Simulation-based analysis of threats to location privacy in fog computing

Theresa Wettig and Zoltán Ádám Mann
University of Duisburg-Essen, Essen, Germany

***Abstract*—In fog computing, end devices can benefit from low-latency access to computing capacity provided by nearby fog nodes. However, an adversary controlling some fog nodes may infer information about the location of end devices that engage with these fog nodes. The goal of this paper is to analyze the severity of this threat to location privacy. We analyze how precise the information leaked by fog nodes about the location of end devices may be, and how this depends on the ratio of compromised fog nodes and on the adversary's background knowledge. We present the simulator LocPrivFogSim, which extends the existing fog computing simulator MobFogSim with the constructs necessary to model location privacy threats. The findings from preliminary simulations of various attack scenarios show that an attacker controlling even a modest ratio of the fog nodes may be able to infer precise information about the location and trajectory of end devices.**

***Index Terms*—fog computing, edge computing, location privacy**

I. Introduction

In the Internet of Things (IoT), a rapidly growing number of devices produce data that needs to be processed. Fog computing uses geographically distributed devices called fog nodes, which offer cloud-like services, so that end devices can benefit from the processing capacity of a nearby fog node with low latency [1], [2], [3].

Fog computing is associated with security and privacy challenges, from which location privacy is one of the most critical [4]. We consider a user's personal mobile device, e.g., smartphone or smartwatch. A key feature of fog computing is that the mobile device can always connect to a nearby fog node. Thus, the provider of the fog node learns that the end device is near to the fog node. The provider gains information about the user's location, violating location privacy. Beside the provider, also an adversary that successfully compromises the fog node can gain access to user location information [5].

This is known as a potential problem in fog computing, but it is not clear how much an adversary can actually infer about the location or the trajectory of a user based on the information leaked by fog nodes. Also, it is not clear how this depends on parameters like the ratio of compromised fog nodes and the adversary's background knowledge. Designers, operators, and users of fog systems currently have no tool to quantitatively assess the threats to location privacy. However, it would be very important to assess these threats, so that informed decisions can be made to handle the risk.

In this paper, we propose a tool for assessing threats to location privacy in fog computing and we analyze how these threats depend on key parameters. We use a simulation-based approach because simulation gives us the possibility to quickly assess the implications of different parameter settings [6]. We make the following two key contributions:

- We present the simulator LocPrivFogSim, which extends the existing fog simulator MobFogSim [7] with the constructs necessary to model location privacy threats.
- Using LocPrivFogSim, we evaluate the impact of different attack scenarios on location privacy.

Our preliminary findings show that an adversary controlling even a modest ratio of the fog nodes may be able to infer precise information about the location and trajectory of end devices, especially if the adversary knows the location of all fog nodes.

II. Problem description and assumptions

We consider a fog system in a designated region (e.g., a city). We assume that all fog nodes are active and connected to the network at all times, and that the regions covered by the fog nodes altogether cover the entire region. Mobile devices are assumed to always connect to the closest fog node.

An adversary compromised some of the fog nodes. If a mobile device connects to one of these fog nodes, the adversary can determine the approximate location of the mobile device. In addition, the adversary can link the device to a person [8]. From multiple observations, the adversary can also try to reconstruct the path of the person [9], [10]. We investigate what the adversary can learn when a mobile device traverses a path through the fog system.

As shown in Table I, we consider four scenarios with different levels of background knowledge. On the one hand, we consider what the adversary knows about the location of the fog nodes: either the adversary only knows the location of the compromised fog nodes, or he knows the location of all fog nodes. On the other hand, we consider what can be assumed about the state of the mobile device: either the mobile device is on and remains connected to the fog computing system at all times, or it is allowed to be turned off / disconnect.

This work was partially supported by the European Union's Horizon 2020 research and innovation program under grant 871525 (FogProtect).

This paper was published in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 736-741, 2021.

