CHAPTER 4

GEO-KEY METHOD

The enhancement of encryption and decryption key generation process of this research by implementing the adopted method into the key generation process in existing symmetric AES method. Figure 4.4 illustrates the process of generating AES geo-key to be used in the encryption and decryption process which includes the used of Raspberry Pi and GPS module for retrieving the location information. This enhanced AES geo-key method requires three additional parameters which are (i) location information consists of latitude and longitude coordinates, (ii) user password and (iii) device MAC address to generate the encryption key known as geo-key.

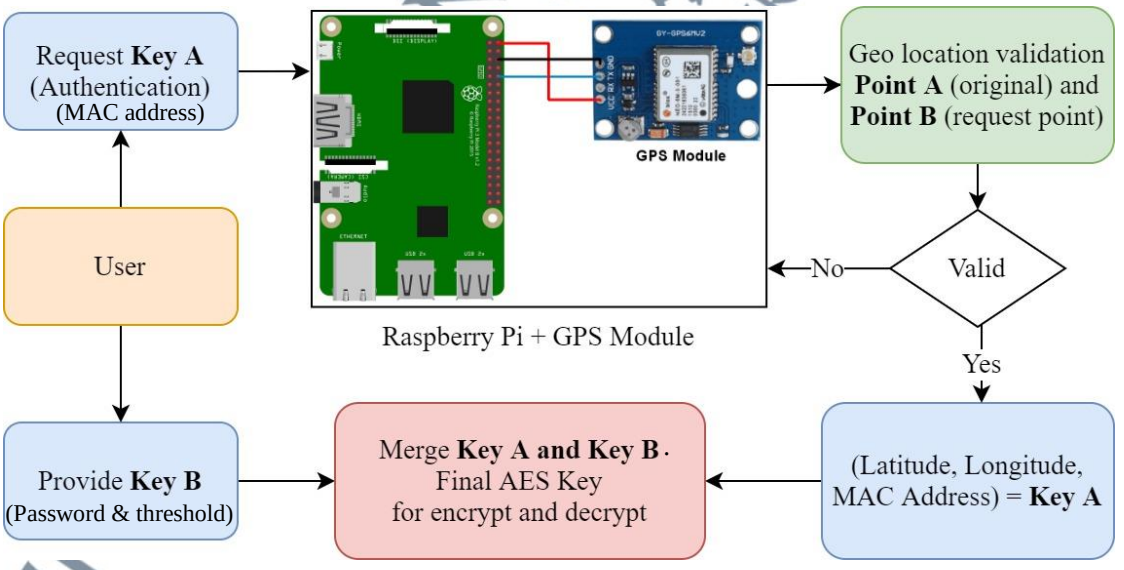

Figure 4.1: Process of generating AES geo-key to be used in encryption and decryption process.

In general, any GPS-enabled device should provide the latitude and longitude of its whereabouts. However, in this research, used a Raspberry Pi 4 with NEO GPS Module board (Pi-GPS). There is no limitation of which hardware can be choose but instead, the important data that it requires are the coordinates of the longitude and latitude from the devices. The main objectives here are method used for generation the AES and validation based on user location to obtain this key for the purpose of encryption and decryption of data. In Figure 4.1, illustrate the data flow, decisions and events for the generating the encryption key using location-based cryptographic.

Based on Figure 4.1, there are three main components in the location-based cryptographic which are discuss in details as the following sub-chapters.

4.1 User's account and connectivity to Pi-GPS

The first part is the initiation of user request of the key to be used for encryption. The pre-requirement at initial stage in Figure 4.1, requires user to register its credential on for the Pi-GPS box to register its initial location. Take note that the Pi-GPS box will not be connected to any LAN or WAN network, however it requires the device to be connected via USB to the user terminal such as computer and server. A local database is created within Pi-GPS for storing the credential and initial GPS location.

