Differentially Private Multi-Agent Planning for Logistic-like Problems

Dayong Ye, Tianqing Zhu*, Sheng Shen, Wanlei Zhou and Philip S. Yu

Abstract— Planning is one of the main approaches used to improve agents' working efficiency by making plans beforehand. However, during planning, agents face the risk of having their private information leaked. This paper proposes a novel strong privacy-preserving planning approach for logistic-like problems. This approach outperforms existing approaches by addressing two challenges: 1) simultaneously achieving strong privacy, completeness and efficiency, and 2) addressing communication constraints. These two challenges are prevalent in many real-world applications including logistics in military environments and packet routing in networks. To tackle these two challenges, our approach adopts the differential privacy technique, which can both guarantee strong privacy and control communication overhead. To the best of our knowledge, this paper is the first to apply differential privacy to the field of multi-agent planning as a means of preserving the privacy of agents for logistic-like problems. We theoretically prove the strong privacy and completeness of our approach and empirically demonstrate its efficiency. We also theoretically analyze the communication overhead of our approach and illustrate how differential privacy can be used to control it.

Index Terms—Multi-Agent Planning, Privacy Preservation, Differential Privacy

1 Introduction

Multi-agent planning is one of the fundamental research problems in multi-agent systems [1], [2]. Multi-agent planning research aims to improve agents' working efficiency by making plans in advance. Research into collaborative multi-agent planning largely focuses on jointly automated planning [3]. During jointly automated planning, agents have to share information. However, this kind of information sharing often results in the leaking of agents' private information. Accordingly, to protect agents' privacy, privacy preservation is introduced into the collaborative multi-agent planning process [4], [5]. The main problem associated with privacy preservation in collaborative multi-agent planning is that of how to make plans for agents while also preserving the privacy of each agent.

Privacy can be roughly classified into four levels: weak privacy, strong privacy, object cardinality privacy, and agent privacy [3]. Strong privacy means that an agent, regardless of its reasoning power, cannot deduce the private information of other agents based on the information available to it. Developing a planning method with strong privacy in distributed and communication-constrained environments is challenging for the following two reasons. First, it is difficult to achieve strong privacy, completeness and efficiency simultaneously [6]. Second, in communication-constrained environments, each agent is allowed to communicate only a limited number of times.

*Tianqing Zhu is the corresponding author. D. Ye, T. Zhu, S. Shen and W. Zhou are with the Centre for Cyber Security and Privacy and the School of Computer Science, University of Technology, Sydney, Australia. Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, USA. Email: {Dayong.Ye, Tianqing.Zhu, Sheng.shen-1, Wanlei.Zhou}@uts.edu.au, psyu@cs.uic.edu. This work is supported by two ARC Projects (LP170100123 and DP190100981) from the Australian Research Council, Australia, and also by NSF under grants III-1526499, III-1763325, III-1909323, and SaTC-1930941, USA.

These two challenges are widespread in many real-world applications. A typical application is military logistics. In military logistics, it is vital that each military unit should strongly protect its private and sensitive facts. Also, plans for military units must be complete and efficient to avoid any delay. In addition, communication between units has to be constrained, since the more communication takes place, the more likely it will be that sensitive information is leaked.

Most existing planning approaches are either weak privacy-preserving or overlook the issue of privacy preservation entirely [3]. Very few approaches are strong privacy-preserving [7]. These strong privacy-preserving planning approaches, however, may not achieve strong privacy, completeness and efficiency simultaneously, as summarized in [6]. Moreover, these approaches also may not work efficiently in distributed and communication-constrained environments, as they implicitly assume that an agent can communicate directly with all other agents, and overlook the analysis of communication overhead.

Accordingly, in this paper, we develop a novel strong privacy-preserving planning approach for distributed and communication-constrained environments. Our approach focuses primarily on logistic-like problems, which are typically used as running examples in multi-agent planning. To achieve strong privacy, completeness and efficiency simultaneously, we adopt the differential privacy technique. Differential privacy is a promising privacy model, which has been mathematically proven that when this model is in use, an individual record being stored in or removed from a dataset makes little difference to the analytical output of the dataset [8], [9]. To the best of our knowledge, we are the first to apply differential privacy to the privacy-preserving planning problem. Using a differential privacy mechanism to obfuscate an agent's private information can strongly preserve the agent's privacy while also having minimal impact on the usability of the agent's private information.

Furthermore, we also address the communication-constrained environment issue by adopting the concept of a 'privacy budget'. In differential privacy, a privacy budget is applied to control privacy levels. In our proposed approach, the privacy budget can naturally be used to control communication overhead, with the result that only a limited number of messages are permitted during a planning phase.

In summary, the contributions of this paper are two-fold:

1) Improving upon existing strong privacy-preserving planning approaches, our approach can achieve strong privacy, completeness and efficiency simultaneously in logistic-like problems using the differential privacy technique.
2) Our approach is more applicable to distributed and communication-constrained logistic-like problems than existing approaches.

The remainder of this paper is organized as follows. In the next section, a detailed review of related work is presented. Then, a motivating example is given in Section 3. Preliminaries are presented in Section 4. After that, the novel planning approach and the theoretical analysis are presented in Sections 5 and 6, respectively. The application of our approach to other domains is illustrated in Section 7. Next, the experimental results are provided in Section 8. Finally, Section 9 concludes this paper.

2 Related Work

2.1 Weak privacy-preserving approaches

Torreno et al. [10] develop a framework known as FMAP (forward multi-agent planning). In FMAP, agents maintain a common open list with unexplored refinement plans. Agents then jointly select an unexplored refinement plan. Each agent then expands the plan using a forward-chaining procedure. Agents exchange these plans and use a distributed heuristic approach to evaluate them. Later, based on the FMAP framework, Torreno et al. [11] develop a set of global heuristic functions: DTG (domain transition graphs) heuristic and landmarks heuristic, in order to improve the efficiency of the FMAP framework.

Stolba and Komenda [12] present a multi-agent distributed and local asynchronous (MADLA) planner. This planner adopts a distributed state-space forward-chaining multi-heuristic search. The multi-heuristic search takes the advantages of both local and distributed heuristic searches by combining them together. As a result, the combination of the two heuristics outperforms the two heuristics separately.

Maliah et al. [13] propose a greedy privacy-preserving planner (GPPP). In GPPP, agents collaboratively generate an abstract global plan based on two privacy-preserving heuristics: landmark-based heuristic and privacy-preserving pattern database heuristic. Each agent generates a local plan by extending the global plan.