TABLE I: Overview of the considered scenarios

	Adversary knows location of compromised fog nodes only	Adversary knows location of all fog nodes
Mobile device is always connected to the fog system	Scenario 1.0	Scenario 2.0
Mobile device is allowed to disconnect	Scenario 1.1	Scenario 2.1

III. Simulation Environment

This section presents *LocPrivFogSim*[1], which we developed for simulating attacks on location privacy in fog computing. LocPrivFogSim is an extension of the existing fog simulator MobFogSim [11], [7]. MobFogSim itself is based on iFogSim [12], extending it with support for location and mobility.

To simulate threats to location privacy, MobFogSim had to be extended in several respects, as described below.

Roles. To simulate attacks on location privacy, various roles are needed. Thus we introduced the new classes `Owner`, `User`, and `Attacker`. Each `FogDevice` has an `Owner`. For a `MobileDevice`, the `User` needs to be distinguished from the `Owner` (e.g., the owner of a business smartphone is a company but the user is an employee). The `User` is assumed to carry the `MobileDevice` with them. The adversary is represented by the `Attacker` class.

Locations. MobFogSim stores locations of devices as Cartesian coordinates. This is not appropriate for storing the location information that the adversary gains about a mobile device, which is not an exact location, but rather a region of possible locations. Therefore, a new class `Position` was created to store this information. The `Position` contains the fog node that provided the information, a flag whether the mobile device entered or left the range of the fog node, and the timestamp.

Obtaining knowledge. The process how an adversary obtains knowledge was implemented using the Observer design pattern. The `FogDevice` manages registered observers. The adversary is registered as an observer for all compromised fog nodes. When a mobile device connects to or disconnects from a compromised fog node, the `Attacker`'s `update` method is triggered, giving the adversary new information about the mobile device. The class `MobileDeviceInformation` represents the information that the adversary obtains about a connected mobile device by listening on a fog node. The adversary stores this information in his knowledge base.

Regions of the fog nodes. The coordinates of each fog node are stored in the class `DeviceMap`, which provides a method to determine the closest fog node to a mobile device, based on Euclidean distance. Based on which the closest fog node is, the map can be divided into regions, as shown in Fig. 1. Each region contains exactly one fog node. For points within the region, the fog node in the region is the closest one. More formally, we are given a set of fog nodes $\{F_1, \ldots, F_m\}$ in a subset of the plane S. The region of fog node F_i is defined as $R(F_i) = \{s \in S \mid d(s, F_i) \leq d(s, F_j) \ \forall j \in \{1, \ldots, m\}\}$. If the mobile device is in $R(F_i)$, it connects to F_i.

[1]The source code of LocPrivFogSim is publicly available from https://git.uni-due.de/snthwett/locprivfogsim

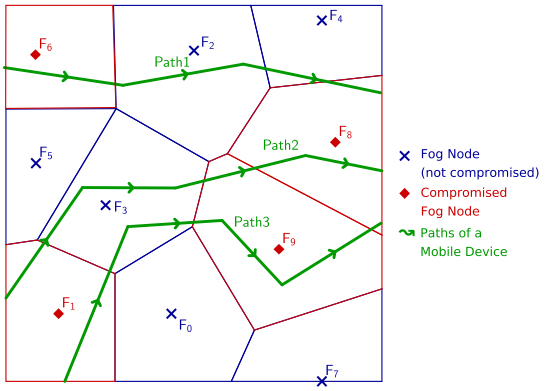


Fig. 1: Paths of a mobile device through a fog system

Computing areas. For quantifying the location information that the adversary can infer, the area of different regions of the plane, bordered by a combination of straight and circular lines, needs to be computed. We approximate the area of an arbitrary region by defining a dense lattice of equidistant points and counting the number of lattice points in the given region.

Trace comparison. For quantifying the leaked location information, we define two metrics [13]. The first is the *trace comparison value*.

