Privacy-Aware Offloading in Mobile-Edge Computing

Xiaofan He
EE Dept., Lamar University
e-mail: xhe1@lamar.edu

Juan Liu
EE Dept., Ningbo University
e-mail: eeliujuan@gmail.com

Richeng Jin and Huaiyu Dai
ECE Dept., North Carolina State University
e-mails: {rjin2,hdai}@ncsu.edu

***Abstract*—Recently, mobile-edge computing (MEC) emerges as a promising paradigm to enable computation-intensive and delay-sensitive applications at resource limited mobile devices by allowing them to offload their heavy computation tasks to nearby MEC servers through wireless communications. A substantial body of literature is devoted to developing efficient scheduling algorithms that can adapt to the dynamics of both the system and the ambient wireless environments. However, the influence of these task offloading schemes to the mobile users' privacy is largely ignored. In this work, two potential privacy issues induced by the wireless task offloading feature of MEC, location privacy and usage pattern privacy, are identified. To address these two privacy issues, a constrained Markov decision process (CMDP) based privacy-aware task offloading scheduling algorithm is proposed, which allows the mobile device to achieve the best possible delay and energy consumption performance while maintain a pre-specified level of privacy. Numerical results are presented to corroborate the effectiveness of the proposed algorithm.**

I. Introduction

Nowadays, mobile devices are being regularly upgraded by their manufacturers for faster data processing, larger storage, longer battery life as well as many other new features. However, they still can hardly catch up with the exponentially growing computation requirement of many existing and emerging applications, such as high-resolution video streaming, online 3D gaming, and augmented reality. To address this challenge and provide an enhanced experience quality for mobile device users, the mobile-edge computing (MEC) paradigm (a.k.a. fog computing) has recently been proposed [1]. Central to the MEC vision is the idea of smartly exploiting the ample computation, communication and storage resources of the MEC servers distributed at the network edge by allowing nearby mobile devices to offload their computation-intensive tasks to the MEC servers via wireless communications. More details about the MEC paradigm can be found in [2, 3] and the references therein.

Since the potential performance gain brought by an MEC system highly hinges on the conditions of the wireless channels between the mobile devices and the MEC server, substantial research efforts have been devoted to studying the scheduling and the resource allocation problems in dynamic wireless environment. For example, Lyapunov optimization-based online algorithms have been employed to facilitate the offloading decision-making when the statistical information regarding task arrival and ambient wireless environment is not available [4]. When such information is available, Markov decision process (MDP) based algorithms can be employed to conduct optimal foresighted scheduling that can adapt to the environmental dynamics for better performance [5, 6]. In addition, both optimization (e.g., [7]) and game-theoretic approaches (e.g., [8]) have been exploited to fulfill efficient scheduling in multiuser MEC systems. Similar communication/computation scheduling problems have also been studied in MEC systems with renewable energy powered mobile devices [4] and/or servers [6].

Nonetheless, the security and privacy aspect of MEC remains less explored. In this direction, a few works have been done but they mainly focused on the conventional security and privacy issues of MEC inherited from the conventional cloud computing framework, such as authentication, secure and private data storage and computing, and intrusion detection (c.f. [9, 10] and the references therein). In this work, two privacy issues related to the unique wireless task offloading feature of the MEC system are investigated, namely, *location privacy* and *usage pattern privacy*. Particularly, when the mobile device focuses solely on optimizing delay and energy consumption, it tends to offload all of its tasks to the MEC when the wireless channel condition is good and only activate local processing when wireless channel condition is not satisfactory. As a result, an honest-but-curious MEC server may not only infer the wireless channel information and hence the user's location information but also monitor the user's actual device usage pattern. Privacy-sensitive users may be deterred from accessing the MEC systems if these two critical privacy concerns are not properly addressed. Despite each of these two privacy issues has already been extensively (and separately) studied for other applications, one challenge to resolve these two privacy issues in MEC systems is how to achieve privacy protection while still maintain the best possible delay and energy consumption performance. To the best of our knowledge, this work is among the first to study this problem. Particularly, by casting the problem as a constrained Markov decision process (CMDP), a privacy-aware task offloading scheme is proposed for the MEC system that enables the mobile device to minimize its delay and energy consumption cost while maintain the user's privacy above a pre-specified level.

