Location Privacy in Mobile Edge Clouds

Ting He*, Ertugrul N. Ciftcioglu†, Shiqiang Wang‡, and Kevin S. Chan†
*Pennsylvania State University, University Park, PA, USA. Email: tzh58@psu.edu
†Army Research Laboratory, Adelphi, MD, USA. Email: ertugrulnc@ieee.org, kevin.s.chan.civ@mail.mil
‡IBM T. J. Watson Research Center, Yorktown, NY, USA. Email: wangshiq@us.ibm.com

***Abstract*—In this paper, we consider user location privacy in mobile edge clouds (MECs). MECs are small clouds deployed at the network edge to offer cloud services close to mobile users, and many solutions have been proposed to maximize service locality by migrating services to follow their users. Co-location of a user and his service, however, implies that a cyber eavesdropper observing service migrations between MECs can localize the user up to one MEC coverage area, which can be fairly small (e.g., a femtocell). We consider using chaff services to defend against such an eavesdropper, with focus on strategies to control the chaffs. Assuming the eavesdropper performs maximum likelihood (ML) detection, we consider both heuristic strategies that mimic the user's mobility and optimized strategies designed to minimize the detection or tracking accuracy. We show that a single chaff controlled by the optimal strategy can drive the eavesdropper's tracking accuracy to zero when the user's mobility is sufficiently random. The efficacy of our solutions is verified through extensive simulations.**

***Index Terms*—Mobile edge cloud, location privacy, chaff service.**

I. Introduction

While improvement in the coverage of wireless communications brings tons of useful applications to the fingertips of mobile users, this trend also imposes a significant threat on user location privacy. *Location privacy* refers to safeguarding a mobile user's location from unintended use. While legitimate use of user location can enable various *location-based services (LBS)*, malicious use of this information can cause harmful consequences such as stalking, blackmailing, and fraud [1].

Existing efforts in protecting user location privacy mostly focus on protecting the information released through the *direct channel*, i.e., location information intentionally revealed by the user. Since the direct channel is controlled by the user, e.g., by configuring how to share his location with an LBS provider, the user can obfuscate his location in the spatial/temporal domain to make sure that his location cannot be distinguished from the locations of many other users [1], [2].

The more challenging problem, however, is how to prevent unintentional release of location information through *side channels*. For example, one side channel is the user's wireless transmission activity, which can be monitored by a wireless eavesdropper to track the user. There have been solutions proposed to protect this side channel, e.g., by introducing intermittent radio silence and reducing the transmission power [3], [4]. In this work, we investigate a new side channel not studied before, which arises in a novel application context of *mobile edge clouds (MECs)* [5].

MECs, as illustrated in Fig. 1, are small clouds that offer a limited set of cloud services from the edge of the mobile network (e.g., base stations). Since its introduction, MEC has attracted tremendous interest from both research communities and industry leaders as a promising approach to improve cloud performance for mobile users [6], [7]. From the perspective of location privacy, however, this technology opens a new side channel, referred to as the *cyber side channel*.

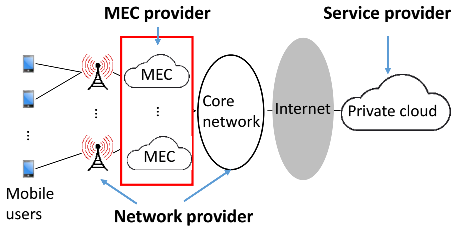


Fig. 1. Providing services to mobile users via MECs.

Specifically, to deliver the promised performance, MECs need to migrate services[1] (e.g., by migrating virtual machines (VMs) encapsulating the services [8]) to follow the mobile users [9], [10], [5], [11], especially for delay-sensitive services (e.g., augmented reality) [8]. Thus, a "cyber eavesdropper", who can observe service migrations among MECs, can track physical movements of the user. Such a cyber eavesdropper can be a hacker that has gained access to the MECs, or an untrusted MEC provider interested in tracking users of certain services. Cyber eavesdropping is a realistic concern in MECs because of the openness of the MEC ecosystem [6], which increases the risk of introducing both unsecured systems and untrusted providers. Note that as shown in Fig. 1, we distinguish between the network provider, the MEC provider, and the service provider, where the network provider and the service provider are trusted, but the MEC provider can be untrusted. Although the spatial resolution of cyber eavesdropping is limited to the coverage of one MEC (e.g., a femtocell), its harm can be severe, as it can be performed without any physical sensing devices, thus potentially at a much lower cost and a much larger scale.

