Differential Privacy in Cooperative Multiagent Planning

Abstract

Privacy-aware multiagent systems must protect agents' sensitive data while simultaneously ensuring that agents accomplish their shared objectives. Towards this goal, we propose a framework to privatize inter-agent communications in cooperative multiagent decision-making problems. We study sequential decision-making problems formulated as cooperative Markov games with reach-avoid objectives. We apply a differential privacy mechanism to privatize agents' communicated symbolic state trajectories, and analyze tradeoffs between the strength of privacy and the team's performance. For a given level of privacy, this tradeoff is shown to depend critically upon the total correlation among agents' state-action processes. We synthesize policies that are robust to privacy by reducing the value of the total correlation. Numerical experiments demonstrate that the team's performance under these policies decreases by only 6 percent when comparing private versus non-private implementations of communication. By contrast, the team's performance decreases by 88 percent when using baseline policies that ignore total correlation and only optimize team performance.

1 INTRODUCTION

In cooperative multiagent systems, a team of decision-making agents interact with a shared environment to accomplish a common objective [Cao et al., 2013, Parker et al., 2016]. In these systems, inter-agent communication is often necessary for the successful coordination of the team; each agent typically relies on information pertaining to its teammates while making its own decisions. However, this communicated information may be sensitive. For example, it may be beneficial for autonomous vehicles to share location data while solving multi-vehicle routing problems. However, this would reveal the passengers' sensitive location data. Privacy-aware multiagent systems should protect the agents' sensitive data, while simultaneously ensuring that the agents are able to accomplish their common objective.

In this work, we develop such privacy-aware multiagent systems. In particular, we study sequential multiagent decision problems formulated as cooperative Markov games with reach-avoid objectives. We assume that a trusted central aggregator is used to synthesize a collection of local policies for the team of agents *a priori*. In general, the local policy of a particular agent maps from the state information of some subset of its teammates, to the agent's local action space. However, during policy execution, the agents want to keep their individual state trajectories private from their teammates and from potential eavesdroppers (the aggregator is not involved at run time). When the local policies do not take privacy into consideration, their performance under private communications can decrease dramatically, as shown by our numerical results. Thus, we develop a framework to privatize the inter-agent communications required to execute the policies, and to synthesize policies that are performant under private communications.

We use *differential privacy* [Dwork et al., 2014] to develop a framework providing formal privacy guarantees in multiagent systems. In the Markov game, each agent is modeled by a Markov decision process (MDP) and we are concerned with privatizing the state trajectories of these MDPs. We implement differential privacy using the Online Mechanism for Markov chains presented by Chen et al. [2022]. Under an assumption on the structure of the dependencies of the agents' local policies on their teammates' states, we show that this mechanism guarantees differential privacy for the symbolic state trajectories produced by the MDPs. The mechanism also provides an efficient method for agents to generate private states in real time, and it ensures that each agent's private trajectory is feasible with respect to the underlying dynamics of its MDP. Finally, the strength of the privacy guarantees can be tuned by each agent.

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

Our specific contributions in this work are as follows:

1. *A framework for differential privacy in multiagent systems.* We propose a framework for differential privacy in multiagent planning problems. The framework allows for the decentralized execution of local policies under private inter-agent communications.
2. *Theoretical results: Analyzing the tradeoff between privacy and performance.* We bound the team's success probability under private communications in terms of the strength of privacy and total correlation of agents' state-action processes.
3. *Synthesis of policies robust to private communications.* By minimizing this total correlation value we synthesize policies for the agents that achieve high performance under strong levels of privacy.

Numerical experiments demonstrate the strong performance of the synthesized policies, even with private communications. We observe that under privatized communication: 1) the proposed minimum-dependency policies are 84 percent more performant than baseline policies that only optimize the team's performance under truthful communications and that ignore total correlation, 2) as the total correlation decreases, the team's performance increases, and 3) the performance of the minimum-dependency policies is robust to the level of privacy enforced by the privacy mechanism.

Despite the importance of privacy in multiagent systems [Such et al., 2014], existing algorithms for multiagent planning and learning typically do not examine the tradeoff between privacy and team performance, and many do not consider privacy at all. Nissim and Brafman [2014], Brafman [2015] introduce the notion of *strong privacy* in mutliagent planning for deterministic problems. These works develop algorithms that ensure agents do not share sensitive states or actions when executing a distributed planning algorithm. Ye et al. [2022] replaces the notion of strong privacy with differential privacy to privatize the information needed for decentralized planning in a deterministic case. Hefner et al. [2022] extends the notion of strong privacy to stochastic systems and develops a distributed value iteration algorithm. These works are concerned with hiding a private portion of each agent's states, and they do not consider mechanisms in which the agents achieve privacy by altering their shared information. By contrast, our work studies a differential privacy mechanism that alters the state trajectories of the agents during multiagent communication in stochastic environments.

Meanwhile, differential privacy has been studied in the context of planning and reinforcement learning for MDPs [Garcelon et al., 2020, Qiao and Wang, 2022, Gohari et al., 2021]. However, these works study single-agent problems and they are mainly concerned with privatizing value functions, reward values, or transition probabilities. Our work instead considers the multiagent setting and we define differential privacy over symbolic state trajectories. In particular, we extend the differential privacy mechanism presented by Chen et al. [2022] to multiagent planning problems, and we study the impact of privacy on the team's performance.

Decentralized policy execution has gained attention for planning and reinforcement learning in multiagent MDPs [Becker et al., 2003, Rashid et al., 2018, Son et al., 2019, Oliehoek and Amato, 2016, Karabag et al., 2022]. As a byproduct of decentralized policy execution, these algorithms may achieve privacy in the sense that agents do not communicate locally available information. However, these works do not explicitly consider privacy or give privacy guarantees. We instead allow for communication and use total correlation as a soft decentralization metric, which enables the synthesis of policies that are performant under private communications.