The remainder of this paper is organized as follows. Sec-

This work was supported in part by the NCSU Science of Security Lablet, in part by the U.S. National Science Foundation under Grants ECCS-1307949 and EARS-1444009, and in part by the China National Science Foundation under Grant 61601255.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

tion II introduces some background of the MEC system and illustrates the corresponding privacy issues. The proposed CMDP based privacy-aware task offloading algorithm is developed in Section III. Numerical results are presented in Section IV to corroborate the effectiveness of proposed algorithm. Related works are discussed in Section V. Conclusions and future works are presented in Section VI.

II. System Model and Problem Formulation

A. The MEC System

The MEC system shown in Fig. 1 is considered. It is assumed that time is slotted and at each timeslot n, the mobile device generates $d_n \in \mathcal{D} \triangleq \{0, 1, ..., d_{max}\}$ computation tasks depending on the user's usage (with d_{max} the maximum possible number of generated tasks) and each task contains M bits. In this work, it is assumed that $\{d_n\}_{n\geq 0}$ follows a Markov process with a transition probability $\mathbb{P}_D(d_{n+1}|d_n)$. Three different scheduling options are assumed to be available for each task: (1) queuing in the buffer, (2) processing by the local unit, or (3) offloading to the MEC server. The scheduling decision has to be made in accordance to the current system state $s_n = (d_n, b_n, h_n) \in \mathcal{S}$, so as to achieve the optimal delay and energy consumption performance. Here, $b_n \in \mathcal{B} \triangleq \{0, 1, ..., b_{max}\}$ represents the current number of tasks in the buffer (with b_{max} the buffer size); h_n represents the condition of the wireless channel between the mobile device and the MEC server. For simplicity, it is assumed in this work that the wireless channel states $\{h_n\}_{n\geq 0}$ follow a Markov process with two states $\mathcal{H} \triangleq \{0, 1\}$ and a state transition probability $\mathbb{P}_H(h_{n+1}|h_n)$.[1] Particularly, $h_n = 1$ ($h_n = 0$) represents a good (bad) channel state at timeslot n, and the corresponding transmit energy for offloading one task to the MEC server is denoted by e_1 (e_0). Based on the current state s_n, the scheduling policy π specifies an action $a_n = (q_n, t_n, l_n)$, with $q_n \in \mathcal{B}$, t_n and $l_n = d_n + b_n - q_n - t_n$ the numbers of tasks to be queued in the buffer, transmitted to the MEC server, and processed by the local processing unit, respectively. Then, the system will transit to a new state $s_{n+1} = (d_{n+1}, b_{n+1}, h_{n+1})$ with d_{n+1} and h_{n+1} determined by $\mathbb{P}_D(\cdot|d_n)$ and $\mathbb{P}_H(\cdot|h_n)$, respectively, and $b_{n+1} = q_n$. At each timeslot, when action a_n is taken under state s_n, the cost to the mobile device is modeled as a weighted combination of the queuing delay and energy consumption, which is given by

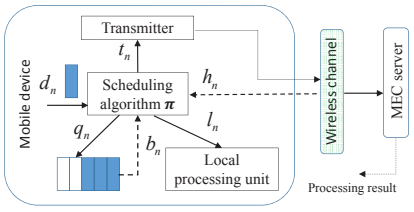

Fig. 1. The considered MEC system.

$$c_n = C(s_n, a_n) \triangleq w_q q_n + e_t(t_n, h_n) + e_l(l_n), \qquad (1)$$

[1]The channel model can be easily extended to the multi-state case.

where $C(\cdot, \cdot)$ is the cost function with the weighting factor w_q reflecting the relative importance of delay over energy consumption. Some further explanations of (1) are in order. The first term captures the queuing delay cost, since from Little's theorem, the average queuing delay is proportional to the average queue length [11, 12]. In addition, it is assumed that both local execution and task offloading can be completed in one timeslot.[2] The second term captures the energy consumption for transmitting t_n tasks to the MEC server for a given channel state h_n. The last term is the energy consumption for locally processing l_n tasks. The objective of the mobile device is to find a suitable scheduling policy π to minimize the expected long-term cost, defined as

$$\bar{C}_{\gamma,\pi,s_0} \triangleq \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n C(s_n, a_n) | s_0 \right], \qquad (2)$$

where $\mathbb{E}_\pi$ means taking expectation according to the law induced by the policy π; s_0 is the initial system state and $\gamma \in [0, 1)$ is the discounting factor; intuitively, a larger γ means that the user is planning for a longer future.