While cyber eavesdropping and wireless eavesdropping are conceptually similar, the defense mechanisms can be quite different. Specifically, the existing defense mechanisms for wireless eavesdropping [3], [4] are *intrusive* in that they modify the user's transmissions. While it is possible to defend against cyber eavesdropping by not letting services follow their users, such a mechanism will cause significantly degraded Quality of Service (QoS) [8]. Instead, we consider a *non-intrusive* mechanism using *chaffs*. Chaffs are legitimate ser-

[1] Here "service" refers to an instance of a given type of service (e.g., a VM instance running the service), which is independently generated/migrated for each user as assumed in existing solutions [8], [9], [10], [5], [11].

vices launched by the user (or by the network provider on behalf of the user) together with the real service to confuse the eavesdropper about which service the user is actually using. For example, they can be implemented by sending fake service requests and handoff signals to user-specified MECs; see Section II-B for details.

To confuse the eavesdropper, the chaffs must be indistinguishable from the real service, e.g., by being independent instances of the same type of service. It is, however, insufficient to only make the chaffs indistinguishable in content. For example, a chaff that never migrates can be easily distinguished from a real service that migrates with the mobile user, and a chaff that randomly migrates among MECs can be easily distinguished from a real service that exhibits temporal correlation in its locations. For a chaff to effectively confuse the eavesdropper, its mobility pattern, i.e., where it is launched and whether/where it is migrated, has to resemble the mobility pattern of the real service. Meanwhile, a chaff that always follows the real service (which follows the user) offers no protection for the user's location privacy. Therefore, the challenge is in controlling the mobility of the chaffs to *maximally resemble the real service while minimally co-locating with the real service*. To address this challenge, we study the following closely related questions: (i) How will an eavesdropper track a user in the presence of chaffs? (ii) How should the user control the chaffs to defend against the eavesdropper?

A. Related Work

Most existing work on location privacy refers to protecting the *direct channel*, where the user intentionally releases his location to access LBS [1]. Most existing solutions, e.g., [2] and references therein, use spatial/temporal cloaking to satisfy a given anonymity requirement (e.g., k-anonymity). The basic idea is to let a trusted server "cloak" a user by replacing the exact user locations by bounding boxes containing sufficiently many other users. While such a strategy can protect the direct channel, it does not protect side channels such as the cyber side channel considered here.

Besides the direct channel, *side channels* can also release location information. An important side channel in wireless networks is the transmission activity, which can be monitored by a wireless eavesdropper to track the user. To defend against wireless eavesdropping, mechanisms are proposed to protect senders/receivers using anonymous routing protocols, frequently changing pseudonyms, silent periods, and reduced transmission power [3], [4]. The above mechanisms are *intrusive* in that they modify the user's behavior. In contrast, we study another side channel arising in MECs due to correlated user mobility and service mobility, and propose a *non-intrusive* defense mechanism using chaffs.

The idea of using chaffs to protect user security/privacy has been explored in other contexts, such as using chaff traffic to protect real traffic [12], using chaff data to protect real data [13], and using chaff applications to protect real applications [14]. However, we are the first to study the use of chaff services to protect user location privacy. Besides the novel application context, our problem also requires new methodology. Specifically, as a real service needs to migrate dynamically to follow a mobile user, its mobility pattern (in addition to its content) can be used to identify the service. To effectively protect the user, the chaff services have to resemble the real service in both content and mobility pattern.

Another line of related work is service migrations in MECs. Service migrations in MECs are primarily driven by the need to keep the service close to a mobile user as the user moves, while considering both migration cost and communication cost. Modeling the user's mobility as a *Markov chain (MC)*, several solutions based on *Markov Decision Processes (MDPs)* have been proposed to minimize the total cost under 1-D [15], [16] or 2-D mobility models [5], [10]. Here we consider the worst case (in terms of location privacy) that the real service *always* follows the user, and focus on protecting the user's location privacy using chaffs. We leave the study of privacy-aware service migration to future work.

B. Summary of Contributions

We consider the problem of protecting user location privacy in the context of MECs using chaffs. Our contributions are:

1) We model the eavesdropper as a *maximum likelihood (ML)* detector that aims at detecting the user's trajectory based on multiple observed trajectories.

2) We propose a suite of increasingly sophisticated chaff control strategies for the user: (i) an *impersonating (IM)* strategy that mimics the user's mobility, (ii) an ML strategy that maximizes the likelihood of the chaff's trajectory to mislead the detector, (iii) an *optimal (OPT)* strategy that minimizes the eavesdropper's tracking accuracy based on the user's trajectory. We show that all the strategies can be computed in polynomial time.

3) We analyze the performance of the proposed chaff control strategies. Our analysis shows that while the eavesdropper's tracking accuracy is always non-zero under the IM or ML strategy, it may decay to zero under the OPT strategy, where we characterize the condition and the decay rate.

4) We evaluate the proposed strategies via extensive simulations. Our evaluations show that beside the chaff control strategy, the user's mobility model also has a significant impact on the tracking accuracy. Nevertheless, our strategies, especially OPT, can significantly reduce the tracking accuracy even for users with highly skewed mobility.

