Establishing Location-Privacy in Decentralized Long-Distance Geocast Services

Martin Florian, Felix Pieper, Ingmar Baumgart
Institute of Telematics, Karlsruhe Institute of Technology (KIT)
76131 Karlsruhe, Germany
Email: {florian,baumgart}@kit.edu, felix.pieper@web.de

***Abstract*—The ability to communicate over long distances is of central importance for smart traffic applications like cooperative route planning or the discovery and reservation of charging stations for electric vehicles. Established approaches are based on centralized architectures with singular service providers. This setup leads to strong privacy concerns, as great amounts of sensitive location data need to be stored at a non-local, centralized entity. Decentralized approaches like the overlay-based geocast service *OverDrive* propose to solve this issue by eliminating the central data sink and sharing location information with a small subset of other participants. In this paper, we propose techniques for further improving the location privacy offered by decentralized long-distance geocast services. Through obfuscation of location data and mechanisms for detecting location spoofing attempts, we can ensure that precise location data is only shared with participants in the physical vicinity. Simulation results show that our extensions render both the large scale surveillance and the targeted tracking of OverDrive users unfeasible even for strong adversaries controlling hundreds of overlay nodes.**

I. Introduction

The availability of Internet access in vehicles offers a variety of new opportunities to assist road users that are not easily realized with short range vehicular networking approaches. Examples include smart-traffic applications like cooperative route planning, where vehicles exchange traffic information for improving route planning decisions, or the localization and reservation of charging stations for electric vehicles. Long-distance communication is also important for vehicular cloud applications like [4], where vehicles act as service providers to which location-based service requests need to be propagated. Current solutions mainly follow a centralized, server-based approach for communication, which, among scalability concerns and the tendency of creating dependencies on individual providers, raises strong privacy concerns. As all communication and service provision is handled by the service provider, he also needs to collect all sensitive user information required for realizing the service. In the context of smart traffic, this specifically includes location data, which was found to enable far-reaching insights into the private life of users [5].

Decentralized overlay networks providing *geocast* services [6], [11] are a recent development with significant potential for resolving these drawbacks. Roughly, the idea is the creation of a logical *overlay network* on top of a cellular communication network based on the *Internet Protocol (IP)*. In the overlay network, nodes propagate their location to other participating nodes and use this information for choosing overlay neighbors and forwarding messages. Thus, neither a central entity nor additional infrastructure support is necessary. With *OverDrive* [6], this approach was specifically adapted to smart traffic scenarios. The evaluation of OverDrive showed [6] the capability to address the scalability and innovation issues of traditional centralized systems. Follow up works [3] demonstrated a series of possible attacks on the OverDrive approach that could allow a strong adversary to break individual pseudonyms with context knowledge and track identified targets.

In this paper, we introduce two techniques for negating such attacks. Our contributions aim at establishing *data locality*, i.e., ensuring that precise location data is only shared with entities that are physically located in the close vicinity. Specifically, we make the following contributions:

1) An *obfuscation* mechanism for OverDrive, that reduces the precision of location data in relationship to the distance at which it is shared. Our approach is resistant to intersection attacks resulting from the combination of data points from multiple observers.
2) Data locality cannot be established if participants can fake their location. Thus, we propose a *location spoofing detection* mechanism based on private proximity testing [9], [8]. Our solution is based on high-entropy data collected from a GSM network and allows proximity checks over multiple kilometers of obstructed terrain.
3) A thorough evaluation of our solutions as extensions to the overlay-based geocast service OverDrive. Through simulation, we evaluate their impact on possible privacy attacks as well as their impact on performance.

While developed with an application in OverDrive in mind, the proposed techniques are widely applicable to other decentralized systems in which location-data needs to be shared with possibly non-local entities.

II. Related Work

Privacy issues in vehicular networking have been be studied thoroughly, focusing mostly on short-range communication and vehicular ad-hoc networking (VANET) scenarios [2], [10]. However, local one- and few-hop communication is insufficient for realizing applications depending on long-distance communication. For realizing these types of services, the existence of dedicated infrastructure support or a trusted centralized service provider is usually assumed. Decentralized

approaches have been proposed, e.g., for traffic information systems [13], but without a serious consideration of location privacy issues.

Location privacy has been a major topic in the context of location based systems (LBS) in general. [14] gives an excellent overview over different techniques, including multiple obfuscation approaches. However, existing approaches are focused on centralized setups and not directly applicable to decentralized systems. Also, location privacy techniques from the LBS domain often lose their efficiency when confronted with continuous updates as required by smart-traffic applications. If location updates can be linked into routes, a subsequent linking to user identities is possible [5].

In [7], the authors describe an approach for the privacy preserving collection of continuous location updates in a vehicular traffic scenario. Location updates are communicated only when a vehicle passes a previously determined virtual trip line. Thus, a spatial sampling of the passed routes is achieved. However, the approach is only suitable for the privacy-preserving collection of location-specific data, e.g., floating car data, and not for the realization of geocast.