As the system states $\{s_n\}_{n\geq 0}$ form a Markov process controlled by the scheduling policy π, the MDP framework can be employed to find the optimal scheduling policy for MEC system [5, 6].

B. Privacy Issues in MEC

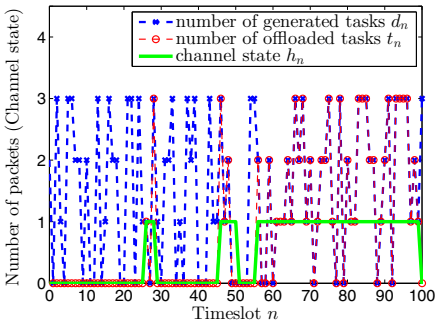

Fig. 2. Optimal scheduling without considering privacy.

Even though the MDP based scheduling policy can help the mobile device optimize delay and energy consumption, as many other existing scheduling policies (e.g., [4, 7, 13]), the corresponding privacy aspect is largely ignored. In this work, two privacy issues caused by the wireless task offloading feature of the MEC system, *location privacy* and *usage pattern privacy*, are identified. To facilitate the discussions in the sequel, a simulation result of a MDP based scheduling policy for the setting discussed in the previous subsection is presented in Fig. 2 (c.f. Section IV for relevant parameters).

[2]In this work, it is assumed that the MEC server has sufficient computation power to execute the offloaded tasks and feedback the processing results in negligible time. Also, for simplicity, it is assumed that the local computation power matches with the task generation rate and buffer size, and hence can process $d_{max} + b_{max}$ tasks in less than one timeslot.

Location privacy: First, it can be noticed from Fig. 2 that, for the considered setting, the mobile device only offloads the tasks to the MEC server when the wireless channel power gain is high (i.e., when $h_n = 1$), while the tasks will be either queued in the buffer or processed locally otherwise (i.e., when $h_n = 0$). Despite the benefits of reducing delay and energy consumption, scheduling policies of this nature may disclose user's location information, since the average wireless channel power gain is highly related to the distance between the user and the MEC server. Therefore, simply by analyzing the offloading pattern, the MEC server may be able to infer the channel state information and hence the distance to the mobile device.[3] In addition, when a mobile user communicates with multiple MEC servers, it is possible that these MEC servers may collude and jointly pinpoint the location of the mobile device to a certain precision.

Usage pattern privacy: In addition to location privacy, the usage pattern privacy is another critical issue ignored by existing literature of MEC. As it can be observed from Fig. 2, when the channel condition is good, the mobile device tends to offload all the generated and buffered tasks at each timeslot to the MEC server so as to minimize latency and energy compution (i.e., $t_n = d_n + b_n$ when $h_n = 1$). When the wireless channel is persistently good for some user (e.g., someone whose office is near to the access point), it always has $t_n = d_n$ (as $b_n = 0$ for $n > 1$). As d_n is determined by the user's actual usage of the device, the MEC server may be able to infer personal information of that user through monitoring t_n. For example, the MEC server may be able to extract statistical information and even patterns of each device's usage based on its task offloading history and use that as the fingerprint to identify the presence of a certain user. In addition, it might even be possible for the MEC server to pinpoint the app running at the user side, when a certain pattern exists in the number of tasks generated by the app [14, 15].

Needless to say, these privacy issues, if not properly addressed, will discourage the usage of MEC systems by privacy-sensitive users. For this reason, the delay and power consumption and privacy should be jointly considered by the mobile devices when designing the scheduling policy.

III. Proposed Solution

In this section, a privacy-aware task offloading scheduling algorithm is proposed for the MEC system based on a CMDP formulation.

A. Privacy Metric

Before introducing the CMDP formulation, the metric for quantifying the privacy achieved by a scheduling policy is discussed first. To the best of our knowledge, there is no benchmark metric available in the literature that can jointly quantify the location and usage pattern privacy discussed in the previous section. Considering this, a heuristic (yet effective, as will be seen in Section IV) privacy metric is considered in this work. Particularly, for a given state-action pair (s_n, a_n), the achieved privacy at timeslot n is modeled as

$$p_n = P(s_n, a_n) \triangleq |d_n - t_n| \cdot \mathbb{1}_{\{h_n=1\}} + w_u \cdot \mathbb{1}_{\{t_n>0\}} \cdot \mathbb{1}_{\{h_n=0\}}, \tag{3}$$

