JES
Journal of Engineering Sciences

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

PASPORT: A SECURE AND PRIVATE LOCATION PROOF GENERATION AND VERIFICATION FRAMEWORK

[1]Kanikelli Anvesh Kumar, [2]Mr.K.Vaddi Kasulu
[1]M.Tech [CSE], Eluru college of Engineering and Technology, Duggirala - 534004, Andhra Pradesh.
[2]Professor, Eluru college of Engineering and Technology, Duggirala - 534004, Andhra Pradesh.

Abstract— Recently, there has been a rapid growth in location-based systems and applications in which users submit their location information to service providers in order to gain access to a service, resource, or reward. We have seen that in these applications, dishonest users have an incentive to cheat on their location. Unfortunately, no effective protection mechanism has been adopted by service providers against these fake location submissions. This is a critical issue that causes severe conse- quences for these applications. Motivated by this, we propose the Privacy-Aware and Secure Proof Of pRoximiTy (PASPORT) scheme in this article to address the problem. Using PASPORT, users submit a location proof (LP) to service providers to prove that their submitted location is true. PASPORT has a decentralized architecture designed for ad hoc scenarios in which mobile users can act as witnesses and generate LPs for each other. It provides user privacy protection as well as security properties, such as unforgeability and on transferability of LPs. Furthermore, the PASPORT scheme is resilient to prover–prover collusions and significantly reduces the success probability of Prover–Witness collusion attacks. To further make the proximity checking process private, we propose P-TREAD, a privacy-aware distance bounding protocol and integrate it into PASPORT. To validate our model, we implement a prototype of the proposed scheme on the Android platform. Extensive experiments indicate that the proposed method can efficiently protect location-based applications against fake submissions.

Index Terms— Distance bounding (DB), location privacy, location proof (LP) location-based services (LBSs).

I. INTRODUCTION

THE recent advances in the smartphone technology and positioning systems has resulted in the emergence of a variety of location-based applications and services [1]–[3], [48], such as activity-tracking applications, location-based services (LBSs), database-driven cognitive radio networks (CRNs), and location-based access control systems. In these applications, mobile users submit their position data to a location-based service provider (LBSP) to gain access to a service, resource, or reward. These applications are very popular due to the useful services they offer. According to recent business reports, the market value of LBSs was U.S. $20.53 billion in 2017 and is anticipated to reach U.S. $133 billion in 2023, with an expected annual growth rate of 36.55% [4]. However, LBSPs are vulnerable to location spoofing attacks since dishonest users are incentivized to lie about their location and submit fake position data [5]–[9].

Now, we present some examples to highlight the relevant issues in these applications. In the current online rating and review applications, users' real location is not verified, which enables them to submit fake positive or negative reviews for their own business or their rivals [10], [11]. Furthermore, in CRNs [6], [8], [16], malicious users can submit fake locations to the database to access channels that are not available in their location. In location-based access control applications [18]–[20], attackers can gain unauthorized

access to a system or resource by submitting fake location claims. In activity-tracking applications, insurance companies may offer health insurance plans in which customers are offered discounts if they have a minimum level of physical activity [7], [12]–[15]. This creates an incentive for dishonest users to cheat on their location data. Thus far, with these examples, it is clear that preventing fake location submissions in these applications is still an open challenge. To protect these applications against location spoofing attacks, a number of location proof (LP) schemes have been proposed. Using these mechanisms, a mobile device (called a prover in the literature) receives one or more LPs from its neighbor devices when it visits a site. The prover then submits the received LPs to the LBSP as a location claim. The LBSP checks the submitted LPs and either accepts or rejects the user's claim. LP schemes is categorized into two groups depending on the system architecture: centralized or distributed. In the centralized mechanisms [21]–[24], a trusted wireless infrastructure [such as a WiFi access point (AP)] is employed to generate LPs for mobile users. In distributed schemes [25]–[30], mobile users act as witnesses and generate LPs for each other. The latter approach is useful for scenarios in which there is no wireless infrastructure at the desired locations or it is expensive to employ a large number of APs for different locations.

