ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Application of Peer to Peer based Geofence aided by Location Based Services (LBS) incorporated with Unique Digital Envelope Security system.

Joy Bhowmick, and V Madhu Vishwanatham*.

VIT, Tamil Nadu, India.

ABSTRACT

Geofencing is one of the key Solution in Location Aware Mobile Technologies. It can be considered as a two-dimensional bounded area. Location Based Peers which are in the province of other peers can be tracked easily with tracking the location of other peers and then using the haversine function to check if that point is within certain distance from the location of the host. But haversine function is not reliable for measurement of very less distance. Calculating every other peer's distance using this function will yield a great deal of computational cost. But here in this paper we are going to build a Peer to Peer Geofencing system which will be used to implement a social communication service. The main challenge of this paper is to provide a framework to have a peer to peer trigger on geo-fences. Also, a parallel implementation to set up Geofences is used to achieve higher throughput Regarding communication service, the transmission of data has to be on a secure channel. To mitigate the security problem in this paper we are going to discuss about a unique digital envelope system that will consist Probabilistic Encryption Algorithm and Secure Hashing Algorithm (SHA) **for** an End-to-End encryption.

Keywords: Android Location Services, Digital Envelope, Geofence, Global Navigational Satellite System, Probabilistic Cryptography

Corresponding author

INTRODUCTION

In the era of mobile based technologies Location Based Services certainly is leaving its trail. Location Aware Technologies are directly associated with the location of the host. Location Aware technology depends on the Position Fix that is detecting the current Latitude and Longitude. This is done using GNSS (Global Navigational Satellite Systems). But except for the Satellite Systems nowadays mobile devices are directly getting Location Data from Assisted GPS (A-GPS) also.

Geofencing is one of the extended Application of Location Aware Technologies. Geofence in simple terms is creating a real-world virtual fence which tests the presence of user's device in the vicinity of the area covered by the perimeter of Geofence. There are three kind of trigger points, whether the host enters into the geofence or exits the Geofence or stays inside the Geofence.

Geofence is of three types. First, Static Geofence, which is set by a single static center in the map and a given radius the virtual circular fence is made up or using multiple points in the map we can set up a bounded region under the geofence. A typical use-case for static Geofence can be to set up a geofence to the destination of a journey and getting notification by entering the province of the destination which will be generated by the geofence.

Second, Dynamic Geofence, where the setting up of Geofence is not based on some predefined points on the map, rather as per events dynamically geofence will be created to trigger notification to host. A typical use-case for dynamic Geofence will be suppose the host wants to find a place for parking, whenever there will be a free zone, dynamically a geofence will be set up on that place, so that if the host is in the province of that geofence a notification will be forwarded to use that parking place.

Third, Peer to Peer Geofence, this is an escalation of Dynamic Geofence, same way as peer to peer network works, here the Geofence will alert a peer of similar application that he or she is in the province of another peer [8].

In this paper setting up peer to peer geofence will be the primary motive. The main agenda is to set up a peer to peer based Social chatting application development. Whenever we consider a social chatting application, security system will be another prioritized module. Here we are going to propose a unique Digital Envelope System which will consist of three algorithms, Probabilistic Encryption Algorithm, Secure Hashing Algorithm (SHA) for one way message digest preparation and for key exchange we will use Elliptic Curve based Diffie-Hellman algorithm. We will also discuss the merits and demerits of our security system from conventional Digital Envelope System made of one Symmetric key encryption and one Asymmetric key encryption algorithm.

MATERIALS AND METHODS

Geofence: A geo-fence is a defined perimeter set up virtually for a real-world geographical region. A geo-fence can be generated with a radius around a store or location, or a geo-fence may be a set of boundaries which is predefined, like a college campus, forest zones or neighborhood boundaries etc.

One of the example of geo-fencing involves a device which can be used to get the current location using “location-based services” entering into or exiting from the virtual perimeter of the geofence. This activity will trigger a Geofence event that will alert the device's user.

Now every geofence event must be triggered by a Location-Aware Device which is somehow getting its co-ordinates or fix of latitude and longitude [1]. There are actually two ways a device can get location fix. It can take help of “Global Navigational Satellite System” or “GNSS” in short. Another one is “Wireless Positioning System” or “WPS”.