2 PRELIMINARIES

2.1 COOPERATIVE MARKOV GAMES

Given a finite collection of N agents indexed by $i \in \{1, 2, \ldots, N\}$, we model the dynamics of agent i with an MDP $\mathcal{M}^i$. An MDP is a tuple $\mathcal{M}^i = (\mathcal{S}^i, s_I^i, \mathcal{A}^i, \mathcal{T}^i)$, where $\mathcal{S}^i$ is agent i's finite set of local states, $s_I^i \in \mathcal{S}^i$ is an initial state, $\mathcal{A}^i$ is agent i's finite set of local actions, and $\mathcal{T}^i : \mathcal{S}^i \times \mathcal{A}^i \to \Delta(\mathcal{S}^i)$ is a transition probability function, where $\Delta(\mathcal{S}^i)$ denotes the set of probability distributions over the state space $\mathcal{S}^i$. For brevity, we use $\mathcal{T}^i(s^i, a^i, y^i)$ to denote the probability of y^i given by the distribution $\mathcal{T}^i(s^i, a^i)$. A state $s_j^i \in \mathcal{S}^i$ is called a *feasible state* of another state $s_k^i \in \mathcal{S}^i$ if there exists an action $a^i \in \mathcal{A}^i$ such that $\mathcal{T}^i(s_k^i, a^i, s_j^i) > 0$.

Given such a collection of agents, we formulate the team's decision problem as a cooperative Markov game $\boldsymbol{\mathcal{M}}$. A cooperative Markov game involving N agents, each of which is modeled by an MDP $\mathcal{M}^i = (\mathcal{S}^i, s_I^i, \mathcal{A}^i, \mathcal{T}^i)$, is given by the tuple $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mathcal{S}}, \boldsymbol{s_I}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{T}})$. Here, $\boldsymbol{\mathcal{S}} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^N$ is the joint state space, $\boldsymbol{s_I} = (s_I^1, \ldots, s_I^N)$ is the joint initial state, $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ is the joint action space, and $\boldsymbol{\mathcal{T}}$ is the joint transition probability function. For brevity, we use $\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y})$ to denote the probability of $\boldsymbol{y}$ given the distribution $\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a})$. Let $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ denote a joint state and action, respectively. $\boldsymbol{\mathcal{T}}$ is defined as $\boldsymbol{\mathcal{T}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{y}) = \prod_{i=1}^N \mathcal{T}^i(s^i, a^i, y^i)$ for all $\boldsymbol{s} = (s^1, \ldots, s^N) \in \boldsymbol{\mathcal{S}}$, $\boldsymbol{y} = (y^1, \ldots, y^N) \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{a} = (a^1, \ldots, a^N) \in \boldsymbol{\mathcal{A}}$.

For notational convenience, we use $\boldsymbol{s}^{-i} \in \boldsymbol{\mathcal{S}}^{-i} = \mathcal{S}^1 \times \ldots \times \mathcal{S}^{i-1} \times \mathcal{S}^{i+1} \times \ldots \times \mathcal{S}^N$ to denote the states of agent i 's teammates, excluding agent i itself. Similarly $\boldsymbol{a}^{-i}$ and $\boldsymbol{\mathcal{A}}^{-i}$ denote the actions of agent i 's teammates and the set of all possible actions of teammates, respectively.

A (stationary) local policy $\pi^i : \boldsymbol{\mathcal{S}} \to \Delta(\mathcal{A}^i)$ of Agent i is a mapping from a particular joint state to a probability distribution over actions of Agent i. Given the team is in joint state $\boldsymbol{s}$, $\pi^i(\boldsymbol{s}, a^i)$ denotes the probability that action a^i is selected by π^i for agent i . We define a (stationary) joint policy $\boldsymbol{\pi}$ to be a collection of local policies, $\{\pi^i\}_{i=1}^N$.

In a truthful communication setting, at each timestep each agent i observes its local state s_t^i, and communicates this information with all of its teammates. Each agent then uses the state information communicated by its teammates, and its local policy π^i, to sample an action $a^i \in \mathcal{A}^i$ to execute.

In this work we consider team reach-avoid problems. The centralized planning problem is to solve for a collection of local policies $\{\pi^i\}_{i=1}^N$ maximizing the probability that the team reaches a target set $\boldsymbol{\mathcal{S}}_{\mathcal{T}} \subseteq \boldsymbol{\mathcal{S}}$ of states from the team's initial joint state $\boldsymbol{s_I}$, while avoiding a set $\boldsymbol{\mathcal{S}}_{\mathcal{A}} \subseteq \boldsymbol{\mathcal{S}}$ of states. We call this probability value the success probability. More formally, we say that a state-action *trajectory* $\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\boldsymbol{a}_1\ldots$ successfully reaches the target set $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$ if there exists some time M such that $\boldsymbol{s}_M \in \boldsymbol{\mathcal{S}}_{\mathcal{T}}$ and for all $t < M$, $\boldsymbol{s}_t \notin \boldsymbol{\mathcal{S}}_{\mathcal{A}}$. While we focus on reach-avoid problems, our framework can be applied to settings with generic rewards.

We use $x_{\boldsymbol{s},\boldsymbol{a}}$ to denote the occupancy measure of the state-action pair $(\boldsymbol{s}, \boldsymbol{a})$, i.e., the expected number of times that action $\boldsymbol{a}$ is taken at state $\boldsymbol{s}$. Similarly, x_{s^i,a^i} denotes the the occupancy measure of the state-action pair $\left(s^i, a^i\right)$ for agent i where $x_{s^i,a^i} = \sum_{\boldsymbol{s}^{-i}\in\boldsymbol{\mathcal{S}}^{-i}} \sum_{\boldsymbol{a}^{-i}\in\boldsymbol{\mathcal{A}}^{-i}} x_{(s^i,\boldsymbol{s}^{-i}),(a^i,\boldsymbol{a}^{-i})}$. Let $\boldsymbol{\mathcal{S}}_{\mathcal{D}}$ be the states from which the probability of reaching $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$ is 0 under any collection of local policies. The following assumption ensures that every trajectory satisfies or violates the reachability specification in finite time.

