Chapter 12
Public Key Cryptosystem for Privacy Sensitive Location-Based Services

K. M. Mahesh Kumar and N. R. Sunitha

Abstract Almost every smartphone and wireless devices are equipped with GPS and other location-enabling technologies, which has enabled users to access location-based services, a popular service offered based on the user's geographical location. In order to get a wide range of location-based services like locating nearby friends and locating nearby places/venues or public places (point of interest), users are forced to reveal their actual location; users are left with no option other than compromise location information causing privacy risk. In this paper, we revisited a protocol proposed by Muhammad N. Sakib and Chin-Tser Huang based on ECC concepts for proximity testing to preserve users location privacy. We made suitable modifications to the existing solution to overcome the false negatives in proximity testing and to reduce the unnecessary communication and computation cost. We have suggested an improvement to enable symmetric key exchange between communicating parties which can be used to securely share the location coordinates to calculate the actual distance between communicating parties. Our scheme withstands triangulation attacks and reveals no information about user's exact location to either service providers or communicating parties or attackers, unless it is revealed by the user himself/herself.

Keywords Elliptic curve cryptography · Location based services · Location privacy · Public key cryptosystem

12.1 Introduction

The smartphones and other wireless devices like tablets and PDA have tremendously grown in the last decade in terms of computation capability and variety of application and services they can support. Location-based services (LBS) is one such application which has gained huge popularity over the recent years; LBS

K. M. Mahesh Kumar (✉) · N. R. Sunitha
Department of CSE, Siddaganga Institute of Technology, Tumakuru, Karnataka, India

N. S. V. Rao et al. (eds.), *Proceedings of International Symposium on Sensor Networks, Systems and Security*, https://doi.org/10.1007/978-3-319-75683-7_12

has been powered by the advances in location-enabling technologies like Global Positioning System (GPS), cell tower-based identification, Internet Protocol (IP) address approximation, and Wi-Fi triangulation. Latest survey reveals that LBS is most popular among the users of social networking applications [6] such as geotagging of photos and videos, check-ins, directory services for nearby places, friend finder, etc. LBS application can be categorized into three main types:

1. Point-of-Interest based (PoI)
2. Friend-Finding (FF)
3. People-Discovery (PD)

PoI applications are used by users to locate nearest places like ATMs, bus station, restaurants, etc. Family and friends can be tracked or located using friend-finder applications; people-discovery applications are useful in case of locating and interacting with new people who are total strangers.

Users using LBS applications are forced to provide access to their location information to the application service provider in order to access the service; this compromises the user's location privacy. Major challenge in LBS application is to preserve the user's location privacy.

12.2 Related Work

The main focus of our paper is to address the problem of location proximity, i.e., dealing with the problem of computing whether user $'A'$ is at a certain distance from user $'B'$ or not; the major challenge here is to preserve the user location information of both user $'A'$ and $'B'$ private to each other and the service provider and just reveal only the proximity and not the actual distance between user $'A'$ and $'B'$.

Disclosing only the proximity rather than the distance between user $'A'$ and $'B'$ helps in preventing external attacks like triangulation effectively. Class of solutions which uses proximity-based approach are referred to as privacy-preserving location-proximity (PPLP) [1, 2, 5, 6, 8–10, 12] protocols. There are several ways in which we can achieve location privacy. Several researchers have used k-anonymity [4, 11], where there exist a set of users and the location of the user is indistinguishable. These solutions focus mainly on hiding the identity of the user rather than location coordinates.

In this paper we try to readdress the issue of location privacy in proximity-based services proposed by Muhammad N. Sakib and Chin-Tser Huang in [7]. We retain the elliptic curve-based proximity test solution provided in [7] and try to make it more efficient.

Contributions Our contributions in this paper are as follows:

- We propose an algorithm and steps to sanitize the GPS coordinates to eliminate false negatives for location proximity.

- We reduce communication and computation cost by eliminating the unnecessary message exchanges suggested by authors in [7].
- We suggest steps to share private key among the communicating parties within the proximity range without incurring overhead.

12.3 Background

12.3.1 Testing Proximity of Users by GPS Coordinates Matching

GPS coordinates are a pair of signed floating-point numbers *(±x, ±y)* which represents latitude and longitude values of the location on the surface of the earth. Say we take two real location values, say $L_A(13.3268, 77.126)$ and $L_B(13.3267, 77.1180)$, by looking at the values we can clearly see that there is partial match among the location coordinates, indicating proximity among the coordinates. Refer to Table 12.1 for details about precision values and proximity range.