2.2 Strong privacy-preserving approaches

Brafman [7] is the first to theoretically prove strong privacy in multi-agent planning. He proposes an approach referred to as Secure-MAFS (secure multi-agent forward search). Secure-MAFS extends the MAFS approach [14] by reducing the amount of information exchanged between agents. In Secure-MAFS, agents protect their privacy by opting not to communicate a given two states to others if these two states differ only in their private elements. This is because other agents could possibly deduce private information through the non-private or public part of the states.

Tozicka et al. [6] investigate the limits of strong privacy-preserving planning. They formulate three aspects of strong privacy-preserving planning: privacy, completeness, and efficiency. They theoretically find that these three aspects are difficult to achieve at the same time for a wide class of planning algorithms. Also, they develop a strong privacy-preserving planner that embodies a family of planning algorithms. The planner is based on private set intersection, which has been proven to be computationally secure.

Stolba et al. [15], [16], [17] refine privacy metrics by quantifying the amount of privacy loss. In this case, their analysis of privacy loss is conducted by assessing information leakage [18], [19]. The amount of information leakage is measured as the difference between initial uncertainty and remaining uncertainty. They also develop a general approach to compute the privacy loss of search-based multi-agent planners. This computation is based on search tree reconstruction and classification of leaked information pertaining to the applicability of actions.

2.3 Other privacy-preserving approaches

Some other existing works seem to be related to ours, such as differentially private networks [20] and privacy-preserving distributed constraint optimization [21]. However, the research aims of these works differ from ours.

The research of differentially private networks mainly aims at hiding specific information contained in a network, which may be disclosed by answering queries regarding that network. By contrast, multi-agent privacy-preserving planning aims at collaboratively making plans without revealing the private facts of each participating agent. In [22], Kasiviswanathan et al. develop a set of node-differentially private algorithms to engage in the private analysis of network data. The key concept here is to obfuscate the input graph onto the set of graphs with maximum degree below a certain threshold. Blocki et al. [23] improve accuracy in differentially private data analysis by introducing the notion of restricted sensitivity in order to reduce noise. Restricted sensitivity represents the sensitivity of a query only over a specific subset of all possible networks. Proserpio et al. [24] propose a platform for differentially private data analysis: wPINQ (weighted Privacy Integrated Query). wPINQ treats edges as a weighted dataset on which it performs ϵ-differentially private computations, such as manipulation of records and their weights. Thus, the presence or absence of individual edges can be masked. Fioretto et al. [20] design a privacy-preserving obfuscation mechanism for critical infrastructure networks. Their mechanism consists of three phases: 1) obfuscating the locations of nodes using the exponential mechanism, 2) obfuscating the values of nodes using the Laplace mechanism, and 3) redistributing the noise introduced in the previous two phases using a bi-level optimization problem. These works assume the existence of adversaries while in multi-agent planning, agents are typically assumed to be honest but curious.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3017497, IEEE Transactions on Dependable and Secure Computing

Research into privacy-preserving distributed constraint optimization aims at securely coordinating the value assignment for the variables under a set of constraints in order to optimize a global objective function [25]. By contrast, multi-agent privacy-preserving planning aims at securely making plans that enable individual agents to achieve their goals. Grinshpoun and Tassa [26] devise a novel distributed constraint optimization problem (DCOP) algorithm that preserves constraint privacy. In their problem, a group of agents needs to compare the sum of private inputs possessed by those agents against an upper bound held by another agent. During this comparison, none of these agents learns information on either the sum or the private inputs of other agents. Their algorithm accomplishes this through the use of a secure summation protocol and a secure comparison protocol. Tassa et al. [27] propose a DCOP algorithm that is immune to collusion and offers constraint, topology and decision privacy. To achieve this goal, they adopt a secure multi-party computation protocol [28] which is capable of securely comparing the cost of the current full assignment and the upper bound and guaranteeing the security of collusion of up to half of the total agents. From an examination of the two above-mentioned works, it can be seen that the privacy-preserving DCOP mainly focuses on securely comparing the values of variables against an upper bound, while multi-agent privacy-preserving planning mainly focuses on the secure computation of each individual agent.

3 A Motivating Example

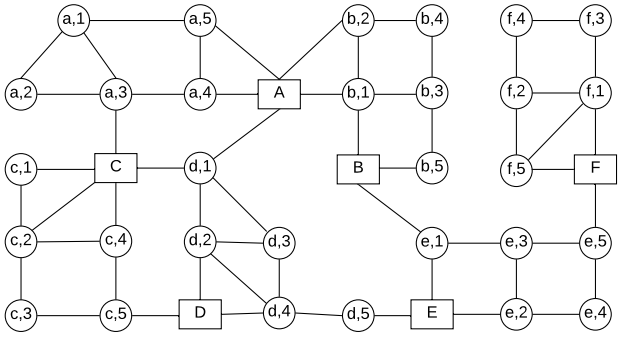


Fig. 1. An example of a logistic map

Fig. 1 presents a military logistic map. In this map, a circle denotes a military base while a rectangle denotes a logistic center. The lines connecting the bases and logistic centers are routes. Each route has a length, which is not indicated on the map in the interests of clarity. Each letter in a circle indicates a military unit's name, while each number in a circle is the index of a base in the military unit's local area. For example, '$(a, 3)$' denotes the third base in military unit a's local area. Six military units are included on this map: a, b, c, d, e, and f. Each unit exclusively operates in a local area of the map.

Information about a local area is private to the corresponding military unit. This information includes 1) the number of military bases in this local area, 2) the number of routes in this local area, 3) the length of these routes in this local area, and 4) the positions of packages in this local area. However, information regarding whether a given package is or is not located in a particular logistic center is public. For example, in Fig. 2, we extract military unit a's local area from Fig. 1. In Fig. 2, there are five bases: $(a, 1)$, $(a, 2)$, $(a, 3)$, $(a, 4)$ and $(a, 5)$. The number of these bases and routes is private to military unit a. Moreover, the length of these routes is also private to unit a. As noted above, the information that a package is located in logistic center A is public and known to all military units.