where $P(\cdot,\cdot)$ is the privacy function; $\mathbb{1}_{\{\cdot\}}$ is the indicator function; and w_u is a weighting factor reflecting the relative importance of location privacy over usage pattern privacy. The intuition behind (3) are explained in the sequel. First notice that, to protect the usage pattern privacy, the mobile device needs to properly adjust the number of tasks to be buffered and locally processed so as to deliberately create a difference between the amount of generated tasks d_n and that of the offloaded tasks t_n when the channel state is relatively good. This is captured by the first term of (3). To protect the location privacy, the mobile device may need to offload some tasks even when the channel condition is not good, which is captured by the second term of (3). Note that this can be achieved by deliberately keeping some of the generated tasks in the local buffer when $h_n = 1$ and offloading them when $h_n = 0$, or directly transmitting some dummy tasks if the buffer is empty and no new task is generated at that timeslot.[4]

Based on the modeling in (3), the average discounted long-term privacy performance of the mobile device is given by

$$\bar{P}_{\gamma,\pi,s_0} \triangleq \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n P(s_n, a_n) | s_0 \right]. \tag{4}$$

B. The Proposed CMDP Formulation

Based on the discussion above, it can be readily noticed that there is a tradeoff between privacy protection and the delay and energy cost. In this work, the objective is to find the best possible task offloading policy that can minimize the delay and energy cost under a given user-specified long-term privacy level $\bar{p}$. Naturally, this can be formulated as a CMDP as follows

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n C(s_n, a_n) | s_0 \right] \qquad (\mathbf{P}_c) \\ \text{s.t.} \quad & \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n P(s_n, a_n) | s_0 \right] \geq \bar{p}. \end{aligned}$$

As to the existence of the solution to the CMDP, it is assumed that there exists at least one feasible policy; otherwise, the user should specify a lower privacy level $\bar{p}$. To solve the above constrained optimization problem, for each state $s \in \mathcal{S}$, the corresponding Lagrangian is defined as

$$\mathcal{L}(s; \lambda, \pi) \triangleq \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n C(s_n, a_n) | s \right]$$

[3] Small-scale fading effects may be eliminated by averaging over a certain time period.

[4] As a result, unlike the MDP formulation discussed in Section II-A, t_n may be larger than $d_n + b_n - q_n - l_n$. In addition, to maintain a bounded action space, it is further assumed that $t_n \leq d_{max} + b_{max}$, since if $t_n > d_{max} + b_{max}$, the MEC server can easily identify that the mobile device is sending dummy tasks.

$$+\lambda \cdot \left(\bar{p} - \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n P(s_n, a_n) | s \right] \right). \quad (5)$$

Further notice that, every fixed Lagrange multiplier λ induces an unconstrained MDP with the corresponding per timeslot cost function C_u given by

$$C_u(s_n, a_n) = C(s_n, a_n) + \lambda \cdot ((1-\gamma)\bar{p} - P(s_n, a_n)). \quad (6)$$

The following fundamental result [16] for CMDP is essential to the two-timescale learning algorithm presented later in this section.

Theorem 1: For a given initial state $s \in \mathcal{S}$ and γ, the optimal value of the CMDP is given by

$$\bar{C}^*_{\gamma,\pi^*,s} = \max_{\lambda \geq 0} \min_{\pi} \mathcal{L}(s; \lambda, \pi), \quad (7)$$

and a policy π^* is optimal if and only if $V(s) = \max_{\lambda \geq 0} \mathcal{L}(s; \lambda, \pi^*)$. That is, the optimal policy π^* and Lagrange multiplier λ^* admits the following *saddle point* property

$$\mathcal{L}(s; \lambda^*, \pi) \geq \mathcal{L}(s; \lambda^*, \pi^*) \geq \mathcal{L}(s; \lambda, \pi^*), \quad \forall \pi, \lambda \geq 0. \quad (8)$$

Note that the Lagrangian defined in (5) admits the following dynamic programming equations

$$Q(s,a) = C(s,a) + \lambda^* \cdot ((1-\gamma)\bar{p} - P(s,a)) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) \mathcal{L}(s'; \lambda^*, \pi^*), \quad (9)$$

$$\mathcal{L}(s; \lambda^*, \pi^*) = \min_{a \in \mathcal{A}} Q(s,a). \quad (10)$$