In trajectory attacks, the adversary aims at determining the path of the mobile device [9]. The adversary's success can be characterized by the level of his uncertainty about the path of the mobile device. To capture uncertainty of the adversary, the size of the "anonymity set" is a useful metric [14]. Here, the "anonymity set" is the set of paths that are consistent with the adversary's observations. The real path of the mobile device is among these paths, but the adversary cannot know which one. The larger this set, the larger the uncertainty of the adversary.

The set of all possible paths is given as $\{p_1, \ldots, p_n\}$. For example, this can be the set of all paths that have been recorded in the past in a given city. The *trace* of a path p, denoted as $tr(p)$, is the sequence of indexes of fog nodes to which a mobile device traversing p connects.

The *observed trace* of a path p, denoted as $otr(p)$, is what the adversary can observe from the trace. If the mobile device is switched on and connected to the fog system at all times (scenarios 1.0 and 2.0 in Table I), the observed trace consists of the compromised fog nodes in the trace. Periods in which the mobile device is connected to a non-compromised fog node are marked with the symbol "$*$" in the observed trace. That is, $otr(p)$ is obtained from $tr(p)$ by replacing contiguous subsequences of non-compromised fog nodes with "$*$".

Fig. 1 shows an example. Path 1 passes the regions of fog nodes F_6, F_2, F_4, F_8; thus, $tr(p_1) = [6, 2, 4, 8]$. From these fog nodes, F_6 and F_8 are compromised, leading to $otr(p_1) = [6, *, 8]$. Path 2 passes the regions of the fog nodes F_1, F_3,

F_9, F_8; thus, $tr(p_2) = [1, 3, 9, 8]$ and $otr(p_2) = [1, *, 9, 8]$.

If the mobile device may temporarily switch off or disconnect from the fog system (scenarios 1.1 and 2.1), also periods in which the mobile device is not connected to the fog system are marked by a "$*$" in the observed trace. We use the same symbol because the adversary cannot distinguish whether the mobile device is connected to a non-compromised fog node or not connected to any fog node.

We assume the adversary knows the observed trace $otr(p_i)$ of each possible path p_i. E.g., the adversary could record this information by walking p_i with his own mobile device.

For a mobile device following a path p, the adversary records its observed trace $otr(p)$. The adversary checks which of the $otr(p_i)$ the observed $otr(p)$ is *consistent* with, i.e., which of the p_i could lead to the same observed trace. Determining if $otr(p)$ and $otr(p_i)$ are consistent, denoted as $otr(p) \sim otr(p_i)$, is different depending on the scenario. If the mobile device is always connected to the fog system, two observed traces are consistent if and only if they are the same. Otherwise, a "$*$" in the observed trace might mean that the mobile device was connected to a non-compromised fog node or that it was not connected to any fog node. Therefore, any occurrence of a "$*$" is removed from the observed trace. If this leads to multiple consecutive occurrences of the same index in the observed trace, those occurrences are replaced by a single occurrence of that index. In scenarios 1.1 and 2.1, two observed traces are considered consistent if and only if, after these changes, they are equal.

For example, path 2 in Fig. 1 has $otr(p_2) = [1, *, 9, 8]$. If a mobile device traverses this path but briefly disconnects in the region of fog node F_1, this leads to $otr(p) = [1, *, 1, *, 9, 8]$. Performing the above changes transforms both sequences to $[1, 9, 8]$, showing that the two observed traces are consistent.

Let $K = \{p_i \mid 1 \leq i \leq n,\ otr(p) \sim otr(p_i)\}$ be the set of paths consistent with the observations. The adversary knows that the mobile device followed one of the paths in K. The adversary's success can be measured by the *trace comparison value*, defined as $TrC = |K|$. The lower the value of TrC, the more successful the adversary is in learning the path of the mobile device. The best case for the adversary is $TrC = 1$, when the exact path of the mobile device was determined. The worst case for the adversary is $TrC = n$, when no information could be inferred about the path of the mobile device.

Accuracy value. Another way of measuring the adversary's knowledge of the mobile device's location is by the area to which the adversary can narrow down the position of the mobile device. This is also referred to as the size of the *uncertainty region* [14] or the cloaking granularity [9].

