.

- Handles API requests and validation
- Stores data in SQLite database
- Ensures all operations pass through controlled logic

This ensures consistency, security, and auditability.

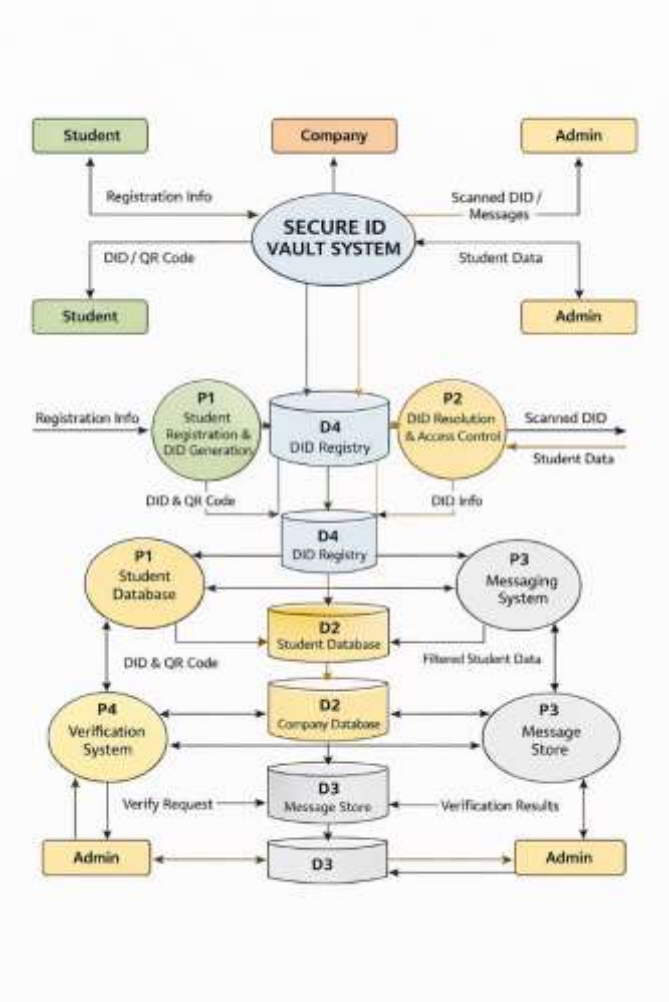

Fig:1

Secure ID Workflow (Fig:1)
(Student Registration → DID Generation → QR Display → QR Scan → Backend Validation → Data Access → Messaging)

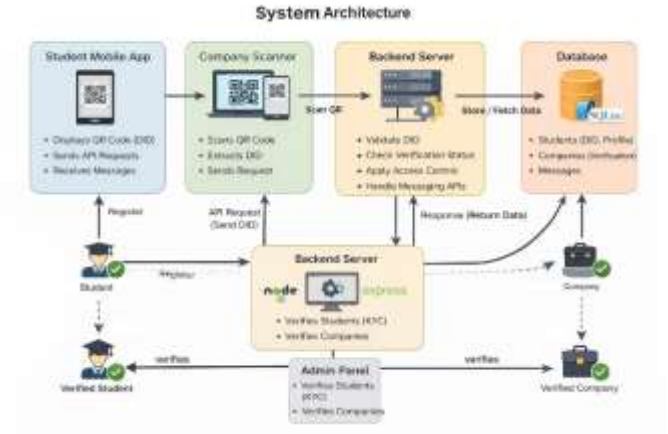

Fig:2

Secure ID System Architecture (Fig:2) *(Mobile App, Company Scanner, Admin Panel connected to Backend Server and Database)*

SOFTWARES AND LIBRARIES DESCRIPTION

The Secure Id Vault system is implemented using a client-server architecture with a mobile frontend and a centralized backend. The system uses modern web and mobile development technologies to ensure scalability, performance, and ease of integration.

PROGRAMMING LANGUAGE

- **Dart (Flutter)**
  Used for developing the cross-platform mobile application. It enables efficient UI rendering and seamless integration with backend APIs.
- **JavaScript (Node.js)**
  Used for backend development, enabling fast and scalable server-side processing using asynchronous programming.

CORE LIBRARIES AND TOOLS

Component : Libraries/Tools

Frontend (Mobile Application)

- **Flutter – Cross-platform UI framework for Android/iOS**
- **Dart – Programming language for Flutter**
- **qr_flutter – Library for generating QR codes from DID**
- **HTTP / Dio – For API communication with backend**
- **Firebase Cloud Messaging (optional) – Push notifications for messaging**

Backend Server

- **Node.js – Runtime environment for server-side logic**
- **Express.js – Web framework for building REST APIs**
- **SQLite – Lightweight relational database for data storage**
- **JWT (JSON Web Token) – Authentication and session management**
- **bcrypt – Password hashing for secure storage**
- **CORS Middleware – Enables secure cross-origin API access**