The rest of the paper is organized as follows. Section II formulates the problem. Section III specifies the model for the eavesdropper. Section IV presents chaff control strategies for the user, whose effectiveness is analyzed in Section V. Section VI evaluates the performance through simulations. Then Section VII concludes the paper. *All the proofs are provided in our technical report*[2] *[17].*

II. Problem Formulation

A. Network Model

Given a network field deployed with multiple MECs, we quantize the space into *cells* such that each cell corresponds to the coverage area of one MEC. Let $\mathcal{L}$ denote the set of cells, which also specifies the set of possible user locations from the perspective of a cyber eavesdropper; let $L := |\mathcal{L}|$.

[2] The OPT strategy is referred to as the *optimal offline (OO) strategy* in [17].

Suppose that there is a user of interest running a delay-sensitive service (e.g., augmented reality or virtual desktop) that must be co-located with the user. We consider delay-sensitive service as it has been identified as one of the most promising applications in future wireless networks [18], while establishing the worst case for location privacy. We leave the study of more flexible services to future work. Note that although our analysis focuses on the single-user scenario, our solution can be independently applied to protect multiple users in a multi-user scenario, where our results provide performance lower bounds as other coexisting users (and their chaffs) offer additional protection.

B. Eavesdropper and Chaffs

We consider a cyber eavesdropper that observes the trajectories of services as they migrate among the MECs. Such an eavesdropper can be a hacker inside the MEC system, or an untrusted MEC provider that operates the MECs. Under the assumption of delay-sensitive services as in Section II-A, the eavesdropper can track the user by detecting the trajectory of his service.

To prevent detection, the user generates $N-1$ ($N>1$) additional trajectories using chaff services. Each chaff service is an independent instance of the same service that the user is accessing, thus indistinguishable from the real service in content. The chaff services will consume MEC resources, and the cost incurred by these services is the responsibility of the user. In this regard, the parameter N captures the user's budget for running chaff services. With assistance of the network provider, the user can make a chaff service follow an arbitrary trajectory by sending fake service requests and handoff signals to the corresponding MECs (which cause the chaff service to be instantiated or migrated). Since for a cyber eavesdropper, tracking a user is equivalent to tracking his service, we simply refer to the user's service as "the user" and the chaff services as "the chaffs".

C. Mobility Model

Assume that the user follows a discrete-time ergodic *Markovian chain (MC)* as in [15], [16], [5], with transition matrix $P=(P(x_t|x_{t-1}))_{x_t,x_{t-1}\in\mathcal{L}}$. Let $\pi:=(\pi(x))_{x\in\mathcal{L}}$ denote his steady-state distribution. Assume that $\pi(x)>0$ for all $x\in\mathcal{L}$. Mobility of the chaffs (i.e., migration of chaff services) is controlled by the user and will be studied later.

For each $u=1,\ldots,N$, let $x_{u,t}\in\mathcal{L}$ denote the location of the u-th service in time slot t, and $\mathbf{x}_u:=(x_{u,t})_{t=1}^T$ the trajectory over T slots. Here $u=1$ corresponds to the user, $u=2,\ldots,N$ correspond to the chaffs, and $T\geq 1$ represents the duration of the user's service.

D. Location Privacy in the Presence of Chaffs

Our goal is to understand the efficacy of protecting user location privacy using chaffs. We achieve this by studying two closely-related problems:

(i) From the eavesdropper's perspective: Given N trajectories generated by a user and $N-1$ chaffs, which trajectory belongs to the user?

(ii) From the user's perspective: Given $N-1$ chaffs, what trajectories should the chaffs follow to cause the worst performance for the eavesdropper?

We measure the eavesdropper's performance by his *tracking accuracy*, defined as the time-average probability of correctly tracking the user, i.e., if the eavesdropper believes that the u-th trajectory belongs to the user, then his tracking accuracy equals $\frac{1}{T}\sum_{t=1}^T \Pr\{x_{u,t}=x_{1,t}\}$. Note that this is different from the detection accuracy, as $u=1$ is sufficient but not necessary for $x_{u,t}=x_{1,t}$.

III. Eavesdropper's Strategy

Given multiple trajectories $\mathbf{x}_u:=(x_{u,t})_{t=1}^T$ ($u=1,\ldots,N$), the eavesdropper wants to determine which trajectory belongs to the user of interest. We assume that the eavesdropper knows the user's mobility model, i.e., the transition matrix P. For example, the eavesdropper can obtain this information by profiling how typical users move in the network field.