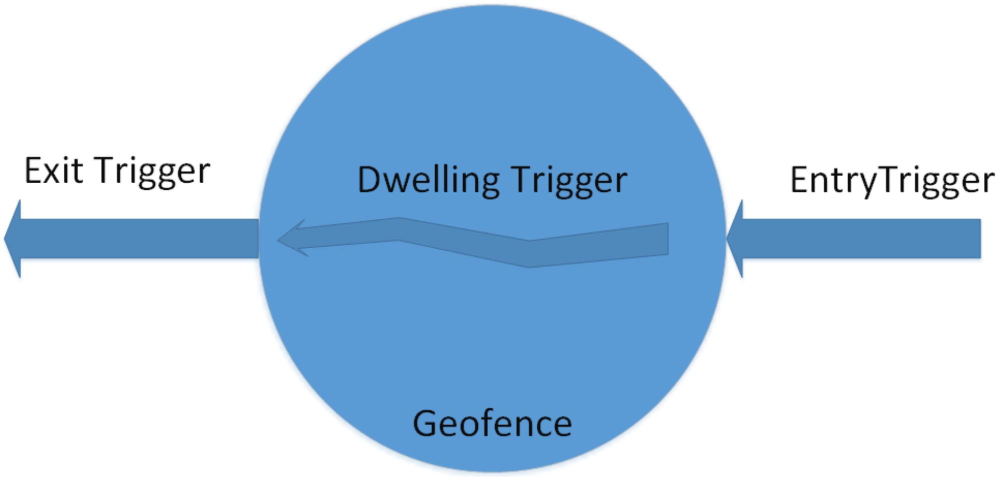

Figure 1: Geofence Triggers

In case of position fixing directly using satellite, we get two kind of messages from it, Almanac and Ephemeris Data. Using this two messages from multiple satellites the device gets the idea about the position of the satellites in their constellation and as per this notion which satellites are in line of sight. After this computation if the device can get at least 4 satellites at line of sight, then as per their messages it calculates the distance from those satellites and get a location fix in real 3D-World.

There are a number of GNSS systems surrounding the Earth such as Global Positioning System (GPS), Global Satellite Navigation System (GLONASS), BeiDou Navigation Satellite System (BDS), Galileo, Compass Indian Regional Navigational Satellite System (IRNSS) etc. These systems were designed with a revolutionary positioning system using triangulation or trilateration. For instance, GPS is consisting of 24 satellites, each of them broadcasts Radio Waves.

Now just imagine you are on a 2-D plane, and you know that you are suppose x unit away from a well-known point A and y unit away from another well-known point B. With this information, you can make two circles with center as those two points each with radius of corresponding distance. Now we all know two circles with different centers, if intersects each other then there will be two intersecting points. So, if you want to calculate your location with this information then you will be either of those two points. If now you get information of a third point C with distance z, then using C as a center making a circle of z radius whichever of those two points is being intersected by these three circles will be your position.

These scenario in real life is handled in 3-D world, where the Radio Frequency information notifies the receiver about the distance information from the satellites, and it calculates taking spheres with radius as that distance centering those satellites, two satellite's sphere will intersect at a circle. Now the third satellite sphere will narrow down our location of two intersecting points of the circle and the third sphere. So in this way we will need at least another satellite that is fourth one to pin point our location.

In order to download the complete GPS messages that is almanac or ephemeris data it takes considerable amount of time, nearly 38secs when the device is in cold start mode. That is why Assisted GPS has come into the picture to reduce the time of getting a location fix. Assisted GPS/GNSS or A-GPS significantly improves the Time-To-First-Fix (TTFF). A-GPS takes the help of cell tower data to enhance the quality of the fix and to be more precise about the location fix [15]. A-GPS has to take help from internet network or carrier network.

In terms of modes of operation, A_GPS can be categorized into two types. Mobile Station Assisted (MSA) and Mobile Station Based (MSB).

Digital Envelope:

In Cryptography "*Symmetric Key*" algorithms are very strong and their complexity is also less, so they take less time to compute the cipher by encrypting even longer plain texts. But the only weakness these algorithms have is its inability to exchange the private key which both of the end user, that is sender and receiver has to have the same key.