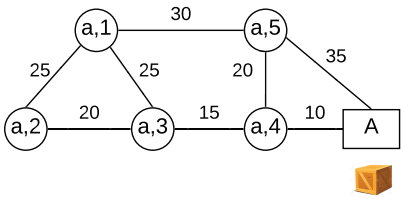


Fig. 2. Unit a's local area

The problem in this example is as follows: how should a plan be made for a military unit to transport a package from one base to another, while strongly preserving each military unit's privacy? For example, unit a wants to transport a package from $(a, 2)$ to $(f, 4)$, but $(f, 4)$ is located in military unit f's local area. Thus, multiple units must collaborate to make a plan to deliver the package, while each unit's privacy is required to be strongly preserved during this process. This problem therefore includes the above-mentioned two challenges. First, planning for military units is highly expected to achieve strong privacy, completeness and efficiency simultaneously, especially when military units are involved in a war. Second, the communication of each military unit may be constrained, as increased level of communication may result in a higher chance of private information being leaked [29].

As the above two challenges have not been adequately addressed by existing approaches, these approaches may not be suitable for this environment. Accordingly, in this paper, a novel strong privacy-preserving planning approach is proposed that takes these two challenges into account.

4 Preliminaries

4.1 The planning model

We propose a multi-agent planning model, Graph-STRIPS, which is based on a widely used privacy-aware planning model, MA-STRIPS [30]. Graph-STRIPS is defined by a 12-tuple: $\langle \mathcal{AG}, \mathcal{V}, \{\mathcal{V}_i\}_{i=1}^m, \mathcal{V}_{Pub}, \mathcal{E}, \{\mathcal{E}_i\}_{i=1}^m, \mathcal{P}, \{\mathcal{P}_i\}_{i=1}^m, \mathcal{A}, \{\mathcal{A}_i\}_{i=1}^m, \mathcal{I}, \mathcal{G}\rangle$:

- $\mathcal{AG}$ is a set of agents in the environment;
- $\mathcal{V}$ is a set of nodes (e.g., physical entities) in the environment;
- $\mathcal{V}_i$ is the set of nodes private to agent i;
- $\mathcal{V}_{Pub}$ is the set of public nodes in the environment, $\mathcal{V}_{Pub} = \mathcal{V} - \cup_{l=1}^{|\mathcal{AG}|}\mathcal{V}_l$;
- $\mathcal{E}$ is a set of edges (e.g., the relationships between physical entities) in the environment;
- $\mathcal{E}_i$ is the set of edges private to agent i;
- $\mathcal{P}$ is a set of possible facts about the environment;
- $\mathcal{P}_i$ is the set of private facts of agent i;

- $\mathcal{A}$ is a set of possible actions of all the agents;
- $\mathcal{A}_i$ is the set of private actions of agent i;
- m is the number of agents in the environment;
- $\mathcal{I}$ is the initial state of the environment;
- $\mathcal{G}$ is the goal state.

For example, in Fig. 1, each military unit is modelled as an agent. In this case, we have the following:

- $\mathcal{AG} = \{a, b, c, d, e, f\}$ and $m = 6$;
- $\mathcal{V}$ is the set of military bases and logistic centers;
- $\mathcal{V}_i$ denotes the set of bases in the local area of agent i; for example, in agent a's local area, $\mathcal{V}_a = \{(a,1),(a,2),(a,3),(a,4),(a,5)\}$;
- $\mathcal{V}_{Pub}$ denotes the set of logistic centers;
- $\mathcal{E}$ is the set of routes connecting bases and centers;
- $\mathcal{E}_i$ denotes the set of routes in the local area of agent i; for example, in agent a's local area, $\mathcal{E}_a = \{(a,1) \sim (a,2),(a,2) \sim (a,3),(a,3) \sim (a,4),...\}$;
- $\mathcal{P}$ includes the position of bases, logistic centers and packages;
- $\mathcal{P}_i$ includes 1) the position of packages in the local area of agent i; for example, if agent a has a package in $(a,1)$, then $\mathcal{P}_a = \{package\_in\_(a,1)\}$; 2) the number of bases in the local area of agent i; 3) the number of routes in the local area of agent i and 4) the length of these routes.
- $\mathcal{A}$ includes the actions of moving from a base or a logistic center to another base or logistic center;
- $\mathcal{A}_i$ includes the actions of moving from a base or a logistic center to another base or logistic center in the local area of agent i; for example, an action of agent a can be: *moving from* $(a,1)$ *to* $(a,2)$ which is abbreviated as $(a,1) \rightarrow (a,2)$, where the pre-condition of this action is $package\_in\_(a,1)$ and the effect of this action is $package\_in\_(a,2)$;
- If a wants to transport a package from $(a,3)$ to $(e,2)$, then $\mathcal{I} = \{package\_in\_(a,3)\}$ and $\mathcal{G} = \{package\_in\_(e,2)\}$: $\mathcal{V}_\mathcal{I} = (a,3)$ and $\mathcal{V}_\mathcal{G} = (e,2)$.

If agent a is to transport a package from $(a,3)$ to $(e,2)$, the associated plan could be $\Pi_a^\triangleright = \langle \mathcal{V}_\mathcal{I} \rightarrow (a,4),(a,4) \rightarrow A, A \rightarrow B, B \rightarrow E, E \rightarrow \mathcal{V}_\mathcal{G} \rangle$. In plan $\Pi_a^\triangleright$, the details of how to move from A to B, from B to E and from E to $(e,2)$ are not included, as these details involve other agents' private information that is unknown to agent a. In fact, as $(e,2)$ is private to agent e, agent a is unaware of the existence of $(e,2)$. Agent a, however, knows that the destination is in agent e's local area.

Specifically, each agent's private information includes two parts: private facts and private actions. An agent's private facts include four components: 1) the number of nodes in its local area, i.e., the number of military bases in the logistic example, 2) the number of edges in its local area, i.e., the number of routes in the logistic example, 3) the length of these edges, i.e., the length of routes in the logistic example and 4) the positions of any items in its local area, i.e., the positions of packages in the logistic example. An agent's private actions are the movements of items in its local area. In this private information, the positions and movements of items are not required by other agents. Thus, these two pieces of information will not be disclosed to other agents. For the other three pieces of information: the number of nodes, the number of edges and the length of edges, since agents have to share the three pieces of information for planning, we need to develop a privacy-preserving mechanism to protect them.

Formally, we have the following definition.