In our extensive literature review and to the best of our knowledge, we observed that all the current LP schemes suffer from at least one key drawback. First, some of these schemes are vulnerable to prover–prover (P–P) collusions [22], [25], [27]. In this attack, a remote malicious prover colludes with a dishonest user (located at a desired site) to obtain an LP. The dishonest user submits an LP request to the neighbor witness devices on behalf of the remote prover. This security threat is called terrorist fraud in the literature [31], [32] (see Section III-A for more details). Second, none of the current distributed schemes offer a reliable solution for Prover–Witness (P–W) collusions. In this attack, a dishonest user acts as a witness for a remote malicious prover and generates a fake LP for him [25]. Note that this security threat is specific to the distributed LP schemes only since witnesses are not trusted in this type of scheme. Finally, in some schemes, location privacy has not been considered [21], [23], [28], i.e., users broadcast their identity for neighbor devices or a third party server during the LP generation or submission process. In addition, there are other challenges with the current schemes, such as high level of communication and computation overheads [26] and expensive implementation [21], [24].

As far as we know, no LP scheme has been introduced to address all these challenges at the same time. Motivated by this, to address these key concerns, we propose a distributed LP scheme, Privacy-Aware and Secure Proof Of pRoximiTy (PASPORT), which performs LP generation and verification for mobile users in a secure and privacy-aware manner. The proposed scheme provides the integrity and non transferablity of generated LPs. To make PASPORT resistant to P–P collusions and perform private proximity checking, we develop a privacy-aware distance bounding (DB) protocol P-TREAD and integrate it into PASPORT. P-TREAD is a modified version of TREAD [33], a state of the art and secure DB protocol without privacy consideration. Our customization does not affect TREAD's main structure and features. Thus, PASPORT benefits from its security guarantees. By employing P-TREAD as the DB mech- anism, a malicious prover colluding with an adversary can easily be impersonated by the adversary later. Generally, users do not take such a risk by initiating a prover–prover collusion. The contributions of this article are threefold.

1) We design PASPORT, a secure, privacy-aware and collusion-resistant LP scheme for mobile users. PASPORT has a decentralized architecture suitable for scenarios in which a fixed wireless infrastructure does not exist.

2) To privately perform the procedure of proximity check- ing, we propose P-TREAD, a privacy-aware and secure DB mechanism and integrate it into PASPORT.

3) We perform a prototype implementation of PASPORT on the Android platform. Our experimental results show that the proposed scheme works faster than the existing distributed LP schemes and requires low computational resources.

II. LITERATURE SURVEY

Centralized LP Schemes

In this approach, a central trusted node, such as a wireless AP, is utilized to generate LPs for users in a specific site. The idea of employing wireless APs as LP generators was introduced by Waters and Felten [22] for the first time. They measure the round-trip signal propagation latency to decide on the proximity of a user to a trusted AP referred to as the location manager. However, the proposed scheme is vulnerable against relay attacks and specifically against terrorist frauds. In other words, their algorithm lacks a mechanism by which the location manager ensures that the received ID is really for the user who has submitted the LP request. To address this issue, Saroiu and Wolman [23] proposed a technique in which the AP broadcasts beacon frames consisted of a sequence number. To obtain an LP, users must sign the last transmitted sequence number with their private key and send it back to the AP along with their public key (the access point broadcasts beacons every 100 ms). This makes the system resistant against terrorist frauds since the malicious prover does not have enough time to receive the sequence number from the adversary and sign and send it back to the adversary. However, the proposed algorithm has privacy issues because users must reveal their identity publicly. Javali et al. [21] have used the same idea to make their algorithm resistant against relay attacks. They also utilize the unique wireless channel characteristics, i.e., channel state information (CSI) to decide on users' proximity. The proposed scheme consists of three entities, i.e., AP, verifier, and server, which make the system expensive. In addition, the user's identity is revealed publicly, which might cause privacy issues.

B. Distributed LP Schemes

In distributed scenarios, users collaborate with the system to generate LPs. In other words, users act as witnesses for each other. The main advantage of this approach is that there is no need for a trusted AP to issue LPs. Therefore, this type of systems can be used in locations where users are far from a trusted entity. APPLAUS introduced by Zhu and Cao [26] is one of the pioneer research works on distributed LP systems. In APPLAUS, mobile devices use their short- range Bluetooth interface to communicate with their nearby devices who request an LP. To preserve users' location privacy, they need to select a set of M pseudonyms and change them periodically. These pseudonyms are considered as users' public keys, which are required to be registered with a trusted certificate authority (CA) along with the associated private keys. However, changing pseudonyms regularly creates a high level of computation and communication overhead. In addi- tion, the users are required to generate dummy LPs as well. Davis et al. [27] proposed a privacy-preserving alibi (LP) scheme that has a distributed architecture. To preserve users' location privacy, in the introduced scheme, their identity is not revealed, while an alibi is being created. Thus, only a judge with

