GoogleApple token generating process.md

# Google/Apple token generating process

#### Workflow:

1. .
   1. [CRNG](https://en.wikipedia.org/wiki/Cryptographically_secure_pseudorandom_number_generator) stands for cryptographic random number generator
   2. RPIK stands for Rolling Proximity Identifier Key, will be used to generate RPI
   3. [HKDF](https://en.wikipedia.org/wiki/HKDF) is a key derivation function based on a [hash-based message authentication](https://en.wikipedia.org/wiki/Hash-based_message_authentication_code)[[1]](https://en.wikipedia.org/wiki/HKDF#cite_note-iacr-1) [code](https://en.wikipedia.org/wiki/Hash-based_message_authentication_code) (HMAC). It says if two RPIK are the same, their corresponding tek should be the same
   4. is a time related data and encoded in 32 bits:
   5. is symmetric

#### What will happen after user receive a RPI:

1. Pre: Use list downloaded from server to generate list.
2. . This is done via and key used is
3. See if there is a match of

#### What Google provides in Github:

* Google does not provide implementation details. Instead, it provides high level APIs

import com.google.android.gms.nearby.exposurenotification.ExposureConfiguration;  
import com.google.android.gms.nearby.exposurenotification.ExposureInformation;  
import com.google.android.gms.nearby.exposurenotification.ExposureNotificationClient;  
import com.google.android.gms.nearby.exposurenotification.ExposureSummary;  
import com.google.android.gms.nearby.exposurenotification.TemporaryExposureKey;

#### My Plan accordingly:

* Declare all the methods involved. These methods will not change the input and will return directly. We are going to implement cryptographic details in the future.
* This week will focus on the workflow:
  + How to broadcast and scan periodically.
  + How to upload information to the server
  + How to download informaiton from server periodically.
  + Database design on the server