Definition 1 (Agents' privacy). *An agent i's privacy is defined as a 3-tuple: $\langle \mathcal{V}_i, \mathcal{E}_i, L(\mathcal{E}_i) \rangle$, where $\mathcal{V}_i$ is the set of nodes in agent i's local area, $\mathcal{E}_i$ is the set of edges and $L(\mathcal{E}_i)$ denotes the set of length of the edges.*

To protect the privacy of $\mathcal{V}_i$ and $\mathcal{E}_i$, we adopt the node-differential privacy technique and uses the Laplace mechanism to mask the number of both nodes and edges. To protect the privacy of $L(\mathcal{E}_i)$, we adopt the exponential mechanism along with a reinforcement learning algorithm.

4.2 Privacy-preserving multi-agent planning

The idea behind privacy-preserving multi-agent planning is based mainly on research in the field of secure multi-party computation [31], where multiple agents jointly compute a function while each agent possesses private input data. The goal is to compute the function without revealing agents' private input data.

One intuitive solution would be to simply not disclose any private information to others. However, since an agent must collaborate with other agents in order to achieve its goals, it is infeasible to hide all private information completely. To ensure that this private information is disclosed securely to the other agents, it is necessary to use privacy-preserving techniques.

Definition 2 (Strong Privacy [3]). *A multi-agent planning approach is strong privacy-preserving if none of the agents is able to infer any private facts regarding an agent's tasks from the public information it obtains during planning.*

In this paper, we adopt differential privacy, which is one of the most promising techniques in this field [8], to achieve strong privacy.

In addition to a privacy guarantee, a planning approach also needs soundness and completeness guarantees.

Definition 3 (Soundness [13]). *A planning approach is sound iff, for a given task, there is at least one valid plan followed by all participating agents to reach the goal state.*

Definition 4 (Completeness [6]). *A planning approach is complete iff, for a given task, 1) the approach is sound and 2) the approach can guarantee to create a valid plan.*

4.3 Differential privacy

Differential privacy (DP) can guarantee that any individual record being stored in or removed from a dataset will make little difference to the analytical output of the dataset [8], [32]. DP has already been successfully applied to agent advising [33], [34] and model publishing [35], [36]. Therefore, this property may also be suitable for application to the planning problem.

In differential privacy, two datasets D and D' are deemed neighboring datasets if they differ in only one record. A query f is a function that maps dataset D to

an abstract range $\mathbb{R}$: $f : D \to \mathbb{R}$. The goal of differential privacy is to mask the differences in the answers to query f between the neighboring datasets. In ϵ-differential privacy, parameter ϵ is defined as the privacy budget, which controls the privacy guarantee level of mechanism $\mathcal{M}$. A smaller ϵ represents stronger privacy. The formal definition of ϵ-differential privacy is as follows:

Definition 5 (ϵ-Differential Privacy [37]). *A mechanism $\mathcal{M}$ gives ϵ-differential privacy for any input pair of neighboring datasets D and D', and for any possible output set Ω, if $\mathcal{M}$ satisfies:*

$$Pr[\mathcal{M}(D) \in \Omega] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(D') \in \Omega] \tag{1}$$

In Definition 5, mechanism $\mathcal{M}$ is a function that takes a dataset as input and outputs a query result. Definition 5 states that if a mechanism, applied on two neighboring datasets, can obtain very similar results, then this mechanism is a differential privacy mechanism.

Sensitivity is a parameter that captures the magnitude by which a single individual's data can change the function f in the worst case.

Definition 6 (Sensitivity [37]). *For a query $f : D \to \mathbb{R}$, the sensitivity of f is defined as*

$$\Delta S = \max_{D,D'} ||f(D) - f(D')||_1 \tag{2}$$

Two of the most widely used differential privacy mechanisms are the Laplace mechanism and the exponential mechanism. The Laplace mechanism adds Laplace noise to the true answer. We use $Lap(b)$ to represent the noise sampled from the Laplace distribution with scaling b.

Definition 7 (Laplace mechanism [37]). *Given a function $f : D \to \mathbb{R}$ over a dataset D, Equation 3 is the Laplace mechanism that provides the ϵ-differential privacy [37].*

$$\widehat{f}(D) = f(D) + Lap(\frac{\Delta S}{\epsilon}) \tag{3}$$

Definition 8 (The Exponential Mechanism [37]). *The exponential mechanism $\mathcal{M}_E$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $exp(\frac{\epsilon u(D,r)}{2\Delta u})$, where $u(D, r)$ is the utility of a pair of dataset and output, and $\Delta u = \max_{r\in\mathcal{R}} \max_{D,D':||D-D'||_1\leq 1} |u(D, r) - u(D', r)|$ is the sensitivity of utility.*

If a graph is treated as a dataset, a given node in the graph can be interpreted as a record in the dataset. According to Definition 5, we can have a similar definition for ϵ-node-differential privacy as follows.

Definition 9 (ϵ-node-Differential Privacy [38]). *A mechanism $\mathcal{M}$ gives ϵ-node-differential privacy for any input pair of neighboring graphs G and G', where G and G' differ by at most one node, and for any possible output set, Ω, if $\mathcal{M}$ satisfies:*

$$Pr[\mathcal{M}(G) \in \Omega] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(G') \in \Omega] \tag{4}$$

Node-differential privacy guarantees similar output distributions on any pair of neighboring graphs that differ in one node and the edges adjacent to that node. Thus, the privacy of both nodes and edges can be preserved.

5 The Strong Privacy-Preserving Planning Approach

In this section, we first outline our approach in a general form, then use the aforementioned logistic example to instantiate our approach. A generalized form of our approach is presented in Algorithm 1. In Line 5 of Algorithm 1, agent i takes all the available public nodes into account to create a plan. These available public nodes are on the way from the initial state to the goal state and found by agent i during its searching phase. However, some of these available public nodes are not needed in the final plan. Then, in Line 8, agent i uses a reinforcement learning algorithm to find the shortest route from the initial state to the goal state, and selects the public nodes on the shortest route to create a plan. The learning is based on the information obtained in Lines 6 and 7.