whom a user submits his/her alibi can see the user's identity. However, collusions and other security threats have not been considered in this article. In the distributed solutions, Prover-Witness collusions are possible because witness devices are not always trusted. A witness device can issue an LP for a dishonest user, while one of them (or both) is not located at the claimed location. This is one of the major challenges of these schemes. For example, in PROPS that has been proposed by Gambs et al. [30], Prover-Witness collusions have not been discussed although it provides an efficient and privacy-aware platform for users to create LPs for other users.

To the best of our knowledge, there is no efficient and reliable solution proposed in the literature to resolve the Prover Witness collusions issue with a high level of reliability even though some significant efforts have been made so far. For example, in LINK introduced by Talasila et al. [28], a group of users collaboratively verify a user's location upon his/her request sent through a short-range Bluetooth interface. It is assumed that there is a trusted location CA (LCA) to which the verifying users (located in the vicinity of the requesting user) send their verification messages. Then, the LCA checks the validity of the claim in case of a Prover-Witness collusion.

This is done by checking three parameters: the spatiotemporal correlation between the prover and verifiers, the trust scores of the users, and the history of the trust scores. However, it does not detect and prevent Prover-Witness collusions with a high level of reliability. Moreover, in the LINK scheme, users' location privacy has not been considered in the scheme design since a user needs to broadcast his/her ID to the neighbor verifiers. STAMP introduced by Wang et al. [25] is another example in which an entropy-based trust model is proposed to address the Prover-Witness collusions issue. This method is also unable to provide the necessary reliability to detect Prover- Witness collusions. In addition, to address terrorist frauds, STAMP employs the Bussard-Bagga protocol [31] as the DB protocol that has already been shown to be unsafe [34]-[36]. Moreover, the computation time required by STAMP to create an LP is long when users have a large private key [25]. Although different novel methods have been introduced so far, each of them has its own constraints, i.e., privacy issues [21], [23], [28], vulnerability against collusions [22], [25]–[28], [30], high level of communication and computation overheads [26], and expensive for implementation [21], [24]. The scheme proposed in [29] prevents P–W collusions only in crowded scenarios.

III. IMPLEMENTATION

In this section, we present our proposed scheme for secure LP generation and verification. First, we present the framework and its entities. Second, we present the trust and threat model which we have considered in this article. Following this, we introduce P-TREAD. Finally, the full framework of the PASPORT scheme is presented.

A. Architecture and Entities

The proposed system architecture is shown in Fig. 3. As we see, the system has a distributed architecture and consists of three types of entities, i.e., prover, witness, and verifier. A prover is a mobile user who requires to prove his/her location to a verifier. A witness is the entity that accepts to issue an LP for a neighboring prover upon request. We assume that service providers create sufficient incentives for mobile users to become a witness and certify other users' location.

In PASPORT, we consider witnesses as mobile users. Finally, a verifier is the unit that is authorized by the service provider to verify LPs

claimed by provers. We assume that provers communicate with witnesses through a short-range communication interface, such as Wi-Fi or Bluetooth. This short- range communication channel is supposed to be anonymous such that users can broadcast their messages over it without revealing their identifying data, such as IP or MAC address.

B. Trust and Threat Model

We assume that mobile users are registered with the service provider. Each user has a unique public-private pair key stored on his/her mobile device and certified by a CA. Users' identity is determined through their public key, and we assume that users never share their private key with other users because they do not give their mobile devices to others [21], [24], [25]. Thus, in a collusion scenario, we suppose a malicious prover never goes that far to provide another party with his/her private key. We also assume that all the messages exchanged between the entities might be eavesdropped by passive eavesdroppers. In the following, we discuss the trust and threat model for each entity individually.

1) Prover: It is assumed that the prover makes an effort to obtain false LPs. This can be done through different scenarios in which a prover might try to provide the witnesses with fake information about his/her location to convince them to generate LPs for him/her, manipulate the LP issued for him/her to change its location or time field, attempt to steal an LP issued for another user and use it for him/herself, and collude with other users (provers or witnesses) to obtain LPs. Moreover, we assume that provers try to obtain the identity of witnesses.

