UCLA

ProtectedU

Chiao Lu, Shen Teng, Mu-Te Shen, SangJi Lyu

CS M117
Spring 2018
June 6th

UCLA

Abstract

Through continuous invention and renovation of technology, identifying yourself has become easier than ever before. Unfortunately, impersonating someone has become easier as well. As long as one has your physical identification card, they can pretend to be you. For example, the barcode on our UCLA ID card is simply plain text encoding of our ID number. To make the hacking process even simpler, one can take a picture of that barcode, and do a lot of things in the name of you, without being noticed since the card is not “lost.” Our goal is to propose a new way of identification with the security that algorithm can provide. Each time a student needs to pull out their ID, like in the dining hall or at libraries, they can either use the generated QR code, or use the NFC (simply tap) to connect with the receiver. With our app, one doesn’t have to worry about lost ID, or malicious copying of the ID.

Theory

- Near Field Communication (NFC). Specifically, Peer to Peer NFC. This was done to achieve maximum security for our app. In our application, we worked closely with Android; thus, used a feature in Android (mobile) operating system called Android Beam. Android Beam authorize data exchange via NFC, and allows one to read NDEF messages from other NFC device.
- SHA-256 Cryptographic Hash Algorithm. We used SHA-256 to generate a unique 32 byte signature for the student's ID. This is useful as hash is a one way function; thus, near impossible to be decrypted to the text it once was. Our app also adds a timestamp and a secret salt to the hashing, and the salt is known only to the server and the specific student. The timestamp adds the security so that the generated hash string cannot be reused by a theft.
- Google Cloud Platform and NoSQL were both used for, mainly, data storage and retrieval of data. Google Cloud Platform was used for database (students information) and server purpose.

User flow

- Registrar create user
  - With student’s name and email
- Student verify their identity
  - The vcode is only valid for 5 minutes
  - Obtain student id and salt after successfully verified
- While student wants to checkout books
  - Student sent the primaryKey and hashed student ID to scanner
  - Scanner sent the information to the server to verify, and return with true/false.

Video Demo

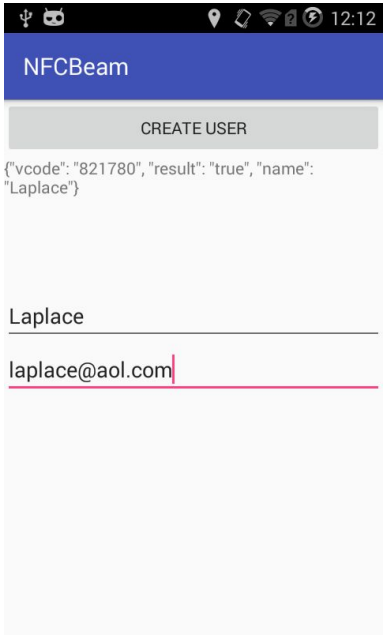
- https://youtu.be/bVTEiDR5GSE

Frontend - Design

- Register (put student information in database)
- Scanner
  - Scanner (authenticate student id)
- Scennee
  - Student (go to register to sign up; go to scanner to authenticate)

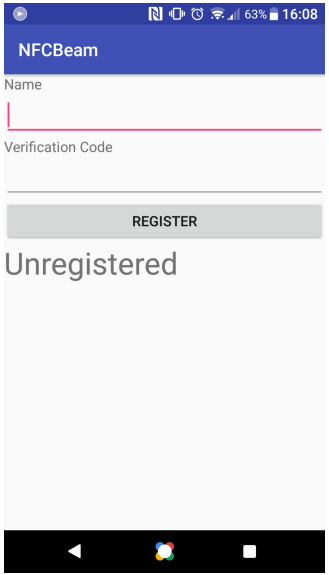
Frontend - Register

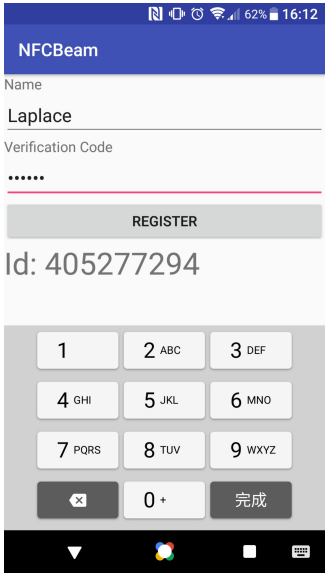
- A “name” field and an “email” field
- Click “CREATE USER” button, and the register app will contact Google Cloud Platform (GCP) to input the student info.
- GCP will return a “vcode”. This “vcode” is then handed to the student. The student uses this vcode to setup the scannee (student id).
- “Result” field tells the register if “create user” is successful



Frontend - Scannee

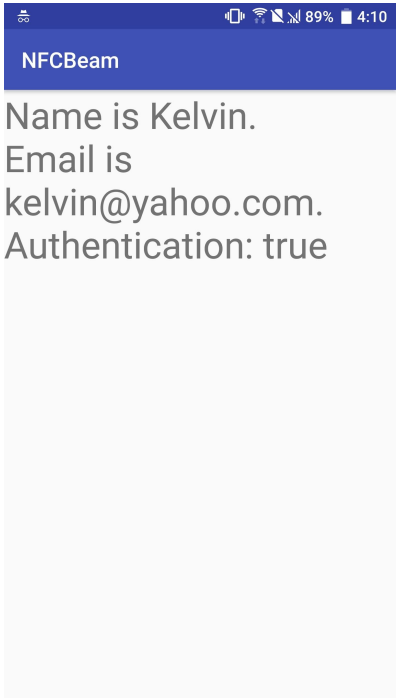
- First time open app (left pic): student will take the vcode from register (prev slide) and setup his app (student id)
- Student id number will be returned from the server and displayed on the text box below.
- After setting up, when the student wants to have his ID scanned, for example to check out a book from library, he can open his app and bring the phone close to the scanner. The scanner will verify if the ID is authentic or not.





Frontend - Scanner

- The student will bring his phone physically in contact with the scanner. The scanner will use NFC to scan the student's phone.
- After scanning the student's phone, the scanner app will contact GCP to verify if the student's ID is authentic or forged.
- We prevent fraudulent verification by sending a SHA256 hashed string which includes the student id number, secret salt, and timestamp.



Backend - Design

- Server
  - Python Flask
    - URL routing
  - HTTP server
    - Google Cloud Platform App Engine
- Database
  - Google cloud storage
  - NoSQL

Backend - API

- /createuser
  - Input: name, email
  - Output: result, vcode
- /verifyuser
  - Input: name, vcode
  - Output: studentID, primaryKey, salt
- /verifyid
  - Input: primaryKey, IDHash
  - Output: result, name, email

Discussion

- Grace period during the first 5 seconds of each minute
- Secret salt to add security
- Hash results change with time

Conclusion

Overall, our application is built to keep student’s life and identity safe. At school no one should worry about their safety. With combination of Near Field Communication (NFC) and SHA-256 our application is built to have security, and in turn keep your Student ID safe.

Questions?

UCLA

Thank you!