Here, the Q-function $Q(s,a)$ represents the minimum cost, when action a is taken at state s and the optimal strategy π^* is followed in the rest of the time. Based on these dynamic programming equations, the following two-timescale learning algorithm [11] can be used to learn λ^* and π^*. Particularly, at each timeslot n, the mobile device observes the current state s_n, and then, with probability $(1-\epsilon_n)$, takes an action $a_n = \arg\min_{a \in \mathcal{A}} Q_n(s_n, a)$ based on the current estimate $Q_n(s,a)$ of the Q-function $Q(s,a)$, and with probability ϵ_n, takes an action uniformly at random (i.e., exploration). In practice, the exploration rates $\{\epsilon_n\}_{n \geq 0}$ are usually chosen to be small positive numbers vanishing to zero. For example, it is suggested in [17] that, for each current state s, set $\epsilon_n(s) = \epsilon_0^{n(s)}$ with $\epsilon_0 \in (0,1)$ and $n(s)$ the number of times state s has occurred up to current timeslot; the underlying rationale is to allow higher exploration rates for the less familiar states. After this, it further observes the next state s_{n+1}. Upon collecting such information, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, the estimate $Q_n(s,a)$ of the Q-function $Q(s,a)$ and the estimate $\mathcal{L}_n(s)$ of the value function $\mathcal{L}(s; \lambda^*, \pi^*)$ is updated by

$$Q_{n+1}(s,a) = Q_n(s,a) + \{(s,a)=(s_n,a_n)\} \cdot \alpha_n \cdot [c_n + \lambda_n \cdot (\bar{p} - p_n) + \gamma \cdot \mathcal{L}_n(s_{n+1}) - Q_n(s,a)], \quad (11)$$

$$\mathcal{L}_{n+1}(s) = \min_{a \in \mathcal{A}} Q_{n+1}(s,a), \forall s \in \mathcal{S}, \quad (12)$$

and the Lagrange multiplier is updated by

$$\lambda_{n+1} = \max\{0, \lambda_n + \beta_n \cdot [(1-\gamma)\bar{p} - p_n]\}. \quad (13)$$

In the above equations, α_n and β_n are the two learning rates corresponding to the fast- and slow-timescale, respectively, and admit following standard conditions: $\sum_n \alpha_n = \sum_n \beta_n = \infty$, $\sum_n (\alpha_n^2 + \beta_n^2) < \infty$, and $\lim_{n \to \infty} \beta_n / \alpha_n = 0$.

The above algorithm is summarized in Algorithm 1. The rational behind the two-timescale learning algorithm is the following. From the perspective of the Q- and the value functions updated in the (virtually) faster timescale, the Lagrange multiplier in the slower timescale is a fixed constant, and hence (11) and (12) can be treated as the classic Q-learning for the unconstrained MDP induced by a fixed Lagrange multiplier, with the corresponding cost function given by (6). Viewing from the Lagrange multiplier's slower timescale, the value of the unconstrained MDP quickly converges to $\min_{\pi} \mathcal{L}(s; \lambda, \pi)$. In addition, the term $[(1-\gamma)\bar{p} - p_n]$ in (13) is a stochastic sub-gradient of $\min_{\pi} \mathcal{L}(s; \lambda, \pi)$ with respect to λ (c.f. (5)), and hence (13) is a stochastic sub-gradient update on λ with a corresponding objective function (7); its convergence is ensured by the fact that $\min_{\pi} \mathcal{L}(s; \lambda, \pi)$ is piecewise linear and concave in λ [17]. Formalizing the above argument, the following result from [11] is recapped here.

Theorem 2: In Algorithm 1, $\lambda_n \to \lambda^*$ and $\mathcal{L}_n(s) \to \mathcal{L}(s; \lambda^*, \pi^*)$ for all $s \in \mathcal{S}$, as $n \to \infty$.

Remark 1: Note that when the state transition probability $\mathbb{P}(s'|s,a)$ is known, the CMDP can be solved as a Linear Programming problem as discussed in [16], which is an offline approach. The above learning algorithm is more general in the sense that it can be applied to not only the offline setting but also the online setting where the state transition probability is unknown. Also, if the learning coefficients α_n and β_n are properly kept away from zero, the above learning algorithm can track the non-stationary environment dynamics (i.e., a (slowly) varying $\mathbb{P}(s'|s,a)$). In addition, it is worth noting that a similar online learning algorithm was developed in [17] for average-cost CMDP.

Algorithm 1 Privacy-Aware Task Offloading.