In [3], the privacy characteristics of overlay-based geocast services are analyzed and evaluated, leading to the discovery of possible attacks. No detailed countermeasures were proposed or evaluated. The reduction of the precision of shared location information as well as the protection against location cheating remained open questions. Solutions for the latter exist that require additional infrastructure support or spot checks [12]. If the problem can be reduced to proximity checks, short range radio beacons can be used, as well as private proximity testings mechanisms as proposed in [9] and [8]. However, no approach for integrating any of these solutions into a long-distance geocast system has previously been proposed.

III. OverDrive

A. Functionality

Originally proposed in [6], the main service provided by *OverDrive* is the delivery of messages to nodes in a given geographic region. Technically, OverDrive is based around two concepts:

- An overlay neighborhood structure based on a partitioning of geographic space into concentric *rings*, as well as mechanisms for maintaining this structure.
- A routing mechanism for forwarding messages to nodes in a desired geographic area. Messages are forwarded using connections from the overlay neighborhood structure.

An overview of the functioning of OverDrive is given in Fig. 1. The figure depicts a possible application for the geocast service, namely the sending of a geographic query to a point in geographic space (e.g., a road segment) lying ahead of the requester. From all of its neighbors, which are chosen based on a partitioning of geographic space into concentric rings, the requester greedily chooses the one neighbor that is closest to the destination region in terms of geographic distance. The request is sent via the cellular network and standard IP to this

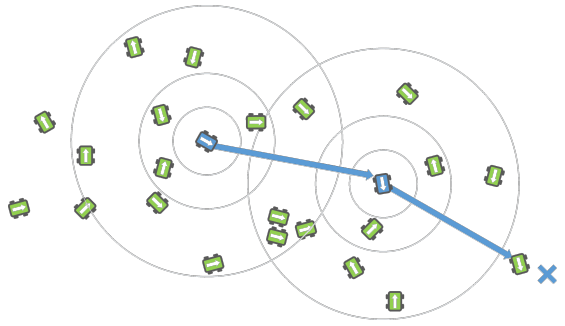


Fig. 1. Geographic routing and neighborhood structure in OverDrive.

neighbor, who then forwards it according to the same rule, sending it to the one of its overlay neighbors that is closest to the destination region. Once the message arrives at a node residing in the target area, that node might, depending on the application, decide to answer the query by directly sending a response (via IP) to the requester. The design of the overlay neighborhood structure is critical in regard to user privacy: For maintaining the neighborhood structure nodes need to continuously communicate their locations to their neighbors.

B. Use of Pseudonyms

As in other vehicular networking approaches [10], OverDrive nodes use pseudonyms to protect the true identity of their drivers. Nodes choose new pseudonyms at the beginning of each trip. As an additional protection measure, the changing of pseudonyms during trips, or mixing, can be implemented as well. It is important to note that as OverDrive nodes communicate with each other over IP, their IP addresses are changed as well during pseudonym changes in order for the unlinkability between pseudonyms to be ensured. With pseudonyms for every participant, a central challenge for an adversary interested in large scale surveillance or targeted tracking becomes the linking of pseudonyms to real-world identities, i.e., the *breaking* of pseudonyms.

IV. Attacker Model

For evaluating the effect of the techniques proposed in this paper, we adapt the attacker model proposed in [3].

A. Assumptions and Attacker Goal

Our attacker model is based on following assumptions:

- The attacker is able to control multiple *attacker nodes* in the geocast network. Sybil attacks are not possible and the maximum number of attacker nodes is limited.
- Attacker nodes are able to lie about their position. Apart from that, they run the OverDrive protocol like regular nodes.

The goal of the attacker is to trace the movement of a real-world entity that uses OverDrive. This involves two steps: the linking of the victim's identity to a pseudonym and the subsequent collection of location updates from that node, thus keeping the victim under surveillance.

B. Establishment of a Global View

A straightforward attacker approach is the establishment of a global view over the whole overlay network including the pseudonyms and locations of a large number of regular nodes. Given a global view, pseudonyms can be broken using known techniques, e.g., using context knowledge about the victims ([5]). The establishment of a global view is possible using the following attack:

1) The attacker controls multiple attacker nodes that behave like regular traffic participants. As attacker nodes can lie about their location, no physical nodes need to be involved and all node movement can be simulated by the attacker.
2) The attacker nodes attempt to become overlay neighbors with as many regular overlay nodes as they can.
3) The attacker nodes forward all location updates they receive to the attacker who combines them into one global view.

C. Surveillance of an Individual Target

A more sophisticated attack approach is the exploitation of inherent properties of the geocast system for identifying and tracking specific targets with far less resources. In the following, we present a representative attack of this class. The approach assumes that the attacker targets one user about which he has *context knowledge* in the form of the location at which he will start his trip.

