**Project Architecture**

**Project Overview** As messaging applications gain widespread popularity, concerns about data privacy and security are increasing. Traditional messaging platforms often rely on centralized servers, which are vulnerable to unauthorized data leaks and security breaches. CryptConnect addresses these issues by creating a secure messaging website focused on end-to-end encryption, giving users greater control over their data and secure key management.

The project aims to provide a robust and customizable communication platform that leverages trusted technologies and advanced security measures. Key features include user authentication via Firebase, encrypted messaging, and secure key exchange using Diffie-Hellman and AES encryption.

**Features**

* **User Registration:** New users can create accounts with credentials (username, email, password) stored securely using Firebase Authentication.
* **Login:** Previously registered users can log in with their credentials to access chats.
* **Password Reset:** Users can reset their passwords
* **Chat Functionality:** Users can search for other users by username to initiate conversations.
* **Profile Editing:** Logged-in users can edit their profiles.
* **End-to-End Encryption:** All messages are encrypted during transit and at rest in the Firebase database.
* **Key Management:** Public-private key pairs are generated for each user, with private keys encrypted using the user’s password for added security.
* **Secure Key Exchange:** Diffie-Hellman is implemented for deriving shared keys, ensuring secure communication.
* **File Upload and Storage**: Users can securely upload files (.pdf, .docx, .txt), which are encrypted using AES and stored in the Firebase database. Files can be downloaded and decrypted on the user's device.

**Technical Details**

CryptConnect employs the following architecture:

**Frontend:**

For the frontend, we have used React (JavaScript). The frontend handles user interactions, such as registration, login, password reset, chat functionality, etc.

**Backend:**

For the backend, the technology we have used is Firebase. It handles the data storage, user authentication, and message delivery. Messages and private keys are stored in encrypted formats to ensure the security.

**Encryption Mechanisms:**

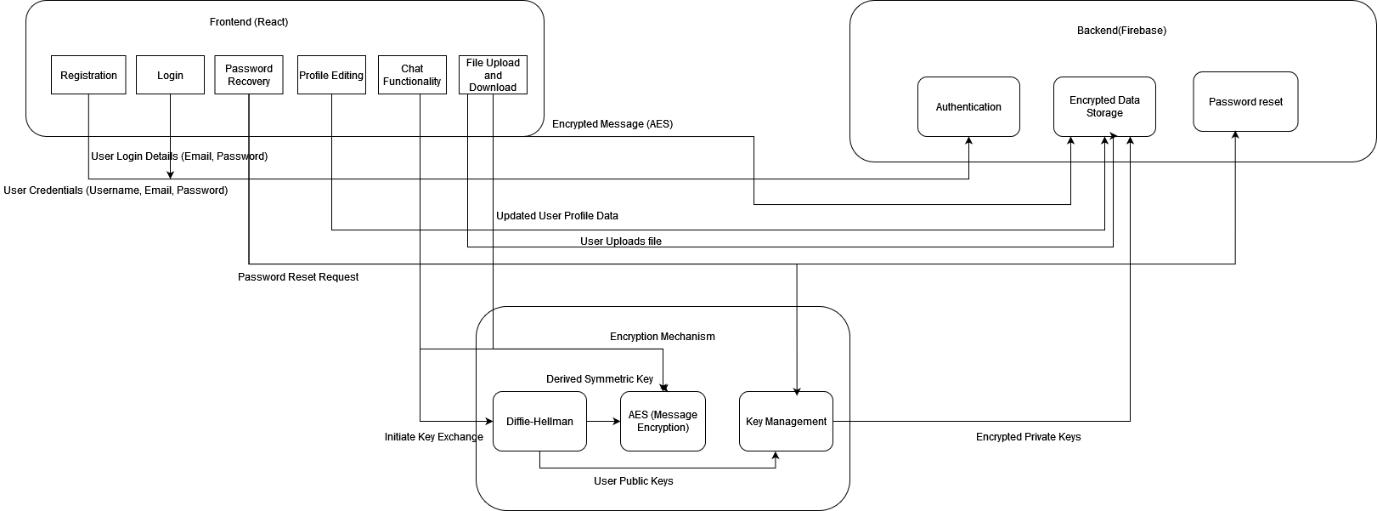
In our project, we have used 2 primary encryption mechanisms which are Diffie-Hellman and AES. Diffie-Hellman is used to establish shared secret symmetric keys between the users without transmitting any sensitive information over the network. During the registration process, public and private keys are generated for each user where private key is encrypted with the password. In addition to this, messages are encrypted using the derived symmetric keys before being sent to the Firebase database. AES ensures confidentiality and integrity of these messages.

**Inner Workings**

Our platform employs a robust and secure workflow to ensure user data privacy and integrity. During registration, users provide their credentials, which are securely managed using Firebase Authentication. A unique public-private key pair is generated for each user, with the private key encrypted using their password for additional security. This encryption ensures that sensitive information remains protected even in case of unauthorized access. The messaging process involves users initiating chats by searching for contacts via usernames/emails. A shared symmetric key is derived using the Diffie-Hellman key exchange, which facilitates secure communication. Messages are encrypted using AES with this symmetric key and securely stored in the Firebase database. For file upload functionality, users can upload files in supported formats (.pdf, .docx, .txt). Each file is encrypted using AES with a shared secret key before being securely stored in the Firebase database. When a user downloads a file, the encrypted file is fetched from the database and decrypted on the user’s device using the shared secret key. This ensures that file storage and retrieval maintain the same level of security as the messaging system.

Additionally, CryptConnect ensures seamless account recovery and user control. Password recovery is facilitated by Firebase Authentication, which sends a secure link to registered emails, enabling users to reset passwords. Upon resetting, the new password is used to re-encrypt private keys, maintaining the continuity of secure communications. Logged-in users can also edit their profiles, with all updates securely stored in the database. This workflow ensures that users benefit from a secure, reliable, and privacy-focused messaging experience.

**CryptConnect Architecture Block Diagram**

****