Initialization: $n = 0$, $Q_0(s,a) = \mathbf{0}$, $\mathcal{L}_0(s) = \mathbf{0}$ and $\lambda_0 = 0$ uniform.

1. Observe the current buffer and channel state $s_n = (b_n, h_n)$
2. Taking action $a_n = (q_n, l_n, t_n)$ at current state s_n
   - uniformly at random with probability ϵ_n;
   - otherwise, $a_n = \arg\min_{a \in \mathcal{A}} Q_n(s_n, a)$.
3. Keep q_n tasks in the buffer, use local resource process l_n tasks, and offload t_n tasks to the MEC server
4. After observing the cost c_n (given by (1)), privacy payoff p_n (given by (3)), and the next state s_{n+1}
   - Compute Q_{n+1}, $\mathcal{L}_{n+1}$, and λ_{n+1} using (11), (12) and (13), respectively;
5. $n = n + 1$, then repeat till convergence.

IV. Numerical Results

In this section, some numerical results are presented to validate the effectiveness of the proposed algorithm. In the simulations, it is assumed that, in (1), $e_t(t_n, h_n) = t_n \cdot e_{h_n}$ with $e_{h_n} \in \{e_0, e_1\}$ the consumed energy for transmitting one task to the server under channel state h_n, and that $e_l(l_n) = l_n \cdot e_l$ with e_l the required local processing energy per task. In addition, it is assumed that each timeslot is one second, and that the local CPU operates at $f = 2$ (GHz) and the corresponding operation power is given by $P_{cpu} = \kappa \cdot f^3 = 8$ (W) with $\kappa = 1 \times 10^{-27}$ [18]. It is futher assumed that each task contains $M = 500K$ bits. The energy consumption of locally processing a task is $e_l = \frac{M \cdot \eta}{f} \times P_{cpu} = 1$ (J), where it is assumed that $\eta = 500$ cycles/bit [19]. In addition, assume that the bandwidth is $B = 5$ (MHz), the offloading time for each task is 0.1 (s), and the channel power gain to noise power $N_0 B$ ratio is 0.2 (0.05) when $h_n = 1$ $(h_n = 0)$; hence, the corresponding per task transmit energy is $e_1 = 0.5$ (J) $(e_0 = 2$ (J)). The channel state transition probability is set to $\mathbb{P}_H(h_{n+1} = 1|h_n = 1) = \mathbb{P}_H(h_{n+1} = 0|h_n = 0) = 0.95$. In addition, for the task generation process, the transition probability is set to $\mathbb{P}_D(d_{n+1}|d_n) = 1/(1 + d_{max})$. The weighting factor w_q in (1) is set to 1.5, which means the user concerns more about queuing delay than energy consumption when the channel gain is low; the weighting factor w_u in (3) is set ot 0.8. The simulation is conducted over $T = 3 \times 10^4$ timeslots. The discounting factor is set to $\gamma = 0.9$; in general, a larger γ can lead to a better scheduling at the cost of a longer learning time. The learning parameters are set to $\alpha_n = (1 + \nu_n(s_n, a_n))^{-0.55}$ and $\beta_n = (1 + n)^{-0.6}$, with $\nu_n(s, a)$ the number of times that the state-action pair (s, a) appears up to timeslot n.

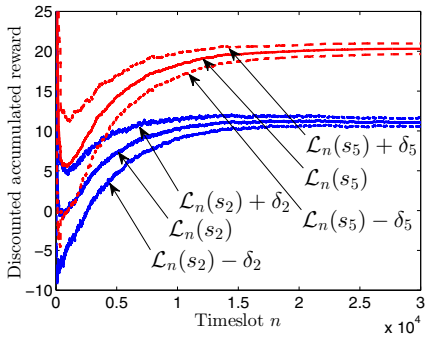

Fig. 3. Convergence of the Lagrangian ($\bar{p} = 0.6/(1-\gamma)$, $d_{max} = 3$, $b_{max} = 5$).

The learning curves of the value function $\mathcal{L}_n(s_i)$'s are shown in Fig. 3, where δ_i denotes the standard deviation of $\mathcal{L}_n(s_i)$ over the conducted 100 Monte Carlo runs.[5] It can be seen that, as n increases, the value functions gradually converge as indicated by Theorem 2.

[5]Since it is not convenient to show the value functions for all the 48 states, two states $s_2 = (0, 0, 1)$ and $s_5 = (0, 2, 0)$ are chosen for the demonstration here.