Algorithm 1: The general form of our approach

1 /*Take agent $i \in \mathcal{AG}$ as an example;*/
2 **Input**: agent i's local sets: $\mathcal{V}_i$, $\mathcal{E}_i$, $\mathcal{P}_i$, $\mathcal{A}_i$, and all the public facts and actions; also, the initial state $\mathcal{I}$ and the goal state $\mathcal{G}$;
3 **Output**: a complete plan $\Pi_i^{\triangleright}$ from $\mathcal{I}$ to $\mathcal{G}$;
4 Agent i identifies $\mathcal{V}_\mathcal{I}$ and $\mathcal{V}_\mathcal{G}$ from the initial state $\mathcal{I}$ and the goal state $\mathcal{G}$, respectively, and initializes plan: $\Pi_i^{\triangleright} = \langle \mathcal{V}_\mathcal{I} \to \mathcal{V}_\mathcal{G} \rangle$;
5 Agent i searches the goal state, and details plan $\Pi_i^{\triangleright}$ by adding the available public actions into plan $\Pi_i^{\triangleright}$: $\Pi_i^{\triangleright} = \langle \mathcal{V}_\mathcal{I} \to v_j, ..., v_k \to \mathcal{V}_\mathcal{G} \rangle$, where $\{v_j, ..., v_k\} \subset \mathcal{V}_{Pub}$;
6 Agent i queries the intermediate agents to request local private facts;
7 Each of these intermediate agents obfuscates its local private facts using the differential privacy technique;
8 Agent i uses the obfuscated facts to refine the plan by removing unnecessary public actions by means of a reinforcement learning algorithm: $\Pi_i^{\triangleright} = \langle \mathcal{V}_\mathcal{I} \to v_x, ..., v_y \to \mathcal{V}_\mathcal{G} \rangle$, where $j \leq x, y \leq k$;
9 Each action in plan $\Pi_i^{\triangleright}$ is further refined by each agent creating a local plan; for example, action $\mathcal{V}_\mathcal{I} \to v_x$ is refined by agent i creating a local plan as $\langle \mathcal{V}_\mathcal{I} \to v_{i_a}, ..., v_{i_b} \to v_x \rangle$, where $\{v_{i_a}, ..., v_{i_b}\} \subset \mathcal{V}_i$;
10 Agent i merges these local plans to form a complete plan: $\Pi_i^{\triangleright} = \langle \mathcal{V}_\mathcal{I} \to v_{i_a}, ..., v_{i_b} \to v_x, ..., v_y \to \mathcal{V}_\mathcal{G} \rangle$; note that the details of local plans, created by intermediate agents, are not shown in plan $\Pi_i^{\triangleright}$, since they contain non-obfuscated private facts belonging to the intermediate agents;

To instantiate this general approach, we use the logistic example given in Section 3. In this example, we assume that 1) all routes in the logistic map are bi-directional; 2) each individual agent controls only one local area; and 3) there are no isolated nodes on the map. An agent follows three steps to create a plan:

- Step 1: the agent creates a high-level logistic map;
- Step 2: the agent asks the agents in the intermediate areas to provide route and map information;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3017497, IEEE Transactions on Dependable and Secure Computing

- Step 3: the agent uses the received information to create a complete plan.

5.1 Step 1: Creating a high-level map

In Fig. 1, it is supposed that agent a has a package to transport from $(a, 2)$ to $(f, 4)$. As agent f is not agent a's neighbor, a must query its neighbors, b, c, and d, regarding the position of f. Two agents are deemed neighbors if there is at least one logistic center connecting two military bases, such that one of these bases belongs to each of the agents.

In the case that agents, b, c and d, also do not have f as a neighbor, they pass this query on to their neighbors, e.g., agent e. Finally, agent f is found through agent e. By using the information acquired while finding agent f, agent a can create a high-level logistic map, as shown in Fig. 3.

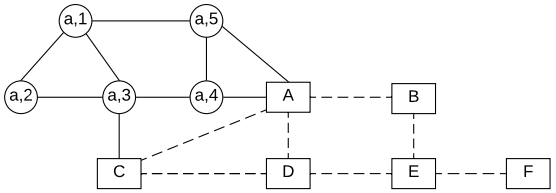


Fig. 3. A high-level logistic map from agent a's perspective

5.2 Step 2: Each intermediate agent provides map and route information

After creating the high-level map, agent a asks the agents in the intermediate areas to provide route and map information. In Fig. 3, the intermediate agents are b, c, d and e. To protect the topological privacy of local maps, each intermediate agent uses the Laplace mechanism to obfuscate its local map, i.e., modify the number of bases and routes. Moreover, to protect length privacy, each intermediate agent uses the exponential mechanism, along with a reinforcement learning algorithm, to assign probability distributions over the routes on its obfuscated local map while removing the distance information. Finally, each intermediate agent presents an obfuscated local map, with probability distributions over routes, to agent a. An example explaining this process is presented below.

In this example, Fig. 4(a) is agent b's local map with route length. Fig. 4(b) is agent b's obfuscated local map. Referring to the obfuscated local map, agent b calculates the shortest route between logistic centers A and B. Then, agent b marks the probability distributions over the routes, as shown in Fig. 4(c). Each probability on a route indicates the probability of that route being selected. To guarantee the route length privacy, agent b uses the exponential mechanism to redistribute these probabilities over the routes, as shown in Fig. 4(d). Agent b then sends Fig. 4(d) to agent a. Finally, agent a receives a map where the topology has been obfuscated and the distance information has been replaced by probability distributions.

5.2.1 Using the Laplace mechanism to obfuscate topologies

The Laplace mechanism is applied to the statistical information contained in a map. We utilize a $1K$-distribution [39] to obtain the statistical information. More specifically, the $1K$-distribution is used to calculate the node degree distribution of a given graph. To describe how the $1K$-distribution is utilized for this purpose, we employ the following example. In Fig. 4(a), the number of nodes with 1 degree is 0; the number of nodes with 2 degrees is 4, (i.e., nodes A, B, $(b, 4)$ and $(b, 5)$); the number of nodes with 3 degrees is 2, (i.e., nodes $(b, 2)$ and $(b, 3)$); and the number of nodes with 4 degrees is 1, (i.e., node $(b, 1)$). Thus, the $1K$-distribution, i.e., the node degree distribution, of Fig. 4(a) is: $P(1) = 0$, $P(2) = 4$, $P(3) = 2$, and $P(4) = 1$.