2) Witness: A witness might collude with a prover to generate a fake LP for him/her. In addition, a witness may try to deny an LP that has been issued by himself/herself. Witnesses are assumed to be curious about the provers' identity.

3) Verifier: We suppose that the verifier is trusted and never leaks users' identity and their spatiotemporal data. It is assumed that the verifier keeps a regularly updated list of witnesses who are present at the given location and have accepted to generate LPs for other users. The verifier accepts the LPs issued by these witnesses only. We suppose that service providers create necessary incentives to encourage selfish users to collaborate with the system. Otherwise, they might not generate LPs to save their battery power or reduce their communication costs. Regarding collusions, we consider both prover–prover and prover–witness collusions in our threat model as it can be directly derived from the above-mentioned assumptions., we introduce the proposed privacy-aware DB protocol P-TREAD.

C. P-TREAD

In this section, we present P-TREAD, a modified version of TREAD, for private proximity checking in the PASPORT architecture.

to protect users' privacy, we need to customize TREAD in such a way that provers can anonymously submit an LP request to neighbor witnesses. For this reason, in P-TREAD, we limit a witness' role to only collecting (not verifying) the required data from the prover (the verification is performed by the remote trusted verifier). All the privacy-sensitive data are encrypted by the prover and sent to a witness who signs and sends them back to the prover as an LP. Then, after the claim (received LP) is submitted to the verifier by the prover, verification of the claim can be performed by the trusted verifier in the next phase. We divide the whole procedure into two phases: 1) data

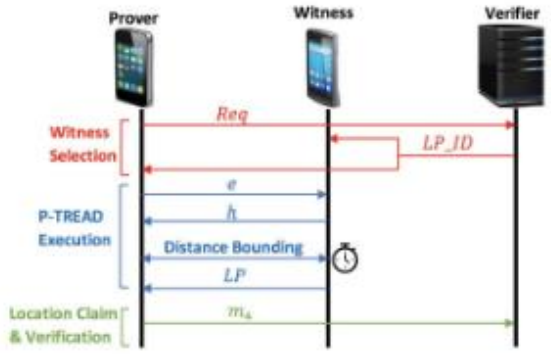
collection and LP generation and 2) authentication and verification.

1) Phase 1 (Data Collection and LP Generation): In this part of the protocol, the initialization phase of TREAD is performed with the following exceptions.

1) The prover device does not send IDP to the witnesses as a plain text message (it only sends e to the witnesses).

2) e is computed by the prover device using the verifier's public key. Therefore, the witnesses cannot decrypt it and deanonymize the prover. We assume that the verifier publishes its public key for the users. Moreover, every user has registered a public/private key pair with the verifier.

3) The witness devices do not check the prover's signature σP since the prover must be anonymous (in addition, they cannot decrypt e and obtain the signature). Later, σP will be checked by the verifier in the next phase.



$Req = E_{Verifier}(ID_P \| Loc)$
$e = LP\_ID \| E_{Verifier}(m_1 \| S_{Prover}(m_1)), m_1 = a \| b \| ID_P \| Loc$
$LP = E_{Verifier}(m_2 \| S_{Witness}(m_2)), m_2 = r \| c \| h \| e \| ID_W \| Loc' \| time$
$m_4 = E_{Verifier}(LP_1 \| LP_2 \| \cdots \| LP_K \| a \| b)$

Message flow between the three entities of the proposed scheme.

D. Workflow of PASPORT Framework :

The proposed LP scheme consists of three main phases: initialization, LP generation, and location claim and verification.

summarizes the cryptographic notations that we use in this article.

1) Initialization: In this phase, users register with the system and the CA certifies users' public-private key pairs. Moreover, the verifier creates a witness table in which it keeps the identity and location of mobile users who accept to be a witness. This table is regularly updated as witnesses sign on or off at every site. Furthermore, for every registered user in the system, the verifier records a list of provers for which the user generates an LP. These lists are used by the verifier to select which witnesses are qualified to generate LPs for a specific prover. This is done to prevent prover–witness collusions.