12.3.2 Distance Calculation Using GPS Coordinates

We can make use of the following equation to compute the distance between two location coordinates in kilometers.

$$\begin{aligned}\text{Distance} = \text{acos}(&\cos(\text{radians}(90 - \text{lat1})) * \cos(\text{radians}(90 - \text{lat2})) \\ &+\sin(\text{radians}(90 - \text{lat1})) * \sin(\text{radians}(90 - \text{lon2})) \\ &* \cos(\text{radians}(\text{lat1} - \text{lat2}))) \;*\; 6371\end{aligned} \tag{12.1}$$

Example: distance between say $L_A(13.3268, 77.126)$ and $L_B(13.3267, 77.1180)$ using Eq. (12.1) is 876 m.

Table 12.1 Various precision values and corresponding distance ranges

Decimal places	Decimal degrees	N/S or E/W distance at equator	E/W distance at 45 N/S
5	0.00001	1.1132 m	787.1 mm
4	0.0001	11.132 m	7.871 m
3	0.001	111.32 m	78.71 m
2	0.01	1.1132 km	787.1 m
1	0.1	11.132 km	7.871 km
0	1.0	111.32 km	78.71 km

12.3.3 Elliptic Curve Basics

Elliptic curve cryptography is a public key cryptosystem. Generally, an elliptic curve is defined over a finite field consisting of finite points satisfying the below equation:

$$y^2 = x^3 + ax + b \tag{12.2}$$

The equation will be defined over a large finite field denoted by prime number P. Elliptic curve contains numerous points satisfying the elliptic curve along with a special point called point at infinity (Θ).

The following operations are possible on an elliptic curve:

- *Point addition:* adding two points on the curve results in a third point which satisfies the curve.
- *Point multiplication:* multiplying a point on the curve with a scalar (integer) value results in a point which satisfies the curve (i.e., repeated addition of given point, also referred to as point doubling).

12.4 Proposed Work

In this section we revisit the proposed work of Muhammad N. Sakib and Chin-Tser Huang in [7] and propose an algorithm to minimize the false negative results of the existing solution and to reduce the communication and computation overhead for proximity test plus suggest an improvement of private key exchange. The proposed work is as shown in Fig. 12.1 and is divided into two parts:

1. Proximity test
2. ECDHE private key exchange

```
Algorithm-1    Sanitization of GPS coordinates
Purpose:       Avoids false negatives by sanitizing GPS coordinates
Input:         Latitude or longitude value in floating-point format
               $\pm abc.xyz$
Output:        Latitude or longitude value in floating-point format
               $\pm abc.000$ or $\pm abd.000$ or $\pm abc.xyz$
Steps:
  if           $'xyz' == '99*'$ (* indicate any digit between 0 and 9)
    return:      $\pm abd.000$ (where $|abd = abc + 1|$)
  else if      $'xyz' == '00*'$
    return:      $\pm abc.000$
  else
    return:      $\pm abc.xyz$
```

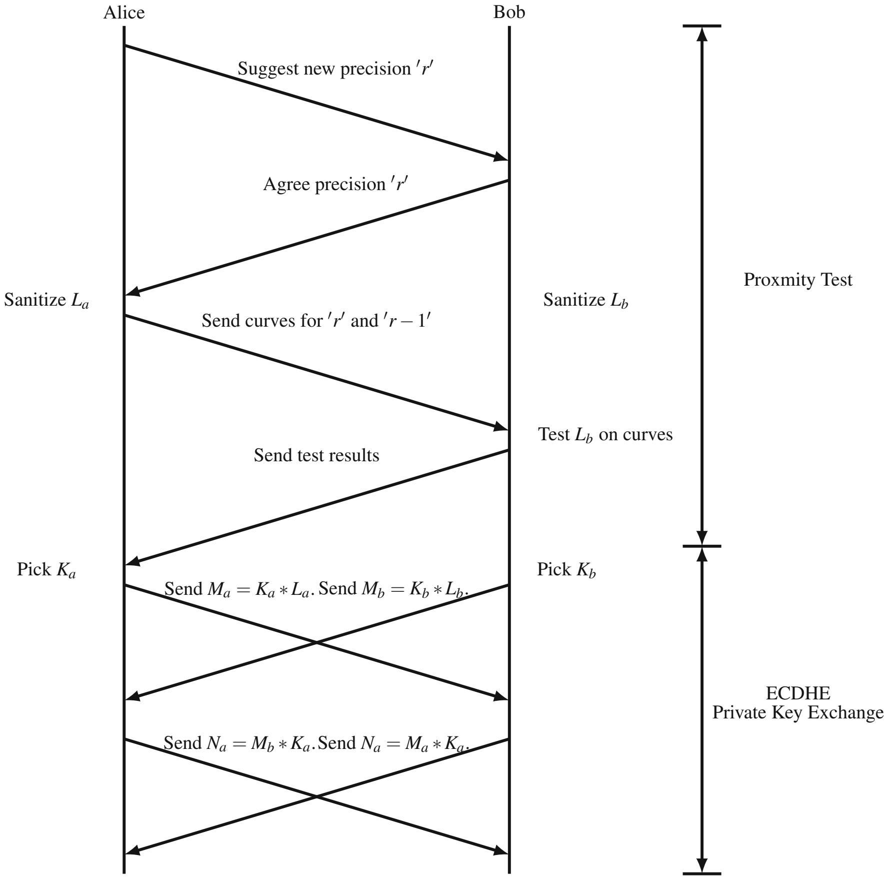
Alice
Bob
Suggest new precision $'r'$
Agree precision $'r'$
Sanitize L_a
Sanitize L_b
Send curves for $'r'$ and $'r-1'$
Test L_b on curves
Send test results
Pick K_a
Pick K_b
Send $M_a = K_a * L_a$. Send $M_b = K_b * L_b$.
Send $N_a = M_b * K_a$. Send $N_a = M_a * K_a$.
Proxmity Test
ECDHE
Private Key Exchange

