Data Security and Privacy Final Report

**Team-1: Members**

Salman Mohammed Parki ([mparki@kent.edu](file:///Users/rasulkhanpatan/Downloads/mparki@kent.edu))

Rasul Khan Patan ([rpatan@kent.edu](mailto:rpatan@kent.edu))

Anu Poojitha Pusuluri ([apusulu1@kent.edu](file:///Users/rasulkhanpatan/Downloads/apusulu1@kent.edu))

**Instructor:** Xu Lei

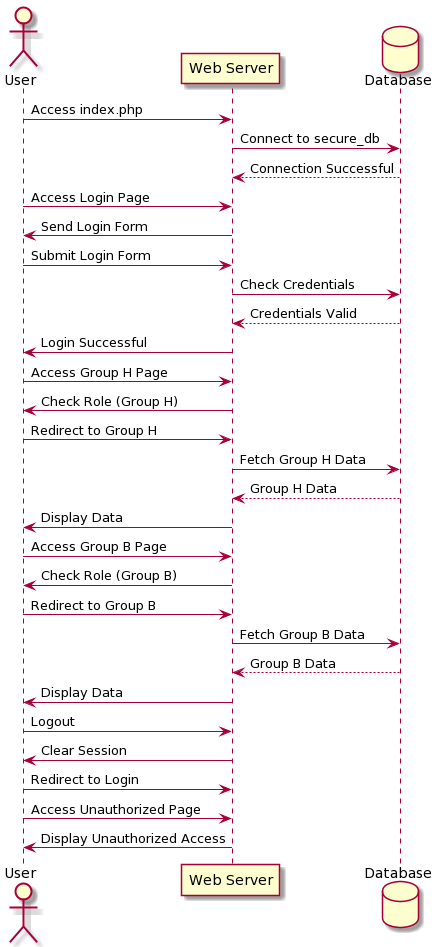
**KENT STATE UNIVERSITY**

Department of Computer Science

**Project Report**

**System Architecture**

The system architecture of the Secure Database-as-a-Service (DBaaS) consists of several components that interact to provide secure access to a healthcare database. Below is a high-level overview of the system architecture:



Key Components and Their Roles:

1. **User Interface:** This component represents the web interface that users interact with. It provides access to the system and presents data to authorized users.
2. **Web Server:** The web server handles incoming requests from users and routes them to the appropriate parts of the system. It also serves web pages to users and ensures secure communication.
3. **Database-as-a-Service (DBaaS):** The core component of the system is the DBaaS. It includes features for user authentication, access control, and data protection.
4. **Authentication & Access Control:** This part of the DBaaS is responsible for user authentication and authorization. It ensures that only authenticated users can access the system and enforces role-based access control.
5. **healthcare\_info (Database):** The database stores sensitive healthcare information. Access to this data is strictly controlled based on user roles.

**Security Feature Implementation**

**User Authentication**

User authentication is implemented in the system to verify the identity of users. It is achieved using the following mechanisms:

* User credentials (username and password) are securely stored in the database.
* Passwords are hashed using the bcrypt algorithm before storing to protect them from exposure. Bcrypt is a strong cryptographic hash function that makes it computationally expensive for attackers to crack passwords.

**Basic Access Control Mechanism**

The basic access control mechanism differentiates between two user groups, Group H and Group B. This distinction ensures that users can only access permitted attributes. The steps taken to implement this mechanism are:

* User roles (Group H and Group B) are defined during user registration.
* The system checks the user's role before allowing access to specific database attributes.
* Role-based access control is enforced to restrict access to the database, ensuring that each user group can only access their respective authorized attributes.

**Basic Query Integrity Protection**

Query integrity protection is implemented to ensure users can detect modified or fake query results. The system achieves this by:

* Implementing mechanisms for verifying the integrity of query results.
* Using cryptographic methods or checksums to verify the authenticity of data returned from the database.
* Regularly checking the consistency of query results to detect anomalies or unauthorized changes.

**Single Data Item Integrity**

Users must be able to detect modifications in individual data items to maintain data integrity and trust in the system. The system employs:

* Hashing or digital signatures to protect individual data items.
* These integrity checks are applied to critical data fields, such as patient records, to ensure that any unauthorized changes are detected.

**Query Completeness**

Ensuring query completeness is crucial for maintaining data consistency and privacy. The system addresses this by:

* Implementing mechanisms to verify that query results contain all expected data items.
* Detecting if one or more data items are missing from query results.
* Protecting against data omissions, which could compromise patient records.

**Basic Data Confidentiality Protection**

Sensitive attributes, such as gender and age, are protected to prevent unauthorized access. This protection extends to both the cloud and local database management systems. The system ensures confidentiality by:

* Implementing encryption mechanisms to protect sensitive data both in transit and at rest.
* Access controls and encryption techniques are applied to safeguard this information from unauthorized access.

**Member Contributions**

Our team collaborated on this project, with each member making significant contributions. Here's an overview of the team members' contributions:

* **Member 1: Salman Mohammed Parki (811260513)**
  + Implemented query integrity protection.
  + Created the system's web interface.
  + Worked on final report.
* **Member 2: Patan.Rasulkhan(811236865)**
  + Collected all the required data.
  + Worked on data creation and implementation.
  + Worked on final report.
* **Member 3: Anupoojitha Pusuluri (811294951)**
  + Ensured single data item integrity.
  + Worked on data confidentiality protection.
  + Worked on backend coding.
  + Did research on security features by collecting various research papers.
  + Worked on final report.

GitHub Commit History for Member 1

<https://github.com/salmanpmd/DSP.git>

GitHub Commit History for Member 2:

<https://github.com/patanrasulkhan/DSP.git>

GitHub Commit History for Member 3:

<https://github.com/Anupoojitha/datasecurity.git>

**Limitations of the Project**

While our system provides strong security features, there are some limitations to consider:

1. **Vulnerabilities and Security Risks:** No system is completely immune to vulnerabilities. As new threats emerge, vulnerabilities may be discovered. Regular security updates and testing are essential.
2. **Enhancements in Security Features:** The current system provides basic security features. There is room for improvement, such as implementing advanced threat detection and mitigation mechanisms.
3. **Privacy Concerns:** Despite our efforts to protect sensitive data, there is always a risk of privacy breaches. Users must remain vigilant and follow best practices to protect their data.
4. **Technical Limitations:** Technical limitations may hinder the achievement of certain security goals. These limitations can include hardware constraints, compatibility issues, and resource constraints.

By acknowledging these limitations, we demonstrate a proactive approach to security and a commitment to continuous improvement in our system.