As the mobile device moves, the size of the uncertainty region varies. For an overall view of the adversary's knowledge of the device's location, we average the size of the uncertainty region over time. For this purpose, time is divided into intervals of size Δt, leading to the points in time $t_1, \ldots, t_r$ where $t_j = t_{j-1} + \Delta t$. For a point in time t_j, Q_{t_j} denotes the area to which the adversary can narrow down the location of the mobile device. The total time is $T = r \cdot \Delta t$, and the total considered area is V. The *accuracy value* is defined as $Acc = (\sum_{j=1}^{r} Q_{t_j} \cdot \Delta t)/(T \cdot V)$.

The value of Acc is between 0 and 1. The adversary's knowledge increases with decreasing value of Acc.

The area Q_{t_j} to which the adversary can narrow down the mobile device's location depends on the adversary's background knowledge, as shown in Table I. Q_{t_j} also depends on whether the mobile device is connected to a compromised or a non-compromised fog node at time t_j. These two situations, together with the four scenarios, lead to eight cases. However, some of these cases result in the same area Q_{t_j}, leading to four possible outcomes regarding Q_{t_j}. These are summarized in Fig. 2 and explained below.

Fig. 2a: the adversary only knows the position of compromised fog nodes (scenarios 1.0 and 1.1), and the mobile device is connected to a compromised fog node F. The adversary knows that the mobile device is in the region of F (here: the circle around F_8). The adversary also knows that the mobile device is nearer to F than to other compromised fog nodes (here: nearer to F_8 than to F_9). Thus, some segments can be cut off from the circle. Although the adversary also knows that the mobile device is nearer to F than to any non-compromised fog node, this information does not help him, since he does not know the location of non-compromised fog nodes.

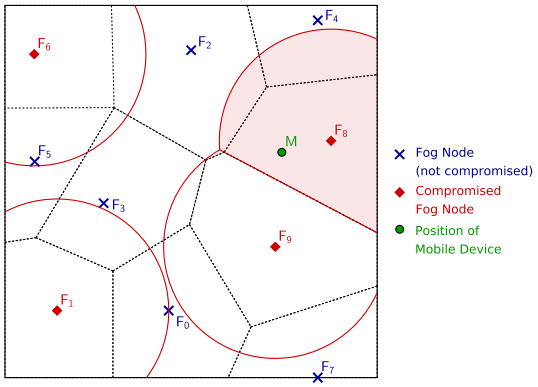
Fig. 2b: the adversary either only knows the position of compromised fog nodes (scenarios 1.0 and 1.1) or knows the position of all fog nodes but the mobile device is not guaranteed to be always on (scenario 2.1), and the mobile device is not connected to a compromised fog node. In this case, the adversary cannot narrow down the region in which the mobile device may be. In scenarios 1.0 and 1.1, the adversary knows that the mobile device is nearer to one of the non-compromised fog nodes than to any other fog node, but he does not know the position of non-compromised fog nodes, so this does not help him. In scenario 2.1, the adversary cannot infer anything about the location of the mobile device, since the mobile device may be offline and could be anywhere.

Fig. 2c: the adversary knows the position of all fog nodes (scenarios 2.0 and 2.1), and the mobile device is connected to a compromised fog node F. This is the best case for the adversary. Knowing the position of all fog nodes, he can build a Voronoi diagram of the regions of each fog node, and he knows that the mobile device is in the region of F. This region is a subset of the region identified in the case of Fig. 2a.