Fig. 12.1 Proximity test and ECDHE private key exchange between Alice and Bob

12.4.1 Proximity Test

12.4.1.1 Decimal Precision Agreement (Optional)

To begin with, Alice and Bob can agree upon a decimal precision value based upon their proximity search requirement; this is an optional step; we can avoid this if the default value for decimal precision value r is set (generally we use $r = 2$).

12.4.1.2 Sanitization of GPS Coordinates

Sanitization of GPS coordinates is done at both ends of communication, i.e., Alice and Bob sanitize their coordinates independently using the agreed precision value

r and $r-1$ using Algorithm-1, e.g., if the agreed precision value $r = 2$, then the coordinate value $L_A(13.3268, 77.126)$ becomes $L_A(13.32, 77.12)$, since elliptic curves do not deal with floating points, we convert it to whole number by removing decimal points. Final coordinates after sanitization will be $L_A(1332, 7712)$ for $r = 2$ and $L_A(133, 771)$ for precision $r - 1 = 1$.

At Bob's end we introduce additional sanitization steps for precision $r - 1 = 1$ as follows:

- case 1: $L_B(x, y - 1)$ i.e., (original,low)
- case 2: $L_B(x, y + 1)$ i.e., (original,high)
- case 3: $L_B(x - 1, y)$ i.e., (low,original)
- case 4: $L_B(x + 1, y)$ i.e., (high,original)
- case 5: $L_B(x - 1, y - 1)$ i.e., (low,low)
- case 6: $L_B(x - 1, y + 1)$ i.e., (low,high)
- case 7: $L_B(x + 1, y - 1)$ i.e., (high,low)
- case 8: $L_B(x + 1, y + 1)$ i.e., (high,high)

For example, $L_B(133, 771)$ for precision $r - 1 = 1$ becomes as follows:

- case 1: $L_B(133, 770)$
- case 2: $L_B(133, 772)$
- case 3: $L_B(132, 771)$
- case 4: $L_B(134, 771)$
- case 5: $L_B(132, 770)$
- case 6: $L_B(132, 772)$
- case 7: $L_B(134, 770)$
- case 8: $L_B(134, 772)$

12.4.1.3 Elliptic Curve Generation

Using Eq. (12.2), coefficient a, large prime number p, and sanitized values, Alice generates elliptic curves for precision value r and $r - 1$ as follows:

$$b_i = (y_i^2 - x_i^3 - ax_i) \bmod p \tag{12.3}$$

Alice then forwards the curve parameters p, a, b_r, and b_{r-1} to Bob.