Assumption 1. *The total occupancy measure is finite at states $\boldsymbol{\mathcal{S}}\setminus(\boldsymbol{\mathcal{S}}_{\mathcal{T}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{D}})$, i.e., $\sum_{\boldsymbol{s}\in\boldsymbol{\mathcal{S}}\setminus(\boldsymbol{\mathcal{S}}_{\mathcal{T}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{D}}),\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} x_{\boldsymbol{s},\boldsymbol{a}} < \infty$.*

A state-action trajectory ξ^i of the MDP $\mathcal{M}^i$ is a sequence $\xi^i = s_0^i a_0^i s_1^i a_1^i \ldots$ such that for all $t = 0, 1, \ldots$, $\mathcal{T}(s_t^i, a_t^i, s_{t+1}^i) > 0$. We use $\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\boldsymbol{s}_1\ldots$ to denote the joint state-action trajectory of all agents and $\boldsymbol{\xi}^{-i} = \boldsymbol{s}_0^{-i}\boldsymbol{a}_0^{-i}\boldsymbol{s}_1^{-i}\boldsymbol{a}_1^{-i}\ldots$ to denote joint state-action trajectory with agent i excluded. Note that $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^{-i}$ are both strings of vectors. We define the effective length of strings $len(\boldsymbol{\xi} = \boldsymbol{s}_0\boldsymbol{a}_0\ldots) = \min\{t+1|\boldsymbol{s}_t \in \boldsymbol{\mathcal{S}}_{\mathcal{T}}\cup\boldsymbol{\mathcal{S}}_{\mathcal{D}}\}$. Let agent i's state trajectory up to time t be $h_t^i = s_0^i s_1^i \ldots s_t^i$. We are concerned with the privacy of h_t^i so that agents can execute their policy without revealing sensitive information.

2.2 DIFFERENTIAL PRIVACY

Differential privacy is enforced by a *mechanism*, which is a randomized map. We enforce differential privacy on a per-agent basis, an approach sometimes called "local differential privacy" [Duchi et al., 2013]. For nearby local state trajectories, a mechanism must produce local private trajectories that are approximately indistinguishable. Let $D_{\boldsymbol{\pi},T} = \{(\boldsymbol{s}_0\ldots\boldsymbol{s}_T) \mid \forall t, \exists \boldsymbol{a}_t, \boldsymbol{\mathcal{T}}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})\boldsymbol{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) > 0\}$ denote a set of all feasible joint state trajectories of $\boldsymbol{\mathcal{M}}$ under a joint policy $\boldsymbol{\pi}$. The definition of "nearby" is given by an adjacency relation using the Hamming distance [Schulz and Mihov, 2003] denoted by $d(v^i, w^i)$, which is a metric that measures the minimum number of substitutions that can be applied to a local trajectory v^i of agent i to convert it to w^i.

Definition 1 (Adjacency)**.** *Fix a length $T \in \mathbb{N}^+$ and an adjacency parameter $k \in \mathbb{N}^+$. For a Markov game $\boldsymbol{\mathcal{M}}$ with state space $\boldsymbol{\mathcal{S}}$ and a joint policy $\boldsymbol{\pi}$, the adjacency relation for agent i is $\text{Adj}_{T,k}^i = \{(\boldsymbol{v}, \boldsymbol{w}) \in D_{\boldsymbol{\pi},T} \times D_{\boldsymbol{\pi},T} \mid d(v^i, w^i) \leq k,$ and $\forall j \neq i, v^j = w^j\}$.*

For agent i, two T-length joint trajectories are adjacent if the Hamming distance between the ith local trajectory is less than or equal to k and the local trajectories of the rest of the team remain the same. The adjacency relation of agent i must account for the behavior of the rest of the network: a malicious agent may infer the local trajectories of one agent using others' trajectories. We next introduce the definition of word local differential privacy.

Definition 2 (Word Local Differential Privacy)**.** *Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, an adjacency parameter $k \in \mathbb{N}^+$, a length $T \in \mathbb{N}^+$, and a privacy parameter $\epsilon > 0$. For a Markov game $\boldsymbol{\mathcal{M}}$ with state space $\boldsymbol{\mathcal{S}}$ and a joint policy $\boldsymbol{\pi}$, a mechanism $M : D_{\boldsymbol{\pi},T} \times \Omega \to \Delta((\mathcal{S}^i)^T)$ is ϵ-word local differentially private for agent i if, for all trajectories $(\boldsymbol{v}, \boldsymbol{w}) \in \text{Adj}_{T,k}^i$ and all $L \subseteq (\mathcal{S}^i)^T$, it satisfies $\mathbb{P}[M(\boldsymbol{v}) \in L] \leq e^\epsilon \mathbb{P}[M(\boldsymbol{w}) \in L]$.*

Definition 2 is an extension of Definition 2 from Chen et al. [2022] for the multiagent setting. The privacy parameter ϵ controls the strength of privacy and a smaller ϵ implies stronger privacy. In the literature, ϵ typically ranges from 0.01 to 10 [Hsu et al., 2014].

3 PROBLEM FORMULATION AND ASSUMPTIONS

In this section, we state and analyze the problem of privatizing inter-agent communications in a cooperative Markov game. We begin with the problem statements. Consider N agents playing a cooperative Markov game with a reach-avoid objective as introduced in §2.1.

Problem 1. *Design an online privacy mechanism that provides ϵ-word local differential privacy (Definition 2) for the state trajectory $h_t^i = s_1^i s_2^i \ldots s_t^i$ of agent i in real time, i.e., without knowledge of $s_{t+1}^i, s_{t+2}^i, \ldots$ at time t.*

Problem 2. *Define an algorithm for the decentralized execution of policies $\{\pi^i\}_{i=1}^N$ under private communications.*

Problem 3. *Given a collection of local policies $\{\pi^i\}_{i=1}^N$, provide a bound on the probability of success under private communications $\boldsymbol{v}^{pr}$. Use this bound to analyze the tradeoffs between privacy and performance in the multiagent system.*

Problem 4. *Synthesize policies for the multiagent system that achieve high performance under strong levels of privacy, by taking into account the tradeoffs analyzed in Problem 3.*

Privacy Assumptions: We formalize what information agents provide to the central planner and what information they hide. We then illustrate this setting with an example.

We assume that each agent trusts a central planner to design local policies. Each agent allows the planner to access its individual MDP, denoted as $\mathcal{M}^i$ for each $i \in [N]$. The planner also has knowledge of the game's objective, which can be specified as reach and avoid sets $\boldsymbol{\mathcal{S}}_{\mathcal{T}}$ and $\boldsymbol{\mathcal{S}}_{\mathcal{A}}$ or a reward function. The central planner uses this information to provide each agent with a local policy π^i. These local policies are assumed to be stationary and the action distribution of each agent is independent of the actions of its teammates given the joint state. This means that the central planner will not synthesize policies that compromise privacy: agent i does not gain knowledge of any other agent's actions by sampling its own local policy π^i. We additionally assume that the initial joint state, $\boldsymbol{s_I}$, is public information.