The attacker first attempts to map a victim to an overlay node by placing attacker nodes around the start point of the victim (e.g., by instructing them to lie about their positions). The attacker nodes report all new nodes they discover to the attacker. Whenever a new node X is discovered in the vicinity of the victim start position, it is likely that it has just started its trip. Having just started its trip in the vicinity of the victim start point, the attacker marks it as potentially belonging to the victim. In [3] it was found that with this reasoning, victim nodes are correctly identified in around 90% of cases. Having acquired a likely victim node, the attacker can track it using a *follower attack*. Here, one attacker node continuously fakes its position so as to appear in the vicinity of the victim node. Being in the vicinity of the victim node, it is very likely to remain in the victim node's neighborhood and continuously receive location updates from it.

V. Location Privacy Enhancements

Based on the discovered weaknesses of overlay-based geocast services in terms of the protection of location privacy, we now propose two enhancements that tackle the major problems enabling attacks like the ones described in Sec. IV. For one, we develop a mechanism that decreases the accuracy of the location information an attacker is able to acquire (*location obfuscation*). Secondly, we develop a countermeasure against malicious nodes that fake their location information in order to receive detailed location updates from targets (*location spoofing detection*).

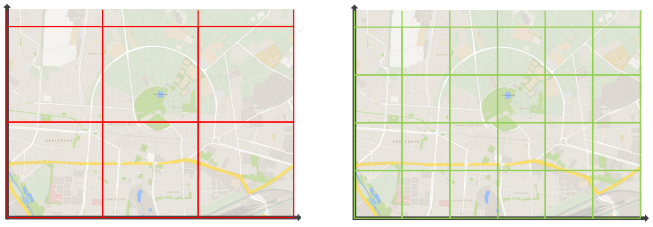
A. Location Obfuscation

For establishing a global view on the location of all nodes, an attacker's goal is to gain as accurate location information as possible about as many nodes in the network as possible. Therefore, an efficient way to defend against this attack is to avoid delivering accurate location information to the attacker. This does not prevent an attacker from collecting location data about many nodes but it will decrease the value of the collected information, e.g., its suitability for breaking pseudonyms or determining the exact location of a given node.

1) General Approach: Our basic approach is to decrease the accuracy of the location information shared between two nodes A and B with growing distances between those nodes. OverDrive uses a greedy forwarding algorithm where each node forwards a message to the node that it believes to be closest to the destination location of the message. As in key-based routing schemes, the distances between individual hops decrease with each routing step and the distance to the destination decreases in smaller and smaller steps. Thus, our proposed enhancement is not expected to impact the performance of the geographic routing in a significant way.

2) Obfuscation Regions: Our obfuscation approach is based on the concept of *obfuscation regions*. An obfuscation region is a quadratic geographic region with an edge length of l_{edge}. Instead of transmitting precise location information, the nodes A and B share the center position of an obfuscation region they currently reside in. In order to allow different levels of obfuscation based on the distance between two nodes, the size of the obfuscation region can be varied. For denoting the desired degree of obfuscation, we define the *zoom level* z so that $l_{\text{edge}} = 2^z$.

3) Obfuscation Grid: Given the zoom level and the accurate location of a node, an obfuscation region can be constructed. If each node calculates its obfuscation region by choosing a random quadratic region around its position, an attacker might break the obfuscation by intersecting multiple views collected from different nodes under the attacker's control. To avoid this kind of attack, obfuscation regions must be constructed in such a way that the information gained from combining multiple received obfuscation regions for the same location never exceeds the information contained in the received obfuscation region with the lowest zoom factor.



(a) obfuscation grid at level z (b) obfuscation grid at level $z-1$

Fig. 2. Obfuscation grid.

To achieve this, we propose the use of an *obfuscation grid*. An obfuscation grid is a division of geographic spaces into

disjoint squares as shown in Fig. 2. Each of the squares represents a single quadratic obfuscation region as described in Sec. V-A2. Every node in the overlay uses the same origin for the obfuscation grid, regardless of the used zoom factor. Since $l_{\text{edge}} = 2^z$, each obfuscation region at zoom level z can be divided into four disjoint regions at zoom level $z-1$, as shown in Figures 2(a) and 2(b). Thus, obfuscation regions never intersect and two obfuscation regions for the same location are either identical (in case their zoom level matches) or the region with the lower zoom level is contained in the other.