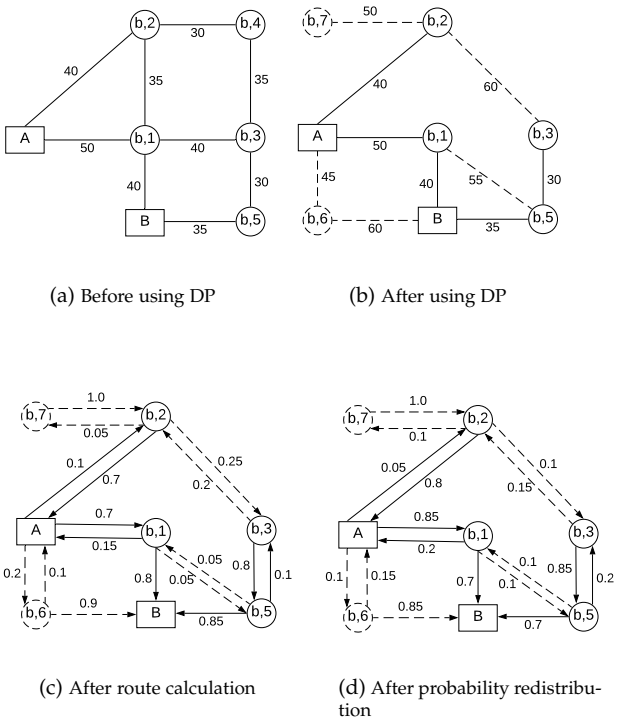


Fig. 4. Obfuscation of agent b's local map

Algorithm 2: The Laplace mechanism-based obfuscation

```
1 /*Take agent b as an example*/
2 Input: agent b's map (Fig. 4(a));
3 Output: agent b's obfuscated map (Fig. 4(b));
4 Use 1K-distribution to obtain the statistical
  information of b's map;
5 for k = 1 to d_max do
6     P̃(k) ← P(k) + ⌈Lap(ΔS·d_max / ε)⌉;
7 Rewire nodes to satisfy each P̃(k);
```

The Laplace mechanism-based obfuscation is outlined in Algorithm 2. In Line 4, the statistical information of b's map is obtained using the $1K$-distribution. In Lines 5-6, the Laplace noise is added to each $P(k)$ in order to randomize the node degree distribution; accordingly the number of nodes now becomes $\sum_{1 \le k \le d_{max}} \tilde{P}(k)$. Here, d_{max} is the maximum node degree in a map, and $d_{max} = 4$ in the example of Fig. 4(a). After adding Laplace noise, the node

degree distribution could be as follows: $\tilde{P}(1) = 1$, $\tilde{P}(2) = 2$, $\tilde{P}(3) = 5$, and $\tilde{P}(4) = 0$. Next, in Line 7, nodes are rewired to satisfy each $\tilde{P}(k)$, where $k \in \{1, ..., d_{max}\}$. The node rewiring is carried out using the graph model generator provided in [39]. After node rewiring is complete, fake routes may be introduced, such as route $A \rightarrow (b, 6)$ in Fig. 4(b). The length of a fake route is randomly generated based on the average length of the existing real routes.

The reason why the Laplace mechanism is used here is that our aim is to obfuscate the topology of each agent's local map by modifying the degree distribution. Since a degree distribution consists of a set of numbers, the Laplace mechanism is more appropriate here than the exponential mechanism which is mainly used for proportionally selecting an element from a set. It should also be noted at this point that the Laplace mechanism may generate negative numbers. This, however, is not a problem in this paper, as we need both positive and negative Laplace noise to ensure that our approach satisfies ϵ-differential privacy. Moreover, we adopt the Laplace mechanism to add noise to node degree distributions rather than directly adding noise to the number of nodes or edges. By adding noise to node degree distributions, our approach can not only guarantee the node and edge privacy of agents, but also guarantee the connection of an obfuscated graph. The connection of an obfuscated graph is a necessity for the completeness of our planning approach. The detailed theoretical analysis will be given in the next section.

The rationale behind Algorithm 2 is as follows. According to the definitions of differential privacy, a map is interpreted as a dataset D, while a node on a map is interpreted as a record in a dataset. As with the concept of neighboring datasets, two maps are deemed neighbors if they differ by only one node. Thus, using $1K$-distribution to obtain a map's statistical information can be thought of as querying some interesting information from a dataset, $f(D)$. If we compare Definition 7 to Line 6 in Algorithm 2, we can see that just as the Laplace mechanism can guarantee the privacy of a dataset, it can also guarantee the privacy of a map. More discussion about the preservation of privacy will be provided in the next section.

In Algorithm 2, ΔS represents the sensitivity of the degree distribution in a map. The value of ΔS is determined by the maximum change in degree distribution when a node is added into or removed from the map. For example, in Fig. 4(a), the degree scaling is from 1 to 4: $P(1), P(2), P(3), P(4)$. According to Algorithm 2, Line 6, when a node is added into or removed from the map, one of the four values, $P(1), P(2), P(3), P(4)$, will be incremented or decremented by 1. Thus, the maximum change of degree distribution is 1, i.e., $\Delta S = 1$ in Algorithm 2.

5.2.2 Using reinforcement learning to compute probability distributions

In a local area, such as the one in Fig. 4(a), there is a set of local military bases and logistic centers, along with a set of routes connecting these bases and centers. As discussed in Section 4, in the Graph-STRIPS model, $\mathcal{V}$ and $\mathcal{E}$ can be used to represent the topology of a map. Accordingly, we use $\mathcal{V}$ to represent the military bases and logistic centers, while $\mathcal{E}$ is used to denote the set of routes connecting these bases and centers. Specifically, in Fig. 4(a), $\mathcal{V}_b = \{(b,1),(b,2),(b,3),(b,4),(b,5)\}$, and $\mathcal{E}_b = \{A \sim (b,1), A \sim (b,2), ..., (b,1) \sim B, (b,5) \sim B\}$. Moreover, different bases or centers will have different routes available to them. For example, in base $(b,1)$, there are four available routes: $\mathcal{E}_{(b,1)} = \{(b,1) \sim A, (b,1) \sim (b,2), (b,1) \sim (b,3), (b,1) \sim B\}$. Furthermore, in center A, there are two available routes: $\mathcal{E}_A = \{A \sim (b,1), A \sim (b,2)\}$.