Furthermore, we assume that the local policies synthesized by the central planner have an *acyclic dependency structure*. Let $G = ([N], E)$ be a directed graph, where $E \subseteq [N] \times [N]$. We use G to define the dependency structure of the local policies in the sense that an edge $(i, j) \notin E$ if and only if $\pi^i(\boldsymbol{s}, a^i) = \pi^i(\boldsymbol{y}, a^i)$ for all $\boldsymbol{s} = (s^1, \ldots, s^N) \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{y} = (y^1, \ldots, y^N) \in \boldsymbol{\mathcal{S}}$ with $s^k = y^k$ for all $k \neq j$. In words, $(i, j) \notin E$ if the local policy of agent i does not depend on the state of agent j. The assumption on the acyclic dependency structure of the agents' policies can then be defined as follows.

Assumption 2. *The directed graph $G' = ([N], E')$ is an acyclic directed graph where $E' = E \setminus \{(i, i) | i \in [N]\}$.*

We also assume that the agents do not trust each other. Therefore, we assume that the agents do not have access to each other's transition probabilities or actions. The agents also do not observe whether the reach-avoid specification is satisfied or violated. Each agent *only* receives a local policy from the central planner and private state information from the agents that this policy depends on, as defined by G.

Lastly, the methods presented in this paper can be applied when each agent has a different privacy level, i.e., different values of ϵ. However, for convenience we assume that each agent has the same privacy parameter ϵ.

Example 1. *In this example, the sensitive information is the location of two rideshare drivers, Alice and Bob. A central planner generates local policies for Alice and Bob. To protect their privacy, Alice and Bob use differential privacy to communicate their locations to each other. For example, they can randomize their location data before sharing it so that their true locations are not revealed. Alice and Bob do not need to know each other's preferences or constraints, only the private state information that they communicate.*

With this private information, Alice and Bob can then execute the local policies synthesized by the central planner. However, because they are sharing perturbed location data, the local policies may not be executed as efficiently as they could be if they had access to each other's true locations.

To mitigate this loss in performance, the central planner can use the methods developed in our paper to synthesize a collection of local policies that takes into account the effects of privacy on performance. This will allow them to balance the need for privacy with the need for efficient policy execution, and ensure that the passengers are picked up as quickly as possible while preserving the drivers' privacy.

4 IMPLEMENTING LOCAL POLICIES WITH PRIVATE COMMUNICATIONS

In this section, we solve Problems 1 and 2. Specifically, in §4.1, we modify the online mechanism for Markov chains from Chen et al. [2022] to privatize state trajectories of an MDP. Then, in §4.2 we detail how each agent can use other agents' private state information to execute its local policy.

4.1 IMPLEMENTING DIFFERENTIAL PRIVACY

We enforce privacy on a per-agent basis. That is, we develop a mechanism for agent i to share its local state trajectory $h_t^i = s_1^i s_2^i \ldots s_t^i \in (\mathcal{S}^i)^t$ during policy execution while satisfying word local differential privacy from Definition 3. To achieve this, agent i will only share a private state trajectory $\tilde{h}_t^i = \tilde{s}_1^i \tilde{s}_2^i \ldots \tilde{s}_t^i \in (\mathcal{S}^i)^t$. To generate $\tilde{h}_t^i$ in real time, agent i uses an online mechanism $M_{h_t^i}$ to generate an individual private state $\tilde{s}_t^i$ at each time step t.

At every time step t, each agent needs to communicate its private state with the agents corresponding to its predecessors in the acyclic graph G, to allow them to execute their local policies. However, the differential privacy guarantee of Definition 2 holds over the entire $T-$length state trajectory. This means that even though agents are communicating at each time step, we provide privacy to their entire $T-$length trajectories. We now define the online privacy mechanism.

Definition 3 (Online Mechanism [Chen et al., 2022])**.** *Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and an MDP $\mathcal{M}^i = (\mathcal{S}^i, s_I^i, \mathcal{A}^i, \mathcal{T}^i)$. Given a state trajectory $h_t^i = s_1^i s_2^i \ldots s_t^i \in (\mathcal{S}^i)^t$, with an initial state s_I^i, define the online mechanism $M_{h_t^i}$ that generates a private trajectory $\tilde{h}_t^i = \tilde{s}_1^i \tilde{s}_2^i \ldots \tilde{s}_t^i \in (\mathcal{S}^i)^t$ such that $\tilde{s}_t^i$ is sampled from the distribution $\mathbb{P}[\tilde{s}_t^i] = \mu_\epsilon^i(\tilde{s}_t^i | s_t^i, \tilde{s}_{t-1}^i)$ where μ_ϵ^i is computed by Algorithm 1.*

Algorithm 1: Online Mechanism Construction

Input: Probability of true transition τ_ϵ
Output: μ^i_ϵ
for $(s^i_t, \tilde{s}^i_{t-1}, \tilde{s}^i_t) \in \mathcal{S}^i \times \mathcal{S}^i \times \mathcal{S}^i$ **do**
  if $s^i_t = \tilde{s}^i_t$ *and* $\beta(\tilde{s}^i_t, \tilde{s}^i_{t-1}) = 1$ **then**
    $\mu^i_\epsilon(\tilde{s}^i_t \mid s^i_t, \tilde{s}^i_{t-1}) = \tau_\epsilon(\tilde{s}^i_{t-1})$.
  else if $s^i_t \neq \tilde{s}^i_t$ *and* $\beta(\tilde{s}^i_t, \tilde{s}^i_{t-1}) = 1$ **then**
    $\mu^i_\epsilon(\tilde{s}^i_t \mid s^i_t, \tilde{s}^i_{t-1}) = \frac{1-\tau_\epsilon(\tilde{s}^i_{t-1})\beta(s^i_t, \tilde{s}^i_{t-1})}{\rho(\tilde{s}^i_{t-1}) - \beta(s^i_t, \tilde{s}^i_{t-1})}$.
  else
    $\mu^i_\epsilon(\tilde{s}^i_t \mid s^i_t, \tilde{s}^i_{t-1}) = 0$.

In Algorithm 1, the feasibility indicator function β is defined for all $s^i, y^i \in \mathcal{S}^i$ as

$$\beta(s^i, y^i) = \begin{cases} 1, \text{ if } \exists a^i \in \mathcal{A}^i \ s.t. \ \mathcal{T}^i(y^i, a^i, s^i) > 0, \\ 0, \ otherwise, \end{cases}$$

and the out-degree ρ is defined for each state $s^i \in \mathcal{S}$ as $\rho(s^i) = |\{y^i \in \mathcal{S}^i \mid \exists a^i \in \mathcal{A}^i \ s.t. \ \mathcal{T}^i(s^i, a^i, y^i) > 0\}|$.

