Improving Rider Safety Using QR Code & Fingerprint Biometrics

Raghu Nandan Avula
Department of Electrical & Computer Engineering
University of Central Florida,
Orlando, USA
Raghu.avula@ucf.edu

Cliff Zou
Department of Computer Science
University of Central Florida,
Orlando, USA
czou@cs.ucf.edu

***Abstract*— Transport Network companies (TNCs) have changed the way we travel in the last five years where a rider can book a ride using her smartphone. However, TNC doesn't provide any robust mechanism to validate the driver or the rider before the ride. This has led to many violent incidents ranging from assault, kidnap of the riders by fake ride-hailing drivers. The most recent one that shook the entire nation is the murder of a USC student when the rider got into the wrong car thinking that it is her Uber [1]. To address this problem, we have proposed a solution that adds an extra security layer in authenticating both rider and driver before initiating a ride. In this solution, both rider and driver will authenticate themselves using technologies like QR Code and fingerprint biometrics supported by modern smartphones before they take the ride.**

Keywords*— *Rider safety, Driver safety, QR Code, Finger-print Biometric, Smartphone;

I. INTRODUCTION

Uber, the most popular peer-to-peer Transport Network Company has over 75 million riders and 15 million trips completed each day. Over 10 billion trips completed so far by the drivers that use Uber [2]. On the other hand, Lyft which is the second largest ride-hailing company in the US has over 550 million rides in the year 2018 [3]. Current security measures provided by these ride-hailing companies for the rider safety while taking a trip are 1) show the driver's face, 2) show the driver's license plate and 3) show the driver's car model in the mobile app. It is the rider's responsibility to check these parameters before she gets into the car. There is no trustworthy mechanism provided by these ride-hailing companies to authenticate the driver or the rider before they start the ride, which has led to many violent incidents ranging from assault, kidnap of the riders by fake ride-hailing drivers.

In this paper we have proposed a solution that will use technologies like QR Code and finger print scan on the smartphone to authenticate both the rider and the driver before they take the ride. QR Code is used by Service Provider (a ride-hailing company) to implement the two-way trust between the rider and the driver and make sure that correct driver and rider are taking the trip, whereas fingerprint sensor is used to authenticate both the rider and the driver before they initiate the trip. In the next sections we have discussed about the QR code, Fingerprint scan technologies followed by the design of our solution.

II. FUNCTIONS SUPPORTED BY SMARTPHONES

A. QR CODE

QR Code is the most popular matrix barcode identification system which was used initially in the automotive industry. QR code stores the information both horizontally & vertically in a square format. Any human-readable data like URL, text, phone number or address value can be encoded and stored into the QR Code. Nowadays any smartphone camera can decode the QR code into the original text. In our solution, Service Provider smartphone app is going to generate a unique QR Code for each trip.

B. FINGERPRINT BIOMETRIC

Fingerprint is an important biometric technique for personal identification. Smartphone companies like Apple, Samsung or Google have introduced and utilized fingerprint scan to unlock the phone. They have extended this feature to perform certain financial transactions using the services like Apply pay, Google pay, or Samsung pay. Nowadays, many third-party applications use fingerprint scan instead of password to enable a user to unlock and use their app. In our solution we have used fingerprint scan to identify the rider and driver by the Service Provider before initiating each ride.

III. SYSTEM DESIGN

Our proposed solution is based on the QR Code and Fingerprint sensing and we have assumed that both riders and drivers have smartphones with Service Provider's mobile app installed and fingerprint scan enabled. Both the rider and the driver authenticate each other using the fingerprint along with QR code generated for the trip by the Service Provider. Many successful implementations like Apple Pay use a fingerprint scan of the user to perform financial transaction by the phone. We have utilized similar functionality where the rider and the driver must use fingerprint authentication to prove themselves.