4) Creating Obfuscation Regions: The information needed to calculate an obfuscation region is the location L that is to be obfuscated, as well as the zoom level z determining the size of the obfuscation region. The zoom factor is determined by the index r of the ring in which neighbor B resides. As a parameter to our system, we introduce the *downscaling factor* d so that $z = r - d$. Thus, if B resides in the r'th ring of A neighborhood structure, A will share its location with B using an obfuscation region with edge length $l_{\text{edge}} = 2^{(r-d)}$ (in kilometers). By varying the value for d, we can test different degrees of obfuscation. To calculate the correct obfuscation region using the given information, we first transform the latitude/longitude-based location L into the coordinate space of the obfuscation grid to facilitate the calculation, yielding the grid point L'[1]. Using L', the points P_{min} and P_{max} defining opposite corners of the resulting region can be calculated as:

$$P_{\text{min}} = \left(2^z * \lfloor \frac{x_{L'}}{2^z} \rfloor, 2^z * \lfloor \frac{y_{L'}}{2^z} \rfloor\right)$$

$$P_{\text{max}} = \left(2^z * \lceil \frac{x_{L'}}{2^z} \rceil, 2^z * \lceil \frac{y_{L'}}{2^z} \rceil\right)$$

Once these points are known, the center point of the obfuscation region can be calculated as

$$P_{\text{center}} = (x, y) = \left(\frac{(x_{P_{\text{min}}} + x_{P_{\text{max}}})}{2}, \frac{(y_{P_{\text{min}}} + y_{P_{\text{max}}})}{2}\right)$$

5) Determining the Ring Index: Whenever A wants to share location information with B, it has to determine the correct level of obfuscation to be applied to the location data. Simply calculating a ring index based on A's real position and B's reported position might lead to inconsistencies in cases where A is located close to a ring boundary. A's obfuscated position might not be within the same ring of B's neighborhood structure as A's real position. Thus, the correct ring index is dependent on the distance between B's reported position and the center of the correct obfuscation region for A (see Fig. 3). The correct obfuscation region is determined iteratively: For each ring index r beginning at 0, an obfuscation region for the corresponding zoom level is calculated. If the center of that obfuscation region lies within the ring with the current index r, this is the correct obfuscation region. If not, the check continues with the next highest ring index $r + 1$.

[1]The precise calculations based on the Haversine formula are omitted here for lack of space.

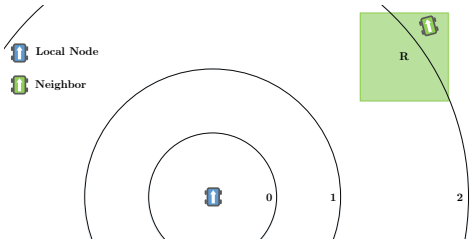


Fig. 3. The correct ring is 2, even though the actual location is in ring 3.

6) Neighborhood Structure and Neighbor Scoring: The general neighborhood structure concept remains the same as presented in [6] and Sec. III. Changes include the type of information shared with neighbors: instead of precise location, bearing and speed, nodes only share the center of the correct obfuscation region. Consequently, we also simplified the scoring function used to determine which nodes to add to the neighborhood structure. Scores are no longer based on bearing and speed, but only on the number of neighbors in the vicinity of the scored node (fewer neighbors in the vicinity lead to higher scores).

B. Location Spoofing Detection

In the following, we present an approach for identifying malicious nodes that spoof their location, so that adversaries need to to be physically close to their victims in order to receive precise location data.

1) Private Proximity Testing: Our approach is based on works on *private proximity testing*. Specifically, we use the *location tag* and *location sketch* concepts as proposed in [9] and [8]. A *location tag* is a set of features that are unique in space and time. The generation of a correct location tag for a location is only possible if an entity is physically present at that location. In [8], location tags are constructed from GSM broadcast traffic. By collecting *immediate assignment* (IA) messages, location tags specific to individual GSM *cells* can be constructed. Using signaling traffic from the *broadcast paging channel* (PCCH), the same is possible for GSM *location areas*, i.e., groups of multiple cells. By comparing location tags generated in this way, reliable proximity tests over distances of 10 km and more are possible.

A *location sketch* is a single value generated from a location tag using the *shingling* technique [8]. It enables the efficient comparison of location tags using *private equality testing (PET)*, i.e., verifying the equality of another party's location tag without either party needing to disclose its location or location tag. Here, we assume the use of a synchronous PET protocol based on El Gamal encryption as proposed in [9]. Due to space constraints, we will omit an in-detail explanation of location sketch generation and the PET protocol here. Our main focus in the scope of this paper is on the use of these techniques and their integration into OverDrive.

2) Integration into OverDrive: For enabling location spoofing detection using private proximity tests, OverDrive nodes need to continuously collect local GSM broadcast traffic - IA

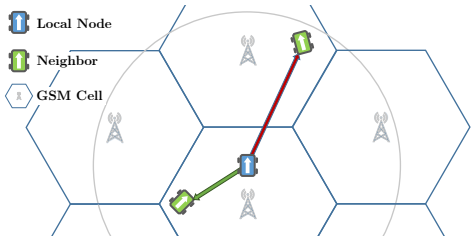


Fig. 4. Cell verification between a node and its neighbors.

messages and traffic on the PCCH. From the collected data they can create location sketches proving their location in a cell (using IA traffic) and location area (using PCCH traffic). Using PET, two nodes can check if their location sketches match without having to share the actual sketches. If their IA-based sketches match, the nodes assume that they reside in the same GSM cell and are therefore not significantly more than 4 km apart[2]. If only their PCCH-based sketches match, they assume to be located in the same location area and not significantly more than 10 km apart[2].