2) LP Generation: This phase is run in two stages: witness selection and P-TREAD Execution. a) Witness selection: In this stage, the prover submits an LP request to the verifier. Upon receiving the prover's request, the verifier selects K witnesses from its witness table to generate LPs for the prover. This is done to neutralize prover–witness collusions because, in this case, the prover does not have control over the witness selection process. However, to further protect PASPORT against prover–witness collusions, we integrate an entropy-based trust model as a supplementary method into the witness selection mechanism. Using this trust model, a trust score is computed by the verifier for every available witness device w based on its LP generation history and the number of LPs that w and the prover have issued for each other in the past. If the obtained score is above a threshold, the device is selected to witness for a requesting prover. The following step-by-step activities are performed in this stage:

1) Prover: First, the prover sends the following message Req to the verifier to inform it that he/she wants to start requesting an LP. This message can be sent to the verifier through the prover's Internet connection

2) Verifier: Upon receiving the prover's message, the verifier extracts all the witnesses who have recently (in a reasonable period of time) proved that they are in an acceptable distance to location Loc from its witness table (this acceptable distance is defined depending on the application). Then, K witnesses are selected among the shortlisted witnesses using the proposed trust model. These K witnesses are then qualified to generate LPs for this prover. If there are not enough qualified witnesses, the verifier suspends this request until the necessary number of qualified witnesses becomes available. Then, the verifier generates a unique ID for this LP (LP_ID) and sends it to the selected witnesses and the prover as well.

b) P-TREAD execution: In this stage, the prover starts to perform the P-TREAD protocol.

1) Prover: The prover generates two n-bit random numbers a and b and then computes the following message e and broadcasts it through the predefined short-range communication interface (Wi-Fi or Bluetooth)

2) Witness: A witness upon receiving e extracts the LP_ID and compares it with the one received from the verifier. If they are not same, it discards e. Otherwise, it generates an n-bit random number h and sends it to the prover.

3) Prover: The prover computes ($z_i = b_i \oplus h_i$) for i = 1, 2,..., n and sends an Ack to the witness.

4) Witness: The witness starts an n-stage time-sensitive DB process by generating a random bit ci at each stage i and sending it to the prover. It also starts a timer immediately after sending ci .

5) Prover: Upon receiving ci , the prover instantly sends the following response ri to the witness:

6) Witness: The witness stops the timer when the response ri is received from the prover. The timer must show a time less than the predefined threshold (2dmax)/C + to, where dmax is the maximum allowable distance between the prover and the witness, C is the speed of light, and tO is the overhead time required by the prover to compute the response bit ri upon receiving ci . If all the n responses are received in the correct time, the witness issues the following LP and sends it to the prover:

IV. PERFORMANCE EVALUATION

To study the feasibility of the proposed scheme, we implemented a Java prototype of the proposed scheme on the Android platform. Our experiments were performed on two Android mobile devices: 1) an LG G4-H818P equipped with a Hexa-Core 1.8-GHz processor, 3 GB of RAM, and running Android OS 5.1, acting as a prover and 2) a Sony Xperia Z1 equipped with a Quad-Core 2.2-GHz processor, 2 GB of RAM, with Android OS 4.4.4, acting as a witness. We adopted Bluetooth as the communication interface between the mobile devices and conducted the tests in both indoor and outdoor environments. Each measurement shown in this section has been obtained by averaging the results of ten independent tests. We used the RSA key pairs for encryption and SHA1 as the one-way hash function to compute user's signatures. Since the LP verification phase is performed by the verifier server that has a high level of storage and computational power, we focus our experiments on the P-TREAD Execution phase that is performed by mobile devices with limited resources.

During the application runtime, we measured the CPU utilization of the

implemented code by installing a monitoring application that reports the amount of CPU usage of the processes running on the device. the CPU usage for a user in standby mode is almost 0.5% and independent of the key size. However, due to heavy computations required for encryption and signature calculations in the LP generation phases, the average CPU usage increases to 2.5%, 8%, and 19% for key sizes 1024, 2048, and 3072, respectively.

We also recorded the amount of time that PASPORT requires to generate an LP after the prover device receives LP_ID from the verifier. We compared the results to the decentralized schemes STAMP [25] and APPLAUS [26]. Fig. 6(b) and (c) shows the results for different key sizes (in APPLAUS, the authors have not implemented their scheme for key sizes larger than 256). As expected, longer times were recorded for larger key sizes. The reason is that the DB phase is performed for n challenge bits. Thus, for larger values of n, it takes longer for the DB phase to be performed. As shown in Fig. 6(b) and (c), PASPORT provides faster responses than