Definition 3 and Algorithm 1 define a privacy mechanism in the form of a conditional probability distribution μ^i_ϵ. To implement the mechanism agent i samples a private output $\tilde{s}^i_t$ from the probability distribution $\mu^i_\epsilon(\cdot \mid s^i_t, \tilde{s}^i_{t-1})$ at each time step t. The mechanism is constructed such that the probability $\mu^i_\epsilon(\tilde{s}^i_t \mid s^i_t, \tilde{s}^i_{t-1})$ is positive if $\tilde{s}^i_t$ is feasible from the most recent private state $\tilde{s}^i_{t-1}$, and 0 otherwise. This prevents the mechanism from outputting private trajectories that are not feasible with respect to the dynamics of $\mathcal{M}^i$. When the true, sensitive state s^i_t is feasible from the previous private output $\tilde{s}^i_{t-1}$, the mechanism outputs s^i_t with probability $\tau_\epsilon(\tilde{s}^i_{t-1})$ and outputs any other feasible state with a uniform probability whose sum is equal to $1 - \tau_\epsilon(\tilde{s}^i_{t-1})$. We refer to the event of outputting the sensitive state s^i_t at time t as a "true transition" and $\tau_\epsilon(\tilde{s}^i_{t-1})$ as the "probability of true transition". In §5, we establish a requirement for this mechanism to achieve word local differential privacy.

4.2 PRIVATE POLICY EXECUTION

In this section, we solve Problem 2 and define an algorithm for the decentralized execution of local policies $\{\pi^i\}_{i=1}^N$ under private communications (Algorithm 2).

Recall that we assume the agents use local policies with an acyclic dependency structure defined by the directed graph $G = ([N], E)$. Let $Pred(i)$ and $Succ(i)$ denote the set of predecessors and succesors of node i in G, respectively. Let $\tilde{\boldsymbol{s}}^{Succ}_{t,i} = (\tilde{s}^j_t)$ for all $j \in Succ(i)$ be the tuple containing the private states of agent i's successors. Note that formally, π^i is defined on the joint state space of the entire team. However, Assumption 2 ensures that the action distribution defined by π^i only depends on $(\tilde{\boldsymbol{s}}^{Succ}_{t,i}, s^i_t)$, the local states of some subset of the agents. To more readily highlight the information dependencies in the problem, it is with a slight abuse of notation that we use $(\tilde{\boldsymbol{s}}^{Succ}_{t,i}, s^i_t)$ to denote the inputs provided to each local policy π^i in Algorithm 2.

Algorithm 2: Privatized Policy Execution

Input for every agent i: Local policy π^i
Set $\tilde{s}^i_0 = s^i_I$ for all $i \in [N]$.
for $t = 0, 1, \ldots$ **every agent i does in parallel**
  Set $\hat{\boldsymbol{s}}_{t,i} = (\tilde{\boldsymbol{s}}^{Succ}_{t,i}, s^i_t)$.
  Sample an action $a^i_t \sim \pi^i(\hat{\boldsymbol{s}}_{t,i})$.
  Execute a^i_t and transition to $s^i_{t+1} \sim \mathcal{T}^i(s^i_t, a^i_t)$.
  Share $\tilde{s}^i_{t+1} \sim \mu^i_\epsilon(\cdot | s^i_{t+1}, \tilde{s}^i_t)$ with agents in $Pred(i)$.

During private policy execution, the agents communicate potentially false information. To sample actions from their local policies, each agent must therefore maintain an estimate of the states of its successors in G. In this work, we assume that each agent constructs this estimate using the privatized information it receives from its teammates. In detail, agent i knows its own local state s^i_t and the private state of its successors, $\tilde{s}^j_t$ for $j \in Succ(i)$ at time t. Agent i's estimate of the relevant teammate states is thus given by $\hat{\boldsymbol{s}}_{t,i} = (\tilde{\boldsymbol{s}}^{Succ}_{t,i}, s^i_t)$. Then, agent i samples an action a^i_t for itself from π^i using the estimate $\hat{s}_{t,i}$. We note that the agents do not communicate during the action selection phase since the local policies are independent given the joint state. After choosing an action a^i_t, agent i executes this action and transitions to a next state s^i_{t+1}. In the next time step $t+1$, agent i samples a private state $\tilde{s}^i_t$ using μ^i_ϵ and shares this private state with the agents corresponding to its predecessors in G.

5 PRIVACY AND PERFORMANCE TRADEOFFS

In this section, we address Problem 3 and analyze the tradeoff between performance and privacy when executing a collection of local policies with private communications.

In the single-agent setting, Chen et al. [2022] showed word differential privacy of the agents' state trajectories generated by the online mechanism. We extend this result to the multiagent setting using Assumption 2, which ensures that at time t the future true state trajectory $s^i_{t+1} \ldots$ of agent i is statistically independent from its past private state trajectory $\tilde{h}^i_t = \tilde{s}^i_1 \tilde{s}^i_2 \ldots \tilde{s}^i_t$ given $h^i_t = s^i_1 s^i_2 \ldots s^i_t$. We discuss the necessity of this independence in the proof of Theorem 1, which is provided in the supplementary material.

Theorem 1. *Fix a length $T \in \mathbb{N}^+$, an adjacency parameter $k \in \mathbb{N}^+$, and a privacy parameter $\epsilon \geq 0$. Under Assumption 2, the online mechanism (Definition 3) is ϵ-word locally differentially private (Definition 2) with respect to the Adjacency relation $\mathrm{Adj}^i_{T,k}$ in Definition 1 if $\tau_\epsilon(\tilde{s}^i_{t-1})$ satisfies*

$$\tau_\epsilon(\tilde{s}^i_{t-1}) = 1/((\rho(\tilde{s}^i_{t-1}) - 1)e^{-\epsilon/k+1}).$$

Having established the differential privacy guarantees of Algorithm 2, we now focus on performance guarantees. In order to succeed under private communications, the agents' local policies should be as indifferent as possible to the other agents' states. In other words, agents' behaviors should be made nearly independent from each other.