With a base ring radius of 2 km (as proposed in [6]), we propose that nodes use the location verification mechanism for neighbors in the three innermost rings of their neighborhood structure. For the innermost ring (i.e. for neighbors up to 2 km away), it will try to perform a *cell verification*, thus trying to verify that it is located in the same GSM cell as the neighboring node (see Fig 4). For nodes in the second and third ring (up to 4 km and 8 km away, respectively), *location area verification* is used. For more faraway nodes, no location verification is used. The location information shared with nodes in the outer rings is heavily obfuscated and thereby only of limited value to an attacker. Without verification, no neighbor receives location updates with a precision exceeding that of the location updates for ring 4. Likewise, if a node B shares a location that implies that it needs to be allocated to the innermost ring of a node A, but has only proven that it resides in the same location area as A, it only receives location updates with the precision corresponding to the second ring.

3) Verification Process: Each node A periodically performs checks about the verification status of all of its neighbors in its innermost three rings. If a cell or location area verification is pending for a neighbor B a *location verification request* is sent to it. The request message contains the node handle identifying node A as well as two encrypted location sketches according to the synchronous PET protocol outlined in [9]. One sketch is based on cellular-level broadcast data, the other on location area-level data. Upon receiving the verification request, neighbor B combines its own location sketches with the ones he received, according to the PET protocol. He sends the result of the operation back to A in a *location verification response*. Based on *B*'s response, A can now check if *B* is in the same location area or even in the same cell as itself.

Once the proximity to a neighbor is verified, more accurate location can be shared with him accordingly. The verification process is repeated periodically in order to protect against follower attacks. Specifically, without a periodic reverification of neighbors, an attacker needs to be physically close to his victim only once, after which he can track the victim's movement by faking his location.

4) Dealing with Identified Attacker Nodes: If, despite repeated attempts, a node A was not able to successfully verify its proximity to a node B claiming to reside within A's innermost 3 rings, the *maximum verification delay* will be reached. In this case, A assumes that B is a malicious node that has spoofed its location data. A then evicts B from its neighborhood structure and adds in to a list of identified malicious nodes, effectively ignoring any messages from B from that time on. After a *retention period*, B is removed from that list again. This approach prevents a malicious node from quickly regaining access to A's neighborhood, while at the same time minimizing the impact on falsely accused nodes.

5) Practical Considerations and Alternative Approaches: Based on the proof of concept provided in [8], we assume that the continuous collection of both IA and PCCH traffic and the efficient generation of location sketches from collected messages is possible for traffic participants. As the authors point out, however, changes to the GSM stack implementation might be necessary on client devices for the collection of the required broadcast traffic. An additional open question is whether the same networking interface used for data communication can be used for collecting GSM broadcast messages.

As an alternative to GSM-based private proximity testing, location tags can also be generated using military-grade GPS receivers as proposed in [9]. While very promising in terms of availability and the entropy of resulting location tags, commercially available receivers are not yet suited for resolving GPS signals at a sufficiently high precision. We decided against the use of radio-based proximity verification, e.g., via short-range radio beacons. Reasons include the short range of these techniques, as we would like to be able verify distances of 1 km and more and ideally up to 10 km.

VI. Evaluation

We implemented location obfuscation and location spoofing detection as extensions to OverDrive. Here, we present a detailed evaluation of these extensions focusing on privacy gains and performance impact.

A. General Evaluation Setup

In order to provide comparability of results with the ones in [6] and [3], the general setup of the testing environment is the same as in these publications. Our proposed enhancements are realized as extensions to the OverDrive prototype presented in [6]. Thus, our implementation is embedded into the OverSim simulation framework [1], which we also use for evaluation. For simulating mobile nodes (OverDrive-enabled vehicles), we use the mobility and communication models proposed in [6] and the highway network of the German state of Baden-Württemberg as an underlying road network. For

[2]These values can be fine-tuned with more specific information about the used GSM network.

each simulated parameter combination, we performed four independent simulation runs, each covering a period of 1400 seconds. Unless otherwise noted, the presented plots show average values over these runs, with error bars indicating 95% confidence intervals.

For assessing the impact of the Location Spoofing Detection mechanism on OverDrive, we implemented an abstract model for the GSM-based private proximity testing technique proposed in [8]. Specifically, we use *oracles* per node that, given a location tag, can determine if it was generated in the same GSM cell or location area as the node that the oracle belongs to. We use a hexagonal grid with a cell radius of 2 km to model the cell structure of the used GSM network, and a hexagonal grid with a cell radius of 5 km to model the partitioning of the network into location areas. This model represents a conservative approximation to real GSM networks, which have a high variance in cell sizes and location area span (both usually larger than in our model).