Intuitively, the eavesdropper should pick the trajectory that best matches the user's mobility model. Mathematically, this is the trajectory that has the *maximum likelihood (ML)* among all the trajectories. Under the assumption that all the trajectories have equal prior probability of belonging to the user, the ML trajectory has the maximum posterior probability of belonging to the user. Under the Makovian user mobility model in Section II-C, the ML detector is given by ($[N]:=\{1,\ldots,N\}$)

$$u^{\mathrm{ML}}=\arg\max_{u\in[N]} p(\mathbf{x}_u)=\arg\max_{u\in[N]} \pi(x_{u,1})\prod_{t=2}^{T}P(x_{u,t}|x_{u,t-1}). \quad (1)$$

The optimization in (1) can be easily solved in $O(NT)$ time.

IV. User's Strategy

The problem faced by the user is that given $N-1$ chaffs, how to control the mobility of the chaffs, i.e., how to generate the trajectories $\mathbf{x}_u$ ($u=2,\ldots,N$), to maximally confuse the eavesdropper. Depending on the precise definition of "confusion", we have the following chaff control strategies.

A. Impersonating Strategy

If the eavesdropper's strategy is unknown, a safe choice for the user is to make the chaffs appear similar to himself, a strategy referred to as the *impersonating (IM) strategy*. Under Markovian user mobility, this strategy makes each chaff follow a trajectory generated independently from the same transition matrix P as followed by the user, which naturally mimics the user's mobility. Under this strategy, all the N trajectories are statistically identical, and therefore any detector, including the ML detector (1), can only make a random guess.

B. Maximum Likelihood Strategy

1) The Strategy: If the user knows that the eavesdropper uses the ML detector (1), then he can design trajectories for the chaffs to intentionally mislead the detector. A chaff's trajectory can mislead the ML detector only if its likelihood (based on the user's mobility model) is no smaller than the likelihood of the user's trajectory. Since the detector is deterministic, it suffices to use a single chaff as at most one chaff (the one with the ML trajectory) will have effect even if multiple chaffs are used.

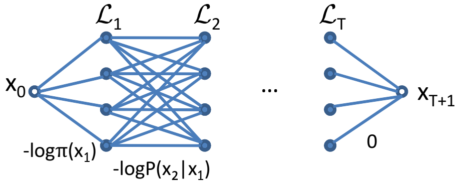


Fig. 2. Auxiliary graph for computing the ML trajectory.

This idea inspires a strategy referred to as the *maximum likelihood (ML) strategy*. Letting $\mathcal{L}^T$ denote all possible trajectories of length T, this strategy controls the chaff to follow a trajectory $\mathbf{x}_2$ that achieves the following optimization:

$$\mathbf{x}_2 = \arg\max_{\mathbf{x}\in\mathcal{L}^T} p(\mathbf{x}) = \arg\max_{\mathbf{x}\in\mathcal{L}^T} \pi(x_1)\prod_{t=2}^{T} P(x_t|x_{t-1}). \quad (2)$$

2) The Algorithm: While the space of all possible trajectories ($\mathcal{L}^T$) is too large to explore exhaustively, the optimization problem in (2) has a physical interpretation that allows a more efficient solution. We will show that problem (2) can be converted to a *shortest-path problem* as follows.

The key is to rewrite the optimization (2) as

$$\mathbf{x}_2 = \arg\min_{\mathbf{x}\in\mathcal{L}^T} -\log\pi(x_1) + \sum_{t=2}^{T}(-\log P(x_t|x_{t-1})). \quad (3)$$

Let $\mathcal{L}_t$ ($t = 1,\ldots,T$) be a set of vertices representing all possible chaff locations at time t ($|\mathcal{L}_t| = |\mathcal{L}|$). As illustrated in Fig. 2, we construct a graph $\mathcal{G} = (V, E)$, with vertices $V = \{x_0\}\cup\{x_{T+1}\}\cup\bigcup_{t=1}^{T}\mathcal{L}_t$ denoting possible chaff locations at different times (x_0 and x_{T+1} are virtual locations) and edges $E = (\{x_0\}\times\mathcal{L}_1)\cup(\mathcal{L}_T\times\{x_{T+1}\})\cup\bigcup_{t=2}^{T}(\mathcal{L}_{t-1}\times\mathcal{L}_t)$ denoting possible movements. We assign each edge a cost[3]:

1) edge (x_0, x) for each $x\in\mathcal{L}_1$ has cost $-\log\pi(x)$;
2) edge (x, x') for each $x\in\mathcal{L}_{t-1}$ and $x'\in\mathcal{L}_t$ ($t = 2,\ldots,T$) has cost $-\log P(x'|x)$;
3) edge (x, x_{T+1}) for each $x\in\mathcal{L}_T$ has zero cost.

Each possible trajectory $\mathbf{x} = (x_t)_{t=1}^{T}$ corresponds to a path $(x_0, x_1,\ldots,x_T, x_{T+1})$ from x_0 to x_{T+1} in $\mathcal{G}$, and the cost of this path, given by the sum of its edge costs, equals the value of the objective function (3) at $\mathbf{x}$. Thus the solution to (3) is essentially the path from x_0 to x_{T+1} that has the *minimum cost*, which can be computed by Dijkstra's algorithm[4] at complexity $O(TL^2)$. Note that this trajectory only depends on the user's mobility model and can thus be computed beforehand.

