Resilient Privacy Protection for Location-Based Services Through Decentralization

Hongyu Jin
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
hongyuj@kth.se

Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

ABSTRACT

Location-based Services (LBSs) provide valuable features but can also reveal sensitive user information. Decentralized privacy protection removes the need for a so-called anonymizer, but relying on peers is a double-edged sword: adversaries could mislead with fictitious responses or even collude to compromise their peers' privacy. We address here exactly this problem: we strengthen the decentralized LBS privacy approach, securing peer-to-peer (P2P) interactions. Our scheme can provide precise timely P2P responses by passing proactively cached Point of Interest (POI) information. It reduces the exposure both to the *honest-but-curious* LBS servers and peer nodes. Our scheme allows P2P responses to be validated with very low fraction of queries affected even if a significant fraction of nodes are compromised. The exposure can be kept very low even if the LBS server or a large set of colluding curious nodes collude with curious identity management entities.

1 INTRODUCTION

A Location-based Service (LBS) query targets a location/region and has a specific interest; the LBS server responds with the most up-to-date relevant information, e.g., the latest menu of a restaurant, movies at a cinema, or remaining parking slots at a shopping mall. During this process, users' current or future whereabouts and interests are disclosed to the LBS server through their queries. Access to all submitted information is deemed necessary to best serve users, and the LBS server is entrusted with rich information. However, many studies reveal service providers can be honest-but-curious, aggressively collecting information to profile users, identifying home or working places or inferring interests towards commercial purposes.

LBS privacy has been widely studied. Location k-anonymity [3] ensures that at least $k-1$ other users are involved in an obfuscated region, $\mathcal{R}$, used as the querier's location. Therefore, even in the presence of an observer in $\mathcal{R}$, the query could not be linked to a certain user; the LBS server only learns the querier is one among k users in $\mathcal{R}$. Such protection could be achieved by centralized schemes [3, 10, 11] that introduce an anonymizer, a proxy between users and the LBS server that anonymizes user queries before sending them to the LBS server. However, the assumed trustworthiness of the anonymizer merely "shifts" the trust from the LBS server to the anonymizer, which obtains rich information the same way that the LBS server did and can also be honest-but-curious.

Decentralized k-anonymity [4, 12] eliminate the need of an anonymizer and protects user privacy in a collaborative manner: e.g., an obfuscated area is formed by k users within each other's communication range [4]. However, if such k users are too close, e.g., in a church, a shopping mall or a cinema, such symbolic "addresses" can still be disclosed. Thus, it is hard to define how large k should be to ensure an appropriate level of protection.

An alternative collaborative privacy protection approach is to pass/share LBS-obtained information among users, to decrease exposure to the LBS server [5, 13]. This is orthogonal to location obfuscation, both in terms of location and query privacy; the two could complement each other. The sharing approach requires nodes to cache information received from the LBS and pass it to neighbors when requested.

Nonetheless, opening up the system functionality is a double-edged sword: it reduces user exposure to the curious provider (LBS or anonymizer) but it also exposes her to possibly faulty or misbehaving peers. In fact, P2P systems [6, 9, 15] show that insecure decentralized schemes face serious problems. For example, sensitive information could be exposed or malicious nodes could share bogus data. Signed LBS server responses can be self-verifiable when passed to peers [13]. However, queries and cached information from different users could be diverse, making it necessary to share multiple complete LBS responses (each with a signature attached), even though only a subset of each LBS response might be needed by the querying peer. Moreover, it is hard to decide whether the peer responses cover the same information that could be received directly from the LBS server, thus, guaranteeing the quality of service. Last but not least, peer queries, openly submitted to a node's neighborhood, could expose users to other nodes and passive eavesdroppers.

These problems are exactly addressed in this paper: we propose a security architecture for decentralized/collaborative privacy protection for LBSs. We propose new components that are orthogonal to the LBS servers. We leverage pseudonymous authentication to provide privacy-enhancing message authentication and integrity for communication with other users (nodes) and with the infrastructure entities. We leverage proactive caching of POI data by a small fraction of users that serve others, sharing the cached POI data. This ensures peer responses can provide the same quality as LBS responses do. The burden is balanced among users through a

WiSec '17 , Boston, MA, USA

DOI: 10.1145/3098243.3098268

periodical randomized role assignment by the infrastructure. While users benefit from the information sharing system, we minimize their exposure to the LBS server and curious peers, and thwart security threats (from active malicious peers). Our evaluation shows both effective exposure reduction and highly successful valid POI provision in a realistic intelligent transportation setting even with a huge (in practice) fraction (e.g., 20 %) of malicious nodes.