We evaluate the impact of our privacy extensions using an evaluation approach based on the one used in [3]. Specifically, we construct simulation models for the attack scenarios presented in Sec. IV: (1) the establishment of a global view with as accurate location information as possible about as many nodes in the network as possible and (2) the identification and tracking of a single victim using context knowledge.

B. Extension Parametrization

As a preliminary step to the evaluation of the proposed extensions, we conducted an extensive simulation study to determine a suitable parametrization of the enhanced OverDrive system that strikes a balance between privacy gain and performance impact. This is especially challenging considering the large amount of existing parameters and possible parameter combinations for OverDrive and the proposed extensions. In order to keep the simulation overhead at a reasonable level, we first determined suitable values for parameters that have no significant impact on the systems' privacy characteristics. This is, for example, the number of nodes a node will accept as neighbors. As in [6], we used a *performance versus cost* (PVC) evaluation to determine parameter combinations with a good trade-off between routing success and bandwidth consumption.

We then determined a suitable obfuscation level for the obfuscation extension, i.e. a value for the downscaling factor d. We considered the impact of the parametrization on the difficulty for an attacker to establish a global view on the network (see Sec. VI-C for details on the evaluation scenario). Our results confirm that the average error in the location information known to the attacker grows with the degree of obfuscation applied. However, using a high degree of obfuscation also tampers with the system's performance in delivering geocast messages. Based on our results, we settled on a downscaling factor of $d = 1$, leading to an improvement to the regular OverDrive design in both performance and attacker uncertainty. Lastly we also needed to find suitable parameters for the location spoofing detection extension. Here, our main optimization goal was to decrease the additional communication overhead while increasing the chance that two proximate nodes will correctly verify each other as such. A location verification can fail if the two nodes happen to reside in different cells or location areas despite of their proximity. Based on simulations, we settled on a parametrization in which nodes attempt a mutual verification every 15 seconds and consider a node malicious if the verification has failed for 150 seconds, i.e. after 10 attempts.

C. Establishment of a Global View

In the following, we present our evaluation of the difficulty for an attacker to construct a global view of the OverDrive network with the positions of all (pseudonymized) nodes.

1) Evaluation Scenario and Metrics: We simulate a network with 10000 mobile nodes and an additional population of *attacker nodes* that exhibit the same mobility pattern as regular nodes. We evaluated different sizes of the attacker node population up to a maximum of 100 nodes. We consider all attacker nodes to be under the control of one attacker entity, that combines their views on the overlay network into one global view. For evaluating the location spoofing detection, we additionally assume these nodes to be lying about their location, i.e., never being physically present at the locations they claim to be. This models an attacker without the resources to use actual vehicles for gathering surveillance data.

In [3], the main evaluation metric for evaluating the attacker success in this scenario is the percentage of nodes known to the attacker, referred to as the *surveillance coverage*. Here, we additionally introduce the *distance disparity* metric, which describes the distance between the node position known to the attacker pos_{att} and the actual position of the node pos_{real} at any given time. Given the distance (in km) between two geographic locations P and Q as *d(P,Q)*, the distance disparity $disp$ for node X can be calculated as:

$$disp(x) = d(pos_{\text{att}}(x), pos_{\text{real}}(x)).$$

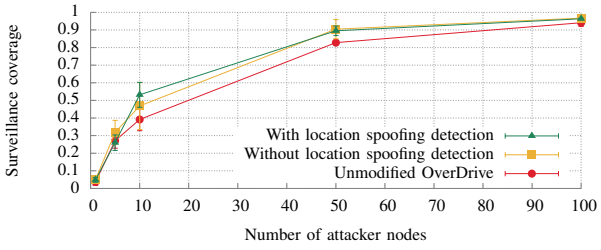


Fig. 5. Surveillance coverage in relation to the number of attacker nodes.

2) Results: Fig. 5 shows the measured average surveillance coverage in scenarios using the unmodified OverDrive from [6], OverDrive with enabled obfuscation and OverDrive with both obfuscation and location spoofing detection. Since the surveillance coverage shows only the percentage of nodes for which the attacker has location data but gives no information about that location data's precision, the impact of applying obfuscation and location spoofing detection is negligible. The

use of obfuscation even leads to an increase in surveillance coverage, as with the parametrization used for the obfuscation-enabled OverDrive more nodes are accepted as neighbors.

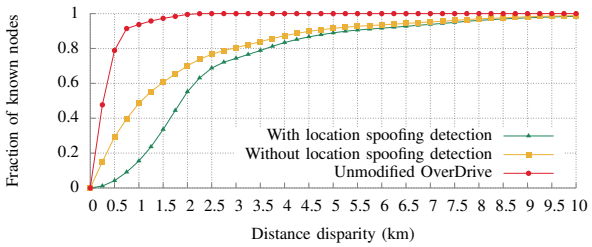


Fig. 6. Cumulative histogram of the distance disparity of all nodes known to an attacker with 100 attacker nodes.