Remark: The ML strategy is clearly optimal against the ML detector (1) in minimizing the detection accuracy. This is, however, different from minimizing the tracking accuracy, as the chaff's trajectory may coincide with the user's trajectory at times, when the eavesdropper can track the user perfectly.

C. Optimal Strategy

1) The Strategy: The ultimate goal of the user is to prevent himself from being tracked by the eavesdropper. To this end, the chaff's trajectory not only needs to mislead the detector, but also needs to be as disjoint as possible from the user's trajectory. For the ML detector (1), the optimal strategy is to let the chaff follow a trajectory that is as disjoint as possible from the user's trajectory while having a higher likelihood, i.e., the solution $\mathbf{x}_2 := (x_{2,t})_{t=1}^{T}$ to the following optimization[5]

$$\min\sum_{t=1}^{T}\mathbb{1}_{\{x_{2,t}=x_{1,t}\}} \quad (4)$$

$$\text{s.t.}\,\pi(x_{2,1})\prod_{t=2}^{T}P(x_{2,t}|x_{2,t-1})>\pi(x_{1,1})\prod_{t=2}^{T}P(x_{1,t}|x_{1,t-1}), \quad (5)$$

where the constraint (5) guarantees that the ML detector will pick the chaff's trajectory, and the objective (4) minimizes the number of times that the chaff's trajectory coincides with the user's trajectory. Again, a single chaff suffices as the detector is deterministic. We refer to this strategy as the *optimal (OPT) strategy*, as it is optimal in minimizing the tracking accuracy of an eavesdropper using the ML detector (1).

Note that (5) will be infeasible if the user's trajectory has the maximum likelihood among all the trajectories. In this case, we change the "$>$" in (5) to "$=$" to force the ML detector to make a random guess, but the objective (4) remains valid as we want to minimize the number of times the eavesdropper tracks the user correctly when the detector guesses wrong.

2) The Algorithm: While a brute-force solution to (4) is infeasible due to the exponentially large solution space, we can solve it by dynamic programming over the weighted graph introduced in Fig. 2. Let $p_{\mathbf{x}_1}$ denote the path in this graph corresponding to the user's trajectory, and $K(p_{\mathbf{x}_1})$ the length (sum of edge costs) of this path. Then optimizing (4) subject to (5) is equivalent to finding a path from x_0 to x_{T+1} with a length less than $K(p_{\mathbf{x}_1})$ (or equal to $K(p_{\mathbf{x}_1})$ if $p_{\mathbf{x}_1}$ is a shortest path) that is as disjoint as possible from $p_{\mathbf{x}_1}$. To this end, we introduce $K_t(x, i)$ to denote the length of the shortest path from $x\in\mathcal{L}_t$ to x_{T+1} that intersects (i.e., sharing vertices) with $p_{\mathbf{x}_1}$ at most i times ($0\le i\le T-t+1$), and $n_t(x, i)$ to denote the next hop neighbor of x on this path.

Initially, $K_T(x, 1)\equiv 0$ for all $x\in\mathcal{L}_T$,

$$K_T(x, 0) = \begin{cases} 0 & \text{if } x\neq x_{1,T}, \\ \infty & \text{o.w.,}\end{cases} \quad (6)$$

and $n_T(x, i)\equiv x_{T+1}$ for all $x\in\mathcal{L}_T$ and $i\in\{0, 1\}$. For $t = T-1,\ldots,1$,

$$K_t(x,i)=\begin{cases}\min\limits_{x'\in\mathcal{L}_{t+1}}-\log P(x'|x)+K_{t+1}(x',i) \text{ if } x\neq x_{1,t},\\ \min\limits_{x'\in\mathcal{L}_{t+1}}-\log P(x'|x)+K_{t+1}(x',i-1) \text{ o.w.,}\end{cases}$$
$$\forall x\in\mathcal{L}_t,\ i\in\{0,\ldots,T-t+1\}, \quad (7)$$

and $n_t(x, i)$ is the value of $x'\in\mathcal{L}_{t+1}$ achieving the minimum. By definition, $K_t(x, i)\equiv K_t(x, T-t+1)$ for all $i > T-t+1$, and $K_t(x, i) = \infty$ for $i < 0$ (infeasible). At $t = 0$, we have

$$K_0(x_0, i)=\min_{x\in\mathcal{L}_1} -\log\pi(x) + K_1(x, i),\ \forall i\in\{0,\ldots,T\}, \quad (8)$$

and $n_0(x_0, i)$ is the $x\in\mathcal{L}_1$ achieving the minimum.

