Locanyms: Towards Privacy-Preserving Location-Based Services [*]

Sebastien Gambs
Univ. de Rennes 1 - INRIA
IRISA

Marc-Olivier Killijian
Université de Toulouse
LAAS-CNRS

Matthieu Roy
Université de Toulouse
LAAS-CNRS

Moussa Traore[†]
Université de Toulouse
LAAS-CNRS

ABSTRACT

Recent advances in geolocated capacities, secure and verified positioning techniques, ubiquitous connectivity, as well as mobile and embedded systems, have led to the development of a plethora of Location-Based Services (LBS), personalizing the services they deliver according to the location of the user querying the service. However, the widespread use of mobile equipments, with ever increasing availability, precision, performance and connectivity have introduced the creepy feeling of being continuously monitored, in particular by the providers of the LBS. Thus, beyond the benefits they provide, users have started to be worried about the privacy breaches caused by such systems. The main objective of this paper is to discuss the privacy issues raised by LBS and the challenges of implementing privacy-preserving location-aware systems. Moreover, we also give a brief overview of positioning techniques used by LBS and we introduce the novel concept of *locanym*, which corresponds to a pseudonym linked to a particular location that could be used as a basis for developing privacy-preserving LBS.

Keywords

Privacy, Location-based services, Ubiquitous computing.

1. INTRODUCTION

Due to the rapid advances in positioning technologies such as Global Positioning System (GPS), Global System for Mobile Communication (GSM), Radio Frequency IDentification (RFID), and WiFi (802.11b/g/n) and the widespread deployment of wireless local area networks, mobiles devices are often equipped with geolocated and wireless communication capacities. These recent development of ubiquitous devices have lead to the development of a new class of services known as *Location-Based Services* (LBS), that are tailored to the current location of the individual querying the service. LBS can access, combine, and transform contextual information, and more specifically location information, in order to personalize the service provided to the user. For instance, a LBS can be used for resource discovery (*e.g.*, finding the closest restroom from my position[1]), path-finding (*e.g.*, computing the shortest route to a gas station), real-time social applications (*e.g.*, informing me about the presence of my friends in the vicinity[2]) or location-based gaming (*e.g.*, playing with the nearest challenger).

When people use LBS to support them in their daily tasks, their position is usually acquire automatically through mobile equipments they carry with them. Thus, these systems continuously monitor and reveal information about the location of their users as the position of these mobile systems is essentially the same as the users of such system (*e.g.*, which could be a single individual or a small group of persons such as a family). In most of the cases, the collected location data is transmitted to another system (typically a centralized server or another mobile equipment), which needs this information to provide the LBS (*e.g.*, to generate the list of nearby restaurants[3]) or to participate to its computation (*e.g.*, to help two people to meet at the optimal rendezvous point[4]). However, the collection and transmission of such data can also be used against the privacy of a user, either at the time of transmission (*e.g.*, to send unwanted advertisement), or later in the future (*e.g.*, to detect that the user has violated the speed limit while driving his car [8]). Moreover, *inference attacks* [9] can be used to extract personal information from the observed mobility travels of an individual such as the Points Of Interests (POIs) characterizing his mobility (*e.g.*, home, place of work or even the hospital that he often visits), to build mobility models that can predict with an high accuracy his past, current and future locations, as well as to deduce a part of his social graph by inferring that

[*] This work is partially supported by LAAS, CNRS and ANR French national program for Security and Informatics (grant #ANR-11-INSE-010, project AMORES[1]).

[†] Corresponding author, email: mtraore@laas.fr

[1] www.have2p.com

[2] www.loopt.com

[3] www.have2eat.com

[4] www.rendevousSpot.com

he has a social relationship with the individuals with whom he shares often the same physical location.

In order to address and mitigate these privacy issues, recently there has been a huge interest in the design of privacy-preserving versions of LBSs providing high quality of service while preserving the privacy of their users. In this paper, we elaborate on how privacy can be integrated in location-aware systems through a few examples highlighting the complexity of addressing such issues. We also argue that privacy needs to be taken into account in LBS by grounding in fundamental privacy principles capturing the privacy needs of users of such systems. Additionally, we believe that addressing privacy in LBS should embed privacy protection and control mechanisms as fundamental requirements on all the levels of the system. The outline of this paper is the following. First in Section 2, we review the concepts of location-based services and secure positioning. Afterwards, in Section 3, we conduct a privacy analysis of some existing LBS. Finally, in Section 4 we define some desirable properties that any LBS should fulfill to protect the privacy of their users. We also introduce the notion of *locanyms*, which captures most of these privacy requirements, before concluding through an illustration on how these properties apply to a specific LBS.