12.4.1.4 Elliptic Curve Evaluation

Bob upon receiving the elliptic curve parameters p, a, b_r, and b_{r-1} verifies his sanitized value $L_b(x_b, y_b)$ with precision r by substituting values into Eq. (12.2). Upon successful verification, he replies back with a positive result, else he continues with verification with precision $r - 1$ by substituting values into Eq. (12.2). If the test is successful, he replies back with a positive result, else he continues with verification of $L_b(x_b, y_b - 1)$, $L_b(x_b, y_b + 1)$, $L_b(x_b - 1, y_b)$, $L_b(x_b + 1, y_b)$, $L_b(x_b - 1, y_b - 1)$, $L_b(x_b - 1, y_b + 1)$, $L_b(x_b + 1, y_b - 1)$, and $L_b(x_b + 1, y_b + 1)$ by substituting values into Eq. (12.2), one set of coordinates at a time in case of success, he replies back with a positive result and discontinues the test. If no match found, he replies back with negative result. These test results are sufficient for proximity testing.

We demonstrate our technique using sample values as follows:

12.4.2 ECDHE Private Key (Symmetric Key) Exchange

In [7] they have proposed additional steps of ECDHE for verification of location proximity which we feel is an unnecessary burden; instead the same can be used for symmetric key exchange between Alice and Bob. ECDHE symmetric key exchange steps are as follows:

Alice:

- Selects a secret value K_a randomly
- Computes $M_a = K_a * L_a$ using the verified curve and sanitized coordinate L_a
- Sends M_a to Bob

Bob:

- Selects a secret value K_b randomly
- Computes $M_b = K_b * L_b$ using the verified curve and sanitized coordinate L_b
- Sends M_b to Alice (similar to Alice)

Alice:

- Receives M_b from Bob
- Computes $N_a = K_a * M_b = K_a * K_b * L_b$
- Uses N_a as symmetric key

Bob:

- Receives M_a from Alice
- Computes $N_b = K_b * M_a = K_b * K_a * L_a$
- Uses N_b as symmetric key ($N_a = N_b$ if all steps are correct)

Repeat the above steps with different random secrets to arrive at a different secret key when required.

Alice	Bob
$L_a(37.295, 28.135)$ Sanitize L_a using default precision $r = 2$ and $r - 1 = 1$	$L_b(37.321, 28.138)$ Sanitize L_a using default precision $r = 2$ and $r - 1 = 1$
$L_a(3729, 2813)$ for $r = 2$ and $L_a(372, 281)$ for $r - 1 = 1$ $a = 5$ and $P = 1{,}000{,}003$ in Eq. (12.3) for $r = 2$ $b_2 = 660{,}373$ for $r = 1$ $b_1 = 598{,}409$ send $a = 5,\ P = 1{,}000{,}003,\ b_2 = 660{,}373$ *and* $b_1 = 598{,}409$	$L_b(3732, 2813)$ for $r = 2$ and $L_b(373, 281)$ for $r - 1 = 1$
	Substitute $a = 5,\ P = 1{,}000{,}003,\ b_2 = 660{,}373,\ L_b(3732, 2813)$ in Eq. (12.2) and evaluate the curve $912{,}948 \not\equiv 162{,}264$ (indicates curve does not satisfy) Substitute $a = 5,\ P = 1{,}000{,}003,\ b_1 = 598{,}409,\ L_b(373, 281)$ in Eq. (12.2) and evaluate the curve $78{,}961 \not\equiv 495{,}235$ (indicates curve does not satisfy) Our proposed extension Case 1: substitute $a = 5,\ P = 1{,}000{,}003,\ b_1 = 598{,}409,\ L_b(373, 280)$ in Eq. (12.2) and evaluate the curve $78{,}400 \not\equiv 495{,}235$ (indicates curve does not satisfy) Case 2: substitute $a = 5,\ P = 1{,}000{,}003,\ b_1 = 598{,}409,\ L_b(373, 282)$ in Eq. (12.2) and evaluate the curve $79{,}524 \not\equiv 495{,}235$ (indicates curve does not satisfy) Case 3: substitute $a = 5,\ P = 1{,}000{,}003,\ b_1 = 598{,}409,\ L_b(372, 281)$ in Eq. (12.2) and evaluate the curve $78{,}961 \equiv 78{,}961$ (indicates curve values satisfied for precision 1 for case 3)
$a = 5,\ P = 1{,}000{,}003,\ b_1 = 598{,}409,\ L_a(372, 281)$ in Eq. (12.2)	Send positive result for precision value 1
$78{,}961 \equiv 78{,}961$	Same value at both ends indicate proximity for precision value 1

12.5 Security Analysis

Security of our work relies on the hardness of elliptic curve discrete log problem (ECDLP). Let M is point on the curve and k be a secret integer $N = [k] * M$, given

M and N it is hard to reveal k if the finite field value P is sufficiently large. Please refer [3] for more details about ECDLP.