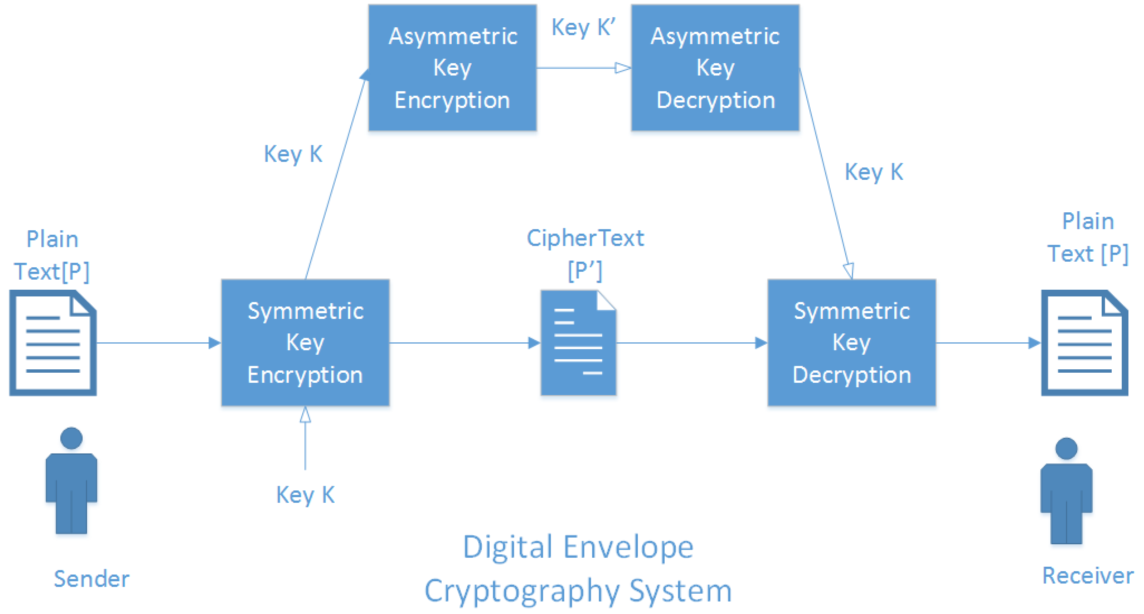

Figure 2: Digital Envelope Cryptography System

Now for this reason "*Asymmetric key*" algorithms have two kind of keys, *public key* and *private key*. Public-key is used by the sender to encrypt the plain text and private key is used by the recipient for decrypting the cipher text. Now though asymmetric key algorithms also known as public key algorithms are not having the problem of key exchanging but its computation relies on modulo arithmetic calculations which takes significant amount of time to complete. So, it is obvious using public key algorithm to encipher long plain texts is not optimal [11].

For this reason, in cryptography digital envelope system which uses both of the above-mentioned techniques that is private key algorithm as well as private key algorithm.

Private-key algorithm is used for long messages and the key used for the private key algorithm is encrypted using public key algorithm.

Probabilistic Encryption Algorithm:

Probabilistic encryption algorithm has a very strong foundation based on the concept of "*quadratic residue*". This algorithm provides encryption which is "Semantically Secure" [12]. Semantically secure means that even if the plain text is known to the cracker as well the cipher text whatever he/she can compute from this information is the same amount of computation without knowing the Plain Text. In order to do this probabilistic encryption scheme provides a ½ probable chance for every return to be "0" or "1". In this way for the same plain text applying this same scheme will generate different ciphers.

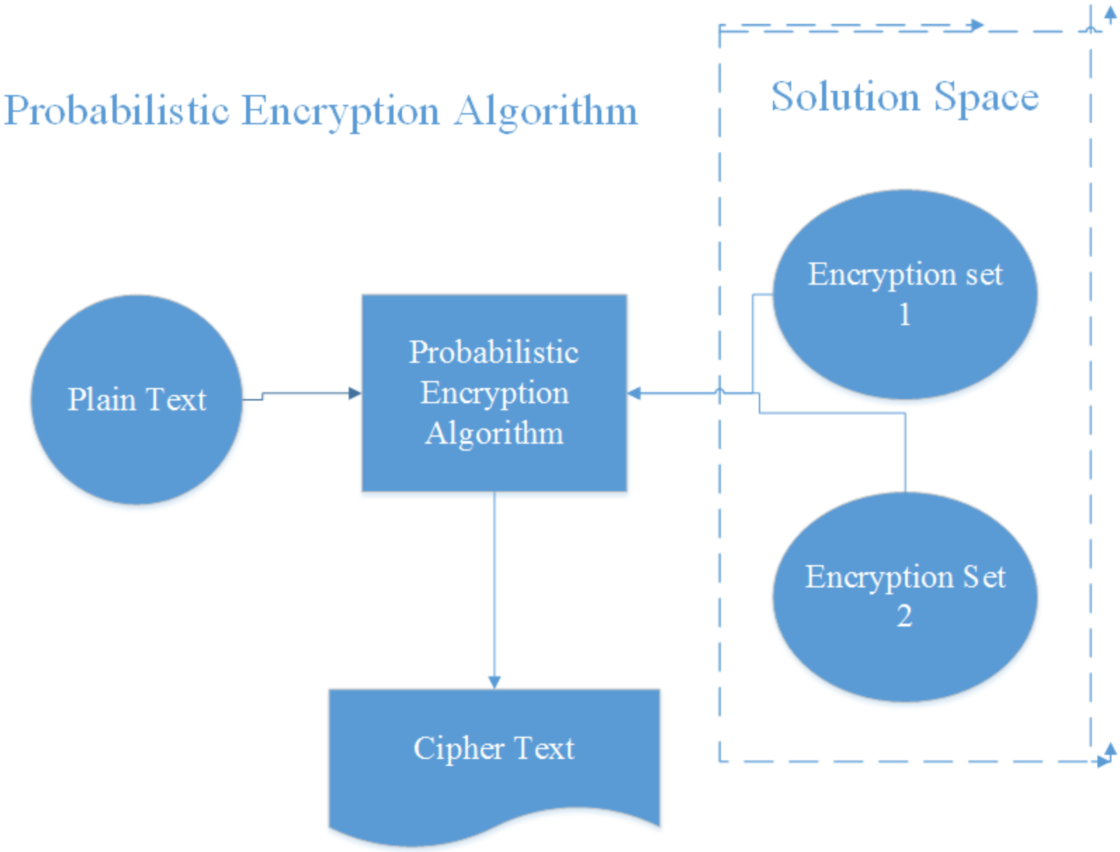
Probabilistic Encryption Algorithm
Solution Space
Plain Text
Probabilistic Encryption Algorithm
Encryption set 1
Encryption Set 2
Cipher Text

Figure 3: Probabilistic Encryption Solution Space

Quadratic Residue can be understood by the following example. Suppose there is one integer x lies within (0, p) such that $x^2 \equiv q \pmod{p}$, if there exists any solution for this congruence then q is considered as the quadratic residue (mod p) and if there does not exist any solution for this congruence then q is considered as quadratic nonresidue.