More importantly, a significant improvement can be noted concerning the precision of the locations known by the attacker. Fig. 6 shows a cumulative histogram of measured distance disparity values, in a scenario with 100 attacker nodes and averaged between simulation runs with identical parameters. The plot shows the distance disparity plotted against the sum of all known nodes with a smaller or equal distance disparity. We can see that with the unmodified OverDrive system, the attacker knows the positions of 80% of the nodes known to him with a precision of less than 500 m. When using the obfuscation-based privacy enhancement, the attacker reaches this accuracy with only about 30% of the nodes known to him. With location spoofing detection, more than 54% of the node positions known by the attacker are wrong by more than 1.5 km. Here, location spoofing detection prevents nodes from sharing accurate location information with attacker nodes, as the latter always fake their location. We argue that, especially in populated areas, the measured levels of uncertainty make the collected location data unusable for breaking pseudonyms or determining the destinations of pseudonymized nodes. Thus, establishing a global view becomes completely unprofitable.

D. Identification of an Individual Target

Here, we present our evaluation of the difficulty for an attacker to identify the pseudonymized node belonging to a specific victim. The attacker is assumed to have context knowledge about his victim in the form of the location at which it will start its trip.

1) Evaluation Scenario and Metrics: We constructed an evaluation scenario based on the attack described in Sec. IV-C. We simulated a network with 10000 regular OverDrive nodes and an additional population of 100 *victim nodes*. Victim nodes behave like regular OverDrive nodes, but enter the network at wider intervals and at a common fixed *victim start point*. The start point is chosen randomly at the beginning of each simulation and is known by the attacker. Thus, the attacker introduces a set of up to 10 stationary attacker nodes to the network, that fake their location to random positions within a radius of 1 km around the victim start point. The attacker nodes continuously report new nodes they discover via overlay maintenance traffic like neighbor discovery messages. Based on the reasoning in Sec. IV-C, if the attacker learns about a node for the first time while that node is within 1 km of the victim start point, that node is marked as a potential victim. Based on this, we can measure the *victim recognition rate* of the attacker - the ratio of victim nodes that were correctly identified as such. As the attacker success greatly depends on the choice of a victim start point, we performed four times as many simulation runs for this experiment, i.e., a total of 16 per configuration.

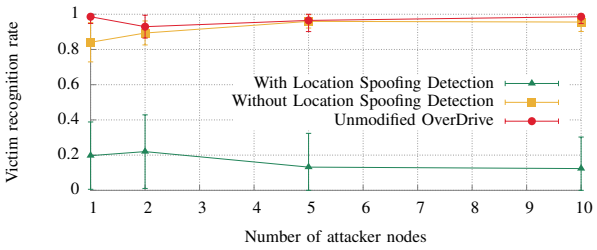


Fig. 7. Attacker success for identifying individual target.

2) Results: Fig. 7 shows the results for this experiment. In line with the results from [3], the attacker achieves a victim recognition rate of above 90% for the unmodified version of OverDrive, due to the unrestricted sharing of accurate location information by victim nodes. The recognition rate is not 100% because victim nodes move away from their start position and are not always immediately discovered by attacker nodes. When using the privacy aware OverDrive system without location spoofing detection, the recognition rate remains similarly high. This is due to the fact that the attacker nodes pretend to be very close to the victim, which causes the victim node to share more accurate location information. When using the location spoofing detection mechanism, the attacker scores a much lower recognition rate of only about 20%. Here, nodes will not share accurate location information with neighbors with whom the physical proximity has not been verified. Since attacker nodes are not physically in the area of the victim start point, the attacker will receive location information with an average error of around 1 km (corresponding to an obfuscation area with a edge length of 4 km), which significantly hinders a successful identification.

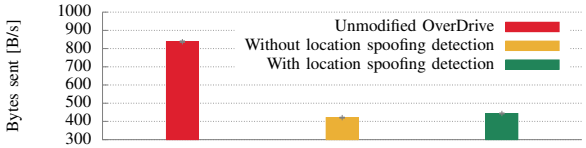
Location spoofing detection also hinders a subsequent tracking of victims. Malicious neighbors in the innermost rings are blacklisted after multiple verification attempts have failed. Nodes in the outer rings, on the other hand, have a lower chance of remaining in the neighborhood due to overlay maintenance logic. Thus, even if an attacker achieves a higher recognition rate by using additional side channels, the subsequent tracking via a follower attack is no longer practical.

E. Impact on Performance

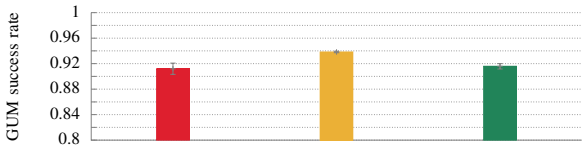
Here, we present results concerning the impact of the proposed extensions on system performance.

