Milestone

Google Cloud Speech API Based Encrypted Notes

Harsha Kota

Abstract:
This paper talks about construction of a mobile application for Android that is safe and secure to store your sensitive private information such as login credentials and credit card information by implementing AES (Advanced Encryption Standard) to encrypt and decrypt data and additional features such as voice transcription powered by Google's Cloud Speech API for ease of use.

Introduction:
Motivated by the goal of constructing a secure encrypted notes/memo's application that can save any text data while providing useful features such a voice transcription, this paper discusses what has been implemented so far into the application and later what will be implemented further and also talks about many other features, not all of which will be implemented into the application but provide useful ways to protect the application.

Background and Related Work:
There are several applications on Google's Android Play Store that provide a way to save notes/memo's, but none have been found that provide voice transcription using Google's Cloud Speech API. None of them show data in their encrypted state; most of the authentication is done beforehand on the lock screen or before the application is accessed, while these provide enough security, displaying an encrypted message has a certain appeal to it.

Methodology:
The approach taken in building this application is as follows:

- First the landing page for the application ie. the main screen is built which upon start will show users with any previously saved notes in their encrypted form. The notes are displayed from recently created/modified to old.
  This page has buttons to Encrypt and Decrypt notes if there are any, and also a button to create new notes.
- The create new notes button launches another page which lets users enter data/notes.
  New notes can be entered in two ways, typing it in using any keyboard application or by using voice transcription.
  The voice transcription is powered by Google's Cloud Speech-To-Text API, with their machine learning technology it uses powerful neural network models to convert audio to text. It can recognize 120 languages and variants, to support global user base. This application is implemented with their real-time streaming process that transcribes your voice as you speak directly into the application.
  This paper also considers the problem of the application needing to have a constant internet connection for voice transcription to work and later discuss in the paper an alternative and

layout its pros and cons, in deciding to implement Google's Cloud Speech-To-Text API over Android's onboard Speech-To-Text API.

- Currently, the data; each note is stored as a serialized Java object, that stores the notes creation date and the note itself on the applications private space in the internal storage of the phone. Serialization is a way that developers turn their data structures into a stream of bytes for transport or storage. Deserialization is the reverse process that happens when data is retrieved.

  AES will be implemented to encrypt the Java serialized object before saving it to memory, that will help against exploitation of stolen data. It will use a 256-bit key size for encryption. Java security *package java.security* and the Java cryptography extensions (JCE) *package.javax.crypto* contain a pair of classes designed to address these challenges. These classes protect the integrity of the serialized objects.

Experiments:

- Test the display order of the notes on the main page, which must show notes with a sorted filter from recently created/modified to last, so you always see your recently created/modified note at the top of the list.
- Display voice transcription in real time and on the final result of the transcription, the text is added to the body of the note.
- Simultaneous use of keyboard and voice transcription to enter data without any conflicts.

Discussion and/or Analysis:

- A sort function has been applied to each note's date field to sort the notes before they are populated on the main page.
- The real-time transcription has been separated and displayed temporarily on the note's date field so as to not add any intermediate results to the note. Upon receiving the final result from the API, the text will be added to the note.
- While it may seem like an unlikely use case scenario, it has been thoroughly tested and allows the simultaneous use of keyboard and voice transcription to enter data.

  Possible additional features:

  - As an alternative to voice transcription using Google's Cloud Speech API, Android's onboard Speech-To-Text API (Recognizer Intent) is taken into consideration.
    While Android's onboard implementation doesn't provide a powerful speech recognition when compared to the cloud-based service, weighing on the pros and cons it poses as a potential additional feature that will help a small group of users when the device is offline.
    Pros:
    - It is available even when the application is offline
    - It is free

    Cons:
    - Need to pass local language to convert speech to text.

    - Only works with Android Devices
    - Not all devices support offline speech input.

- Using Shake Detection through the phones inbuilt accelerometer to lock down access to unencrypted data, so when the phone detects a shake, it switches the display of any unencrypted data to their encrypted counterparts as a quick way to secure the data.

- If the phone has a fingerprint hardware, an extension to the simple use of AES with the fingerprint to encrypt and decrypt data can be implemented.

Conclusion:

Encryption provides an extra layer of protection when saving usernames, passwords, credit card information and all sorts of other login credentials and private information on mobile phones.
In the future, this application can be implemented on various other platforms, including its Google Cloud API feature, as it is not platform dependent. To improve on the existing storage solution, a cloud-based database service can be used to store the data as a backup in case the phone's data is lost. Text notes can be further extended to include images, videos to cater to bigger audiences.

References:

1. Ekta Agrawal, Dr. Parashu Ram Pal. (2017) *A More Effective Approach Securing Text Data Based On Private Key Cryptography.* In: International Journal on Recent and Innovation Trends in Computing and Communication, vol. 5, issue. 3.

2. Suchita Tayde*. (2015) *File Encryption, Decryption Using AES Algorithm in Android Phone.* In: International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, issue. 5.

3. Veton Këpuska, Gamal Bohouta. (2017) *Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx).* In: Journal of Engineering Research and Application, vol. 7, issue. 3, pp. 20-24