For instance,

$1^2 \equiv 1 \pmod{8}$
$2^2 \equiv 4 \pmod{8}$
$3^2 \equiv 1 \pmod{8}$
$4^2 \equiv 0 \pmod{8}$
$5^2 \equiv 1 \pmod{8}$
$6^2 \equiv 4 \pmod{8}$
$7^2 \equiv 1 \pmod{8}$

Now as we can see for mod 8
Quadratic residues are 1, 4 (trivial case q= 0 is not considered)
Quadratic nonresidues are 2,3,5,6,7

Now the "*Quadratic Residuosity Problem*" states that, with a sufficiently large p in the previous problem there is no efficient procedure to know q is quadratic residue or not.

Now we take Zn* as a set of numbers less than n and have GCD as 1 with n, that is if q∈Zn* then gcd(q,n)=1.

As per Godwasser-Micali Encryption Scheme:

i) Select two random and significantly (for security) large prime numbers c and d such as $c \neq d$
ii) Set a variable n = cd
iii) Now a pseudo-square has to be selected from Zn
iv) The (n, y) in this procedure will be the public key for encryption and (c, d) will be the private key.

Now for instance the message is binary string m = M1M2M3…Mk

Encryption:
For i=1 to k do:
  i) Select x at random from Zn*
  ii) If Mi = 0 set Ci = x^2 mod n; else set Ci = yx^2
The cipher text is c = (C1C2C3…Ck)

Decryption:
For i=1 to k do:
i) Compute if Ci is Quadratic residue of mod p or not.
ii) Using the prime factorization of p, q it is checked whether Ci is quadratic residue or not. If it is then set Mi=0; else set Mi = 1

The decrypted text is m = (M1M2M3…Mk)

This way while encrypting the plain text the randomness of the x chosen from Zn* makes the cipher a probable set of ciphers for a single plain text [2].

This scheme keeps a noticeable amount of redundancy to the initial message. Each bit of the plain text is encrypted as a bit Ci of Zn. For instance, the expansion of 1 to 1024 bits.

In this algorithm, another noticeable information is that every bit of the message is encrypted completely without any correspondence to the previous computation. So it can be considered more of a stream cipher algorithm, rather than a block cipher.

Godwasser-Micali scheme has an improved version on the basis of the redundancy for each of the bit of the message. The other version (Blum-Goldwaser 1984) reduces the additional redundancy by selecting n that have special characteristics.

Secure Hashing Algorithm:

Secure Hashing Algorithm is used not for encryption but Authentication. In a very simple way to describe Hashing algorithms would be it is a one-way encryption. Hash also known as Digest of the input is received at the output of this kind of algorithm.

This hash is sent along with the message to the other end. Now the receiver again computes the hash from received message and compares it with the received hash. If both the generated Hash and Received hash are same at receiver end then only the received message is accepted.

Even for a single bit change in Plain Text there is a huge change noticed in the generated hash, this is called *Avalanche Effect.*

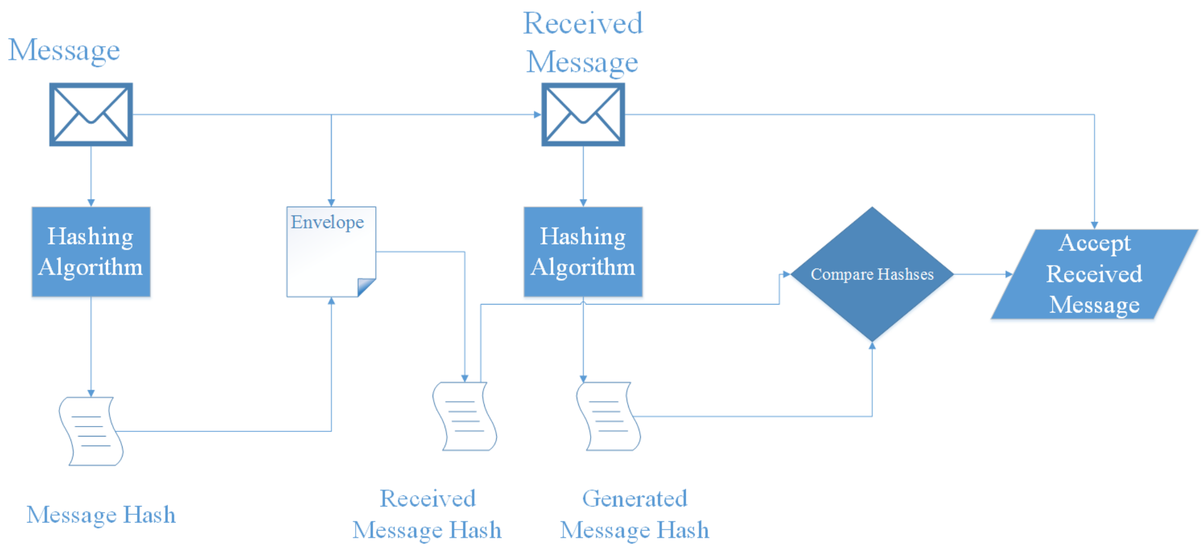

Figure 4: Use of Hashing Algorithm

This methodology ensures the integrity of the message is intact [13].

