Privacy Preserving for Location-Based IoT Services

Yue Qiu(✉) and Maode Ma

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore, Singapore
QIUY0005@e.ntu.edu.sg, emdma@ntu.edu.sg

Abstract. In recent years, the applications of location-based Internet of Things (IoT) services change the way of people's lives and works. However, these applications may disclose some private location information of users due to lack of privacy protection mechanism, which could result in serious security issues. To protect users' confidential data, an efficient and secure private proximity testing (ESPT) scheme is designed for location-based IoT services to improve the efficiency while maintaining the privacy of the location of the users. The proposed scheme enables a user to query a service provider whether some people are within a given search range without disclosing any private location information of the user. The security analysis and the simulation results demonstrate that the proposed scheme could not only implement a privacy-preserving proximity test, but also has less computational overheads.

Keywords: Bloom filter · Location privacy · Proximity testing
Security

1 Introduction

In recent years, smartphones, tablets and other wearable devices have become ubiquitous due to rapid development of electronic industry. Internet of Things (IoT), which is going to connect everything in the world, has provided an infrastructure to connect those mobile devices to make people be able to access various wireless network services at anytime and anywhere. Among various network services, location-based service (LBS) has attracted considerable attention, which can be applied into various areas of human life such as social networking applications, healthcare services, financial services and etc. Users' locations are the key enabler of the LBS and can be easily measured by various mobile IoT devices using Global Positioning System (GPS). With the locations information of users, the LBS could provide more valuable services, such as private proximity test, which enables a user to get the information whether some people are within a given geometric range.

Although the LBS is a promising application in IoT and can offer more precise and valuable services based on the location information of users, it should be cautiously used due to privacy concerns [1, 2]. However, most of the applications require users to

A.-S. K. Pathan et al. (Eds.): SGIoT 2018, LNICST 256, pp. 36–45, 2019.
https://doi.org/10.1007/978-3-030-05928-6_4

upload their current locations through different IoT devices without strong privacy protection. The leakage of the private location information of users may be maliciously taken to track users, which could lead to severe consequences.

Motivated by the above-mentioned privacy concerns, many privacy-preserving proximity tests have been proposed to protect users' locations. A private proximity test using private equality testing (PET) and location tags has been proposed in [3]. To improve the efficiency, a vectorial private equality testing (VPET) with an untrusted server has been proposed based on linear algebra in [4]. Another privacy-preserving proximity test with an untrusted server using ElGamal encryption has been proposed in [5] to allow users to verify the correctness of test result instead of a LBS server. A private proximity based location sharing scheme, named Near-pri, using Paillier encryption has been proposed in [6] to allow users to maintain their own security policy. Compared to Near-pri, a more efficient and privacy-preserving proximity testing with differential privacy techniques (EPPD) has been proposed in [7]. Two schemes to support location based handshakes and private proximity testing with location tags using Bloom filter and Bose-Chaudhuri-Hocquenghem (BCH) code have been proposed in [8, 9]. While most schemes only support one coordinate system, a privacy-preserving distance computation and proximity testing has been proposed to support three different coordinate systems on earth in [10]. Although the above-mentioned proximity testing can protect the information of users' locations, most of them only supports proximity testing between two parties, which may result in a high computational overhead when deployed in the real world.

In this paper, as our major contribution, an efficient and secure private proximity testing (ESPT) with location tags is proposed for location-based IoT applications to improve the efficiency while maintaining privacy of the location information of users. The users' location information uploaded to an untrusted third party will be protected by differential privacy techniques for the LBS [7, 11]. The private proximity testing is performed based on the cloaking location tags, a Bloom filter [12] and a secure dot product protocol [13] without disclosing users' confidential data to adversaries. The security functionality of the proposal is evaluated by formal verification and its efficiency is demonstrated by the simulation results. It can be shown that the ESPT scheme could not only preserve privacy, but also incur less computational overhead.

The remainder of this paper is organized as follows. In Sect. 2, the system model is introduced. In Sect. 3, the proposed scheme is described in details. The security analysis and the performance evaluation of the proposed scheme is presented in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 System Model

The system model under this study is shown in Fig. 1, including many registered users, a service provider and a trusted authority.