2. LOCATION-BASED SERVICES AND SECURE POSITIONING

A LBS can be defined as a service that takes as input the current location of a user (generally acquired through a mobile device carried by this user) and tailors its output depending on the acquired location data. For instance, a user visiting a shopping mall may call a LBS to locate the closest shop that matches his budget and its clothing preferences. Therefore, location data are usually augmented with complementary information related to the user, thus further increasing the privacy risks. The ability to provide the user with a customized service depending on his location could also be used by companies to send targeted advertising and for billing purposes, by banks to perform authentication based on the location, and by restaurant owners to propose discount to users passing nearby. The above list is far from being exhaustive, as one could think about position-based access control in which the access to a particular resource is granted only to persons that are physically located inside a predefined perimeter. For instance, a printer or fax machine could be accessible only to persons located within a set of offices, or a pizza delivery service might first verify if the person placing the order is indeed located at the specified delivery address.

One of the first question that naturally arises when dealing with LBS is how a particular user can convince others about the validity of its current position. More precisely, the user can be viewed as a *prover*, who claims to be currently at a particular location, and which wants to convince a set of remote *verifiers* that he is indeed at the claimed position. Thereafter, we will refer to this problem as *Secure Positioning* or sometimes as *Secure Position Verification.* Secure Positioning is a fundamental problem that has to be tackled when designing a secure LBS and that can be addressed by designing a technique enabling the prover (*i.e.*, the user) to prove its position through interactions with a group of verifiers. In the following, we review the two main families of approaches that have been proposed to tackle this problem (*i.e.*, distance-bounding protocols and received signal strength), and we briefly discuss their pros and cons.

2.1 Distance-bounding Protocol

The approach based on *Distance-Bounding Protocol* (DBP) [3] (sometimes also called *Time-of-Flight* (TOF), *Round-Trip Time* (RTT) or *Round Time-of-Flight* (RToF)) aims at measuring the relative proximity of two devices using physical limits on information propagation speeds. A DBP protocol involves generally two participants, a prover and a verifier, and enables the verifier to place an upper bound on the physical distance separating him from the prover without requiring the assistance of a third party. The general schema of a DBP is the following: first, the verifier sends a challenge to the prover and starts his own timer. Upon reception of the challenge, the prover performs some computation (in some scheme, the computation simply consists in sending back the message [17]) in order to construct the response to the received challenge and then sends it to the verifier, which stops his timer upon reception of the answer. By multiplying the elapsed time with the propagation speed of the signal (*e.g.*, ultrasound or electromagnetic signals), the verifier can deduce an upper bound on his distance to the prover. Moreover, in addition to the DBP, it is possible to add a layer of authentication by having the prover authenticate himself to the verifier by using some secret shared between the prover and the verifier, thus proving that the entity responding to the challenge is indeed the prover that has initiated the DBP.

The security of DBP is based on the assumption that it should be impossible for the prover to send the response before receiving the challenge [3]. In addition, it is assumed that the processing time needed to compute the response upon reception of the challenge should be negligible compared to the propagation time of the message in order to estimate an accurate upper bound on the location of the prover. This requirement can be easily met for DBP based on ultrasounds in which the processing time of the prover needs to be in the order of microseconds to attain reasonable precision [16]. However, in this context some security problems arise since (ultra-)sounds are not resistant to active attackers that can physically alter the signals. For instance, such an attacker can modify the medium (*e.g.*, sound travels faster through metal than through the air) or use Radio Frequency (RF) wormholes (*e.g.*, by retransmitting the signal using electromagnetic waves) to claim that it is closer from the verifier than he really is. Current knowledge of physics ensures that nothing can travel faster than light, and hence RF-based DBP (whose travel speed is very close to that of light) seems more robust, at least in the sense that they forbid such wormhole attacks. In this situation, the only threat left is that the attacker can claim to be further away than he really is by delaying his response. This does not contradict the main objective of the DBP that is to derive an upper (and not a lower) bound on the distance from the prover to the verifier. However, with RF-based DBP, the prover's processing time needs to be in the order of nanoseconds, which in the worst case allows a malicious prover to pretend be closer to the verifier by approximatively 15 centimeters (assuming that the malicious prover is able to process signals instantaneously).