QR code is readable by any modern-day smartphone, hence we are concerned about the security of data that we need to store in the QR Code used to build the two-way trust

978-1-7281-3885-5/19/$31.00 ©2019 IEEE

between the rider and the driver. We have decided to go with a separate QR code for each trip to avoid QR code reuse attack. This will make sure that no QR code will ever be the same between two trips. Each QR code contains the hash value of the trip that is sent by the Service Provider to the driver's mobile app. Service Provider's mobile app will create a new QR code for every trip based on the tripID received by the Service Provider. A symmetric key is created by the Service Provider for each driver during his account creation. Every tripID is encoded by Service Provider using the driver's symmetric key before it is sent to the driver's smartphone. This tripID is later decrypted to original tripID by using the driver's symmetric key on Service Provider's cloud.

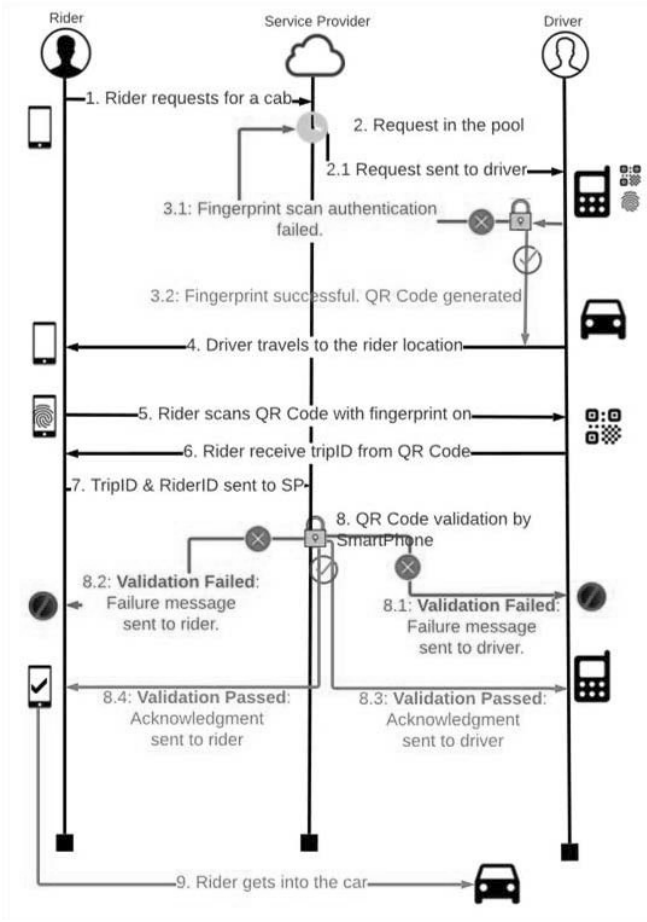

Figure 1: Rider and Driver authentication workflow

Figure 1 represents the complete lifecycle of the rider and driver authentication mechanism right from the rider booking the ride to the rider getting into the car. In step 1, the rider requests for a cab in the Service Provider's mobile app. In step 2, Service Provider sends the request to the driver. In step 3, the driver receives the trip information from Service Provider and the driver must use his fingerprint to view the QR code that is generated based on the tripID sent by Service provider. A response is sent back to Service Provider if the driver is unable to scan his fingerprint to generate the QR code. This is an important step in our design as it authenticates the driver of the trip and generates the QR code which needs to be scanned by the rider. In step 4, the driver travels to the rider location. In steps 5, 6, 7 the rider scans the QR code using her smartphone with her fingerprint on and the retrieved tripID information is sent to Service Provider cloud from the rider's mobile app.

In step 8, using the driver's symmetric key, Service Provider will decode the encoded tripID to get the original tripID. From the original tripID, Service Provider will get the information about the trip, driverID, and riderID. We are relying upon smartphone ability to validate driver or rider's identity using the fingerprint scan. Validation of the driver is done by the tripID received from the rider's smartphone. The tripID is shared only to the driver by the service provider and QR code is generated only with the driver's fingerprint verified by biometric scan sensor on the driver's smartphone. Validation of the rider is done by the riderID received by Service Provider in step 7. If Service Provider is unable authenticate either the driver or the rider a failure message is sent to both requesting not to proceed with the drive as shown in the steps 8.1 & 8.2. In steps 8.3 & 8.4, an approval acknowledgement is sent to both the rider and the driver if Service Provider can authenticate tripID and riderID sent by the rider. It will notify both the rider and the driver to proceed with the ride.