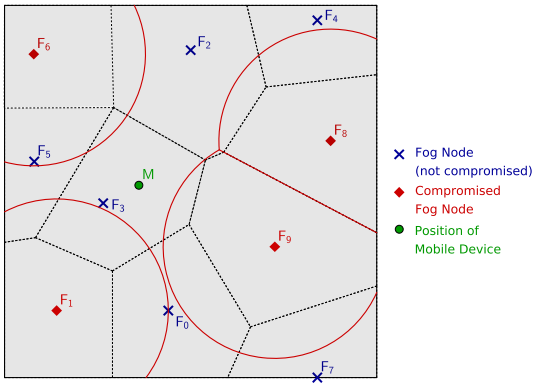
Fig. 2d: the adversary knows the position of all fog nodes, the mobile device is always on (scenario 2.0), and the mobile device is not connected to a compromised fog node. The adversary infers that the mobile device is connected to a non-compromised fog node. Knowing the position of all fog nodes, he narrows down the possible location of the mobile device to the union of the regions of non-compromised fog nodes.

IV. Simulation experiments

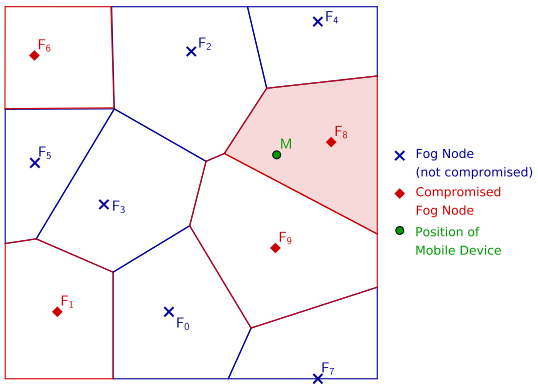
We created a test system in LocPrivFogSim, based on a rectangular field of size $50 * 50$ (e.g., representing a city). This field contains 20 fog nodes, which are the same for each



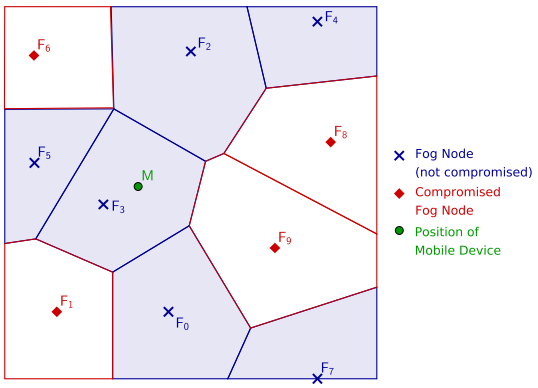
(a) Scenarios 1.0 and 1.1; mobile device is connected to a compromised fog node. The adversary can narrow down the device's location to the coloured circle section.



(b) Scenarios 1.0, 1.1 and 2.1; mobile device is not connected to a compromised fog node. The adversary is not able to narrow down the mobile device's location.



(c) Scenarios 2.0 and 2.1; mobile device is connected to a compromised fog node. The adversary can narrow down the device's location to the region of this fog node (the coloured polygon).



(d) Scenario 2.0; mobile device is not connected to a compromised fog node. The adversary can narrow down the device's location to the region of the non-compromised fog nodes (coloured region).

Fig. 2: The adversary's knowledge about a mobile device's position

test run. Each fog node covers a circle with radius 15. An adversary controls a given ratio of the fog nodes. For every test run, the compromised fog nodes are determined randomly.

50 possible paths are defined. Each path consists of up to 50 steps through the test system. One of these paths is randomly selected for every test run as the path of the mobile device.

To examine the effect of the adversary's background knowledge, we perform simulations according to each of the four scenarios of Table I. In addition, we vary the number of compromised fog nodes from 1 to 20, so that the ratio of compromised fog nodes varies from 5% to 100%. To cope with random variations, the simulation is repeated 100 times for each parameter configuration.