The rest of the paper is organized as follows: we explain the system and adversarial models, and requirements in Sec. 2. Then, we present the proposed scheme in Sec. 3. Sec. 4 begins with a qualitative analysis on security and privacy, followed by a quantitative, simulation-based evaluation before we conclude (Sec. 5).

2 SYSTEM MODEL AND REQUIREMENTS

System Model: Fig. 1 shows the considered system architecture. Mobile devices (termed *nodes* in the rest of the paper), e.g., smartphones and vehicular On-board Units (OBUs), are equipped with various communication interfaces, e.g., Wi-Fi and cellular. They can access LBSs, submitting queries regarding their current locations/regions. They also communicate in a P2P manner over a wireless ad hoc (e.g., IEEE 802.11p) or cellular (e.g., LTE direct) network. Nodes can share POI information and choose to query the LBS server only when no response is received from their peers. Nodes are registered with an *identity and credential management* system (i.e., a Public-Key Infrastructure (PKI)). Certification Authorities (CAs) (see Sec. 3 for more details) issue credentials to the registered nodes and the Service Providers (SPs) (i.e., LBS servers here), so that SPs and nodes can interact securely.

Adversary Model: We assume LBS servers are honest-but-curious: they follow the protocols, responding faithfully to queries, but they may trace the nodes (linking their queries) or even deanonymize them and infer sensitive data (e.g., home and work sites). Queries sent to the LBS servers expose user locations and interests, and can be used to infer sensitive data. We maintain the honest-but-curious assumption for any trusted third party, including the ones we introduce in our scheme (Sec. 3).

Nodes can be also honest-but-curious or outright compromised. The P2P interactions allow nodes in the system to aggressively collect all the peer queries and responses. Such transcripts from multiple honest-but-curious nodes could be merged and used by the adversary. Furthermore, nodes can deviate from the collaborative protocol functionality and policies, and attack the system, notably their peer nodes. They can forge or tamper with responses, and masquerade other nodes. This could, in turn, affect quality of service and force honest nodes to expose themselves to the LBS server(s).

Requirements: We require that peer-provided information be verifiable and the nodes be accountable for their messages. The nodes should be able to efficiently obtain POI data from their peers with the same quality as that obtained directly from the LBS server. While the nodes benefit from P2P POI sharing, node exposure to neighboring assisting peers (and definitely, to the LBS server) should be minimized. Towards this, the following security and privacy requirements need to be met:

Authentication and integrity - Node messages should allow their receivers to corroborate the legitimacy of their senders and verify they were not modified or replayed.

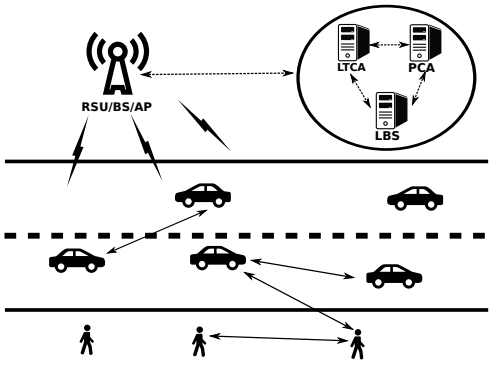

Figure 1: System Architecture (Icons by Freepik, freepik.com)

Accountability - Message senders cannot deny having sent a message (non-repudiation). Any node can be tied to its actions, and, if need arises, be held accountable and possibly evicted.

Anonymity/Pseudonymity and unlinkability - Node actual (long-term) identities should not be linked to their messages. Anonymity should be conditional, allowing the system to identify a misbehaving node and evict it. Ideally, it should be impossible for any observer to link any two messages (e.g., queries) by the same node. However, for practical reasons, messages can be linkable at most over a protocol selectable period, τ.

