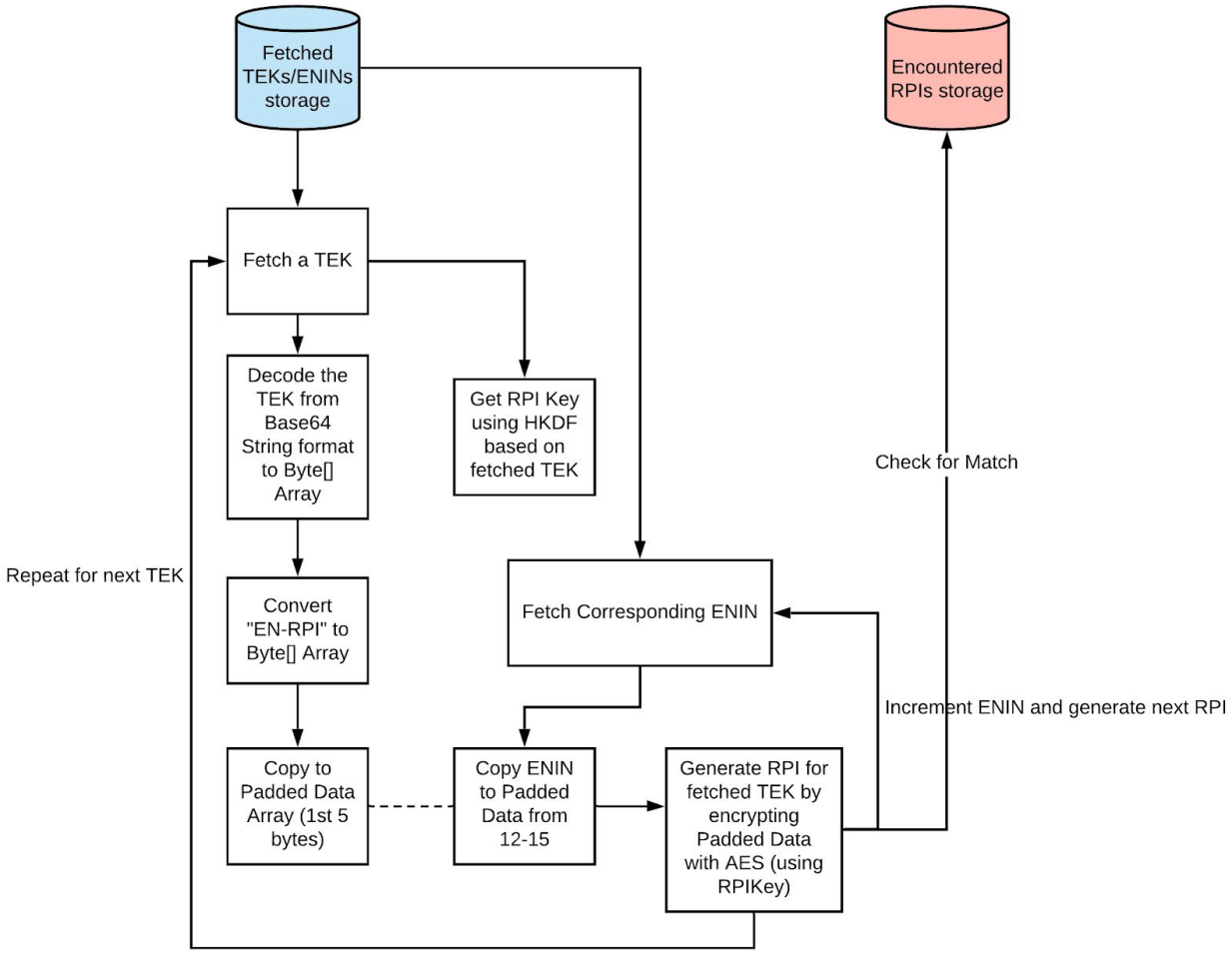
Weekly Blog 8

This week I mainly started working on the regeneration of RPIs from fetched TEKs which is a crucial step to match with the encountered RPIs. The following diagram illustrates the step visually.