Brands and Chaum [3] were the first to introduce the concept of DBP that can be used to verify the proximity of a device in a cryptographic manner. This seminal work has lead to the design of a following DBP from Hancke and Kuhn more appropriate for RFID tags and dealing with noisy environments [11], and a plethora of other protocols [2, 18, 20, 21]. Despite their accuracy and well-founded security models, most DBP suffer from location privacy leakage [15]. More precisely, they always leak some information about the location and distance of the different communicating partners even to passive attackers that only eavesdroppes the communications.

2.2 Received Signal Strength Indicator

An approach used by several Wifi-based localization system is to measure the *Received Signal Strength Indicator* (RSSI) of the radio signals used. The RSSI approach is based on two observations: RF signal strength decreases 1) when the transmitter and the receiver are further apart and 2) when there are obstacles between the transmitter and the receiver. Based on these two observations, different readings of the signal strength are measured on different points of the location site and then recorded in a database. When the system receives a location query from a user, the system compares the user's current signal strength values with the values stored in the database. Based on this comparison, the system is able to deduce the most probable location of the user and returns it. In [14], the authors have proposed a RF-based indoor location tracking system that processes the signal strength information at multiple base stations. Another localization system, WHAM! (Where AM I!) [12], continuously records signal strengths received by a user's device, and disambiguates the current location of the user by backtracking to the user's previous locations in the floor model, eliminating candidate locations that are not likely to be reachable from its previous known locations. In [10], the authors describe a practical robust Bayesian method for topological localization that reduces the time required to train the localizer while keeping the localization accuracy good enough so that it can be used by an LBS. Many other RSSI schemes exist in the literature and we refer the reader to the following survey for more details [13].

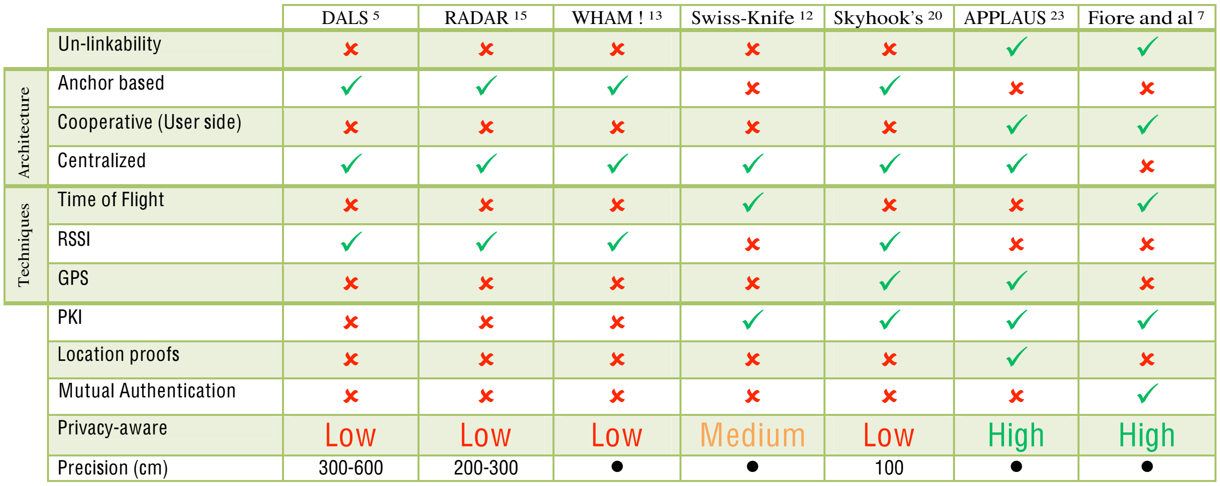
Most RSSI schemes implicitly assume that the prover uses a standard and unmodified wireless card. However, it is not very difficult for an attacker to build a directional antenna that can largely increase the sending or receiving range, and therefore in this case measuring the signal strength does not lead to a good level of security. In addition, by jamming and replaying localization signals, an attacker can convince a device to be in a location different from its actual physical position [19]. As a countermeasure, it was proposed to design a system based on collaborative localization in order to enhance the accuracy of the position estimation by leveraging and combining on the information gathered by neighboring nodes [4]. However, anonymization is a challenging task for the design of Wifi-based localization systems. Indeed, users are primarily authenticated through their MAC address in order to avoid undesirable deliveries of messages to unappropriate nodes. Instead of using the true MAC address, some systems [7] frequently and randomly change the MAC address of the node (which can be seen as pseudonym) to reduce the linkability risks.