The collection of local policies induce a joint policy $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$. To measure the dependencies between the agents, we use a quantity called the "total correlation" of the joint policy [Karabag et al., 2022]. Let $\boldsymbol{S}_t$ be a random variable denoting the joint state of the agents at time t under the joint policy $\boldsymbol{\pi}$ with no privatization, $\boldsymbol{A}_t$ be a random variable denoting the joint action of the agents at time t, S_t^i be a random variable denoting the state of Agent i at time t, and let A_t^i be a random variable denoting the action of Agent i at time t. The total correlation $C_{\boldsymbol{\pi}}$ of a joint policy $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$ is

$$C_{\boldsymbol{\pi}} = \Sigma_{i=1}^N H(S_0^i A_0^i \dots S_\eta^i) - H(\boldsymbol{S}_0 \boldsymbol{A}_0 \dots \boldsymbol{S}_\eta) \quad (1)$$

where $H(Y) := -\sum_{y \in \mathcal{Y}} \Pr(Y = y) \log(\Pr(Y = y))$ is the entropy of a discrete random variable Y with support $\mathcal{Y}$, and η denotes the random hitting time to $\boldsymbol{\mathcal{S}}_{\mathcal{T}} \cup \boldsymbol{\mathcal{S}}_{\mathcal{D}}$, i.e., the effective end of the trajectory in terms of the reach-avoid specification [Karabag et al., 2022].

The following result relates the success probability under private communications to the success probability under truthful communications (i.e., without privacy). The proof of the theorem is included in the supplementary material.

Theorem 2. *Fix a privacy parameter $\epsilon > 0$ and adjacency parameter k. Given N agents implementing local policies $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$ with private communications according to Algorithm 2, let $\boldsymbol{v}^{pr}$ be the success probability under private communications and let $\boldsymbol{v}^{tr}$ be the success probability under truthful communications (no privacy). Then,*

$$\boldsymbol{v}^{pr} \geq \boldsymbol{v}^{tr} - \sqrt{1 - e^{-C_{\boldsymbol{\pi}}} \left((\rho_m - 1)e^{-\epsilon/k} + 1\right)^{N l^{tr}}}, \quad (2)$$

where $C_{\boldsymbol{\pi}}$ is defined in (1)*, $\rho_m = \max_{i \in [N], s^i \in \mathcal{S}^i} \rho(s^i)$ is the max out-degree, $l^{tr} = \mathbb{E}_{\boldsymbol{\xi} \sim \boldsymbol{\Gamma}^{tr}}[len(\boldsymbol{\xi})]$ is the expected joint trajectory length when $\boldsymbol{\pi}$ is executed with no privacy, and Γ^{tr} is the probability distribution over joint trajectories induced by the joint policy executed with no privacy.*

The term $\left((\rho_m - 1)e^{-\epsilon/k} + 1\right)^{N l^{tr}}$ in Theorem 2 represents the probability of events where the private state trajectories are the same as the true state trajectories. The term $e^{-C_{\boldsymbol{\pi}}}$ in Theorem 2 is a proxy to account for the events where the private state trajectories are different from the true state trajectories. In these events, the agents can still succeed if the local policies are independent of the other agents' states. A lower total correlation implies lower dependencies between the agents, and that the agents are more likely to succeed. We note that the equality holds in (2) when agents communicate truthfully, i.e., $\epsilon = \infty$, and each agent acts totally independently from other agents, i.e., $C_{\boldsymbol{\pi}} = 0$.

6 POLICY SYNTHESIS

In this section, we present an algorithm for the synthesis of a collection of local policies $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^N$ that remains performant, even under private communications.

We aim to maximize the reach-avoid probability under private communications by minimizing the lower bound on $\boldsymbol{v}^{pr}$ given in Theorem 2. Since the bound is complex in nature and it is a monotone function of its variables, we instead solve the following optimization problem involving constants $\delta > 0$ and $\beta > 0$:

$$\sup_{\boldsymbol{\pi}} \boldsymbol{v}^{tr} - \delta l^{tr} - \beta C_{\boldsymbol{\pi}}. \quad (3)$$

Using the stationarity of $\boldsymbol{\pi}$, the optimization problem given in (3) can be represented with occupancy measure variables $x_{\boldsymbol{s},\boldsymbol{a}}$ of the joint state-action space [Karabag et al., 2022]. We refer interested readers to Karabag et al. [2022] for the details of this optimization problem. The objective function of (3) contains convex and concave functions of the occupancy measure variables that can be solved using the concave-convex procedure [Lanckriet and Sriperumbudur, 2009, Yuille and Rangarajan, 2001] for a local optimum.

After solving for the optimal $x^*_{\boldsymbol{s},\boldsymbol{a}}$ of the occupancy measure variables, we compute the local policies. Recall that in Assumption 2, we assumed the policy dependencies between the agents are acyclic. In order to compute local policies $\pi^i(\boldsymbol{s}, a^i)$ that satisfy Assumption 2, we marginalize the joint occupancy measure using a desired $G = ([N], E)$ dependency graph. Formally, for every $\boldsymbol{s} = (s^1, \dots, s^N) \in \boldsymbol{\mathcal{S}}$,

$$\pi^i(\boldsymbol{s}, a^i) = \frac{\sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}_{\boldsymbol{s}}^i} \sum_{\boldsymbol{a}^{-1} \in \boldsymbol{\mathcal{A}}^{-1}} x^*_{\boldsymbol{y},(a^i,\boldsymbol{a}^{-i})}}{\sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}_{\boldsymbol{s}}^i} \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} x^*_{\boldsymbol{y},\boldsymbol{a}}}$$

where $\boldsymbol{\mathcal{Y}}_{\boldsymbol{s}}^i = \{(y^1, \dots, y^N) \in \boldsymbol{\mathcal{S}} \mid \forall (i,j) \in E, y^j = s^j\}$. We note that, instead of postprocessing joint the occupancy variables, we can alternatively enforce this assumption by including additional bilinear equality constraints in the policy synthesis optimization problem.

7 NUMERICAL EXPERIMENTS

Numerical experiments demonstrate the robustness to private communication enjoyed by the policies synthesized using the procedure described in §6. In each experiment, we solve (3) to synthesize minimum-dependency local policies $\{\pi_{MD}^i\}_{i=1}^N$ for the agents in the team. We use $\boldsymbol{\pi}_{MD}$ to denote the joint policy that results from the concurrent execution of these local policies, described in §4.2.