Confidentiality and reduced exposure - POI data should be accessible only by legitimate participants. Sensitive information (e.g., node queries) should be accessible only by authorized entities, and the amount of information revealed to peers and the LBS server should be minimal.

Resilience - Nodes should be able to validate authenticated information to reject bogus POI data from malicious nodes.

Sybil-resistance - A node should be able to participate only with a single identity (pseudonym) at any point in time.

3 OUR SCHEME

Our decentralized privacy protection scheme design is driven by privacy, resilience and efficiency considerations. Our approach significantly extends P2P LBS privacy schemes [5, 13] with the following main ideas: (1) Each node is equipped with short-term anonymous credentials, used to authenticate all node-to-node (and node-to-LBS) interactions. (2) Peer-provided POI can be drawn from a larger volume of POI data, proactively distributed by the LBS server to a small fraction of randomly chosen nodes, termed *serving nodes*. (3) Nodes submit queries to serving nodes, which periodically announce their presence and available POIs (i.e., POI for their regions). Table 1 summarizes the used notation.

3.1 Pseudonymous Authentication

The LBS server and all nodes are registered with an *identity and credential management facility*. A registered node is issued with a Long-Term Certificate (LTC), used as the long-term node identity, by the Long-Term Certification Authority (LTCA). With the LTC (and the corresponding private key), a node can obtain a *ticket* from the LTCA to be used towards obtaining pseudonyms from a Pseudonymous Certification Authority (PCA). *Pseudonymous*

Table 1: Notation

LTCA	*Long-Term Certification Authority*
PCA	*Pseudonymous Certification Authority*
RA	*Resolution Authority*
LTC	*Long-Term Certificate*
PC	*Pseudonymous(/Short-term) Certificate*
N	*Maximum peer requests per LBS query*
T_{beacon}	*Beacon interval*
T_{POI}	*POI update interval*
Γ/T_{serve}	*Pseudonym request interval/Serving period*
τ	*Pseudonym lifetime*
Pr_{serve}	*Probability of serving node assignment*
T_{query}	*Average query interval*
T_{wait}	*Waiting time before requesting LBS*

Certificates (PCs) (or, for simplicity, *pseudonyms*) can be used to authenticate the nodes to other entities in the system, and establish secure communication channels. Messages are signed with the private keys corresponding to the pseudonyms. All the attached signatures must be verified before the messages can be processed. *Ticket*s are authenticated by the LTCA but *anonymized*: they do not reveal the real identity of the node [7]. Therefore, a single LTCA or a single PCA cannot link the real node identity (i.e., LTC) to the issued pseudonyms (and the corresponding messages signed under the pseudonyms). We adopt the privacy-enhancing pseudonym issuance policy proposed in [7] for pseudonym unlinkability. *Conditional anonymity* allows *revocation of anonymity* and *eviction* of detected misbehaving nodes with the help of a Resolution Authority (RA) [7]. A benefit of leveraging such an approach for identity and credential management is that it has been investigated and adopted by industry and standardization bodies.

3.2 Privacy-enhancing LBS

We mandate proactive caching of POI information by a small fraction of selected nodes, termed *serving nodes*. A *serving node* is responsible for requesting POI data from the LBS server, store this information locally and serve neighboring nodes' (peer) queries. The role (i.e., serving or non-serving node) assignment is done by the PCA at the time of pseudonym acquisition: the PCA assigns a pseudonym requester as a serving node with probability Pr_{serve}. This role is explicitly visible at each issued pseudonym through an attribute set accordingly. To balance the workload among the nodes, a randomly assigned serving node would only have to serve at most for a protocol selectable period T_{serve}, which can coincide with the pseudonym request interval Γ (the period covered by the lifetimes of the set of obtained pseudonyms).

We assume the whole area (e.g., a city under the same LTCA) is divided into (equally sized) regions. A serving node is responsible for requesting all POI data for the region it is located in. Whenever it enters a new region, it has to request the POI data for that new region. Moreover, we assume POI information is refreshed every T_{POI} period. Therefore, whenever a POI refresh point is reached, serving nodes have to request the updated POI data.