3. PRIVACY ANALYSIS OF EXISTING LOCATION-BASED SERVICES

Currently, there is no universal metric to quantify location privacy that reached a consensus in the privacy community. Hence, it is sometimes difficult to compare different approaches aiming at building privacy-preserving LBS. Generally, each approach adopts its own definition of location privacy and defines its own adversary model. Figure 1 provides an overview of several protocols and compares them according to different criteria. In this section, we briefly review their main features and how they address privacy. Thereafter, we assume that the main objective of the attacker is to track the location of a mobile node and this attacker is equipped with eavesdropping capacities.

A Duress Alarm Location System (DALS) [5] was proposed in the early nineties for the sole purpose of determining users location, and therefore does not provide any data networking services or privacy protection. DALS uses RF-signal strengths to determine the location of a user similarly to RSSI localization techniques. Furthermore, DALS makes use of specialized and costly hardware, and therefore the trade-off between the deployment cost and the perceived value of this system is not compelling enough for large-scale adoption. RADAR [14] was designed to overcome the limitations of DALS and can be deployed off-the-shelf over any wireless LAN technology. More precisely, RADAR used a RSSI localization technique and relies on a Viterbi-like algorithm for continuous tracking of users' location and disambiguation of candidate user locations with a precision of a few meters ($2-3$ meters). With respect to privacy, continuous user tracking is a major threat as users may feel that "Big Brothers is continuously watching them". Furthermore, the communications exchanged can be eavesdropped as their content is not encrypted by default for these protocols. Another system, called WHAM! [12], works similarly to RADAR, with the exception of the backtracking technique used, which improves the accuracy of the localization results but causes the same security and privacy problems as previous protocols. SkyHook [19] differs from the previously described systems in the sense that the messages exchanged are encrypted. Therefore, location information can normally only be accessed by authorized entities. However, even if the user knows which entity should in principle be responsible for keeping his data private, he has no guarantee other than the promises of this entity that his location data will not be disclosed to other entities (*e.g.*, for instance to a marketing company for a profiling purpose or to nearby shops for targeted advertising).

Recently, a distributed cooperative scheme for Neighbor Position Verification (NPV) [7] was proposed. It enables a node playing the role of the verifier to discover and ascertain the position of nearby nodes. The verifier can initiate the protocol at any time, by triggering interactive protocol within his 1-hop neighborhood that consists in 4 rounds of communication. The main objective of this protocol is to let the verifier collect enough information so that he can compute by himself the distances between any pair of neighboring nodes. In this protocol, the messages exchanged are made anonymous by taking advantage of the broadcast nature of the wireless medium, thus enabling nodes to record reciprocal timing information without disclosing their iden-



		DALS [5]	RADAR [15]	WHAM ! [13]	Swiss-Knife [12]	Skyhook's [20]	APPLAUS [23]	Fiore and al [7]
	Un-linkability	✗	✗	✗	✗	✗	✓	✓
Architecture	Anchor based	✓	✓	✓	✗	✓	✗	✗
	Cooperative (User side)	✗	✗	✗	✗	✗	✓	✓
	Centralized	✓	✓	✓	✓	✓	✓	✗
Techniques	Time of Flight	✗	✗	✗	✓	✗	✗	✓
	RSSI	✓	✓	✓	✗	✓	✗	✗
	GPS	✗	✗	✗	✗	✓	✓	✗
	PKI	✗	✗	✗	✓	✓	✓	✓
	Location proofs	✗	✗	✗	✗	✗	✓	✗
	Mutual Authentication	✗	✗	✗	✗	✗	✗	✓
	Privacy-aware	Low	Low	Low	Medium	Low	High	High
	Precision (cm)	300-600	200-300	●	●	100	●	●