We compare the performance of $\boldsymbol{\pi}_{MD}$ to that of a collection of baseline local policies $\{\pi_{base}^i\}_{i=1}^N$, which are synthesized by optimizing the team's performance without taking the total correlation value into account. That is, by solving (3) with δ and β set to zero, and subsequently marginalizing

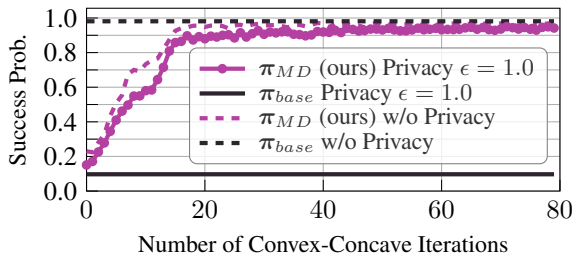

Figure 1: Probability of task success as a function of the number of iterations of the policy synthesis procedure for the two-agent navigation experiment.

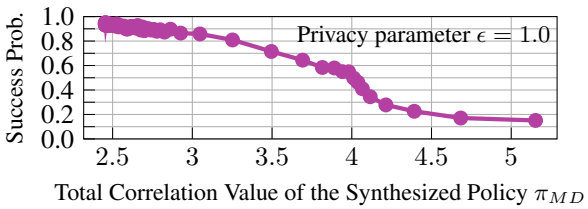

Figure 2: Probability of team success under private communications as a function of the total correlation of the synthesized policies.

the policies to satisfy Assumption 2 (as described at the end of §6). We use $\boldsymbol{\pi}_{base}$ to refer to the joint policy resulting from the concurrent execution of $\{\pi^i_{base}\}_{i=1}^N$. Project code is included in the supplementary material.

7.1 TWO-AGENT NAVIGATION EXAMPLE

We begin by considering the multiagent navigation example introduced by Karabag et al. [2022]. Two agents operate in a common environment, and each must navigate to a target location while avoiding collisions with its teammate. To reach their target locations, the agents must navigate past each other by passing through one of two narrow corridors. The act of jointly navigating the corridors without colliding necessitates coordination between the agents.

The environment is implemented as a grid of cells, each of which corresponds to an individual local state. At any given timestep, each agent can choose to move in any direction, or remain in place. Each agent slips with probability 0.05 when it takes an action, resulting in the agent moving to one of its neighboring states instead of its intended target state.

While synthesizing $\boldsymbol{\pi}_{MD}$, we set $\delta = 0.01$ and $\beta = 0.4$ in (3). These values were selected to strike a balance between the optimization objective's competing terms. We fix an adjacency parameter of $k = 3$ while constructing the privacy mechanisms used in all experiments. To define the acyclic dependencies between the agents' local policies, we use the directed graph G with nodes $\{1, 2\}$ and with an edge $E = \{(1, 2)\}$—the first agent's local policy depends on the local state of the second agent, but not vice versa.

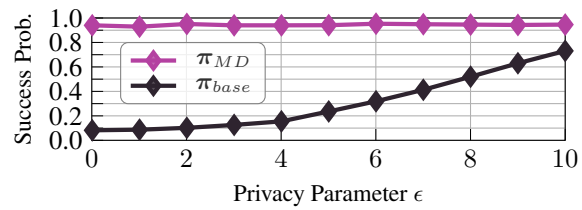

Figure 3: Probability of team success under a variety of levels of privacy. Smaller values of the privacy parameter ϵ correspond to a stronger level of privacy.

Minimum-dependency policies are 84% more performant than the baseline under private communications. Figure 1 illustrates the probability of success of $\boldsymbol{\pi}_{MD}$ throughout policy synthesis. We plot the success probability resulting from both private ($\epsilon = 1.0$) and truthful implementations of communication. For comparison, we also plot the results of $\boldsymbol{\pi}_{base}$. We estimate the plotted probability values by simulating $1,000$ rollouts of the policies at each iteration, and computing the empirical rate at which the team reaches its target set. While the baseline policy achieves a success probability of 0.98 under non-private communication, its success probability drops to 0.10 when communications are privatized. By contrast, even under private communication, $\boldsymbol{\pi}_{MD}$ enjoys a probability of success of 0.94.

Lower total correlation values result in higher success probabilities under private communications. Figure 2 illustrates the team's success probability and the total correlation of each of the joint policies obtained throughout policy synthesis. As the total correlation of $\boldsymbol{\pi}_{MD}$ decreases during policy synthesis, the policy's performance under private communications significantly increases. This result provides a strong empirical justification for the use of total correlation as a regularizer during policy synthesis.

The performance of $\boldsymbol{\pi}_{MD}$ is robust to the level of privacy enforced by the differential privacy mechanism. Recall that the parameter ϵ controls the strength of privacy enforced by the differential privacy mechanism. Lower values of ϵ correspond to stronger levels of privacy—the mechanism is more likely to perturb the state trajectories of the agents. In Figure 3 we observe that the performance of $\boldsymbol{\pi}_{MD}$ remains consistently high, regardless of the value of ϵ. By contrast, the performance of $\boldsymbol{\pi}_{base}$ is highly sensitive to ϵ; it decreases significantly for moderate to strong levels of privacy.

7.2 FOUR-AGENT SYSADMIN EXAMPLE

We now consider a variant of the multiagent system administration example from Guestrin et al. [2003], Choudhury et al. [2021]. A collection of servers must coordinate to provide a consistent level of service, while simultaneously performing necessary maintenance. Each server is modeled as an individual agent with four local states: nominal $s^i = 1$, in need

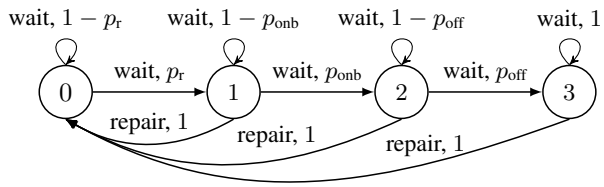

Figure 4: Local transition dynamics of the SysAdmin example. A label (a, p) refers to a transition happening with probability p under action a.

of repairs $s^i = 2$, in repair $s^i = 0$, and offline $s^i = 3$. At any timestep, each agent may choose to continue operation, or to initiate a repair. We assume the local transition dynamics of the agents, illustrated in Figure 4, to be independent.

The team's task is to reach a target joint state in which all of the servers are operating nominally. However, we impose the additional constraints that, at any given time during operation, the team is allowed at most two offline servers and at most two servers in the repair state. If either of these constraints are violated, the team fails the task.

In this example, we set $p_r = 0.9$, $p_{onb} = 0.1$, and $p_{off} = 0.1$, we set the values of the policy synthesis coefficients δ and β to 0.001 and 0.1 respectively, and we use an adjacency parameter of $k = 1$ in the differential privacy mechanism. The acyclic graph G is defined such that the local policy of the first agent depends only on its own local state, the policy of the second agent depends on the its own local states as well as those of the first agent, the third agent depends on the local states of the first three agents, and so on.