The code for the above regeneration part is currently ***under progress***. The algorithm for the above procedure is as follows:

1. All the TEKs submitted by other users is stored in storage (currently in progress by another teammate)
2. Since the fetch part is currently in progress, I have now hardcoded TEKs for testing purposes.
3. Using GAEN's official cryptographic documentation followed earlier, I have clubbed and modified the code for regeneration.

4. Padded data is a byte array that is responsible for RPI generation, as it contains ENIntervalNumber which is a UNIX epoch timestamp based on the time TEKs and RPIs are created.
5. Padded data is a 16-byte array where 0 to 5 bytes are bytes extracted from the string EN-RPI.
    a. The bytes 12 to 15 make up the ENIntervalNumber
    b. Other bytes are set to zeros.
6. The RPIKey is derived from the fetched TEK using the HKDF library which was also previously used for deriving RPIs for transmission in earlier stages.
7. AES algorithm is used for encrypting the padded data corresponding to the fetched TEK.
8. RPI is finally regenerated for the ENIntervalNumber corresponding to the time of TEK creation and creation of the first RPI of the TEK.
    a. Repeat step 8 by incrementing ENIntervalNumber and generating the next RPI.
9. Regenerated RPIs are stored in data structures like an ArrayList and passed on to the matchmaking stage as a data argument.
10. The entire process is repeated for the other TEKs

Analysis of COVID-19 Tracking Tool in India: Case Study of Aarogya Setu Mobile Application.

The paper provides an in-depth analysis of the Arogya Setu app. I have written my weekly blog in a manner that compares Arogya Setu with COVID Guard. This helps me in identifying the strong and weak points and potentially leverage it during my research paper assignment.

Arogya Setu	COVID Guard	Comments
Uses GPS and BLE	Uses BLE only	COVID Guard would be preferred by users as it does not use personal identification information like GPS
Requires Location Services for detection	Requires Location Services for detection	Owing to the mandatory Android requirement
Requires Name, Age, Phone Number, and other profiling information	Does not ask for any profiling information, devices identified through UUID	
Acts as an app to completely social graph the contact's whereabouts	Merely acts as an exposure indicator	COVID Guard: Would be preferred by users due to its privacy by design. Arogya Setu: Would be preferred by authorities for social graphing.
Follows a centralised architecture where both positive and non-positive people would need to upload their identifiers for the match to a central server	Follows a decentralised architecture where only positive cases would need to upload their anonymised cryptographically secure identifiers to the server. They are downloaded and matchmaking happens on the user's device itself	COVID Guard: would be preferred by users as little to no data leaves user's devices
Very little documentation is available	Transparent documentation about how identifiers are anonymised with detailed algorithms and logic	COVID Guard: would be preferred by users due to its transparency.
Chances of false positives are high. For instance, if two users (one infected and other non-infected) are in a building separated by a wall, the GPS based approach would categorise them as infected.	Chances of false positives are less, as BLE is not strong enough to pass through solid walls.	

GPS consumes more power	BLE consumes less power	
Highly invasive: asks for profiling info, GPS location, Centralised.	Highly private: does not ask for profiling info, no GPS location, decentralised.	
Useful for public health authorities for identifying clusters through GPS location, complete contact tracing.	Useful for users as having highly private by design principles. Merely acts as an exposure indicator.	

Overall, I believe COVID Guard stands out in its privacy by design and exposure indication capabilities for users. However, it cannot be useful for public health authorities and for social graphing purposes as it merely acts as an exposure indicator. Users would ultimately prefer the privacy of their information over anything. As such, for making the app to be used by everyone stronger privacy by design paradigm is required for Arogya Setu. This is offered by COVID Guard. As such, COVID Guard stands out.

REFERENCES

1. Rajan Gupta, Manan Bedi, Prashi Goyal, Srishti Wadhera, and Vaishnavi Verma. 2020. Analysis of COVID-19 Tracking Tool in India: Case Study of Aarogya Setu Mobile Application. Digit. Gov.: Res. Pract. 1, 4, Article 28 (August 2020), 8 pages. https://doi.org/10.1145/3416088