

# Smart Metro Rail Ticketing System

**Introduction:** In everyday life, transportation is essential. The main goal of this paper is to make traveling much more relaxing by eliminating the common and significant issues related to having a ticket with you while you're traveling. We are recommending a biometric-based ticketing system for this purpose, however it is not restricted to the metro railway scenario. We take into account each person's fingerprint as soon as they register, purchase tickets, and validate the fingerprint on the day of travel so he or she can travel on a certain day and on the chosen train to his or her selected destination. Arduino will be connected to the fingerprint sensor, and Arduino will then store the fingerprint data online. We suggest using a two-way encryption standard to keep the private fingerprint information on the cloud. The data is encrypted twice as part of this two-way encryption standard: once during data production at the hardware end and once more before being stored in the cloud database.

**User Scenario:** Travelers may lose their tickets, which is an issue in the metro station. While on the trip, the ticket could be taken. They could be receiving inaccurate findings in that circumstance, which would be problematic. When a person is traveling and misses their destination, the RFID tag will not recognize them as a legitimate customer, and they will be charged accordingly. The customer must pay some additional money. There will be a lot of individuals, thus the system needs to be quick to respond. The algorithm should also be quick. The ticket has to be kept safe in case they misplace it and he is unable to pass through the checkout gates. We are creating a method that eliminates the need for a physical ticket and uses the individual's fingerprints instead to solve all of these issues. Travelers can lose their tickets, which is an issue in the metro station[12][13]. While on the trip, the ticket could be taken. They could be receiving inaccurate findings in that circumstance, which would be problematic. The RFID tag will not recognize the user as a legitimate customer while they are traveling if they are missing their destination, and they will be charged accordingly. The traveler must pay some additional money. There will be a lot of individuals, thus the system needs to be quick to respond. The algorithm should also be quick. To avoid being unable to pass through checkpoint gates if they lose their ticket, the passenger must carry it with them at all times. We are creating a method that eliminates the requirement for a physical ticket and uses the user's fingerprints instead to solve all of these issues.

## *Ticket Booking Workflow*

- By providing the source, destination, and date of travel information, the user books the ticket.

- The created data is then delivered to the AWS EC2 instance after the user just enters his or her fingerprint, which is secured with AES.
- In this case, we encrypt it using our encryption method, and our searching algorithm locates the user's information in the DynamoDB database containing the data for the registered user.
- The page will acknowledge the booked ticket and add the ticket's details to a new DynamoDB table if the user is a registered user.
- The registration process may need to be repeated if the user is not already registered.
- The complete table of saved tickets is divided based on the date-of-journey because DynamoDB is a NoSQL database.
- At the time of the trip, the local ticketing kiosk will only get the information from users who are traveling on that specific day.
- The user enters his or her fingerprint on the day of the trip, and if the data matches the data recorded in the ticketing kiosk, the gates open and the user can continue traveling.