A serving node broadcasts beacons every T_{beacon}. A beacon, signed and with the pseudonym attached, includes the identifier of the region of the serving node and the expiration time of the corresponding POI data. An interested node listens to beacons in the network for T_{wait} at maximum. If a beacon from a serving node in the same region is received, it sends a P2P query to the serving node: the node generates a session key and it encrypts the pseudonym and the query with the session key. Then, the session key is encrypted with the public key in the serving node's pseudonym. The whole message (including the encrypted session key, and the encrypted pseudonym and query) is signed by the querying node. The serving node decrypts the session key (with its private key) to decrypt the pseudonym and the query. Once the message is verified, the serving node generates the response and encrypts it with the same session key. The encrypted response is then signed by the serving node and sent back to the querier.

To protect the querying node from accepting false information from malicious serving nodes and reveal such misbehavior, the querying node can query $N > 1$ (discovered) serving nodes, with the same query within a T_{wait} period. Each serving node has cached the same POI for that region, thus their responses to that query should be the same (i.e. a given search on the same data should return the same result). A query succeeds if at least one serving node is discovered and queried. POI from additional serving nodes can be used for cross-checking: any conflicting responses can be reported (with the originally attached signatures and pseudonyms) to the RA. The RA checks with the LBS server the correctness of the responses and reveals any misbehaving node(s) through pseudonym resolution [7]. If a beacon from a non-serving node is received, this is also considered as misbehavior and reported to the RA. Without such a report, the RA cannot initiate a resolution (with a self-evident discrepancy).

If no serving node is discovered until timeout (i.e., no beacon from serving nodes in the same region is received within T_{wait}), the node queries the LBS server directly. The LBS server is also issued an LTC and uses it to authenticate itself to the nodes. The LBS server and the nodes are registered with the same PKI architecture, so that the LBS server can authenticate the nodes. Moreover, we assume the responses to the LBS-submitted queries are signed by the LBS server. However, the responses form the serving nodes need not carry the LBS signatures.

4 SECURITY AND PRIVACY EVALUATION

4.1 Qualitative Analysis

Authentication and integrity: Entity and message authentication, and message integrity are achieved with message signature verification. Message (e.g., beacon) timestamps prevent replays over significant periods. Queries are encrypted and bound to a specific serving node, thus they cannot be meaningfully replayed and any other serving node would fast reject them. More important, the query identifier can trivially allow the (given) same serving node to reject, and not serve replayed attacks, at the expense of modest local memory for overheard recent queries.

Accountability: Upon detection of misbehavior reported to the RA, the RA can reveal actual, long-term real identity of the misbehaving node through pseudonym resolution [7] and possibly evict the node from the system.

Confidentiality: Communication among the nodes and the infrastructure entities (LTCA, PCA and LBS) is kept confidential by using public key cryptography and (symmetric) session keys.

Sybil-resilience: The LTCA and the PCA issue $tickets$ and pseudonyms with non-overlapping lifetimes [7], ensuring a node is equipped with only one valid pseudonym at any point in time.

Response validation (Resilience): Our scheme prevents malicious nodes from aggressively providing false information. First, each node is chosen as serving node probabilistically by the PCA and such role is explicitly bound to the provided pseudonyms. Therefore, a malicious node has no control over becoming a serving node, the only possibility to provide false information to its peers. For example, when $Pr_{serve} = 0.05$, an adversary has to wait for $10 * T_{serve}$ on average before being selected as a serving node. Second, a malicious node has to be chosen by the querying node. Again, this selection is not controllable by the adversary. There may exist multiple serving nodes around the querying node that has the initiative to choosing one or multiple (N) serving node(s). The use of multiple (redundant) serving nodes can reveal a malicious node by cross-checking their responses, given all (benign) serving nodes have the same POI data for the region. [1] A malicious serving node could increase the beacon rate trying to increase the probability to be chosen by a querying node. However, such behavior (i.e., abnormally high beacon rates) could be easily detected and reported to the RA by nearby benign nodes.