- User: Each user should be registered before accessing the services. After registration, the registered users need to periodically upload the information of their current locations through mobile IoT devices. The location information is encrypted and stored in the service provider's database. A user can initiate a request for searching other users within a fixed search area by using the proposed ESPT scheme.
- Service Provider (SP): The service provider is responsible for storing users' encrypted location information and providing secure proximity testing based on users' locations and search range. The SP is an honest-but-curious entity, which means that although it operates through the defined steps of the protocol, it may be curious about the private information of locations of the users.
- Trusted Authority (TA): The TA, which is responsible for system initialization and assigning key materials to registered users and the SP, is fully trusted by other entities.

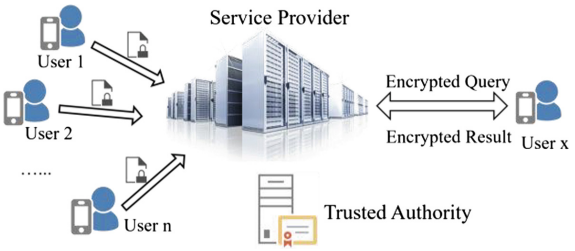


Fig. 1. The system model

3 Proposed ESPT Scheme

In this section, based on differential privacy techniques for location-based services [7, 11] including a bloom filter [12] and a secure dot product protocol [13], an efficient and secure private proximity testing (ESPT) scheme is proposed. It consists of three phases: system initialization phase, user location uploading phase and privacy-preserving proximity testing phase.

3.1 System Initialization Phase

Before accessing the services provided by the SP, the TA is responsible for generating public parameters and assigning keys to the users and the SP in the following steps:

- Given a security parameter k_1, a large prime p is selected, such that $|p| = k_1$. A cyclic group G of prime order p, in which the discrete logarithm problem (DLP) is hard, is generated. The TA also generates a random generator $h \in G$ and chooses hash functions $H : \{0,1\}^* \rightarrow G$.

- Given the parameter $\epsilon \in \mathbb{R}^+$, the TA transfers the map into a map with many grids. The side length μ of a grid depends on the privacy level used by the system. When the side length becomes longer, the privacy level increases. Each grid has a grid tag, such as $tag1_{i,j} = (i, j)$ and each grid is divided into several small grids. The side length of them is l and area tag is $tag2_{i',j'} = (i', j')$ as shown in Fig. 2. The location (x, y) of a user can be mapped into a tag as $tag_{User} = \left(tag1_{i,j}, tag2_{i',j'}\right)$.
- The TA assigns identities to each registered user and the SP, and publishes a parameter list <G, h, p, H, μ, l, ϵ> and area information. Based on the public parameters, the SP selects a random number $sk_{SP} \in \mathbb{Z}_p^*$ as a private key and compute $pk_{SP} = h^{sk_{SP}} mod\, p$. as the corresponding public key. Each user U_i chooses a private key $sk_i \in \mathbb{Z}_p^*$ and computes $pk_i = h^{sk_i} mod\, p$ as his public key.

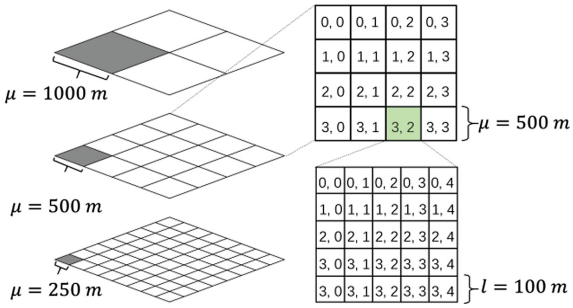


Fig. 2. Grid maps and location tags

3.2 User Location Uploading Phase

Before uploading the location information of the users to the SP, the users need to convert their GPS locations into the form of xy-coordinates in the Cartesian plane by the Miller Projection, which is similar as the procedure in the EPPD scheme in [7]. To protect the information of the current locations of users from being tracking by the adversaries, each user $U_i \in U$ periodically uploads his/her cloaking location data to the SP. The detailed steps of the user location uploading phase are described as follows.

- According to geo-indistinguishability [11], U_i computes a cloaking location coordinates (cx_i, cy_i) based on U_i's current location coordinates (x_i, y_i) as $cx_i = x_i + r\cos\theta$ and $cy_i = y_i + r\sin\theta$, where $r = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right)$, $\theta \in [0, 2\pi)$ and $p \in [0, 1)$. After that, U_i maps his/her current location (x_i, y_i) into area tags $tag_x = \left(tag1_{i,j}, tag2_{i',j'}\right)$, and maps the cloaking location (cx_i, cy_i) into a cloaking location tag $ctag1_{U_i}$.
- U_i calculates a symmetric key established with the SP as $k_{i \leftrightarrow SP} = H\left(pk_{SP}^{sk_i} \| ID_{SP} \| ID_i \| H(tsp)\right)$, where tsp is a timestamp. The cloaking tag is encrypted as $CT_i = ENC_{k_{i \leftrightarrow SP}}(ctag1_{U_i} \| ID_{SP} \| ID_i \| tsp)$ and will be sent to the SP in the message <CT_i||ID_i||tsp>.