1) Evaluation Scenario and Metrics: In line with [6], we mainly focus on two metrics here: the consumed bandwidth of

the system measured in sent bytes per node, and the success rate for *geographic unicast messages* (GUMs). For measuring both in a realistic environment, we use a test application running on each node, that sends GUMs to randomly placed circular areas and tracks successfully delivered messages. In line with [6], we define the GUM success rate as the ratio between the number of messages that were successfully delivered and the number of messages that could have been successfully delivered. Messages sent to areas without any nodes (unavoidable errors) are not counted towards the GUM success rate. The presented results were gathered using simulations with 10000 honest mobile nodes.



(a) bandwidth consumption



(b) GUM success rate

Fig. 8. Performance impact

2) Results: Despite the fact that the modifications presented here are aimed at improving the privacy characteristics of OverDrive, they also have a positive impact on performance. Fig. 8 depicts results measured for the OverDrive system presented in [6] in comparison with values measured for the enhanced versions of the system that were presented here. While the average success rate increases slightly, the bandwidth consumption of the system drops significantly when obfuscation is enabled. With obfuscation, location updates to neighbors need to be sent significantly less often. Thus, nodes are able to maintain more neighbors without a significant increase in bandwidth consumption, while increasing success rates due to a higher interconnection in the overlay. The location spoofing detection mechanism, on the other hand, has only a minor effect on the performance of the geocast service provided by OverDrive. The GUM success rate drops slightly, as proximate nodes start sharing precise location information only after a successful location verification. The bandwidth consumption remains on a low level, as the private proximity test protocol needs to be performed only rarely in comparison to the sending of location updates.

VII. Conclusion

We propose key mechanisms for enabling privacy-preserving long-distance geocast services that do not rely on centralized service providers or dedicated infrastructure support. Through our location obfuscation concept, the precision of location information shared with entities in a decentralized system can be decreased with increasing distances to those entities, thus enforcing *data locality*. Through our location spoofing detection approach using GSM broadcast traffic, the information gain for an attacker from faking his position is reduced significantly. We designed our proposals as extensions to the overlay-based geocast service OverDrive. Through extensive simulation studies, we evaluated their effect on location privacy as well as their impact on performance. The results demonstrate that even strong adversaries controlling hundreds of nodes cannot break pseudonyms or track nodes with an acceptable level of certainty. Directions for future works include evaluating the impact of location spoofing detection on geographic routing correctness, e.g., for realizing a reliable long-distance location verification service.

References

[1] I. Baumgart, B. Heep, and S. Krause, "OverSim: A scalable and flexible overlay framework for simulation and real network applications," in *Proceedings of the 9th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P '09)*, Seattle, WA, USA, Sep. 9–11, 2009, pp. 87–88.

[2] D. Eckhoff and C. Sommer, "Driving for big data? privacy concerns in vehicular networking," *Security & Privacy, IEEE*, vol. 12, no. 1, pp. 77–79, 2014.

[3] M. Florian and I. Baumgart, "Privacy in overlay-based smart traffic systems," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*. IEEE, 2013, pp. 912–917.

[4] M. Gerla, J.-T. Weng, and G. Pau, "Pics-on-wheels: Photo surveillance in the vehicular cloud," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 1123–1127.

[5] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Pervasive Computing*. Springer, 2009, pp. 390–397.

[6] B. Heep, M. Florian, J. Volz, and I. Baumgart, "OverDrive: An Overlay-based Geocast Service for Smart Traffic Applications," in *Proceedings of the 10th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*. IEEE, Mar. 2013, pp. 1–8.

[7] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 2008, pp. 15–28.

[8] Z. Lin, D. F. Kune, and N. Hopper, "Efficient private proximity testing with gsm location sketches," in *Financial Cryptography and Data Security*. Springer, 2012, pp. 73–88.

[9] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing." in *NDSS*, 2011.

[10] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *Telecommunications, 2007. ITST'07. 7th International Conference on ITS*. IEEE, 2007, pp. 1–6.

[11] M. Picone, M. Amoretti, and F. Zanichelli, "GeoKad: A P2P Distributed Localization Protocol," in *Proceedings of the 8^{th} IEEE International Pervasive Computing and Communications Conference (PERCOM 2010) Workshops*, Mannheim, Germany, Mar. 2010, pp. 800–803.

[12] R. A. Popa, H. Balakrishnan, and A. J. Blumberg, "Vpriv: Protecting privacy in location-based vehicular services." in *USENIX security symposium*, 2009, pp. 335–350.

[13] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, and M. Mauve, "Challenge: peers on wheels-a road to new traffic information systems," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 215–221.

[14] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, pp. 1–13, Nov. 2012.