The Key A in Figure 4.1 refers to key that will be provided by the Pi-GPS and user device MAC address while Key B refers to the user's password. These keys combination described in this research known as Geo-key. Next, the setup of the distance threshold. The threshold is recommended to be set up within the range of the user residential or specific building office sizes. In this research, we used to set the threshold in range of 5 meters up to 50 meters.

GPS location retrieved via Pi-GPS is denoted by Point A and Point B where Point A represent origin location of where the file is been encrypted while Point B represent request point of decryption location. For the geo location validation, detail process is briefly described in the following section C.

4.2 Conversion of GPS longitude and latitude to X and Y coordinates

Firstly, on converting GPS coordinated into flattening map projection we will use the Equirectangular projection method (Weisstein, 2011) in the following Equation 3.1. We can simply use the horizontal axis x to denote longitude λ, the vertical axis x to denote latitude φ.

$$x = R(\lambda - \lambda 0)\cos\varphi_1$$
$$y = R(\varphi - \varphi_1) \quad (3.1)$$
$$x = R(\lambda)\cos\varphi_1$$
$$y = R(\varphi)$$

Where: λ : is the longitude in radians of the location to project;

φ : is the latitude in radians of the location to project;

φ_1 : are the standard parallels;

λ_0 : is the central meridian of the map;

x : is the horizontal coordinate on the map;

y : is the vertical coordinate on the map;

R : is the radius of the globe.

The ratio between should use cos φ_1 as the aspect ratio, where φ_1 denotes a latitude close to the centre of your map. Furthermore, to convert from angles which measured in radians to lengths is require to multiply by the radius, R of the earth (which in this model is assumed to be 6371 km). The central meridian λ_0 is 0. The algorithm 1 provide the

pseudocode as a detailed step in the process of developing a program based on equation 4.1.

Algorithm 1 Conversion longitude, latitude to X, Y

```
1: Initialization radius ← 6371 .                          > Earth Radius in KM
2: Input GPS longitude, latitude
3: Output Coordinates X,Y
4: procedure CLASS REFERENCEPOINT
5:     procedure INIT(longitude, latitude) .                > Conversion to radian
6:         radLng ← longitude ∗ pi/180
7:         radLat ← latitude ∗ pi/180
8: procedure LATLNGTOGLOBALXY(radLng,radLat)
9:     X ← radius ∗ radLng ∗ cos(0)
10:    Y ← radius ∗ radLat
11: return X,Y
12: procedure MAIN
13:    p0 ← ref erencePoint(longitude, latitude)
14:    p0 ← latlngToGlobalXY (p0.radLng, p0.radLat)
15:    return p0
```

4.3 Validation of user location within the setup threshold

Next, to create a distance radius threshold based on the x, y coordinates obtained from the longitude and latitude coordinates. The initial encryption location of the person denoted as Point A and the request decryption location of the person denoted as Point B. The validity denoted as V, if V is equal to 1 represent the decryption request was inside the distance radius threshold while if V is equal to 0, it determines the decryption was request outside the distance radius threshold as in Equation 4.2.

(4.2)
$$\Delta T = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

$$V = \begin{cases} 1, & if\ \Delta T \leq D_T \\ 0, & if\ \Delta T > D_T \end{cases}$$

Where: D_T : the distance threshold limit

(x, y) : set as coordinate location (lat, lon)

A : original location

B : requested location

ΔT: variant distance between 2 point of A and B

V : Distance validity

The algorithm 2 provide the pseudocode as a detailed step in the process of validation of user location within the setup threshold based on equation 2.

Algorithm 2 Validation of user location within the setup threshold

1: **Initialization**
2: $DT \leftarrow 1$. > distance threshold in KM
3: $V \leftarrow 1$. > validity of position
4: **Input**
5: $A \leftarrow A.x, A.y$
6: $B \leftarrow B.x, B.y$
7: **Output** variant distance T
8: **procedure** MAIN
9: $T \leftarrow sqrt((A.x - B.x)^2 + (A.y - B.y)^2)$
10: if $T \leq DT$ then
11: $V \leftarrow 1$
12: else
13: $V \leftarrow 0$
14: return T,V