Algorithm 3: The reinforcement learning algorithm

1 /*Take agent b as an example*/
2 **Input**: agent b's obfuscated map (Fig. 4(b));
3 **Output**: agent b's obfuscated map with probability distributions (Fig. 4(c));
4 Initialize probability distributions;
5 Initialize the Q-value of each route;
6 Initialize the current position: $v \leftarrow A$;
7 **while** $v \neq B$ **do**
8 Agent b selects a route, e, based on the probability distribution $\boldsymbol{\pi}(v) = \langle \pi(v, e_1), ..., \pi(v, e_n) \rangle$, where $e \in \mathcal{E}_v = \{e_1, ..., e_n\}$;
9 $r \leftarrow \mathcal{R}(v, e)$;
10 $Q(v,e) \leftarrow (1-\alpha)Q(v,e) + \alpha[r + \gamma max_{\substack{e_i \\ e_i \in \mathcal{E}_{v'}}} Q(v', e_i)]$;
11 $\overline{r} \leftarrow \sum_{e_i \in \mathcal{E}_v} \pi(v, e_i) Q(v, e_i)$;
12 **for** *each route* $e_i \in \mathcal{E}_v$ **do**
13 $\pi(v, e_i) \leftarrow \pi(v, e_i) + \zeta(Q(v, e_i) - \overline{r})$;
14 $\boldsymbol{\pi}(v) \leftarrow Normalise(\boldsymbol{\pi}(v))$;
15 $v \leftarrow v'$;
16 Agent b marks the learned probability distributions over the routes;

The reinforcement learning algorithm is outlined in Algorithm 3. In Line 4, agent b proportionally initializes probability distributions over actions, where each action indicates the selection of a route. The initialization is based on the lengths of the routes. For example, in Fig. 4(b), the probability distribution over routes $A \sim (b,1)$ and $A \sim (b,6)$ can be initialized as $\frac{4}{9}$ and $\frac{5}{9}$, respectively. In Line 5, agent b initializes the Q-value of each route; here, the Q-value is an indication of how good a route is. In this algorithm, the initial Q-value of a route is set based on the length of the route, such that a shorter route is allocated a higher Q-value. For example, in Fig. 4(b), the initial Q-value of route $A \sim (b,1)$ can be set to $\frac{100}{50} = 2$, while the initial Q-value of route $A \sim (b,6)$ can be set to $\frac{100}{40} = 2.5$. In Line 6, agent b sets the initial position to A and the destination to B. This setting is based on the fact that, as an intermediate agent, agent b will help agent a to transport the package from A to B.

Regarding the loop, in Line 8, agent b selects a route e based on the probability distribution over the available routes in base v. After taking route e, agent b receives a reward r (Line 9), which is inversely proportional to the route length. For example, in Fig. 4(b), $r(A \sim (b,1))$ and $r(A \sim (b,6))$ can be set to 4 and 5, respectively. The reward r is used to update the Q-value of route e in base v (Line 10). This update is based on: 1) the current Q-value of e in base v, $Q(v,e)$; 2) the maximum Q-value of the routes in new

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3017497, IEEE Transactions on Dependable and Secure Computing

base v', $max_{e_i \in \mathcal{E}_{v'}} Q(v', e_i)$; 3) the immediate reward r; and 4) a learning rate α and a discount rate γ. In the next step, the updated Q-value and the probability distribution are used to compute the average reward $\overline{r}$ (Line 11), where $\mathcal{E}_v$ is the set of available routes in base v. In Lines 12 and 13, the probability of selecting each route $i \in \mathcal{E}_v$ is updated. This update is based on: 1) the current probability of each route being selected $\pi(v, e_i)$; 2) the current Q-value of each route $Q(v, e_i)$; 3) the average reward $\overline{r}$; and 4) a learning rate ζ. In Line 14, the updated probability distribution is normalized to be valid, meaning that for each $i \in \mathcal{E}_v$, $0 < \pi(v, e_i) < 1$ and $\sum_{e_i \in \mathcal{E}_v} \pi(v, e_i) = 1$. In Line 15, the new base, v', is set as the current base. The above steps are iterated over until the goal state is reached. Finally, in Line 16, agent b marks each of the routes with the learned probability distributions.

5.2.3 Using the exponential mechanism to redistribute probabilities

After using the reinforcement learning algorithm to replace distance information with probability distributions, agents' local distance information can be hidden. Hiding distance information can reduce the risk of leaking this information but cannot guarantee the privacy preservation of this information. Therefore, we adopt the exponential mechanism to redistribute probabilities.

We use an example to explain how to use the exponential mechanism to redistribute probabilities. Suppose a node in a local map has two adjacent edges, x and y, and the probabilities of selecting x and y are 0.7 and 0.3, respectively. Based on the definition of exponential mechanism, the exponential mechanism selects and outputs an element r with probability proportional to $exp(\frac{\epsilon u_r}{2\Delta u})$, where ϵ is the privacy budget, u_r is the utility of selecting r and Δu is the sensitivity of utility. If we set the utility of selecting a route to be the probability of selecting that route, then we have: $u_x = 0.7$ and $u_y = 0.3$, and in this setting, $\Delta u = 1$. Then, if we set $\epsilon = 2$, we have $exp(\frac{\epsilon u_x}{2\Delta u}) = 2.014$ and $exp(\frac{\epsilon u_y}{2\Delta u}) = 1.350$. Finally, the probabilities of selecting x and y become $\frac{2.014}{2.014+1.350} = 0.6$ and $\frac{1.350}{2.014+1.350} = 0.4$, respectively. The above process is performed on each node in the local map.

Another simple way to preserve the distance information privacy is to let each agent use the Dijkstra's algorithm [40] to compute the shortest route length between two logistic centers in its local area and add a Laplace noise to that length. However, other agents may still get an approximate idea about the route length. For example, after adding a Laplace noise, the route length changes from 100 to 105. Although other agents cannot deduce the real length, they can still guess that the real length must be near 105. In some situations, e.g., the military logistic example, an approximate length is good enough for other agents. By contrast, if an agent uses reinforcement learning and shares only probabilities, other agents cannot obtain even an approximate length. This idea is based on the spirit of federated learning by allowing agents to share only parameters [41]. In federated learning, to protect each client's training data privacy, each client only sends the model parameters, trained based on her private data, to the server. The server, thus, has only clients' model parameters without any clients' private data.

5.3 Step 3: Creating a complete plan

After receiving obfuscated local maps from intermediate agents, agent a creates a logistic map by combining these obfuscated local maps, as shown in Fig. 5. On each obfuscated local map, although both real and fake nodes and edges are involved, agent a is unable to determine whether a given node or edge is real. More detailed discussion on this matter will be presented in Section 6.