The task offloading behavior of using the proposed algorithm is shown in Fig. 4. It can be seen that, on the one hand, the mobile device will transmit some (possibly dummy) tasks even when the channel condition is $h_n = 0$ (e.g., when $n = 91$) and hence can protect its location privacy; on the other hand, through proper queuing and local processing, the mobile device also successfully creates a difference between the generated task and the offloaded task (e.g., when $n = 7$ and $n = 100$). This is in contrast to Fig. 2 where t_n is always zero when $h_n = 0$ and equal to d_n when $h_n = 1$.

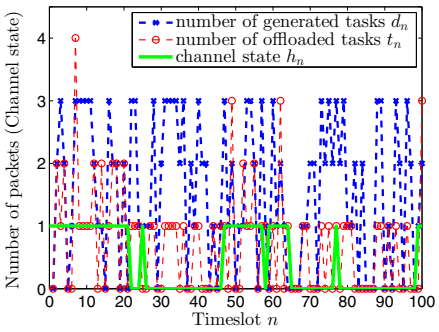

Fig. 4. Improved privacy by the proposed algorithm ($\bar{p} = 0.6/(1-\gamma)$, $d_{max} = 3$, $b_{max} = 5$).

To further illustrate the effectiveness of the proposed algorithm in minimizing the delay and energy consumption, a naive privacy-aware offloading scheme is considered here to serve as the baseline for comparison. Particularly, in the naive scheme, the mobile device follows the optimal MDP strategy without concerning about privacy with probability $(1 - p_d)$ and takes action chosen uniformly at random with probability $p_d \in [0, 1]$; intuitively, the larger p_d is, the better (worse) the privacy (delay and energy consumption) is. The performance of this naive scheme is compared with that of the proposed scheme in Table I, under different privacy requirements $\bar{p}$'s.[6] Note that $\bar{p}$ is the required lower bound for the long-term privacy performance, and $(1-\gamma)\bar{p}$ in Table I can be treated as the corresponding average privacy requirement per timeslot. Also, $\bar{c}$ is the average cost per timeslot, defined as $\bar{c} = \frac{1}{T}\sum_{n=0}^{T} c_n$. It can be seen from Table I that a higher privacy requirement will cause larger delay and energy consumption in both the naive and the proposed schemes (i.e., a privacy-performance tradeoff). However, the cost of the proposed scheme is always much lower than that of the naive scheme. For example, if the user specified privacy level is $(1-\gamma)\bar{p} = 0.6$, the extra delay and energy consumption cost incurred by the naive approach is about 100% (as compared to that for $(1-\gamma)\bar{p} = 0$), while the corresponding extra cost of the proposed scheme is only about 45%.

[6]In Table I, as it is not possible to simulate for all (infinite many) p_d, only a finite set of $p_d \in \{0, 0.05, ..., 0.9\}$ are considered in the simulation as an approximation. For each $(1-\gamma)\bar{p}$, the smallest p_d within this set that can achieve the corresponding privacy requirement is adopted.

TABLE I
PERFORMANCE AND PRIVACY TRADEOFF ($d_{max} = 3, b_{max} = 5$).

$(1-\gamma)\bar{p}$	0	0.1	0.2	0.3
$\bar{c}$ (proposed)	1.12	1.19	1.25	1.38
$\bar{c}$ (naive)	1.12	1.25	1.44	1.74
p_d	0	0.1	0.25	0.35
$(1-\gamma)\bar{p}$	0.4	0.5	0.6	0.7
$\bar{c}$ (proposed)	1.49	1.56	1.62	1.67
$\bar{c}$ (naive)	1.83	2.03	2.26	2.4
p_d	0.45	0.6	0.8	0.9

V. RELATED WORKS

Privacy is an ever presented concern for many information systems and MEC is no exception. The privacy issues of the MEC systems include (at least) the following three aspects: (1) data and computing privacy, (2) usage pattern privacy, and (3) location privacy.

Data and computing privacy mainly concerns about fulfilling the designated computing tasks without disclosing users' sensitive information in the data. Many existing solutions developed in the context of other applications may be adapted to address this issue in the MEC systems. For example, existing secure multiparty computation and homomorphic encryption techniques can be employed to achieve privacy-preserving computation in MEC [20]. Differential privacy is another potential candidate to achieve privacy protection in inquiry-like tasks in MEC [21].