IV. Problem Formulation

We have addressed the following challenges as a part of the solution that involves the authorization of rider and driver during a trip using QR Code and biometrics.

A. Problem 1: Authentication of Driver and Rider

Authentication of Driver and rider is done within the smartphone application using phone login and biometrics. For a rider to book a trip or driver to accept the trip in the app, they need to login into the app using their registered username and password. Upon the first login rider and driver has an option to authenticate using the on-device biometrics. Upon successful on-device biometrics with the smartphone app, rider or driver need to use the biometrics to authenticate themselves during the trip. Biometrics functionality is already provided by the smartphone application and we are using it within our smartphone application.

B. Problem 2: Authorization of the trip by the Rider and Driver.

After a successful login into the app, rider will book the trip by selecting the source and destination. Upon a successful request for the trip, service provider will select and send out the trip information to driver. At this point, driver will authorize themselves by accepting the trip using the on-device biometrics by which they are identifying themselves. After a successful authorization and acceptance of the trip, a QR Code is generated on the driver smartphone. To authorize themselves as the rider who initiated the trip, rider need to scan the QR Code in the app using their biometrics.

C. Problem 3: Generation of driverID, riderID, trip and corresponding symmetric keys.

We have SHA3 as our Secure Hash Algorithm to create the driverID, riderID and their corresponding keys [4].

a. After a successful registration, driverID, riderID are created by the service provider.

$$I_d = SHA3 - 256(UUID())$$

$$I_r = SHA3 - 256(UUID())$$

b. Corresponding keys are created for drivers and riders which will be used during the trip.

$$K_d = SHA3 - 256(I_d \,\&\, UUID())$$

$$K_r = SHA3 - 256(I_r \,\&\, UUID())$$

c. Trip ID is created as hash of UUID. Trip key is generated as mentioned below.

$$I_{trd} = SHA3 - 256(\, UUID())$$

$$K_{trd} = AES - 256_e(AES - 256_e(I_{trd}\,, K_d), K_r)$$

$$R_{trd} = AES - 256_d(AES - 256_d(K_{trd}\,, K_d), K_r)$$

Legend:

- $AES - 256_e$: *AES ENCRYPTION FUNCTION*
- $AES - 256_d$: *AES DECRYPTION FUNCTION*
- $SHA3 - 256$: *SECURE HASH ALGORITHM 3 FUNCTION*

V. EXPERIMENTAL DESIGN

A. Design

Our current implementation is Client/Server model where the client is smartphone app installed in a smartphone and server is cloud-based Google Firebase with Real-time database and Firebase cloud. All the communications happen using Firebase Cloud Sub/Pub triggers. Flutter which is an open-source mobile development framework was used to develop the smartphone app.

1. Authentication mechanism is implemented using Firebase Authentication.
2. As a backend database, we are using Google Firebase Cloud Firestone.
3. To perform the business logic and necessary workflow, we have used Firebase Cloud Functions.

1) Smartphone application using Flutter

We have used Flutter [5] which is one of the most popular mobile application development frameworks to build our smartphone application. Users can export either an iOS or an Android application directly from the framework. Our scope of this project is to export an android app on to an Android mobile. We have built the following modules that reside within the smartphone app.

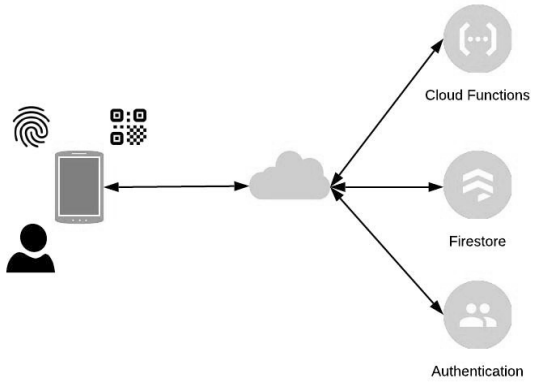

Figure 2: Workflow of our app using Google Firebase Components

a) Login/Signup Module:

Login Signup module is the integral part of our application. In the signup module, user will select whether they want to be a driver or rider along with their proposed email address and password. After a successful signup and the email verification, user can login and access the smartphone application. In the backend Login/Signup module uses Google Firebase Authentication.

b) QR Code Generation and Biometrics Module:

Usage of the Biometrics to authorize both driver and rider is an integral part of the smartphone application. Driver need to use biometrics to accept the trip request sent the service provider. After the acceptance of the trip a QR Code is generated in the Driver's smartphone. Rider needs to scan the QR Code which is enabled with the biometrics of the rider.