[3] Strictly, each vertex $v\in\mathcal{L}_t$ corresponds to a unique cell $f_t(v)\in\mathcal{L}$. Edge (x_0, x) for each $x\in\mathcal{L}_1$ has cost $-\log\pi(f_1(x))$; edge (x, x') for each $x\in\mathcal{L}_{t-1}$ and $x'\in\mathcal{L}_t$ has cost $-\log P(f_t(x')|f_{t-1}(x))$ ($t = 2,\ldots,T$).

[4] Dijkstra's algorithm works in this case since all the edge costs are nonnegative.

[5] Here $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

Then i^*, defined by the smallest value of $i \in \{0, \ldots, T\}$ satisfying $K_0(x_0, i) < K(p_{\mathbf{x}_1})$, is the optimal value of (4) (if infeasible, then i^* is the smallest i satisfying $K_0(x_0, i) = K(p_{\mathbf{x}_1})$). The optimal chaff's trajectory $\mathbf{x}_2$ is given by:

1) $x_{2,1} = n_0(x_0, i^*)$, and $i_1 = i^*$;
2) for $t = 2, \ldots, T$: $x_{2,t} = n_{t-1}(x_{2,t-1}, i_{t-1})$, and $i_t = i_{t-1}$ if $x_{2,t-1} \neq x_{1,t-1}$ or $i_t = i_{t-1} - 1$ otherwise.

The complexity of this dynamic programming is $O(T^2L^2)$.

V. Performance Analysis

We now analyze the performance of the proposed strategies in Section IV in terms of the tracking accuracy of the eavesdropper in Section III. We denote the time-average tracking accuracy under each strategy by P_{IM}, P_{ML}, and P_{OPT}.

A. Tracking Accuracy under IM

Under the IM strategy, the eavesdropper randomly guesses a trajectory for the user. He correctly tracks the user at time t if and only if (i) he guesses the trajectory right, which occurs with probability $1/N$, or (ii) he guesses the trajectory wrong but the guessed trajectory coincides with the user's trajectory at time t. Thus, the overall tracking accuracy equals

$$P_{\text{IM}} = \frac{1}{N} + \frac{N-1}{N} \cdot \frac{1}{T} \sum_{t=1}^{T} \Pr\{x'_t = x_t\}, \tag{9}$$

where $\mathbf{x}' = (x'_t)_{t=1}^T$ and $\mathbf{x} = (x_t)_{t=1}^T$ are two independent instances of the same MC that describes the user's mobility. Given the steady-state distribution π of this MC, it is easy to see that $\Pr\{x'_t = x_t\} = \sum_{x \in \mathcal{L}} \pi^2(x)$. Therefore,

$$P_{\text{IM}} = \Big(\sum_{x \in \mathcal{L}} \pi^2(x)\Big) + \frac{1}{N}\Big(1 - \sum_{x \in \mathcal{L}} \pi^2(x)\Big). \tag{10}$$

B. Tracking Accuracy under ML

Under the ML strategy, the chaff's trajectory $\mathbf{x}_2$ is deterministic and is guaranteed to be selected by the ML detector[6]. The tracking accuracy is therefore determined by the fraction of time that the user's trajectory coincides with $\mathbf{x}_2$, i.e.,

$$P_{\text{ML}} = \frac{1}{T} \sum_{t=1}^{T} \Pr\{x_{1,t} = x_{2,t}\} = \frac{1}{T} \sum_{t=1}^{T} \pi(x_{2,t}), \tag{11}$$

where $\mathbf{x}_2$ is the solution to (2).

C. Tracking Accuracy under OPT

Under the OPT strategy, the chaff's trajectory is designed to yield the minimum tracking accuracy. Therefore, its tracking accuracy is upper-bounded by the tracking accuracy under any suboptimal strategy.

1) Auxiliary Strategy: To bound the tracking accuracy under the OPT strategy, we introduce a suboptimal strategy whose tracking accuracy can be analyzed in closed form. This strategy, referred to as the *constrained maximum likelihood (CML) strategy*, greedily maximizes the likelihood of the chaff's trajectory under the constraint that the chaff cannot co-locate with the user. That is, given the user's trajectory $\mathbf{x}_1$, the chaff's trajectory $\mathbf{x}_2$ is computed by

1) at $t = 1$, $x_{2,1} = \arg\max_{x \in \mathcal{L} \setminus \{x_{1,1}\}} \pi(x)$;
2) at $t > 1$, $x_{2,t} = \arg\max_{x \in \mathcal{L} \setminus \{x_{1,t}\}} P(x|x_{2,t-1})$.

Note that CML is actually an online strategy as it never requires the future trajectory of the user.