Database Layer

- **SQLite – Stores student data, company data, DID mappings, and messages**

API Communication

- **REST APIs (JSON) – Handles communication between frontend and backend**
- **Axios / Fetch – Used for sending HTTP requests**

Messaging System

- **REST-based Messaging APIs – Sends and retrieves messages**
- **Polling Mechanism – Periodic message fetching by client**
- ***(Optional)* Firebase Cloud Messaging for real-time updates**

Utilities and Supporting Tools

- **UUID Library – Generates unique DIDs for students**
- **Nodemon – Development tool for auto-restarting server**
- **Postman / Thunder Client – API testing and debugging**

KEY LIBRARIES EXPLAINED

1. **Flutter** – Provides a responsive UI and allows QR-based identity display within the mobile app.

2. **Node.js & Express.js** – Handle all backend operations including DID generation, validation, access control, and messaging.

3. **SQLite** – Lightweight and efficient database suitable for structured identity storage.

4. **JWT & bcrypt** – Ensure secure authentication and password protection.

5. **qr_flutter** – Generates QR codes that encode only the DID, ensuring no sensitive data exposure.

6. **REST APIs** – Enable secure communication between system components with controlled data exchange

CONCLUSION

This paper presented Secure ID Vault, a secure digital identity platform designed to address the limitations of traditional identity management systems in academic and recruitment environments. The proposed system introduces a Decentralized Identifier (DID)-based approach, where identity representation is separated from actual data, ensuring minimal data exposure.

The system leverages a centralized backend architecture to perform identity resolution, verification, and access control. By ensuring that QR codes contain only the DID and no personal information, the system eliminates the risks associated with direct data sharing. The inclusion of an administrative verification mechanism establishes trust between students and companies, while the access control logic ensures that only authorized entities can retrieve specific information.

In addition, the integration of a secure messaging module enables controlled communication between users without exposing personal contact details. The lightweight anomaly detection mechanism further enhances system security by identifying suspicious activities such as excessive QR scans and invalid identifier requests.

Overall, the Secure ID Vault system provides a practical, scalable, and privacy-preserving solution for digital identity management, demonstrating improved security, controlled data sharing, and enhanced trust in academic ecosystems.

Future work includes:

The current system provides a strong foundation for secure identity management; however, several enhancements can be implemented to improve functionality and scalability:

1. **Role-Based Data Visibility**
   Implement fine-grained control to allow students to customize which fields (e.g., email, phone, skills) are visible to companies.

2. **Real-Time Messaging System**
   Replace polling-based communication with WebSocket or push-based messaging for instant message delivery.

3. **Data Encryption and Security Enhancements**
   Apply encryption techniques for stored data and API communication to further strengthen security.

4. **Decentralized Identity Integration**
   Extend the system to support blockchain-based DID registries for enhanced decentralization and trust.

5. **Advanced Anomaly Detection**
   Introduce machine learning-based models for detecting complex usage patterns and potential threats.

6. **Scalability Improvements**
   Replace SQLite with scalable databases such as PostgreSQL or MongoDB for large-scale deployments.

7. **Multi-Platform Support**
   Extend the application to web platforms for easier accessibility by companies and administrators.

8. **Notification System Integration**
   Integrate email or push notifications to alert

users about messages, verification status, and system updates.

REFERENCES

[1] W3C, "Decentralized Identifiers (DIDs) v1.0," World Wide Web Consortium (W3C), 2022. https://www.w3.org/TR/did-core/

[2] W3C, "Verifiable Credentials Data Model 1.0," W3C Recommendation, 2019. https://www.w3.org/TR/vc-data-model/

[3] M. Sporny, D. Longley, and D. Chadwick, "Decentralized Identifiers (DIDs) Overview," W3C Draft, 2020.

[4] OWASP, "Authentication and Access Control Guidelines," OWASP Foundation, 2023. https://owasp.org/www-project-top-ten/

[5] NIST, "Digital Identity Guidelines," NIST Special Publication 800-63, 2017. https://pages.nist.gov/800-63-3/

[6] Flutter Team, "Flutter Documentation," Google, 2024. https://docs.flutter.dev/

[7] Node.js Foundation, "Node.js Documentation," 2024. https://nodejs.org/en/docs/

[8] Express.js, "Express Web Framework Documentation," 2024. https://expressjs.com/

[9] SQLite Consortium, "SQLite Documentation," 2024. https://www.sqlite.org/docs.html

[10] RFC 7519, "JSON Web Token (JWT)," IETF, 2015. https://datatracker.ietf.org/doc/html/rfc7519

[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. (Referenced for decentralized identity concepts)

[12] OWASP, "Secure Coding Practices and API Security," 2023. https://owasp.org/www-project-api-security/

[13] Firebase, "Firebase Cloud Messaging Documentation," Google, 2024. https://firebase.google.com/docs/cloud-messaging

[14] ISO/IEC, "Information Security Management Systems (ISO 27001)," 2022.