2) Google Firebase

Google Firebase [6] mobile application development platform to build smartphone application quickly and without having to manage backend infrastructure. We have segregated our backend implementation on Firebase into two main modules as mentioned below:

a) Security Module:

Security module takes care of generation of driverID, riderID and the corresponding symmetric keys during their

registration. Generation of the tripID and the corresponding trip key using driver's and rider's asymmetric key. Encoding and Decoding of the trip key is handled by the security module. Security module uses Google Firebase Cloud Functions.

b) Notification Module:

Notification module is an important entity which is used to push various notifications to driver or rider's smartphone. Alert module handles push notifications like:

- Trip notification to driver about a new trip.
- Trip notification to rider about the acceptance of the trip by driver and their information.
- Trip notification to driver and rider about success of failure of trip authorization.
- Biometrics validation success/failure notification to driver.

VI. Experimental Design

User safety in the ride sharing scenarios is one of the most important topics which has been ignored to a greater extent. With the latest happenings in the transport network companies, there is a need to address the user safety issue and look for relevant solutions from technology perspective. After an evaluation of safety features provided by ride sharing apps like Uber and Lyft, it is observed that a major security validation is missing which is rider and driver authorization during the trip. Feeney [7] addressed the possibility of potential violent crimes when rider is taking the rider in a stranger's car which not regulated by any agency. Amey, Attanuci, Mishalani [8] also discussed about the challenges of real-time ridesharing and mainly concerned about the data that is collected by the Service Provider during the day. To our best of knowledge, we are first ones to use on-device biometrics and QR Code technology to validate rider and driver before the start of the trip.

VII. Conclusion

In this paper, we have introduced an extra security layer in the ride-hailing taxi book mechanism that will authenticate both the rider and the driver before proceeding with a ride. Our design uses established technologies like QR code and the fingerprint scanning to build a two-way trust between the rider and the driver. Fingerprint scanning is used in the design to authenticate the rider and the driver within the Service Provider's mobile app. QR code can store any form of data in a matrix barcode format and hence it is used to store the sensitive information that can be shared between the rider and the driver to build the trust. This is currently in the implementation phase and as a future work we would like to complete a prototype and test it with the real users.

VIII. Acknowledgement

This work is supported by the National Science Foundation under grant DGE-1723587.

IX. References

[1] President's Message to the Carolina Family, Available: https://www.sc.edu/about/our_leadership/president/letters/aletter_tothe_carolina_family.php , Accessed: 2019-09-30.

[2] Company Information | Uber Newsroom, Internet, Available: *https://www.uber.com/en-GB/newsroom/company-info/* , Accessed: 2019-09-30

[3] A Closer Look At Lyft's Valuation – Forbes, Available: https://www.forbes.com/sites/greatspeculations/2018/10/10/a-closer-look-at-lyfts-valuation/#76d9d2d84a97, Accessed: 2019-09-30

[4] SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Available: *https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions?pub_id=919061 ,* Accessed: 2019-09-30

[5] Flutter: Google UI Toolkit, Available: *https://flutter.dev,* Accessed: 2019-09-30

[6] Google Firebase, Available: *https://firebase.google.com,* Accessed: 2019-09-30

[7] Andrew Amey, John Attanucci, and Rabi Mishalani, "Real-Time Ridesharing Opportunities and Challenges in Using Mobile Phone Technology to Improve Rideshare Services." Transportation Research Record: Journal of the Transportation Research Board, No. 2217, 2011

[8] Matthew Feeney: *"*Is Ridesharing Safe?" Cato Institute Policy Analysis, 2015