SHA-512 takes input of 1024 bits blocks and outputs 512 bits of hash. This algorithm has 80 rounds. SHA-512 can handle maximum input of bits $2^{128} – 1$.

If the input bits are not in multiple of 1024 bits it continues to divide it in 1024 bits blocks and then last block will have some padding with it. This blocks will be operated one at a time.

In order to determine the actual length of the message that is without the padding the actual length information is put into the buffer. SHA-512 use these operations on 64 bits words:

Bitwise XOR
Bitwise AND
Bitwise OR
Bitwise complement
Mod 264 addition
Right shift by n bits
Right rotation by n bits

SHA-512 consists of two stages SHA-512 compression function, and the SHA-512 message schedule.

EXPERIMENTAL:

Our main goal in this paper is to put a solution to find peers in the proximity and to communicate with them in a secure way. In order to execute this task, we are using the Google APIs for android.

Google provides an extensive set of APIs for android development. By logging into Google API Client interface the Google APIs for location fix, getting map at map fragment, authentication of signing in all this can be done [9].

In order to have Geofence set up dynamically in this application one database is used to store the location with timestamp of the fix. Now when the user is logging into the application, the first thing being conducted is getting connected to GoogleApiClient and using Fused Location Api the location fix is being obtained. This location fix is being refreshed every 60seconds.

The obtained location fix is being uploaded to the Location Database with its timestamp. Now based on the location we are setting up a Geofence of 200 meters. This is to optimize the purpose of the application, even if the location is being updated with new value, while the Geofence set up with the last location fix as center does not trigger a Geofence exit event it will not be updated to the database. This will ensure that if the user is not having much of a displacement or mobility then there will not be any updates on his location simply because of the periodic retracking of the location fix [6]. But the timestamp at the location database will be updated for the entry every 30 minutes even if there is no location update occurs.

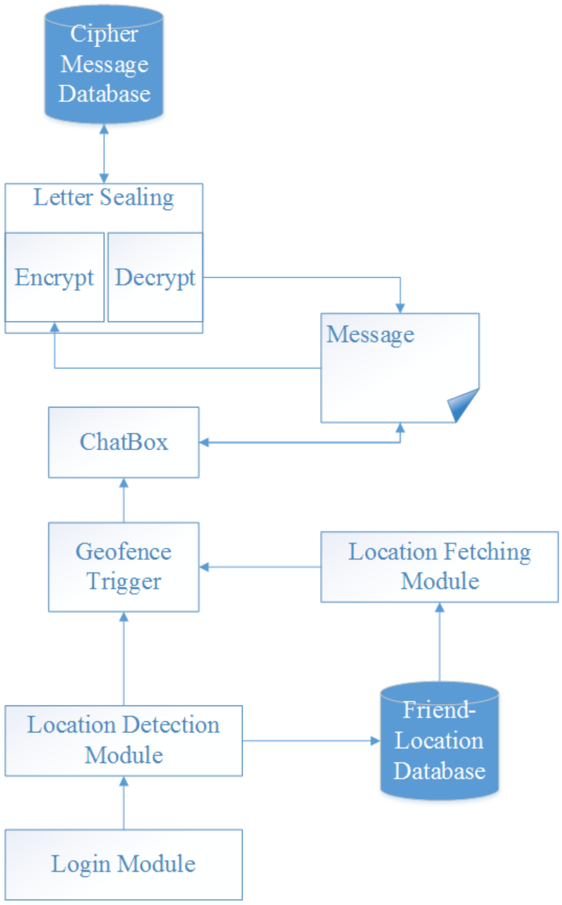

Figure 5: Application Overview

Next, from the Location Database this application will fetch the entries and will check with the timestamps of the entries whether the timestamp is within 30 minutes limit from current time or not. After fetching the location entries from the database it will set up Geofences based on those locations with radius of 800 meters [10].

The data fetching from Friend Location Database, timestamp checking and setting up the Geofence is done using a multi-threaded context so that the Geofence gets initiated almost instantaneously for even huge number of peers.

After the geofences are set up the device will wait for any Geofence trigger. If the device gets any of the Geofence Enter, Exit or Dwell trigger then corresponding geofence id is tracked which is mapped to a peer member.

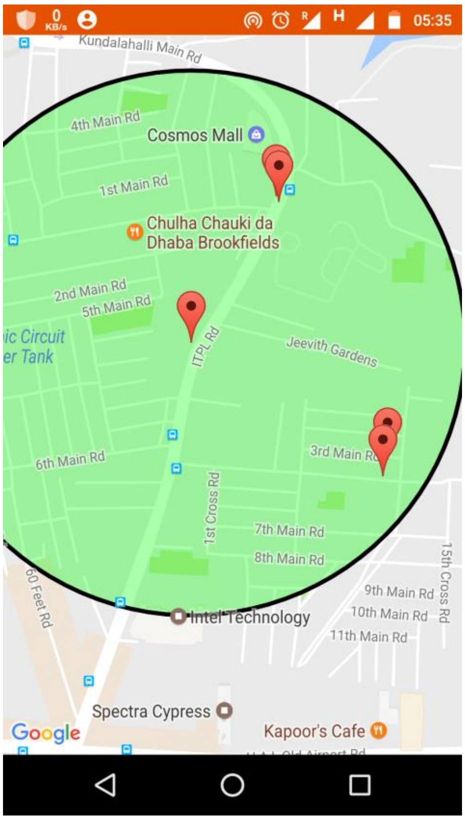

Figure 6: Who's Nearby Screenshot

Every Geofence trigger listening is done by a background service. To optimize the application service for waiting for geofence triggers, after every 15minutes any set geofence will be unset if there is no trigger from that particular geofence.