Results – trace comparison value. For the trace comparison, it is irrelevant whether the adversary knows the location of all fog nodes. However, whether the mobile device is assumed to be always connected to the fog system influences the precision of the observed trace and thus also the trace comparison value. Fig. 3 shows that the number of paths possible for the observed trace decreases rapidly as the number of compromised fog nodes increases. If the mobile device is allowed to disconnect, 5% compromised fog nodes hardly allow the adversary to limit the possible paths. With 15% compromised fog nodes, usually only a few of the paths are

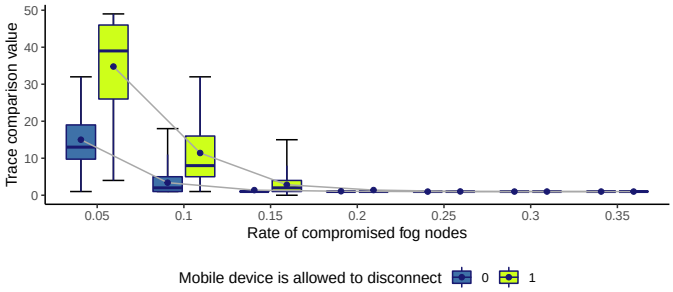


Fig. 3: Results in terms of the trace comparison value

still possible, and 20% compromised fog nodes allow almost always an exact statement about the path of the mobile device.

If the mobile device is always on, 5% compromised fog nodes allow the adversary to limit the device's path to in most cases less than 40% of the possible paths. With 10% compromised fog nodes, only a few of the possible paths are consistent with the recorded trace, and with 15% compromised fog nodes, the adversary can assign the observed trace of the mobile device to one of the known paths. Thus it can be seen that if the mobile device is allowed to disconnect, about 5% more compromised fog nodes are needed to get the same results as if the mobile device is always on.

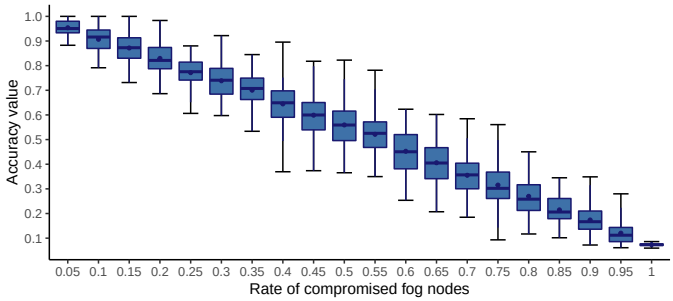


Fig. 4: Accuracy values in scenario 1.0

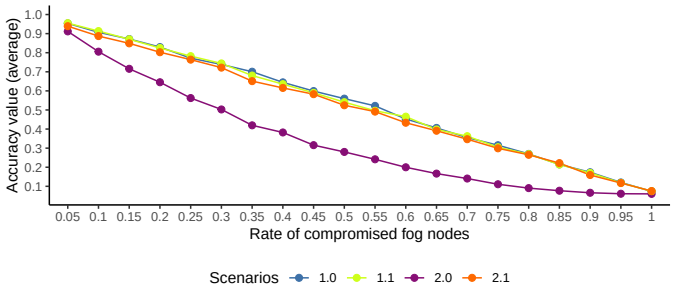


Fig. 5: Comparison of the accuracy value between scenarios

Results – accuracy value. Fig. 4 shows the results in terms of the accuracy value for scenario 1.0. The results for the other scenarios are similar and omitted to save space.

With increasing ratio of compromised fog nodes, the accuracy value decreases steadily, i.e., the accuracy with which the adversary can determine the mobile device's location increases constantly. With 5% compromised fog nodes, hardly any information about the location of the mobile device can be obtained. But when all nodes are compromised, the accuracy value reaches a minimum of about $\frac{1}{20} = 0.05$.

To better visualize the impact of the different scenarios, Fig. 5 shows the average accuracy value of each scenario in a single plot. We can see that scenarios 1.0, 1.1 and 2.1 lead to a similar, roughly linear dependence of the accuracy value on the ratio of compromised fog nodes. From Fig. 2a and 2b, it is clear that the accuracy value is actually the same for scenarios 1.0 and 1.1. From Fig. 2b, it is also clear that, when the mobile device is not connected to a compromised fog node, also scenario 2.1 leads to the same area as scenarios 1.0 and 1.1. Comparing Fig. 2a and 2c, it can be seen that, when the mobile device is connected to a compromised fog node, scenario 2.1 leads to a slightly smaller area than scenarios 1.0 and 1.1. The difference becomes insignificant when there are many compromised fog nodes because of the overlaps between the ranges of the compromised fog nodes. This explains why in Fig. 5 the average accuracy value of scenario 2.1 is slightly lower than that of scenarios 1.0 and 1.1, as long as the ratio of compromised fog nodes is not too high.