- Upon receiving the uploaded location tag from a user, the SP checks the timestamp and calculates the symmetric key $k_{SP\leftrightarrow i} = H\left(pk_i^{sk_{SP}} \| ID_{SP} \| ID_i \| H(tsp)\right)$. CT_i can be successfully decrypted if $k_{SP\leftrightarrow i}$ is correctly computed. The SP stores the information $<ctag1_{U_i}, ID_i, tsp>$ in its database which will be further used in the private proximity testing.

3.3 Private Proximity Testing Phase

During the private proximity testing phase, a user U_x sends a test request to the SP including the location tag and a search area. The detailed steps are described as follows.

- Step 1: If a user U_x would like to perform a private proximity testing, U_x needs to set a search area SA which can be a rectangle area or an arbitrary area which consists of different location tags. Then, a bloom filter is generated with the length of m, and the number of hash functions is k. U_x inserts the location tags to the bloom filter within the SA. It inserts location tags $tag_x = \left(tag1_{xi,xj}, tag2_{xi',xj'}\right)$ into $BF_x(SA) = \left(b_{x,1}, \ldots, b_{x,m}\right)$. Then, the $BF_x(SA)$ is encrypted by the secure dot product protocol. U_x chooses two large prime numbers q and α, where $|q| = k_2$ and $|\alpha| = k_3$. A large random number $s \in Z_q$ and $m+3$ random numbers c_i $(i = 1, .., m+3)$ with $|c_i| = k_4$ are chosen. To encrypt the bloom filter value of $BF_x(SA)$, a vector is created as $\overrightarrow{u_x} = (BF_x(SA), -1) = \left(b_{x,1}, \ldots, b_{x,m}, -1\right) = (u_1, \ldots, u_{m+1})$. An encrypted vector $\overrightarrow{C_x} = (C_1, \ldots, C_{m+3})$ can be computed as:

$$C_i = \begin{cases} s(u_i \cdot \alpha + c_i) \bmod p, & u_i \neq 0 \\ s \cdot c_i \bmod p, & u_i = 0 \end{cases} \tag{1}$$

  for each element $u_i (i = 1, \ldots, m+3)$, where $u_{m+2} = u_{m+3} = 0$. The cloaking search area SA' centered at the $ctag1_{U_x}$ is sent to the SP, which can be the same size as the real search area or bigger than it. U_x computes a session key as $k_{x\leftrightarrow SP} = H\left(pk_{SP}^{sk_x} \| ID_{SP} \| ID_i \| H(tsp_x)\right)$, encrypts the vector as $CT_x = ENC_{k_{x\leftrightarrow SP}}\left(H\left(q \| \alpha \| \overrightarrow{C_x}\right) \| ID_{SP} \| ID_x \| SA' \| tsp_x\right)$, and sends the test request $<q \| \alpha \| \overrightarrow{C_x} \| CT_x \| ID_{SP} \| ID_x \| SA' \| tsp_x>$ to the SP and keep $s^{-1} \bmod q$ secret.
- Step 2: When receiving the request from U_x, the SP first checks if the time interval is below the defined threshold using the timestamp tsp_x and the current time. The SP also computes the session key as $k_{SP\leftrightarrow x} = H\left(pk_x^{sk_{SP}} \| ID_{SP} \| ID_x \| (tsp_x)\right)$, decrypts the received ciphertext as $DEC_{k_{SP\leftrightarrow x}}(CT_x) = H\left(q \| \alpha \| \overrightarrow{C_x}\right) \| ID_{SP} \| ID_i \| SA' \| tsp_x$ and validates the hash value using the received information. According to the U_x's uploaded location tag $ctag_{U_x}$, the SP sends the encrypted vector $\overrightarrow{C_x}$, q and α to users U^* whose location area is within or intersects the search area SA', and users whose location tags are neighbors to U^*, as shown in Fig. 3.