The usage pattern privacy and the location privacy themselves have already been studied in the context of other applications. For example, the usage pattern privacy of smart metering has received substantial amount of research efforts in the past decade. Two common approaches for protecting the privacy of smart metering are reshaping power consumption curves through residential battery and properly adjusting the sampling rate [22]. Location privacy has been studied in an even broader range of applications, including cognitive radio systems [23], wireless sensor networks [24] and location-based service systems [25]. However, existing solutions to these two privacy issues are not readily applicable to the MEC systems, when a joint design of privacy protection and delay-and-energy cost minimization is required. In fact, to the best of our knowledge, these two crucial privacy issues of MEC have only been briefly mentioned in [10] and the corresponding solutions still remain open. In this sense, the algorithm proposed in this work is among the first efforts towards addressing them.

VI. CONCLUSIONS AND FUTURE WORKS

In this work, two potential privacy issues induced by the unique wireless task offloading feature of MEC, location privacy and usage pattern privacy, are identified. To address these two privacy issues, a privacy-aware task offloading scheduling algorithm is proposed based on the CMDP framework. The proposed algorithm allows the mobile device to achieve the best possible delay and energy consumption performance while maintain the pre-specified level of privacy. In addition, the proposed algorithm is applicable to both online and offline settings. The effectiveness of the proposed algorithm is validated by numerical results. Examining the proposed scheme at more general MEC scenarios (e.g., multi-state wireless channels and different task generation processes), exploring other possible privacy metrics, and developing faster (approximate) learning algorithms by exploiting problem-specific structures are all interesting future directions.

REFERENCES

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. Mobile edge computing-A key technology towards 5G. *ETSI White Paper*, 11, 2015.

[2] A. Ahmed and E. Ahmed. A survey on mobile edge computing. In *IEEE ISCO*, 2016.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. Mobile edge computing: Survey and research outlook. *arXiv preprint arXiv:1701.01090*, 2017.

[4] Y. Mao, J. Zhang, and K. B. Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE JSAC*, 34(12):3590–3605, 2016.

[5] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief. Delay-optimal computation task scheduling for mobile-edge computing systems. In *IEEE ISIT*, 2016.

[6] J. Xu and S. Ren. Online learning for offloading and autoscaling in renewable-powered mobile edge computing. In *IEEE GLOBECOM*, 2016.

[7] C. You and K. Huang. Multiuser resource allocation for mobile-edge computation offloading. In *IEEE GLOBECOM*, 2016.

[8] X. Chen, L. Jiao, W. Li, and X. Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, 2016.

[9] R. Roman, J. Lopez, and M. Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Elsevier Future Generation Computer Systems*, 2016.

[10] S. Yi, Z. Qin, and Q. Li. Security and privacy issues of fog computing: A survey. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 685–695. Springer, 2015.

[11] N. Mastronarde and M. Van der Schaar. Fast reinforcement learning for energy-efficient wireless communication. *IEEE Transactions on Signal Processing*, 59(12):6262–6266, 2011.

[12] S. M. Ross. *Introduction to probability models*. Academic press, 2014.

[13] L. Pu, X. Chen, J. Xu, and X. Fu. D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE JSAC*, 34(12):3887–3901, 2016.

[14] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic. Who do you sync you are?: Smartphone fingerprinting via application behaviour. In *ACM WiSec*, 2013.

[15] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde. Can't you hear me knocking: Identification of user actions on android apps via traffic analysis. In *ACM CODASPY*, 2015.

[16] E. Altman. *Constrained Markov decision processes*. CRC Press, 1999.

[17] V. S. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems & control letters*, 54(3):207–213, 2005.

[18] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief. Power-delay tradeoff in multi-user mobile-edge computing systems. In *IEEE GLOBECOM*, 2016.

[19] A. P. Miettinen and J. K. Nurminen. Energy efficiency of mobile clients in cloud computing. In *ACM USENIX*, 2010.

[20] W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In *ACM Workshop on New Security Paradigms*, 2001.

[21] C. Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.

[22] S. Finster and I. Baumgart. Privacy-aware smart metering: A survey. *IEEE Communications Surveys & Tutorials*, 16(3):1732–1745, 2014.

[23] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao. Location privacy leaking from spectrum utilization information in database-driven cognitive radio network. In *ACM CCS*, 2012.

[24] M. Conti, J. Willemsen, and B. Crispo. Providing source location privacy in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(3):1238–1280, 2013.

[25] C. Y. Chow and M. F. Mokbel. Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, 13(1):19–29, 2011.