It is clear from Fig. 5 that scenario 2.0 leads to the best accuracy. Fig. 2b and 2d show that the adversary can more accurately locate the mobile device in scenario 2.0 than in the other scenarios, if the mobile device is not connected to a compromised fog node. With many compromised fog nodes, this happens infrequently, hence the improvement in the accuracy value is not so significant. With few compromised fog nodes, the difference in the area between Fig. 2b and 2d is not so high, hence again the improvement in accuracy is not so significant. This is why the difference between scenario 2.0 and the other scenarios in Fig. 5 is biggest when the ratio of compromised fog nodes is neither too low nor too high.

V. Related work

Location privacy was investigated already before the advent of fog computing, e.g., in wireless networks, mobile computing, and online services [15]. Two different settings can be differentiated: the adversary may be located on the network infrastructure layer or on the application layer.

Adversary on the network infrastructure layer. When an end device connects to a network, the network operator learns something about the device's location [16]. In mobile telecommunication networks, the network operator knows in which cell the mobile phones are located. In wireless local area networks (WLANs), the operator of a WLAN access point knows that connected devices are in the vicinity of the access point. An adversary that can eavesdrop on the messages between end devices and base stations or access points may also be able to infer similar information [17].

The protection of location privacy in mobile and wireless networks received significant research attention. E.g., [18] proposed frequently changing the network identifier of mobile devices to protect privacy. To hinder linkability between a mobile device's sessions with different base stations, [19] proposed using identifier changes and silent periods. [20] addressed location privacy in WLANs and proposed frequent pseudonym changes, silent periods, and a reduction of the devices' transmission range. [16] used blind signatures to create authorized anonymous IDs for mobile devices. [21] devised different policies for changing the identifiers of devices or swapping identifiers between devices.

While these technical measures make it more difficult for an adversary to infer location information about users, they do not offer sufficient protection, and need to be combined with other measures like legislative provisions [15].

Adversary on the application layer. Location-based application services are popular and often very useful. However, they require access to the user's location, thus threatening location privacy [9], [10]. Several approaches were proposed to achieve location privacy while using location-based services. Typical approaches use some obfuscation technique to hide the user's real location or real identity from the provider of the location-based service. The obfuscation logic is typically running in a trusted environment. [22] proposed a location protection broker running on a trusted server that guarantees location k-anonymity by means of message perturbation. [23] also used a broker running on a trusted server to obfuscate location information, also taking into account the movement of devices and the possible continual observation of their location information by adversaries. [24] proposed a set of elementary

location obfuscation operators that can be combined to several different types of location obfuscation operators. [25] proposed the use of a trusted location anonymizer component, which blurs exact locations to cloaked spatial areas. [26] proposed to use historical location information of other end devices to conceal the real location of an end device.

Another possibility is to generate, in addition to real location information, fake or dummy location information, to confuse potential adversaries. For example, [27] proposed an approach in which dummy trajectories are generated to hide users' real trajectories. Similarly, [28] devised an algorithm for creating dummy trajectories based on real ones to ensure k-anonymity.

Location privacy in fog computing. Existing surveys on security and privacy in fog computing feature location privacy as a key issue [4], [5]. Location privacy is particularly challenging in fog computing because an adversary controlling some fog nodes combines the characteristics of the two types of adversaries considered in previous research. An adversary that controls some fog nodes is similar to adversaries on the network layer considered previously, since providers of fog nodes also have to know about the end devices connecting to their fog nodes. Hence, adopting solutions for location privacy from location-based application services, like concealing the real location of end devices or generating dummy locations, is challenging in fog computing. On the other hand, an adversary controlling some fog nodes is also the provider of a location-based service, since fog computing is inherently location-based. Hence, it is not clear how solutions ensuring location privacy in the context of network operators, like changes of identifiers, could be applied in fog computing.