Our paper has achieved the following four security goals which are listed below (interested readers can refer [7] for proofs and additional information):

1. Alice and Bob can perform proximity verification, in order to know if they are located in a certain distance range.
2. Either Alice or Bob cannot narrow down on each other's single specific location information (i.e., region smaller than the Earth) if they are not in the proximity range. But they get to know they are not within the specified distance.
3. Either Alice or Bob cannot narrow down on each other's single specific location information (i.e., region smaller than the Earth) if they are within the proximity range, except if they are in the eyesight distance.
4. None of the third party (intruder/communication server) can narrow down on single specific region smaller than the Earth where Alice or Bob is located.

Our desired goals are achieved using ECC and the hardness of elliptic curve discrete logarithm problem. Our protocol assumes that both Alice and Bob use the protocol to communicate only about their proximity information. If at all they want to share their location information for calculating distance, they can do so by first exchanging symmetric key using ECDHE private key exchange and exchanging the encrypted coordinates using the symmetric key.

12.6 Conclusion

In our paper we revisited the proposed work of Muhammad N. Sakib and Chin-Tser Huang in [7] and propose an algorithm to minimize the false negative results of the existing solution and to reduce the communication and computation overhead for proximity test plus suggest an improvement of symmetric key exchange for secure communication. Our contributions are as follows:

- We proposed an algorithm plus additional steps to sanitize the GPS coordinates to eliminate false negatives for location proximity.
- We reduced communication and computation cost by eliminating the unnecessary message exchanges suggested by the authors in [7].
- We suggested an improvement to share symmetric key among the communicating parties within the proximity range without adding any overhead to the scheme.

References

1. Freni, D., Vicente, C.R., Mascetti, S., Bettini, C., Jensen, C.S.: Preserving location and absence privacy in geo-social networks. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10. ACM Press, New York (2010). https://doi.org/10.1145/1871437.1871480

2. Hallgren, P., Ochoa, M., Sabelfeld, A.: InnerCircle: a parallelizable decentralized privacy-preserving location proximity protocol. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST). IEEE, New York (2015). https://doi.org/10.1109/pst.2015.7232947
3. Hankerson, D., Menezes, A.: Elliptic Curve Discrete Logarithm Problem, pp. 397–400. Springer, Boston (2011). https://doi.org/10.1007/978-1-4419-5906-5_246
4. Magkos, E.: Cryptographic approaches for privacy preservation in location-based services: a survey. Int. J. Inf. Technol. Syst. Approach **4**(2), 48–69 (2011). http://dx.doi.org/10.4018/jitsa.2011070104
5. Mascetti, S., Freni, D., Bettini, C., Wang, X.S., Jajodia, S.: Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. VLDB J. **20**(4), 541–566 (2010). https://doi.org/10.1007/s00778-010-0213-7
6. Narayanan, A., Thiagarajan, N., Lakhani, M., Hamburg, M., Boneh, D.: Location privacy via private proximity testing. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, CA, 6th February - 9th February 2011 (2011). http://www.isoc.org/isoc/conferences/ndss/11/pdf/1_3.pdf
7. Sakib, M.N., Huang, C.T.: Privacy preserving proximity testing using elliptic curves. In: 2016 26th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, New York (2016). https://doi.org/10.1109/atnac.2016.7878794
8. Šeděnka, J., Gasti, P.: Privacy-preserving distance computation and proximity testing on earth, done right. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS ’14, pp. 99–110. ACM, New York (2014). http://doi.acm.org/10.1145/2590296.2590307
9. Šikšnys, L., Thomsen, J.R., Šaltenis, S., Yiu, M.L., Andersen, O.: A Location Privacy Aware Friend Locator, pp. 405–410. Springer, Berlin (2009). http://dx.doi.org/10.1007/978-3-642-02982-0_29
10. Šikšnys, L., Thomsen, J.R., Šaltenis, S., Yiu, M.L.: Private and flexible proximity detection in mobile social networks. In: 2010 Eleventh International Conference on Mobile Data Management. IEEE, New York (2010). https://doi.org/10.1109/mdm.2010.43
11. Talukder, N., Ahamed, S.I.: Preventing multi-query attack in location-based services. In: Proceedings of the Third ACM Conference on Wireless Network Security, WiSec ’10, pp. 25–36. ACM, New York (2010). http://doi.acm.org/10.1145/1741866.1741873
12. Zhong, G., Goldberg, I., Hengartner, U.: Louis, Lester and Pierre: Three Protocols for Location Privacy, pp. 62–76. Springer, Berlin (2007). http://dx.doi.org/10.1007/978-3-540-75551-7_5