2) Analysis of Auxiliary Strategy: Under the CML strategy, the chaff's trajectory is always disjoint from the user's trajectory, and thus the eavesdropper correctly tracks the user if and only if the ML detector is correct, which occurs only if the user's trajectory has a likelihood no smaller than that of the chaff's trajectory. That is, the tracking accuracy under the CML strategy satisfies

$$P_{\text{CML}} \leq \Pr\{p(\mathbf{x}_1) \geq p(\mathbf{x}_2)\}, \tag{12}$$

where $\mathbf{x}_2$ is generated according to the CML strategy.

As $p(\mathbf{x}) = \pi(x_1) \prod_{t=2}^{T} P(x_t|x_{t-1})$, we can define

$$c_1(x_{1,1}, x_{2,1}) := \log \pi(x_{1,1}) - \log \pi(x_{2,1}), \tag{13}$$

$$c_t(x_{1,t}, x_{2,t}, x_{1,t-1}, x_{2,t-1}) := \log P(x_{1,t}|x_{1,t-1}) - \log P(x_{2,t}|x_{2,t-1}), \quad t > 1, \tag{14}$$

and convert $\Pr\{p(\mathbf{x}_1) \geq p(\mathbf{x}_2)\}$ to

$$\Pr\{c_1(x_{1,1}, x_{2,1}) + \sum_{t=2}^{T} c_t(x_{1,t}, x_{2,t}, x_{1,t-1}, x_{2,t-1}) \geq 0\}. \tag{15}$$

The tracking accuracy under the CML strategy is then upper-bounded by (15). The key is to show that even if c_t's are correlated, as long as $\mathbb{E}[c_t] < 0$, $\Pr\{\sum_t c_t \geq 0\}$ decays exponentially with T. See [17] for details.

Specifically, let c_0 be the maximum value of c_1, $c_{\min}$ ($c_{\max}$) be the minimum (maximum) value of c_t for $t > 1$, and w and δ be constants defined as in Lemma V.2 in [17]. Then we have the following result.

Theorem V.1. Let $\mathbb{E}[c_t] := -\mu$ ($t > 1$). If $\exists \epsilon > 0$ such that $\mu - \epsilon\delta - c_0/(T - w) \geq 0$, then the tracking accuracy under the CML strategy (and thus the OPT strategy) satisfies

$$P_{\text{OPT}} \leq P_{\text{CML}} \leq w \cdot \exp\left(-2\Big(\frac{T}{w} - 1\Big) \frac{(\mu - \epsilon\delta - \frac{c_0}{T-w})^2}{(c_{\max} - c_{\min} + 2\epsilon\delta)^2}\right). \tag{16}$$

Remark: A few remarks are in order:

i) In contrast to the previous strategies (IM, ML) where the tracking accuracy is always non-zero, we see that when the condition in Theorem V.1 holds, the CML strategy and the OPT strategy can both reduce the tracking accuracy to zero.

ii) For a sufficiently large T, the condition in Theorem V.1 holds if and only if $\mathbb{E}[c_t] < 0$. This condition has an information-theoretic interpretation: By definition, $\mathbb{E}[c_t] = H(X_{2,t}|X_{2,t-1}) - H(X_{1,t}|X_{1,t-1})$, where $H(X_{1,t}|X_{1,t-1})$ ($H(X_{2,t}|X_{2,t-1})$) is the *conditional entropy* of the user's

[6] We ignore ties as they occur with an exponentially decaying probability (except for i.i.d. uniform mobility).

(chaff's) movement. Thus, the tracking accuracy decays to zero if $H(X_{1,t}|X_{1,t-1}) > H(X_{2,t}|X_{2,t-1})$, i.e., the user has a higher entropy than the chaff.

VI. Performance Evaluation

We use simulations to evaluate the effectiveness of the proposed chaff control strategies by measuring the eavesdropper's tracking accuracy: the lower, the better.

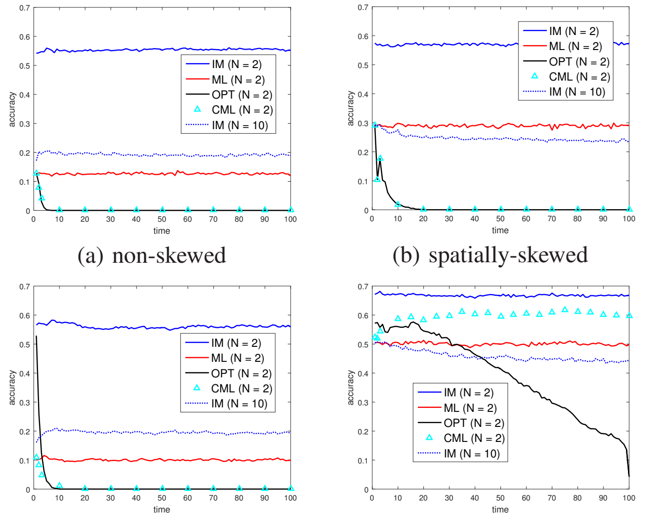
A. Simulation Setting

We generate synthetic mobility traces, where the user follows a MC of L states with transition probabilities specified below, and the chaffs follow one of the proposed strategies. We set $T = 100$, $L = 10$, and vary N from 2 to 10 (recall that $N - 1$ is the number of chaffs). The performance is averaged over 1000 Monte Carlo runs.