Figure 1: Comparison of different localization protocols

tities. Afterwards, following a revelation message broadcasted by the verifier, nodes disclose their identities only to him through secure and authenticated messages, which also contain the anonymous timing information they have collected. Finally, the verifier uses this data to match the different timings and identities and then performs a ToF-based ranging to compute the distances between all pairs of communicating nodes in its neighborhood. From the point of view of privacy, this protocol ensures the anonymity of communication by relying on a generator of MAC address during the discovery phase in order to obfuscate the identity of the prover. However, this protocol assumes that the verifier is trusted (*i.e.*, honest). Indeed, the verifier is granted the privilege to decide which entities are really in his proximity without requiring the help of an external trusted entity to verify the correctness of this proximity map.

In general, a system that can be used to generate a certified proof of the location of a node is called a *location proof system*. Zhu and Goa have proposed a privacy-preserving location proof system called APPLAUS [22], which is composed of the following elements:

1. A *prover* is a node collecting location proofs by broadcasting when needed a location proof request to its neighboring nodes through Bluetooth communications.

2. A *witness* is a node that accepts to generate a location proof upon request and sends it back to the prover.

3. A *location proof server* is required for storing the history of location proofs. It communicates directly with the provers when they submit their location proofs.

4. A *certification authority* (CA) is run by an independent trusted third party. Upon joining the location proof system, each mobile node registers with the CA and pre-loads a set of public/private key pairs before entering the network. The CA is the only entity that knows the correspondence between the real identity of a node and the pseudonyms (public keys) used by this node.

5. A *verifier* is another node or an application that is authorized to verify a prover's location within a specific period of time.

When a prover needs to generate a location proof at particular time, it broadcasts a location proof request to its neighboring nodes through Bluetooth communications. Once a neighboring node agrees to provide a location proof for the prover, this node (now acting as a witness) generates a location proof and sends it back to the prover. The prover is responsible for forwarding this location proof to the location proof server. Finally, an authorized verifier can send a query that contains the real identity of the prover and a time interval. The CA first authenticates the verifier, and then converts the real identity contained in the query to its corresponding pseudonyms during that time period and retrieves their location proofs for this particular time period from the server. Changing pseudonyms on a regular basis provides two benefits. First, it protects the location privacy of each node by enhancing their *unlinkability*. Second, if the location proof server is compromised or monitored, it will be impossible for the attacker to identify the real source of the location proof. More precisely, the privacy of APPLAUS is ensured by the *separation of trust*: (1) the location proof server knows only pseudonyms and their associated locations, (2) the CA knows only the mapping between the real identity and its pseudonyms, and finally (3) the verifier only sees the authorized locations visited by a node for which he knows the real identity. The separation of trust ensures that as long as the attacker does not control more than one of the entity involved in the location proof system, it is impossible for him to learn a user's location because he does not have access to enough knowledge.

4. TOWARDS PRIVACY-PRESERVING LOCATION-BASED SERVICES

Location privacy can be broadly defined as the ability to prevent an unauthorized entity from learning the past, current or future locations of an individual. Based on this definition, most of LBS mentioned previously do not ensure location privacy. In order to reach this goal, we describe thereafter the desirable properties that we believe that a privacy-preserving LBS should fulfill:

1. **Unlinkability and unforgeability.** It should not be possible to trace and link different locations visited by a user. For instance, even if the user proves his locations at different occasions, it should be impossible to link the different location proofs being made by the same user. Similarly, a user should not be able to fool the system by forging a fake location proof.
2. **Accountability and non-repudiation.** The location information collected by the mobile device of a user should serve as a basis for deriving location proofs that are verifiable in order to ensure the validity of the user's position over an elapsed time.
3. **Sovereignty.** The LBS should empower the user with the control on how his location information is used and disseminated in the system. More precisely, users should be able to decide who can access their location information and under which restrictions.