Table 2: Linked quereis for different collusion cases

Case	*Linked queries*	*Colluding entities*
C_1	Same Id_{PC}	No collusion with CA
C_2	Same Id_{ticket}	Collusion with PCA
C_3	Same Id_{LTC}	Collusion with PCA and LTCA

Exposure reduction: Our scheme reduces the exposure to the LBS server by sharing information among the peers, as its predecessors [5, 13] did. In addition, through controlled selection of serving nodes and encrypted P2P interaction, only selected serving nodes learn queries. Queries by one node, V, to a certain serving node, S, can be linked while V uses the same pseudonym. Moreover, V could choose among several S_i nodes even within a given region for successive queries. Mobility of all nodes (serving or not), short-lived pseudonyms, and rotating assignment of serving nodes minimize exposure to any curious serving node acting alone; also thanks to encrypted queries that prevent eavesdropping of other P2P exchanges. However, colluding serving nodes could merge the queries they received, attempting to link them the same way the curious LBS server would do. Moreover, collusion with the CAs could further linking queries by linking pseudonyms of the same node. The RA is not able to initiate a pseudonym resolution without any reported (and confirmed) misbehavior. Table 2 shows the queries that can be linked for different cases. We provide quantitative evaluation on node exposure for different collusion cases in Sec. 4.2. We refer to [7] for the information disclosed to honest-but-curious PKI entities.

[1] Our scheme identifies adversarial serving nodes based on response cross-checking. However, inconsistent honest responses may occur when an LBS query coincides with a POI refresh point, t. For example, if the first response is received before t, and the second response is received after t, then the two responses could be different. To avoid this, an interested node is allowed to query immediately if the remaining time until t (that can be learned from beacons) is larger than a protocol selectable period ($T_{wait}/2$ in our simulation in Sec. 4), and any query should conclude by t (thus, a shorter waiting time than T_{wait} for the LBS queries that started within $[t - T_{wait}, t - T_{wait}/2]$). Otherwise, it will wait until t to query for fresh POI information.

4.2 Quantitative Analysis

We further evaluate our scheme through simulations. Exposure to the LBS server and honest-but-curious nodes is quantitatively evaluated through two metrics: peer hit ratio and exposure degree. The resiliency of our scheme in the presence of malicious nodes is also evaluated.

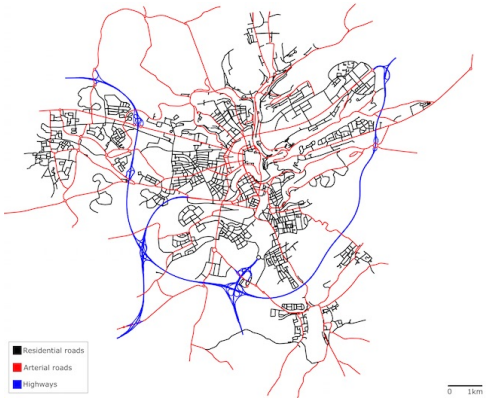

Figure 2: LuST Scenario Topology [2].

Table 3: Simulation Parameters (*Bold* for Default Settings)

Pr_{serve}	0.02, 0.04, ***0.06***, 0.08, 0.1, 0.12
$Ratio_{adv}$	0.05, 0.1, 0.15, ***0.2***, 0.3, 0.4, 0.5

Simulation setup: We simulate our scheme with OmNET++ [1] and the TraCI mobility interface in Veins [14], connected to the SUMO [8] traffic simulator. We use the Luxembourg SUMO Traffic (LuST) internal mobility scenario (i.e., with source, destination, or both points internal to the city) [2] in a 14 $km \times 12$ km area. We assume a penetration ratio of 40%, i.e., 40% of the mobile nodes use LBSs and participate in our collaborative scheme. We use the traces for 12:30 pm – 2:00 pm period and use the 1:00 pm – 2:00 pm part for the evaluation. We assume ideal wireless connection for ad-hoc node-to-node communication, and a range of 200 m. The results are averaged over 5 seeded simulation runs. Fig. 2 shows the LuST scenario topology. We divide the whole area into 2 $km \times 2$ km equally sized regions, thus a 7×6 gridded area. Table 3 shows the parameters of our simulation, bold values indicate default simulation settings. We set the rest of the parameters as: $T_{serve} = \Gamma = 10$ min, $\tau = 5$