We evaluate four different mobility models for the user: (a) *neither* spatially *nor* temporally skewed mobility, represented by a MC with randomly generated transition probabilities, (b) *spatially*-skewed mobility, represented by a MC with a high probability of transiting into a certain cell, (c) *temporally*-skewed mobility, represented by a random walk with a uniform steady-state distribution, and (d) *both* spatially *and* temporally skewed mobility, represented by a random walk with a non-uniform steady-state distribution. See Fig. 4 in [17] for the steady-state distribution under each model.

B. Simulation Results

We evaluate the performance of an eavesdropper using the ML detector (1) in Fig. 3. We see that: (i) while IM and ML always lead to non-zero tracking accuracy, OPT can drive the tracking accuracy to zero; (ii) the more skewed the mobility model (i.e., the more predictable the user's movements), the higher the tracking accuracy; (iii) while the deterministic strategies (ML, OPT) cannot benefit from using more chaffs, the IM strategy can use more chaffs to lower the tracking accuracy. We further simulate the auxiliary strategy CML in Section V-C and verify our analysis that the accuracy under CML decays exponentially if $\mathbb{E}[c_t] < 0$; see Fig. 6 in [17].



(a) non-skewed

(b) spatially-skewed

(c) temporally-skewed

(d) spatially&temporally-skewed

Fig. 3. Tracking accuracy of the eavesdropper.

VII. Conclusion

We studied the problem of protecting the location privacy of a mobile user in MECs using chaff services. Assuming that a cyber eavesdropper tracks the user by performing ML detection among observed service trajectories, we examined a range of chaff control strategies, from a baseline strategy to an optimal strategy. We proved that the optimal strategy can reduce the eavesdropper's tracking accuracy to zero when the user's mobility is sufficiently random, while simpler strategies cannot. Our evaluations highlighted the dependency of the tracking accuracy on the user's mobility model, and verified the efficacy of our solutions in protecting the location privacy, even for users with highly predictable mobility.

References

[1] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM/USENIX MobiSys*, 2003, pp. 31–42.

[2] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios, "Providing k-anonymity in location based services," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 3–10, June 2010.

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

[4] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *ACM/USENIX MobiSys*, 2007, pp. 246–257.

[5] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *IFIP Networking*, May 2015.

[6] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani, "An open ecosystem for mobile-cloud convergence," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 63–70, March 2015.

[7] "Smarter wireless networks," IBM Whitepaper No. WSW14201USEN, Febuary 2013. [Online]. Available: www.ibm.com/services/multimedia/Smarter_wireless_networks.pdf

[8] K. Ha, Y. Abe, Z. Chen, W. He, B. Amos, P. Pillai, and M. Satyanarayanan, "Adaptive VM handoff across cloudlets," Technical Report CMU-CS-15-113, June 2015. [Online]. Available: https://www.cs.cmu.edu/~satya/docdir/CMU-CS-15-113.pdf

[9] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, September 2013.

[10] T. Taleb, A. Ksentini, and P. Frangoudis, "Follow-me cloud: When cloud services follow mobile users," *accepted to IEEE Transactions on Cloud Computing*, February 2016.

[11] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "Dynamic service placement for mobile micro-clouds with predicted future costs," *accepted to IEEE Transactions on Parallel and Distributed Systems*, August 2016.

[12] T. He, L. Tong, and A. Swami, "Maximum throughput of clandestine relay," in *Allerton Conference*, September 2009.

[13] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *IEEE Symposium on Security and Privacy Workshops*, May 2012.

[14] G. Kontaxis, M. Polychronakis, and A. D. Keromytis, "Computational decoys for cloud security," in *Secure Cloud Computing*, December 2013, pp. 261–270.

[15] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for Follow Me cloud," in *IEEE ICC*, June 2014.

[16] S. Wang, R. Urgaonkar, T. He, M. Zafer, K. Chan, and K. K. Leung, "Mobility-induced service migration in mobile micro-clouds," in *IEEE MILCOM*, October 2014.

[17] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds," Technical Report, March 2017. [Online]. Available: https://www.dropbox.com/s/xs67a9zw7d4kwmn/LocationPrivacyInMEC.report.pdf?dl=0

[18] S. Banerjee and D. O. Wu, "Final report from the NSF workshop on future directions in wireless networking," National Science Foundation, November 2013.