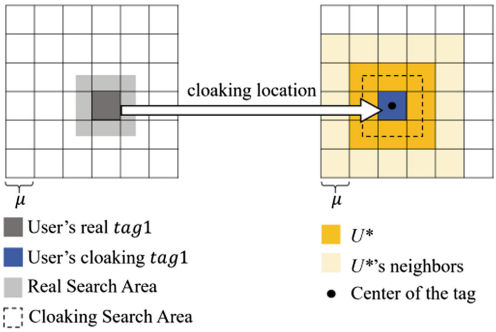


Fig. 3. Private proximity test

- Step 3: When the user U_y receives the $\overrightarrow{C_x}$, U_y creates a bloom filter with the length and the hash functions to be same as $BF_x(SA)$. The location tag $tag_y = \left(tag_{yi,yj}, tag_{yi',yj'}\right)$ is the only element inserted into the bloom filter $BF_y\left(tag_y\right)$. To calculate the dot product, a vector is created as $\overrightarrow{v_y} = \left(BF_y\left(tag_y\right), k\right) = \left(b_{y,1}, \ldots, b_{y,m}, k\right) = (v_1, \ldots, v_{m+1})$. U_y compyutes a encrypted vector $\overrightarrow{D_y}$ based on the $\overrightarrow{C_x}$ as:

$$D_i = \begin{cases} v_i \cdot \alpha \cdot C_i \, mod \, q, & v_i \neq 0 \\ r_i \cdot C_i \, mod \, q, & v_i = 0 \end{cases} \tag{2}$$

  for each element $v_i (i = 1, \ldots, m+3)$, where r_i ($|r_i| = k_5$) is a random number, and $v_{m+2} = v_{m+3} = 0$. After that, U_y calculates the sum of D_i as $D_y = \sum_{i=1}^{m+3} D_i \, mod \, p$, and sends a message $<D_y \| CT_y \| ID_{SP} \| ID_y \| tsp_y>$ to the SP, where $CT_y = ENC_{k_{y \leftrightarrow SP}}\left(H\left(D_y\right) \| ID_{SP} \| ID_y \| tsp_y\right)$ and the session key $k_{y \leftrightarrow SP} = H\left(pk_{SP}^{sk_y} \| ID_{SP} \| ID_y \| H\left(tsp_y\right)\right)$.
- Step 4: After receiving all the response messages, the SP sends a result message $<D_1 \| \ldots \| D_n \| ID_1 \| \ldots \| ID_n \| CT_{SP} \| ID_{SP} \| ID_x \| tsp_{SP}>$ to U_x, where $CT_{SP} = ENC_{k_{SP \leftrightarrow x}}(H(D_1 \| \ldots \| D_n \| ID_1 \| \ldots \| ID_n) \| ID_{SP} \| \, ID_x \| tsp_{SP})$ and a session key $k_{SP \leftrightarrow x} = H\left(pk_x^{sk_{SP}} \| ID_{SP} \| ID_x \| H(tsp_{SP})\right)$.
- Step 5: U_x computes the session key $k_{x \leftrightarrow SP} = H\left(pk_{SP}^{sk_x} \| ID_{SP} \| ID_x \| H(tsp_{SP})\right)$ to decrypt the message from the SP. After successfully obtaining $D_1, \ldots, D_n$, U_x tries to exract the dot products by using $\overrightarrow{u_x} \cdot \overrightarrow{v_j} = \sum_{i=1}^{m+3} \left(u_{xi} \cdot v_{ji}\right) = \frac{E_{xj} - \left(E_{xj} mod \, \alpha^2\right)}{\alpha^2}$, where $E_{xj} = s^{-1} \cdot D_j mod \, p$ $(1 \leq j \leq n)$. If $\overrightarrow{u_x} \cdot \overrightarrow{v_j} = 0$, U_j is within the real search area SA.

4 Security and Performance Evaluation

In this section, the security and privacy properties of the proposed ESPT scheme is analyzed first by using Automated Validation of Internet Security Protocols and Applications (AVISPA). The AVISPA [14] is a formal verification tool that can automatically validate the network security protocols and applications. There are three basic roles in the private proximity testing phase of the ESPT, which are the user U_x who sends a test request, the SP and users U_i who are within the search area. The security goal defined in AVISPA is that the bloom filter value is kept secret during the test, and the participants in the protocol can achieve mutual authentication. The proposed ESPT protocol is analyzed by on-the-fly model checker (OFMC). The intruder model used in the AVISPA is the Dolev-Yao intruder model [15, 16]. The intruder initially has all the public information, its own public/private key pair and its own identity. As shown in Fig. 4, the output result demonstrates that the ESPT is safe under the goals as specified.