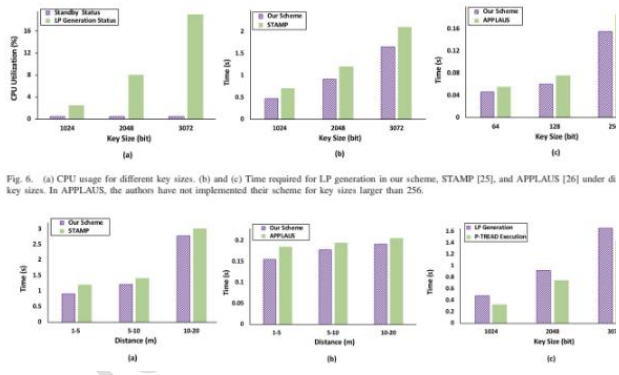


Fig. 6. (a) CPU usage for different key sizes. (b) and (c) Time required for LP generation in our scheme, STAMP [25], and APPLAUS [26] under di key sizes. In APPLAUS, the authors have not implemented their scheme for key sizes larger than 256.

similar schemes. The reason is that in STAMP and APPLAUS, the Bussard–Bagga DB protocol is used for provers' proximity checking, while in PASPORT, we integrate P-TREAD into the scheme to perform this job that is a more lightweight protocol regardless of its security advantages over the Bussard-Bagga protocol. Unlike P-TREAD, in the Bussard–Bagga protocol, different commitment and decommitment computations are needed to be performed by the prover and witness devices, respectively. Moreover, STAMP requires to perform at least two commitment calculations in order to provide location privacy [25]. In APPLAUS, to preserve users' location privacy, they need to select a set of M pseudonyms and change them periodically. This creates a high level of computation and communication overhead. To evaluate the impact of physical distance between the mobile users on LP generation, we conduct our experiments for different distances and compare the results to the performance of STAMP and APPLAUS respectively). As we see, for longer distances, the required time for PASPORT to generate an LP increases since higher communication latencies occurring in this case. Note that distance only affects the Bluetooth communication latency and does not change the amount of time required for computations performed in mobile devices.

V. CONCLUSION

This article proposed a secure and privacy-aware scheme for LP generation and verification. The proposed scheme has a decentralized architecture suitable for ad hoc applications in which mobile users generate LPs for each other. To address terrorist frauds, we developed a DB protocol P-TREAD, that is, a private version of TREAD, and integrated it into PASPORT. Using P-TREAD, a dishonest prover who established a prover-prover collusion with an adversary can easily be impersonated by the adversary later. Thus, no logical user takes such a risk by initiating a prover–prover collusion. Furthermore, we employed a witness selection mechanism to address the prover–witness collusions. Using the proposed mechanism, available witnesses are randomly assigned to requesting provers by the verifier. This prevents

malicious provers from choosing the witnesses themselves.

The main strengths of the proposed scheme are: 1) no central trusted entity is required to operate as a witness device; 2) it has reliable performance against prover–prover and prover– witness collusions to which majority of the current schemes are vulnerable; 3) our prototype implementation shows that the LP generation process in the proposed scheme is faster than the existing schemes; and 4) it preserves users' location privacy as P-TREAD DB protocol enables users to anonymously broadcast their messages for the neighbor witnesses during the LP generation process. As a future work direction, we intend to extend the PASPORT scheme such that it provides location granularity feature. Using these users can select to which level their location data is revealed. Moreover, designing a blockchain based incentive mechanism to encourage users to collaborate with the system can be another research direction for this article.

VI. REFERENCES

[1] P. Asuquo et al., "Security and privacy in location-based services for vehicular and mobile communications: An overview, challenges, and countermeasures," IEEE Internet Things J., vol. 5, no. 6, pp. 4778–4802, Dec. 2018.

[2] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor local- ization," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 491–506, 1st Quart., 2016.

[3] R. Gupta and U. P. Rao, "An exploration to location–based service and its privacy preserving techniques: A survey," Wireless Pers. Commun., vol. 96, no. 2, pp. 1973–2007, 2017.

[4] Global Location–Based Services Market (2018–2023). Accessed: Jul. 20, 2019. [Online]. Available: https://www.businesswire.com/news/home/20180927005490/en/Global–Location-based-Services-Market-2018-2023-Projected-Grow

[5] Y. Zheng, M. Li, W. Lou, and Y. T. Hou, "Location based handshake and private proximity test with location tags," IEEE Trans. Depend. Sec. Comput., vol. 14, no. 4, pp. 406–419, Jul./Aug. 2017.