In case any geofence trigger getting occurs then the application will put up a notification corresponding to that User whose location based geofence trigger an event at this deice. In this way, the user of this device will get to know about the peers of this applications who are in close proximity that is within 800 meters.

Upon tapping on the notification, the user can also see the peers nearby him in the map which is set by Google Map API [4, 7].

The user can also check the routes to the peers and how far is that peer from the user's current location. To get back to the messaging Intent from the Map view this application tracks the intents with a stack [5]. When the user wants to send messages to the peers the security module functionality gets started.

When the user first installs the application in the android device, the user has to sign in the application. While signing in, the application will generate two large prime numbers using the Key generator module. This will be used as the private key for the users end for all the communication.

Whenever any sender wants to initiate a communication with the user first that sender has to obtain the public key of the user. In order to do that the location database entry itself will have the public key associated with each entry.

Now whenever a user gets the Geofence trigger and tries to send a message first it will encrypt that message using the public key obtained from the location entry and encrypt the message using probabilistic encryption scheme. Along with this the message will be passed through SHA-512 to obtain a secure hash of the message.

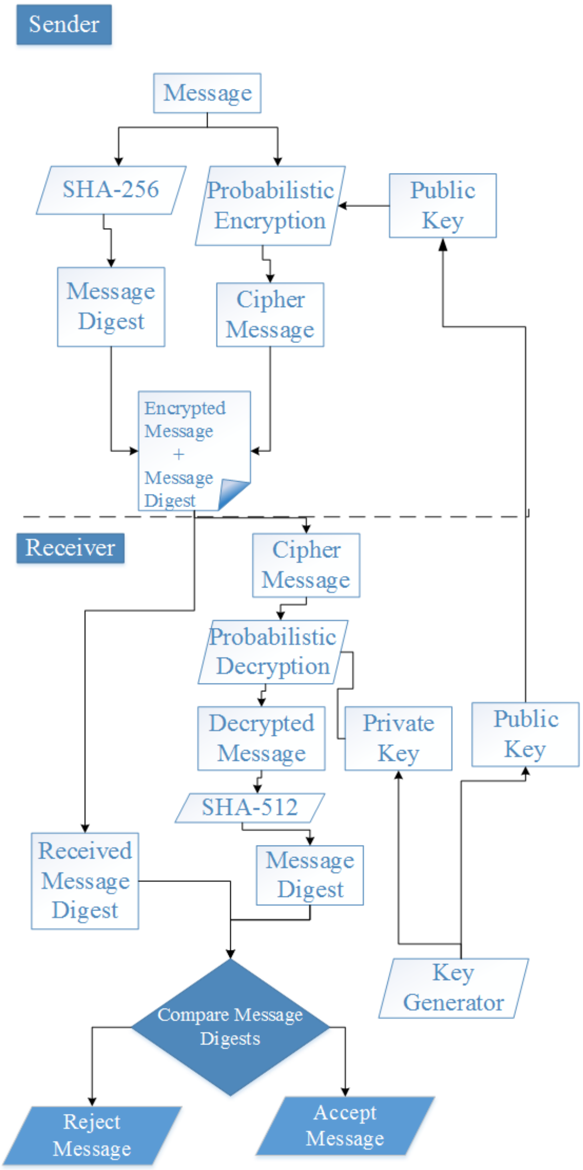

Figure 7: Application Security Module

This secure hash will be attached with the cipher text generated from the encryption algorithm and as a single entry will be put into database of the cipher messages.

The Google Cloud Messaging or GCM client API will be always listening to any database changes made for the user's message database. If it gets any update, the GCM will pull that entry from the database that is the entire envelope including the Encrypted message and Message digest will be downloaded.

Upon receiving a new message the application will first separate out the cipher text from the message digest. For this reason, one data flag will be merged in between this two entities to separate them out from a single buffer.

Next, the cipher message will be passed on to probabilistic decryption process where the user's private key will be used as the key to decipher the content. After the decryption, SHA-512 will be executed on this decrypted message to compute the message digest.

The generated message digest and the received message digest from the envelope will be then compared. If both the digests found to be exactly same then the message will be accepted.

RESULTS AND DISCUSSION

Location based peer to peer application has a number of methodologies to be prepared.

The most naïve approach in such applications may be to get the location fixes from the database and computing the distance from that location to the user's current location.

For instance, suppose one fetched location has latitude and longitude (x1, x2) and the user's current latitude and longitude is (l1, l2).