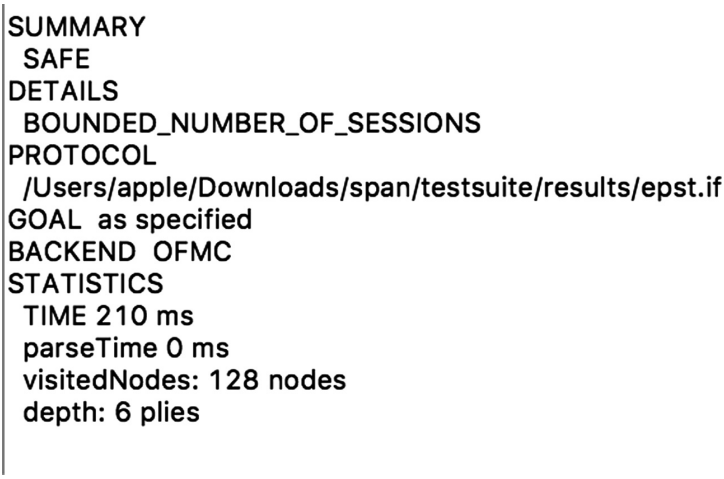
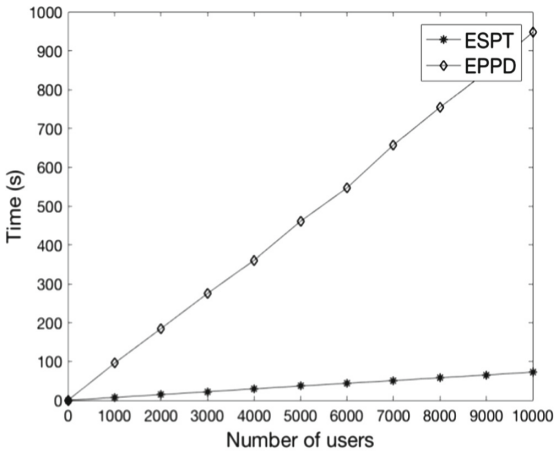


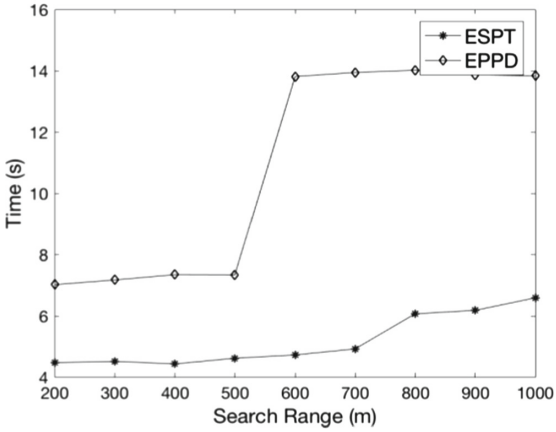
Fig. 4. AVISPA result

Then, the computational overhead is analyzed by JAVA and conducted on Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz, 16 GB Ram, Windows 7 (64-bit). The applied cryptographic algorithms employed in the simulations are hash function, Secure Hash Algorithm-256 (SHA-256), and Advanced Encryption Standard-Cipher Block Chaining-256 (AES-CBC-256). In our simulations, users are uniformly deployed in an area of 20 km $*$ 20 km. The parameters used in the simulations are $\epsilon = 2$, $\mu = 500$, $l = 100$, $k_1 = 1024$, $k_2 = 512$, $k_3 = 200$, $k_4 = 128$, and $k_5 = 128$.

To compare the efficiency of our proposed ESPT scheme with the EPPD scheme, the average computational cost in the user location uploading phase with different numbers of users is presented in Fig. 5(a). The EPPD scheme needs to calculate and upload the cloaking coordinate of each user while the ESPT scheme only uploads the



(a). Average Computation Cost of User Location Uploading Phase



(b). Average Computation Cost of a Private Proximity Test

Fig. 5. Performance evaluation

users' location tag to the SP, which can largely save the computational time in the location uploading process. The average computational time of a single user by the ESPT scheme is about 7.27 ms, which shows that the ESPT scheme is more efficient than the EPPD scheme.