[6] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy–preserving location proof for securing large–scale database–driven cognitive radio networks," IEEE Internet Things J., vol. 3, no. 4, pp. 563–571, Aug. 2016.

[7] A. Pham, K. Huguenin, I. Bilogrevic, I. Dacosta, and J. P. Hubaux, "SecureRun: Cheat–proof and private summaries for location–based activities," IEEE Trans. Mobile Comput., vol. 15, no. 8, pp. 2109–2123, Aug. 2016.

[8] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in Proc. IEEE INFOCOM, Apr. 2013, pp. 2751–2759.

[9] Z. Zhang et al., "On the validity of geosocial mobility traces," in Proc. ACM Workshop Hot Topics Netw. (HotNets), 2013.

[10] D. Bucher, D. Rudi, and R. Buffat, "Captcha your location proof—A novel method for passive location proofs in adversarial environments," in Proc. 14th Int. Conf. Location Based Services, 2018, pp. 269–291.

[11] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in Proc. 21st Int. Conf. World Wide Web (WWW), 2012, pp. 191–200.

[12] Nike+ Badges and Trophies. Accessed: Jul. 20, 2019. [Online]. Available: http://www.garcard.com/nikeplus.php [13] Higi.

Higi: Know Your Numbers. Own Your Health. Accessed: Jul. 20, 2019. [Online]. Available: https://higi.com

[14] Oscar Health Using Misfit Wearables To Reward Fit Customers. Accessed: Jul. 20, 2019. [Online]. Available: http://www.forbes.com/sites/stevenbertoni/2014/12/08/oscarhealth-using-misfit-wearables-to-reward-fit-customers

[15] Health Insurer's App Helps Users Track Themselves. Accessed: Jul. 20, 2019. [Online]. Available: http://www.technologyreview.com/news/516176/healthinsurers-app-helps-users-track-themselves

[16] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Location privacy preservation in database–driven wireless cognitive networks through encrypted probabilistic data structures," IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 2, pp. 255–266, Jun. 2017.

[17] K. Zeng, S. K. Ramesh, and Y. Yang, "Location spoofing attack and its countermeasures in database-driven cognitive radio networks," in Proc. IEEE Commun. Netw. Secur. (CNS), Oct. 2014, pp. 202–210.

[18] A. van Cleeff, W. Pieters, and R. Wieringa, "Benefits of location-based access control: A literature study," in Proc. IEEE/ACM Int. Conf. Green Comput. Commun., Dec. 2010, pp. 739–746.

[19] Y. Baseri, A. Hafid, and S. Cherkaoui, "K-anonymo us location-basedfine-grained access control for mobile cloud," in Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC), Jan. 2016, pp. 720–725.

[20] E. Androulaki, C. Soriente, L. Malisa, and S. Capkun, "Enforcing location and time–based access control on cloud–stored data," in Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2014, pp. 637–648.

[21] C. Javali, G. Revadigar, K. B. Rasmussen, W. Hu, and S. Jha, "I am alice, i was in wonderland: Secure location proof generation and verification protocol," in Proc. IEEE 41st Conf. Local Comput. Netw. (LCN), Nov. 2016, pp. 477–485.

[22] B. Waters and E. Felten, "Secure, private proofs of location," Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep. TR–667–03, 2003.

[23] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in Proc. ACM HotMobile, 2009.

[24] W. Luo and U. Hengartner, "VeriPlace: A privacy–aware location proof architecture," in Proc. ACM GIS, 2010, pp. 23–32.

[25] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, "STAMP: Enabling privacy–preserving location proofs for mobile users," IEEE/ACM Trans. Netw., vol. 24, no. 6, pp. 3276–3289, Dec. 2016.

[26] Z. Zhu and G. Cao, "Towards privacy–preserving and colluding– resistance in location proof updating systems," IEEE Trans. Mobile Comput., vol. 12, no. 1, pp. 51–64, Jan. 2011.

[27] B. Davis, H. Chen, and M. Franklin, "Privacy preserving alibi systems," in Proc. ACM ASIACCS, 2012, pp. 34–35.

[28] M. Talasila, R. Curtmola, and C. Borcea, "Link: Location verification through immediate neighbors knowledge," in Mobile and Ubiquitous Systems: Computing, Networking, and Services (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommuni- cations Engineering). Berlin, Germany: Springer, 2012.