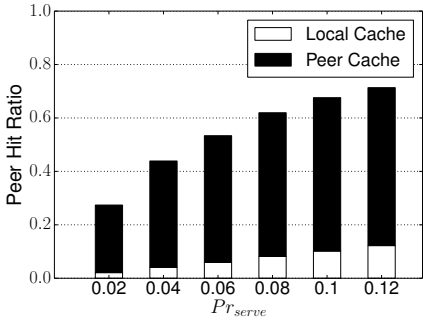

Figure 3: Peer (incl. own cache) hit ratio as a function of Pr_{serve}

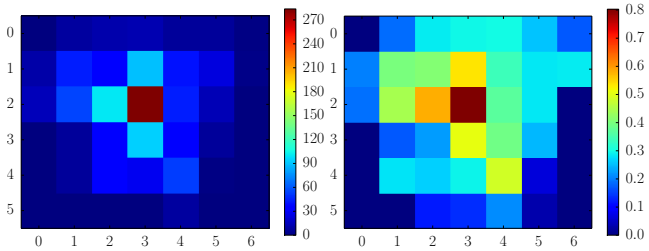

Figure 4: (a) Node density for each region with LuST scenario at 1 pm. (b) Peer hit ratio for each region under the default settings (see Table. 3).

min, $T_{POI} = 20\ min$, $T_{wait} = 60\ s$, $T_{beacon} = 10\ s$, $T_{query} = 3\ min$ and $N = 3$.

Peer Hit Ratio: This is a node exposure measurement, reflecting the degree of query privacy in terms of disclosed node interests. The peer hit ratio is defined as the ratio of LBS queries responded using local or peer caches (of serving nodes), while the remaining is responded by the LBS server itself. Fig. 3 shows the peer hit ratio as a function of Pr_{serve}: it improves with Pr_{serve}, e.g., when $Pr_{serve} = 0.12$, the peer hit ratio is around 0.7. We see a significant increase from $Pr_{serve} = 0.02$ to 0.04, while such improvement becomes moderate for high Pr_{serve} values (e.g., from 0.1 to 0.12). This indicates a modest Pr_{serve} (e.g., 0.06 or 0.08) is enough to hide a significant amount of queries from the LBS server.

Peer hit ratios also depend on node density: higher density results in higher probability of discovering serving nodes. Fig. 4a shows the node density (nodes per $2\ km \times 2\ km$ region) at 1 p.m. for the LuST scenario, and Fig. 4b shows a map of the peer hit ratio under the default settings (see Table 3) for different regions. As expected, the peer hit ratio is roughly proportional to the node density. In the central area, the peer hit ratio exceeds 0.5 and exceeds 0.8 in the densest region; the higher the node density, the higher the exposure reduction. For a low-density region, a local relative increase in Pr_{serve} could increase the peer hit ratio with a modest overhead increase.

Node Exposure: We refine the measurement of node exposure to curious LBS servers and curious nodes, defining the *exposure degree* of a node as:

$$ExpoDeg(Id_{LTC}, C) = \sum_{Id_i \in ID(Id_{LTC}, C)} \frac{T(Id_i)}{T(Id_{LTC})} * \frac{R_H(Id_i)}{R(Id_{LTC})}.$$

Id_{LTC} is the long-term identity of a node, corresponding to a whole series of node actions in the system. $ID(Id_{LTC}, C)$ is a set of identities, corresponding to Id_{LTC}, exposed to the honest-but-curious (possibly colluding) entities for collusion case C (Table 2). $ID(Id_{LTC}, C)$ differs for different collusion cases. $T(Id_i)$ is the corresponding trip duration of a node under identity $Id_i \in ID(Id_{LTC}, C)$. $R(Id_i)$ is the number of regions the node visits during its trip under identity Id_i and $R_H(Id_i)$ is the number of visited regions exposed to honest-but-curious entities under the same identity Id_i. $\frac{R_H(Id_i)}{R(Id_{LTC})}$ indicates the exposure degree under a single identity Id_i. To derive the exposure degree of a node, the exposure degrees under each Id_i are weighted by a time parameter $\frac{T(Id_i)}{T(Id_{LTC})}$: the ratio of the (partial) trip duration under identity Id_i over the total trip time. The exposure degree indicates the accuracy of reconstructed node trajectories based on recorded node queries, taking into consideration the effect of pseudonymous authentication on location privacy protection. We measure the node exposure to the colluding curious nodes through the aggregation of the recorded queries.