π_{MD} consistently outperforms π_{base} under a variety of initial system configurations and privacy levels. Figure 5 compares the probability of success achieved by the proposed minimum-dependency policy π_{MD}, to that achieved by the baseline π_{base}. In this example, under truthful communication, it is possible for the team to achieve a success probability of 1.0 from any initial configuration. However, when communication is private, π_{MD} consistently outperforms π_{base}. In the considered initial configurations, even under the strongest level of privacy, π_{MD} achieves a probability of success of above 97 percent. By contrast, we observe that under private communications π_{base} typically achieves success probabilities of less than 50 percent.

7.3 ADDITIONAL DISCUSSION

In addition to the differences between the values of the team's probability of success under π_{MD} and π_{base}, we also observe a significant change in the expected length of the trajectories that result from these policies. Under truthful communication in the SysAdmin experiments, the expected length of the trajectories induced by π_{base} range from 30 to 40 timesteps, depending on the initial configuration of the

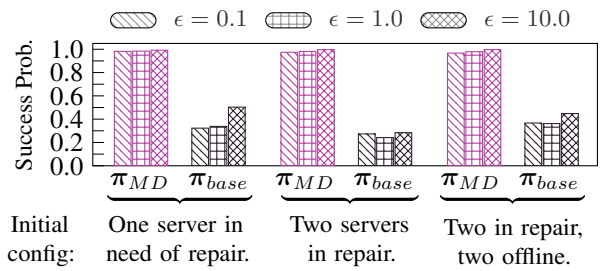

Figure 5: Probability of success in the SysAdmin example under a variety of initial configurations and privacy levels.

system. For π_{MD} these values range from 3 to 6 timesteps.

This observation yields insight into the different qualitative properties of the policies. π_{base} induces conservative behavior that maximizes the team's probability of success (under truthful communication) by requiring the agents to wait for specific joint states before taking certain actions. The actions of each agent are highly dependent on the exact states of its teammates. On the other hand, π_{MD} achieves nearly the same probability of success as π_{base}, but the agents act quickly and accept a small level of risk in order to reduce the dependencies of their actions on their teammates' states.

The inclusion of the total correlation as a regularization term prevents the policy synthesis procedure from making the agents highly interdependent in order to achieve marginally higher performance. This tradeoff becomes relevant when the inter-agent communications are imperfect, which is the case in privatized multiagent systems.

Finally, we remark that in some settings there may not exist a collection of highly independent policies that also achieve a high performance. In such cases, we may not observe a large of a gap in performance between π_{MD} and π_{base} under private communication. However, even in these settings, the total correlation may act as an indicator that it is infeasible to achieve strong performance and privacy simultaneously.

8 CONCLUSIONS

This paper presents a framework to privatize inter-agent communications in cooperative multiagent decision-making problems. Specifically, we adopt a differential privacy mechanism to protect the symbolic state trajectories of agents. We provide theoretical results to analyze the tradeoff between the strength of privacy and the team's performance. We synthesize robust policies for agents by reducing the total correlation among them. Numerical results demonstrate that the minimum-dependency policies achieve high performance under strong levels of privacy, whereas the team performance of baseline policies that ignore total correlation decreases dramatically under private communications.

References

Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V Goldman. Transition-independent decentralized Markov decision processes. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, pages 41–48, 2003.

Ronen Israel Brafman. A privacy preserving algorithm for multi-agent planning and search. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2013. doi: 10.1109/TII.2012.2219061.

Bo Chen, Kevin Leahy, Austin Jones, and Matthew Hale. Differential privacy for symbolic systems with application to markov chains. *arXiv preprint arXiv:2202.03325*, 2022.

Shushman Choudhury, Jayesh K Gupta, Peter Morales, and Mykel J Kochenderfer. Scalable anytime planning for multi-agent mdps. *arXiv preprint arXiv:2101.04788*, 2021.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, 2013. doi: 10.1109/FOCS.2013.53.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differential privacy for regret minimization in reinforcement learning. *arXiv preprint arXiv:2010.07778*, 2020.

Parham Gohari, Bo Wu, Calvin Hawkins, Matthew Hale, and Ufuk Topcu. Differential privacy on the unit simplex via the dirichlet mechanism. *IEEE Transactions on Information Forensics and Security*, 16:2326–2340, 2021.

Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19: 399–468, 2003.

Tommy Hefner, Guy Shani, and Roni Stern. Privacy preserving planning in multi-agent stochastic environments. *Autonomous Agents and Multi-Agent Systems*, 36(1):1–27, 2022.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C. Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410, 2014.

Mustafa O. Karabag, Cyrus Neary, and Ufuk Topcu. Planning not to talk: Multiagent systems that are robust to communication loss. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 705–713, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Gert Lanckriet and Bharath K. Sriperumbudur. On the convergence of the concave-convex procedure. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/8b5040a8a5baf3e0e67386c2e3a9b903-Paper.pdf.

Raz Nissim and Ronen Brafman. Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research*, 51:293–332, 2014.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, Cham, 2016.

Lynne E. Parker, Daniela Rus, and Gaurav S. Sukhatme. *Multiple Mobile Robot Systems*, pages 1335–1384. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_53. URL https://doi.org/10.1007/978-3-319-32552-1_53.

Dan Qiao and Yu-Xiang Wang. Offline reinforcement learning with differential privacy. *arXiv preprint arXiv:2206.00810*, 2022.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4295–4304, 2018.

Klaus U. Schulz and Stoyan Mihov. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition (IJDAR)*, 5(1):67–85, 2003. doi: 10.1007/s10032-002-0082-8.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5887–5896, 2019.

Jose M. Such, Agustín Espinosa, and Ana García-Fornes. A survey of privacy in multi-agent systems. *The Knowledge Engineering Review*, 29(3):314–344, 2014. doi: 10.1017/S0269888913000180.

Dayong Ye, Tianqing Zhu, Sheng Shen, Wanlei Zhou, and Philip S. Yu. Differentially private multi-agent planning for logistic-like problems. *IEEE Transactions on Dependable and Secure Computing*, 19(2):1212–1226, 2022. doi: 10.1109/TDSC.2020.3017497.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/file/a012869311d64a44b5a0d567cd20de04-Paper.pdf.