### UML:

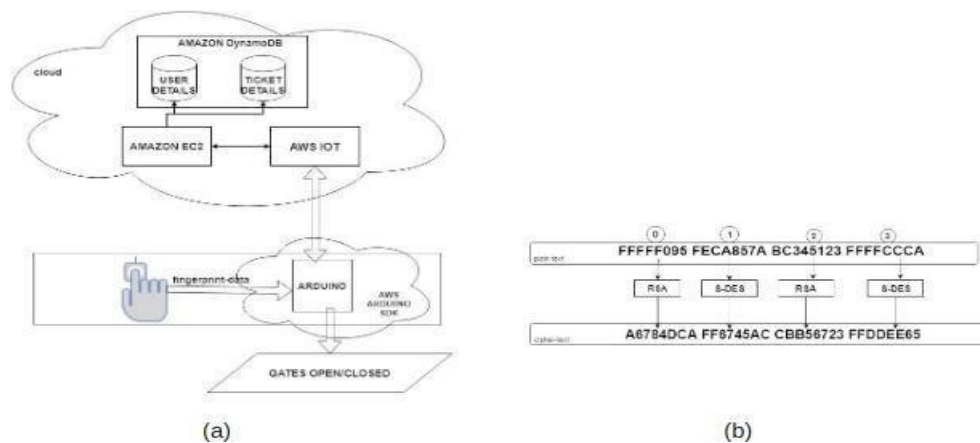


Fig. 1. (a)Proposed Architecture (b) Hybrid Encryption Algorithm

## UI Design Specification:

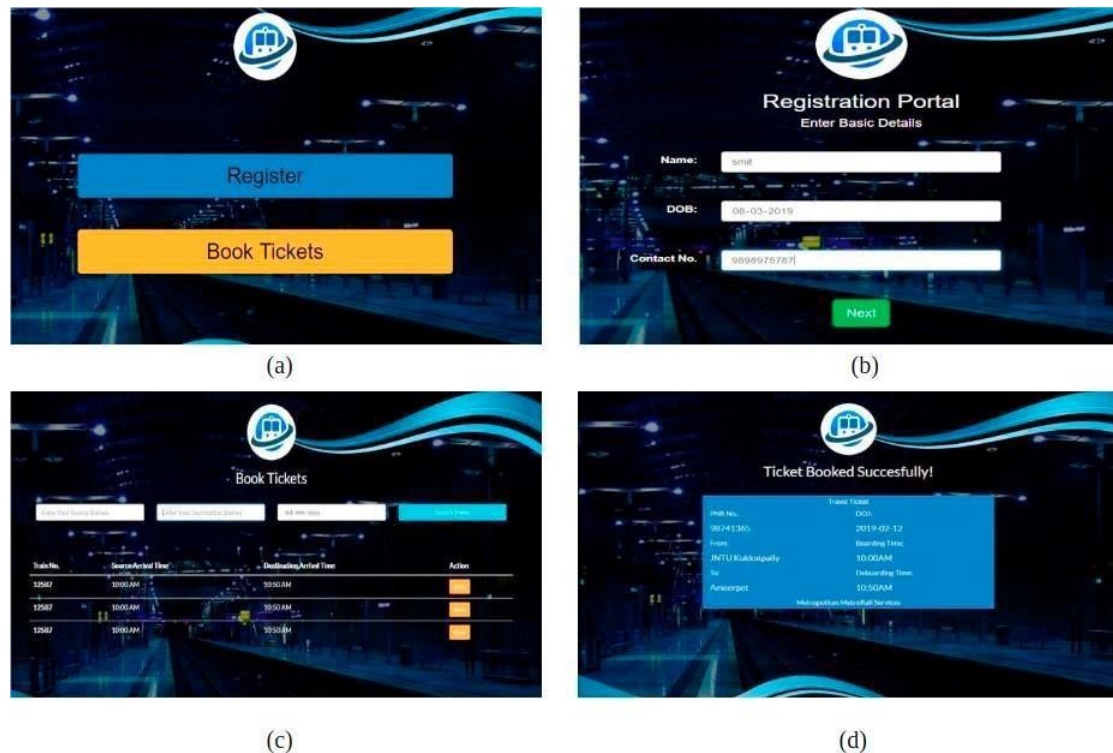


Fig. 2. (a) Webpage start screen (b) User registration screen (c) Ticket-booking page (d) Successfully generated Ticket

The initial page of the website, as seen in figure, allows logged-in users to either book travel tickets or register as a new user (a). When a new user first visits the website, they are greeted by this page, which asks them to provide some basic information. The information they provide on this page is transferred to the python backend and then to the Amazon cloud. A valid user can do so from this website, but they must enter the necessary information first, as shown in figure (b). The information will be transmitted to the cloud, which will then return a list of trains the user can choose from for that particular day of travel. The user chooses any train from the list before entering their fingerprint, as illustrated in figure (c). The web application's final step is to acknowledge the user for the booked ticket, which the user will subsequently save as illustrated in figure (d).

**Challenges:** While attempting to develop this system, we encounter some difficulties. Some users, it appears, do not possess smartphones. We occasionally run across situations when a consumer has an erroneous perception of our service. Therefore, it is crucial to understand their issues, accept them, and offer them appropriate answers. And in order to do this, we need to include a feature on our booking app that allows users to report problems.

**Conclusion:** We eliminated the inconveniences associated with routine travel by using this technique. Physical tick-etc./tokens and other UID cards or travel-related documentation are no longer required. The user will experience a more relaxing and convenient journey thanks to the suggested methods.

**Future Works\_:** The utilization of ECC (Elliptic curve cryptography) to secure the private fingerprint data is a significant future goal for this project. Due to the fact that ECC approach is ideal for protecting the data created by low power devices like Arduino. Additionally, since ECC offers an equivalent level of security to RSA and DES, we are able to completely disregard two-way encryption and simply offer a single-level of encryption with ECC. The implementation of the same program as a mobile application would allow users to book tickets directly from their mobile devices, eliminating the need for a separate kiosk. This would be another potential outcome of this project.