Actual Distance using cosine law:

acos(SIN(x1)*SIN(l1)+COS(x1)*COS(l1)*COS(l2-x2))*6371

In that case using the "*haversine formula*" the distance can be measured [3].

difLat = (l1-x1)
difLon = (l2-x2)
a = sin(difLat/2) * sin(difLat/2) + cos(x1) * cos(l2) * sin(difLon/2) * sin(difLon/2)
distance = 6371 * 2 * atan2(sqrt(a), sqrt(1-a))

Implementing this solution is easier and computationally less costly. But in this way another bigger problem arises. In case of *haversine* distance computation atan2 is used to measure distance from angle. The tangent of small angle x is the value x itself. Now, as a consequence the inverse tangent of a small value is calculated with approximately no loss in terms of precision of the result [3].

This is the reason that even though haversine formula is equivalent to the cosine law, it is far more greater for small distances (small in the sense 1 meter or less)

Below provided a comparison of haversine formula and cosine function using 100 random location pairs on the globe.

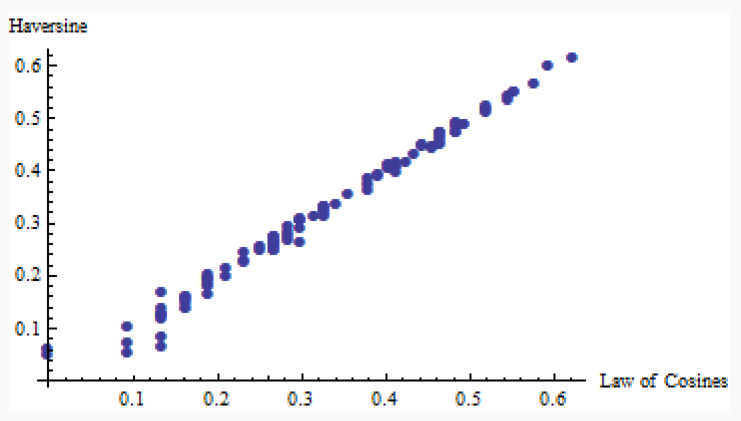

Figure 8: Haversine vs Cosine graph

Even though using Geofence the drawback we get is that we have to rely on the geofence service which will be consistently listening for trigger events and will utilize the CPU and device power, but this is certainly convenient than above approach to be error-free.

The second discussion will be on selection of algorithms for security. Digital Envelope cryptography systems actually was needed to mitigate the shortcoming of public key cryptography's (such as RSA) time complexities. The root of these algorithms on modular arithmetic exponentiation makes it infeasible to encrypt long messages promptly in a block by block manner.

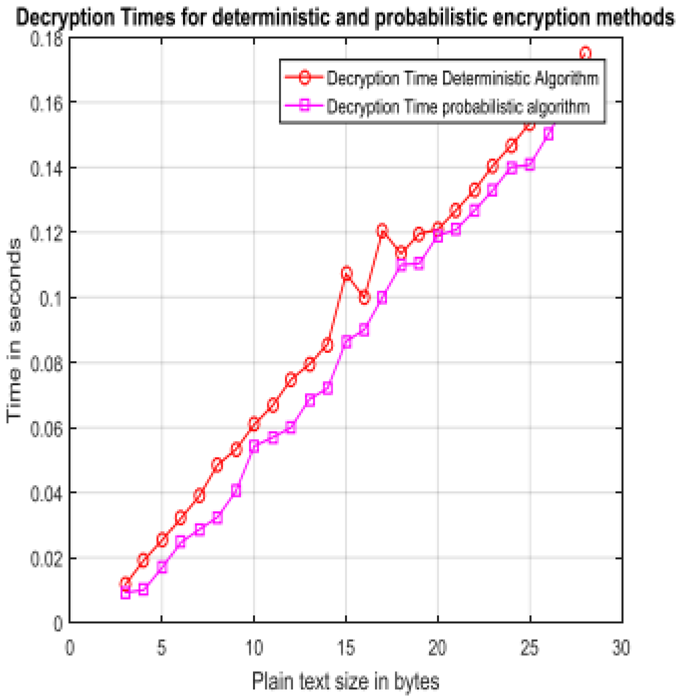

Figure 9: Deterministic Asymmetric key algorithm (RSA) vs Probabilistic Public key algorithm - decryption speed