If the size of the Bloom filter is fixed, the number of false positives is proportional to the number of data inserted. To save the computational time, the size of the Bloom Filter m is dynamic with a given probability of false positives p in the ESPT scheme, which can be computed as $m = -\frac{n \ln p}{(\ln p)^2}$, and the number of hash functions k is computed as $k = -\log_2 p$, where n is the number of inserted elements. In the private proximity testing phase, the total number of users is set to 10000 and the probability of

false positives p is set to be 0.01. A user sends private proximity test requests to the SP with different search range. The advantage of the EPPD scheme is that the search radius can be an arbitrary length. Although the ESPT scheme can only search in units of location tags, the search area can be different shapes. For the comparison purpose, the search range of the EPPD scheme is set as the radius of a circle area while the search range of the ESPT scheme is nearly half of the side length of a square area. As shown in Fig. 5(b), the private proximity testing by the proposed ESPT scheme is much more efficient than that of the EPPD scheme by using the Bloom filter and secure dot product protocol.

5 Conclusion

In this paper, the ESPT algorithm is proposed to protect the location information of users for the LBS of IoT applications. The proposed scheme enables a user to query a SP whether some persons are within a given search range without any location privacy disclosure. The security analysis proves that the ESPT scheme can preserve location privacy for all of the users. The performance evaluation performed by JAVA demonstrates that the proposed scheme could not only implement the privacy-preserving proximity testing, but also significantly improves the efficiency.

Acknowledgment. We appreciate the financial support from Ministry of Education, Singapore through the Academic Research Fund (AcRF) Tier 1 for the project of RG20/15.

References

1. Prigg, M.: Privacy warning over app that can track your location even if you turn GPS off on your phone. http://www.dailymail.co.uk/sciencetech/article-5134219/App-track-location-turn-GPSoff.html. Accessed 10 Dec 2017
2. Chong, Z.: Obike becomes latest victim of global data breach. https://www.cnet.com/news/yellow-bike-sharing-firm-is-new-victim-of-global-data-breach/. Accessed 12 Dec 2017
3. Narayanan, A., Thiagarajan, N., Lakhani, M., Hamburg, M., Boneh, D.: Location privacy via private proximity testing. In: Proceedings of NDSS 2011 (2011)
4. Saldamli, G., Chow, R., Jin, H., Knijnenburg, B.P.: Private proximity testing with an untrusted server. In: Proceedings of 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks 2013, WISEC 2013, pp. 113–118. ACM (2013)
5. Zhuo, G., Jia, Q., Guo, L., Li, M., Fang, Y.: Privacy-preserving verifiable proximity test for location-based services. In: Proceedings of IEEE Global Communications Conference 2015 (GLOBECOM 2015), pp. 1–6. IEEE, USA (2015)
6. Novak, E., Li, Q.: Near-PRI: private, proximity based location sharing. In: Proceedings of IEEE INFOCOM 2014, pp. 37–45. IEEE, Canada (2014)
7. Huang, C., Lu, R., Zhu, H., Shao, J., Alamer, A., Lin, X.: EPPD: efficient and privacy-preserving proximity testing with differential privacy techniques. In: Proceedings of IEEE International Conference on Communications 2016 (ICC 2016), pp. 1–6. EEE, Malaysia (2016)

8. Zheng, Y., Li, M., Lou, W., Hou, Y.T.: SHARP: private proximity test and secure handshake with cheat-proof location tags. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 361–378. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33167-1_21
9. Zheng, Y., Li, M., Lou, W., Hou, Y.T.: Location based handshake and private proximity test with location tags. IEEE Trans. Dependable Secure Comput. **14**(4), 406–419 (2017)
10. Sedenka, 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 2014 (ASIA CCS 2014), pp. 99–110 (2014)
11. Andr´es, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS 2013), pp. 901–914 (2013)
12. Wang, B., Li, M., Wang, H.: Geometric range search on encrypted spatial data. IEEE Trans. Inf. Forensics Secur. **11**(4), 704–719 (2016)
13. Lu, R., Zhu, H., Liu, X., Liu, J.K., Shao, J.: Toward efficient and privacy-preserving computing in big data era. IEEE Netw. **28**(4), 46–50 (2014)
14. The AVISPA Team: AVISPA v1.1 User Manual. http://www.avispa-project.org/package/user-manual.pdf. Accessed 21 Aug 2017
15. Viganò, L.: Automated security protocol analysis with the AVISPA tool. Electron. Notes Theoret. Comput. Sci. **155**, 61–86 (2006)
16. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)