Additionally, we think that it is important to use a distributed architecture, rather than a centralized server, in order to implement the LBS. Most of the systems described previously make use of the past location of user to increase the accuracy when predicting the current location or continuously monitor their positions. Such information is usually centralized by the LBS on a remote database server. However, if the security of this server is damaged, the privacy breach that can occur can be very important. Moreover, the computation of the positions by a central entity raises the possibility of continuously tracking users, which could also lead to a detailed profiling of the user. Therefore, to address this privacy concern, we recommend to build LBS by relying on a distributed architecture. In such an architecture, it is more difficult for an attacker to find the information about the location of a particular user since he does not have a single point of failure to attack to retrieve this information, which is scattered around the network. Moreover, a distributed architecture also preserves the privacy of users from being exposed to large companies, thus avoiding the "Big brother is watching you" syndrome. Finally, we also think that a privacy-preserving LBS should involve the devices present within the current neighborhood of a user to verify his position (*collaborative behavior*), in order to increase the resilience of the system against an active attacker. To encapsulate all these properties, we introduce the concept of *locanym*, which summarize the fundamental privacy requirements required to build a privacy-preserving LBS.

Definition 1 (Locanym) *A* locanym *is a geolocated version of a pseudonym tied to a particular geographical area. More precisely, it aims at deriving, from an accurate and verified positioning, a specific form of pseudonym providing the following properties: unlinkability, unforgeability, accountability, non-repudiation and sovereignty.*

In order to illustrate the importance of the privacy desiderata defined previously for the design of LBS, we will consider a *dynamic carpooling* scenario based on locanyms. Dynamic carpooling is an urban service in which drivers are dynamically match with passengers from sub-urban to urban journeys or intra-urban journeys. Privacy is of paramount importance in such a setting as users move between places but their mobility patterns should be kept private from other entities (even from other mobile users that they periodically meet and cooperate with) as they generally do not know them and therefore cannot blindly trust them with their location data. Consider for instance the scenario in which Bob wishes to drive to a shop situated in the town's outskirts. Before starting his journey, Bob decides to switch on the carpooling application to gain some experiences points (and possibly some money). Approximately at the same period, Alice wants to go to a shopping mall located somewhere on Bob's path.

Unlinkability and unforgeability. After starting her carpool engine by activating the passenger mode, Alice cooperates with the neighboring nodes (*e.g*, bus or other moving nodes) to create her locanym data. By drawing on this locanym, the carpooling application will be able to connect Bob to Alice without disclosing her real identity since the locanym is derived only from her location data. In particular, the request of Alice cannot be linked to her identity before she meets physically with Bob (and *vice-versa*), which ensures the anonymity of the users of the carpooling system. Another issue is for Bob is to be sure of the truthfulness of the location claimed by his potential carpoolmate. Indeed, as the position of Alice is certified by her neighbors, she cannot fool the system by forging a false location proof except if a majority of her neighbors are colluding with her, which should be sufficiently difficult to achieve in a realistic mobile environment in order to ensure the security of the application. Therefore, Bob can trust the carpooling request and decided whether or not to carpool with Alice. In this situation, locanyms seem sufficient to ensure the mutual authentication between Alice and Bob when they establish their first relationship in a privacy-preserving manner.

Accountability and non-repudiation. Once the relationship is established and the two participants physically meet, the anonymity can be lifted and their identities revealed. Moreover suppose that during the carpooling trip, something bad happens to Alice because of Bob. During a police investigation if Bob denies having carpooled with her, Alice can nonetheless prove that she was with Bob during a certain period. Indeed, the location proofs collected by Alice mobile device can be used as an evidence. The accountability property can be important to prove to a third party (such as the administration of the city) that another entity (*e.g.*, Bob) has actually participated in a carpooling activity. Moreover, the non-repudiation property, which can be obtained through a combination of locanyms and digital signatures, will also ensure that an entity cannot deny having participated to a carpooling trip to which he has previously given his explicit consent. The combination of accountability and non-repudiation can also be used by the carpooling service provider to offer some discounts to their regular users (*e.g.*, such as a discount on their transport subscription).

Sovereignty. Continuing on the carpooling scenario, suppose that Alice does not want to carpool with Bob because on some problems during previous trips. However, as locanymity hides the identity of Bob, if such a concern is not managed by the carpool system, Alice may have a bad surprise when she arrives at the meeting place. Ideally it should be possible for Alice to express in her carpooling request (for instance in the form of a blacklist) that she does not want to carpool with Bob. This property is a form a sovereignty in the sense that Alice can decide with whom she wants (or not) to share her location. Therefore, locanyms must take into account privacy policies defined on location data by a particular node. These privacy policies should be customizable by the user depending of the privacy level needed in the spirit of Platform for Privacy Preferences (P3P) [6].