Fig. 5a shows the average exposure degree (i.e., the average of all exposure degrees of the nodes) to the LBS server as a function of Pr_{serve} for different collusion cases. The collusion case C_3 is equivalent to the case that messages are authenticated with nodes' LTCs (i.e., no pseudonymous authentication). $Pr_{serve} = 0$ is equivalent to the case that all queries are sent to the LBS server. The exposure degree is around 0.6 without any protection in place: even if all the queries are sent to the LBS server, the exposure degree is not 1 because a node enters and exits one or more regions between two successive queries. With pseudonymous authentication only, the exposure degree drops below 0.3 and bounces back to around 0.4 for the collusion case C_2. With the decentralized information sharing scheme in use, the exposure decreases further. For example, when $Pr_{serve} = 0.06$, the exposure degree is around 0.15 for C_1, but rises to around 0.37 for C_3.

We evaluate the exposure degree as a function of $Ratio_{adv}$, the ratio of adversarial nodes that merge recorded queries. In Fig. 5b, modest realistic $Ratio_{adv}$ (e.g., 0.05 and 0.1) result in relatively low exposure degrees. For example, when $Ratio_{adv} = 0.05$, exposure degrees are lower than 0.05 for C_1 and C_2, and is slightly higher for C_3. However, without P2P encryption (Fig. 5c), the exposure degree significantly increases, because all queries within an adversarial (serving or non-serving) node's communication range can be recorded. For example, exposure for $Ratio_{adv} = 0.05$ and $Ratio_{adv} = 0.1$ without encryption are almost same as those for $Ratio_{adv} = 0.3$ and $Ratio_{adv} = 0.5$ with encryption, respectively. This shows the importance of query encryption in terms of reducing node exposure.

Resilience: Although a ratio e.g., 20 %, of adversarial nodes (thus, node owners) is unrealistic, we consider such rather extreme cases to capture situations with nodes infected by malware, while the node owners are benign. [2] In the simulation, we assume the

[2]Once misbehavior is reported and the malicious nodes are identified, node owners can be notified and malicious nodes be reinstated as benign nodes through, e.g., diagnostics

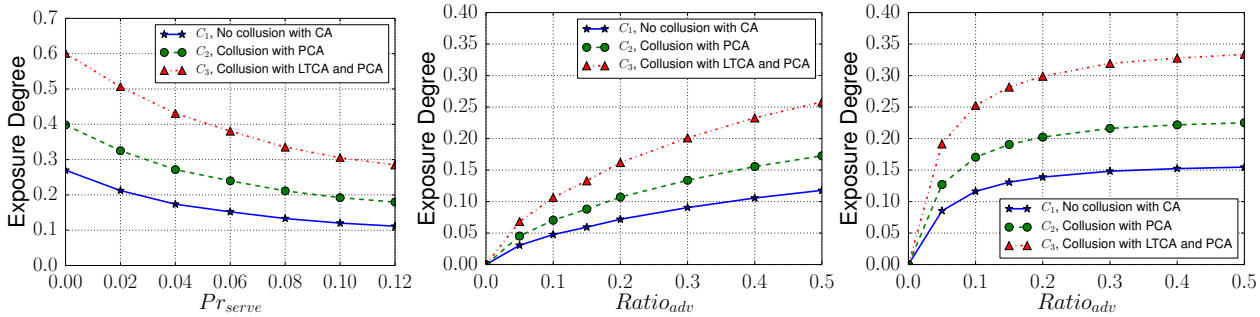

Figure 5: (a) Exposure degree to the LBS server as a function of Pr_{serve}. Exposure degree to different collusion cases (see table 2) as a function of $Ratio_{adv}$ (b) with and (c) without encryption for P2P communication.

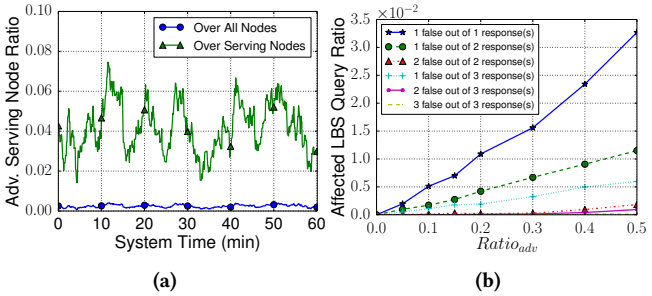

Figure 6: (a) Malicious serving node ratio during simulation (1 p.m. - 2 p.m.) with default settings. (b) Affected LBS query ratio as a function of $Ratio_{adv}$.

malicious serving nodes collude to provide identical false response to a same LBS query in order for the false response to be accepted by the querying node.