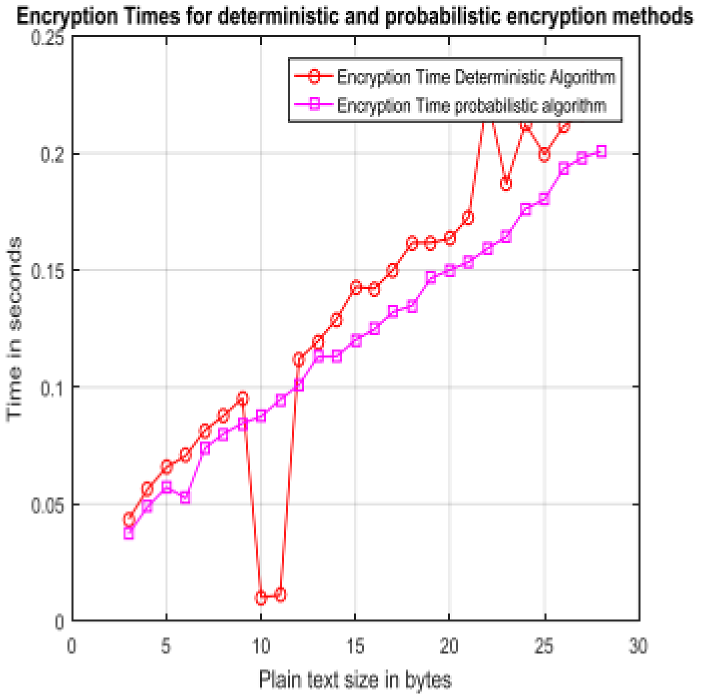

Figure 10: Deterministic Asymmetric key algorithm (RSA) vs Probabilistic Public key algorithm - encryption speed

This is why the long messages are encrypted using symmetric key algorithms and the keys are sent of using the public key algorithms. But probabilistic key algorithm has proven to be much faster than conventional asymmetric key algorithms. The comparison of speed of encryption and decryption of RSA and Probabilistic key algorithm is as per following figures.

Except for this speed property, another advantage of having probabilistic encryption key algorithm to encrypt any message is that the randomness introduced by this algorithm for each bit of the message secures the cryptography system semantically which conventional deterministic symmetric key algorithm (such as DES, IDEA etc.) lacks [14].

CONCLUSION AND FUTURE WORK

The proposed methodology can be improved by optimizing the data fetching from the location database rather than just relying on GCM push. For this purpose Cuckoo Hash searching can be implemented to search through the location database and make another node to download from using GCM. As Cuckoo optimizes the search to O(1) , the relevant location detail downloading will be significantly faster.

Another shortcoming is to set up geofence services to continuously wait to listen for trigger event for geofences. This task can be handled by the server itself, and whenever any geofence event get triggered simply a notification to the corresponding user can be provided to reduce user side computation and increase the productivity of user's device/

ACKNOWLEDGEMENT

The authors are grateful to the authorities of VIT University and School of Computer Science and Engineering, VIT University for all the facilities and research opportunities they have provided.

REFERENCES

[1] Namiot D., Sneps-Sneppe M. (2013) Geofence and Network Proximity. In: Balandin S., Andreev S., Koucheryavy Y. (eds) Internet of Things, Smart Spaces, and Next Generation Networking. Lecture Notes in Computer Science, vol 8121. Springer, Berlin, Heidelberg

[2] Blum, Manuel, and Shafi Goldwasser. "An efficient probabilistic public-key encryption scheme which hides all partial information." Workshop on the Theory and Application of Cryptographic Techniques. Springer Berlin Heidelberg, 1984.

[3] Veness, Chris. "Calculate distance and bearing between two latitude/longitude points using haversine formula in javascript, 2010." URL http://www. movable-type. co. uk/scripts/latlong. html. Fetched January (2012).

[4] Kerr, Michael A., and David Stewart. "User interface for geofence associated content." U.S. Patent No. 9,043,222. 26 May 2015.

[5] Monday, Steven Donald, and Joshua Robert Dalcher. "Geofence system with integrated user interface." U.S. Patent Application No. 12/222,710.

[6] James, Bryan J., and Michael I. Ingrassia Jr. "Geo-Fence Generation and Updating Based on Device Movement Patterns." U.S. Patent Application No. 13/406,406.

[7] Carr, Natalie, and Paul McCullagh. "Geofencing on a mobile platform with alert escalation." International Workshop on Ambient Assisted Living. Springer International Publishing, 2014.

[8] Statler, Stephen. "Geofencing: Everything You Need to Know." Beacon Technologies. Apress, 2016. 307-316.

[9] Kumaresan, Bavya. Android app with GeoFence for Old Town and Related Locations. Diss. San Diego State University, 2016.

[10] Zin, M. S. I. M., et al. "Geofencing-based Auto-Silent Mode Application." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 8.10 (2016): 199-204.

[11] Ganesan, Ramachandran, Mohan Gobi, and Kanniappan Vivekanandan. "A Novel Digital Envelope Approach for A Secure E-Commerce Channel." IJ Network Security 11.3 (2010): 121-127.

[12] Fuchsbauer, Georg J. "An Introduction to Probabilistic Encryption." Osječki matematički list 6.1 (2006): 37-44.

[13] Lakshmanan, Thulasimani, and Madheswaran Muthusamy. "A novel secure hash algorithm for public key digital signature schemes." Int. Arab J. Inf. Technol. 9.3 (2012): 262-267.

[14] Ren, Yonglin, Azzedine Boukerche, and Lynda Mokdad. "Performance analysis of a selective encryption algorithm for wireless ad hoc networks." Wireless Communications and Networking Conference (WCNC), 2011 IEEE. IEEE, 2011.

[15] Saritha, V., and V. Madhu Viswanatham. "An efficient cross layer based channel reservation method for vehicular networks." International Journal of Communication Systems 27.12 (2014): 4249-4264.