VI. Conclusions

We presented LocPrivFogSim, a simulator to assess threats to location privacy in fog computing. We introduced two metrics to measure the location information that an adversary can obtain. Our preliminary simulations showed quantitatively how the ratio of compromised fog nodes, the adversary's knowledge about the fog nodes' position, and whether the mobile device can be assumed to be always connected to the fog system, impact the leaked location information. The results show that a small number of compromised fog nodes suffice to leak precise information about the path taken by a mobile device.

As future work, further simulations with more realistic setups are planned. LocPrivFogSim could be extended with different techniques for preserving location privacy, so that their effect can be evaluated. Moreover, it could be investigated how location privacy can be taken into account in the deployment of fog applications [29].

References

[1] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[2] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[3] Z. Á. Mann, "Optimization problems in fog and edge computing," *Fog and Edge Computing: Principles and Paradigms*, pp. 103–121, 2019.

[4] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Wireless Algorithms, Systems, and Applications*. Springer International Publishing, 2015, pp. 685–695.

[5] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[6] C. Kunde and Z. Á. Mann, "Comparison of simulators for fog computing," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 1792–1795.

[7] C. Puliafito, D. M. Gonçalves, M. M. Lopes, L. L. Martins, E. Madeira, E. Mingozzi, O. Rana, and L. F. Bittencourt, "MobFogSim: Simulation of mobility and migration for fog computing," *Simulation Modelling Practice and Theory*, vol. 101, p. 102062, 2020.

[8] M. Gruteser and B. Hoh, "On the anonymity of periodic location samples," in *International Conference on Security in Pervasive Computing*. Springer, 2005, pp. 179–192.

[9] P. Asuquo, H. Cruickshank, J. Morley, C. P. A. Ogah, A. Lei, W. Hathal, S. Bao, and Z. Sun, "Security and privacy in location-based services for vehicular and mobile communications: An overview, challenges, and countermeasures," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4778–4802, Dec 2018.

[10] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, nov 2012.

[11] https://github.com/diogomg/MobFogSim.

[12] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[13] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 247–262.

[14] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–38, 2018.

[15] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[16] Q. He, D. Wu, and P. Khosla, "The quest for personal control over mobile location privacy," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 130–136, 2004.

[17] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 320–336, 2011.

[18] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis," *Mobile Networks and Applications*, vol. 10, no. 3, pp. 315–325, 2005.

[19] Y.-C. Hu and H. J. Wang, "A framework for location privacy in wireless networks," in *ACM SIGCOMM Asia Workshop*, 2005.

[20] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless LANs," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, 2007, pp. 246–257.

[21] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, "Swing & Swap: user-centric approaches towards maximizing location privacy," in *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, 2006, pp. 19–28.

[22] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2007.

[23] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2011.

[24] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2009.

[25] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new Casper: Query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, 2006, pp. 763–774.

[26] X. Guo, W. Wang, H. Huang, Q. Li, and R. Malekian, "Location privacy-preserving method based on historical proximity location," *Wireless Communications and Mobile Computing*, p. Art. 8892079, 2020.

[27] T. Wang, J. Zeng, M. Z. A. Bhuiyan, H. Tian, Y. Cai, Y. Chen, and B. Zhong, "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, 2017.
[28] G. Li, Y. Yin, J. Wu, S. Zhao, and D. Lin, "Trajectory privacy protection method based on location service in fog computing," *Procedia Computer Science*, vol. 147, pp. 463–467, 2019.
[29] Z. Á. Mann, "Secure software placement and configuration," *Future Generation Computer Systems*, vol. 110, pp. 243–253, 2020.