Fig. 6a shows the ratio of the adversarial serving nodes over time. Even though 20 % of the nodes are compromised, the ratio of non-detected malicious serving nodes (over all serving nodes) is always less than 8 %. The fluctuation of this ratio is due to the eviction of detected malicious nodes and the mobility of (joining and leaving) nodes. Without misbehavior detection (i.e., cross-checking), the ratio would have roughly remained the same as $Ratio_{adv}$ (i.e., 20 %). However, the controlled selection of serving nodes effectively limits active participation of malicious nodes, and the cross-checking mechanism further helps to detect and evict them from the system. Moreover, the ratio of "active" malicious nodes (i.e., ones that could actually provide false responses) is always less than 0.3 %: a significant decrease from the original $Ratio_{adv}$. This shows that even if 20 % of the nodes are compromised (by a "master" adversary), the ratio of actual usable compromised nodes is considerably lower (i.e., lower than 0.3 %). This is reflected on the ratio of the affected LBS query in Fig. 6b: for example, when $Ratio_{adv} = 0.2$, around only 1 % of the LBS queries are served by false P2P responses that are not detected (this happens when all P2P responses to an LBS query are given by malicious nodes, thus, no conflict). If conflicting responses are received, the querying node checks the correctness with the RA and reports the misbehavior accordingly (Sec. 3).

and updates. A recovered benign node would be issued a new LTC and obtain new pseudonyms.

5 CONCLUSION

Our approach extends the recent P2P LBS privacy protection approach, addressing a number of practical open issues. More importantly, it ensures resilience to misbehaving peers and low exposure to curious peers and LBS servers even if they collude with the curious identity management facility. We show that the exposure to curious nodes is low even if 20 % of nodes are compromised and collude, while the same ratio of active malicious nodes could only affect 1 % of the peer-responded LBS queries.

REFERENCES

[1] OMNeT++. https://omnetpp.org/.
[2] L. Codeca, R. Frank, and T. Engel. Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In *IEEE VNC*, Paderborn, Germany, Dec. 2015.
[3] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE TMC*, 7(1):1–18, 2008.
[4] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In *SSTD*. Boston, MA, July 2007.
[5] H. Jin and P. Papadimitratos. Resilient collaborative privacy for location-based services. In *NordSec*. Stockholm, Sweden, Oct. 2015.
[6] M. Johnson, D. McGuire, and N. Willey. The evolution of the peer-to-peer file sharing industry and the security risks for users. In *HICSS*, Waikoloa, Hawaii, Jan. 2008.
[7] M. Khodaei, H. Jin, and P. Papadimitratos. Towards deploying a scalable & robust vehicular identity and credential management infrastructure. In *IEEE VNC*, Paderborn, Germany, Dec. 2014.
[8] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.
[9] S. H. Kwok, K. R. Lang, and K. Y. Tam. Peer-to-peer technology business and service models: risks and opportunities. *Electronic Markets*, 2002.
[10] S. Mascetti, C. Bettini, D. Freni, and X. S. Wang. Spatial generalisation algorithms for lbs privacy preservation. *Journal of Location Based Services*, 2007.
[11] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *VLDB*, Seoul, South Korea, Sept. 2006.
[12] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran. Amoeba: Robust location privacy scheme for vanet. *IEEE JSAC*, 25(8), 2007.
[13] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J.-P. Hubaux. Hiding in the mobile crowd: Location privacy through collaboration. *IEEE TDSC*, 11(3):266–279, 2014.
[14] C. Sommer, R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE TMC*, 10(1):3–15, 2011.
[15] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, and S. Chien. A first look at peer-to-peer worms: threats and defenses. In *Proceedings of the 4th International Conference on Peer-to-Peer Systems*. Konstanz, Germany, Aug. 2005.