5. REFERENCES

[1] E. Anceaume, C. Artigues, C. Bidan, Y. Deswarte, S. Gambs, G. Guette, J. Guiochet, M.-J. Huguet, M. Hurfin, M.-O. Killijian, D. Powell, N. Prigent, M. Roy, and F. Schettini. AMORES: an Architecture for MObiquitous REsilient Systems. TR 12031, LAAS-CNRS, 2012.

[2] G. Avoine and A. Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In P. Samarati, M. Yung, F. Martinelli, and C. Ardagna, editors, *Information Security*, volume 5735 of *Lecture Notes in Computer Science*, pages 250–261. Springer-Verlag, 2009.

[3] S. Brands and D. Chaum. Distance-bounding protocols. In T. Helleseth, editor, *Proceedings of EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer-Verlag, 1994.

[4] L.-W. Chan, J.-R. Chiang, Y.-C. Chen, C.-N. Ke, J. Hsu, and H.-H. Chu. Collaborative localization: Enhancing WiFi-based position estimation with neighborhood links in clusters. In K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, editors, *Pervasive Computing*, volume 3968 of *Lecture Notes in Computer Science*, pages 50–66. Springer-Verlag, 2006.

[5] T. Christ, P. Godwin, and R. Lavigne. A prison guard duress alarm location system. In *Proceedings of Security Technology*, pages 106 –116, oct 1993.

[6] Y. Deswarte and M. Roy. Privacy-enhancing access control enforcement. In *Proceedings of the W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement*, 2006.

[7] M. Fiore, C. Casetti, C.-F. Chiasserini, and P. Papadimitratos. Discovery and verification of neighbor positions in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 99 2011.

[8] J.-C. Freytag. Context quality and privacy - friends or rivals? In K. Rothermel, D. Fritsch, W. Blochinger, and F. Dürr, editors, *Quality of Context*, volume 5786 of *Lecture Notes in Computer Science*, pages 25–40. Springer-Verlag, 2009.

[9] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and I will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL'10)*, pages 34–41, New York, NY, USA, 2010.

[10] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of MobiCom'04*, pages 70–84, New York, NY, USA, 2004.

[11] G. Hancke and M. Kuhn. An rfid distance bounding protocol. In *Proceedings of SecureComm'05*, pages 67 – 73, sept. 2005.

[12] D. L. Lee and Q. Chen. A model-based wifi localization method. In *Proceedings of the 2nd international conference on Scalable information systems (InfoScale'07)*, pages 40:1–40:7, ICST, Brussels, Belgium, Belgium, 2007.

[13] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067 –1080, November 2007.

[14] B. Paramvir, P. Venkata, N., and B. Anand. Enhancements to the radar user location and tracking system. Technical report, Microsoft Research, University of California at San Diego, 2000.

[15] K. B. Rasmussen and S. Čapkun. Location privacy of distance bounding protocols. In *Proceedings of the 15th ACM conference on Computer and communications security (CCS'08)*, pages 149–160, New York, NY, USA, 2008.

[16] K. B. Rasmussen and S. Čapkun. Realization of rf distance bounding. In *Proceedings of the USENIX Security Symposium*, 2010.

[17] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *Proceedings of the 2nd ACM workshop on Wireless security (WiSe'03)* pages 1–10, New York, NY, USA, 2003.

[18] D. Singelee and B. Preneel. Distance bounding in noisy environments. In F. Stajano, C. Meadows, S. Capkun, and T. Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, volume 4572 of *LNCS*, pages 101–115. Springer-Verlag, 2007.

[19] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun. Attacks on public wlan-based positioning systems. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, pages 29–40, New York, NY, USA, 2009.

[20] R. Trujillo-Rasua, B. Martin, and G. Avoine. The Poulidor distance-bounding protocol. In S. Ors Yalcin, editor, *Radio Frequency Identification: Security and Privacy Issues*, volume 6370 of *Lecture Notes in Computer Science*, pages 239–257. Springer-Verlag 2010.

[21] S. Čapkun, L. Buttyán, and J.-P. Hubaux. Sector: secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the 1st ACM workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, pages 21–32, New York, NY, USA, 2003.

[22] Z. Zhu and G. Cao. Towards privacy-preserving and colluding-resistance in location proof updating system. *IEEE Transactions on Mobile Computing*, 99(PrePrints), 2011.