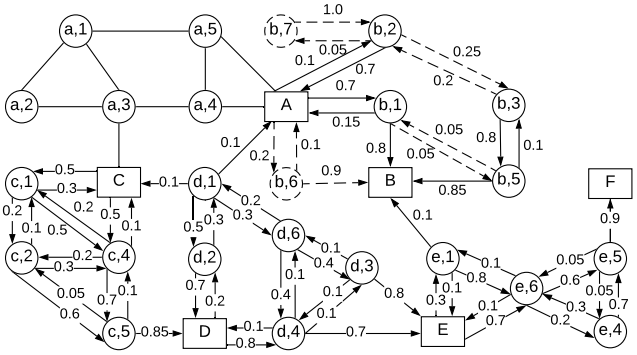


Fig. 5. A logistic map created by obfuscated local maps

Next, agent a uses a reinforcement learning algorithm to calculate the length of the route between each pair of connected logistic centers, e.g., $A \rightarrow B$, $B \rightarrow E$ and so on. The reinforcement learning algorithm is similar to Algorithm 3. Since agent a is only provided with probability distributions about the other areas, agent a must generate the distance information itself based on the probability distributions. Agent a relates the probabilities to the distance based on the average route length in agent a's local area. For example, in Fig. 5, the probabilities of selecting routes $A \sim (b, 1)$ and $A \sim (b, 6)$ are 0.7 and 0.2, respectively. If the average route length in agent a's local area is 45, agent a can simply set the distances from A to $(b, 1)$ and A to $(b, 6)$ to 20 and 70, respectively, whose average is 45. Here, we operate under the assumption that there are no significant differences between the average route length in each local area.

After agent a calculates the length of the shortest route between each pair of connected logistic centers (as shown in Fig. 6), the shortest route from the origin to the destination can also be obtained. It is clear at this point that this calculation is not very accurate, as it is based on estimated length. However, the aim of this calculation is not to find the real shortest route, rather to select the intermediate agents which are located on the shortest route. In Fig 6, the agents on the shortest route are: b, e and f.

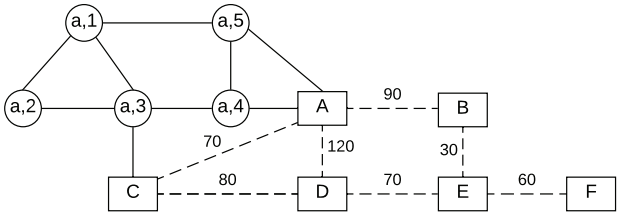


Fig. 6. A high-level map featuring relative distances from agent a's perspective

The final plan, thus, can be expressed as $\Pi_a^{\triangleright} = \langle \mathcal{I} \rightarrow (a,3) \rightarrow (a,4) \rightarrow A \rightarrow B \rightarrow E \rightarrow F \rightarrow \mathcal{G} \rangle$, where $\mathcal{I} = \{package\_in\_(a,2)\}$ and $\mathcal{G} = \{package\_in\_(f,4)\}$. In this plan, $\mathcal{I} \rightarrow (a,3) \rightarrow (a,4) \rightarrow A$ is the local plan formulated and carried out by agent a. At logistic center A, agent a gives its package to agent b, which makes a local plan to transport the package to logistic center B. At center B, agent e takes control of the package and devises a local plan to deliver the package to logistic center F. Finally, agent f picks up the package at center F and makes a local plan to transfer the package to $(f,4)$.

5.4 A simplification of the proposed approach

In some situations, if the distance information is not private, we can let logistic centers do the routing planning and consider the routing only between logistic centers. Each logistic center can directly communicate with the agent that is connected with the logistic center. As the distance information is not private, each logistic center is also aware of the local routing information within the agent. Compared with the proposed approach, this simplified approach can 1) significantly reduce the problem complexity; and 2) enable agents to obtain accurate distance information for further calculation. and 3) fully hide the topology information belonging to each agent from other agents.

A typical example is daily logistic, where the distance information between two public places do not need to be hidden. In daily logistic, packages are transported from their starting points to their destinations across multiple states or provinces. Here, the distance among states/provinces is not a privacy concern and can be considered as public information. The simplified version of our approach can be applied to this example. Each state/province is assumed to have a logistic center. To transport a package, the logistic center at the starting point utilizes the accurate distance information among states/provinces to make an optimal global plan. Then, each logistic center in the global planning path conducts the local routing planning.

6 Theoretical Analysis

6.1 Soundness analysis

Theorem 1. *The proposed approach is sound.*

Proof. We prove this theorem by considering one task, e.g., delivering one package in the logistics example. In Step 1 of our approach, we start from the initial agent which has a task to complete and initializes a plan, each queried agent sets up a link to the querying agent. Thus, all the queried agents are reachable. If a goal agent is identified whose private facts include the goal state, there must be at least one plan connecting the initial agent to the goal agent through some or all of the queried agents. □

6.2 Completeness analysis

Lemma 1. *Obfuscating local maps does not affect the completeness of the proposed approach.*

Proof. In Step 2 of our approach, each intermediate agent obfuscates its local map by adding and/or removing nodes and/or edges (see Algorithm 2). During the obfuscation process, Laplace noise is added to the node degree distribution of the local map: $P(1)$, ..., $P(d_{max})$. As $P(0)$ is not counted, isolated nodes will not be created. Moreover, as the obfuscated map is undirected, it can be guaranteed that the obfuscated map will be connected. Hence, there must be at least one route between the two logistic centers on the local map. Since this property is common to the local maps of all intermediate areas, there must be at least one route from the initial area to the goal area via intermediate logistic centers. Thus, the completeness is not affected. □

Theorem 2. *The proposed approach is complete.*

Proof. Step 1 of our approach guarantees that a goal agent can be found. According to Theorem 1, there must be at least one plan connecting the initial agent to the goal agent. We now need only to prove that our approach is capable of finding at least one of these plans.

According to Lemma 1, there is at least one route from the initial area to the goal area. One of these routes can be treated as a high-level plan, which can be identified using Algorithm 3. Based on the high-level plan, each intermediate agent creates a local plan (Step 3). Given that each agent is honest[1], each local plan is valid, which ensures that the two logistic centers in the local area will be connected. Therefore, a high-level plan and a set of local plans constitute a complete plan. □

6.3 Privacy-preserving analysis

Theorem 3. *The proposed planning approach satisfies ϵ-differential privacy.*

Proof. To analyze the privacy guarantee, we apply two composite properties of the privacy budget: the sequential and the parallel compositions [42]. The sequential composition determines the privacy budget ϵ of each step when a series of private analysis are performed sequentially on a dataset. The parallel composition corresponds to the case in which each private step is applied to disjoint subsets of a dataset. The ultimate privacy guarantee depends on the step which has the maximal ϵ.

In the proposed approach, the Laplace mechanism and the exponential mechanism consumes the privacy budget. In the Laplace mechanism in Algorithm 2, the Laplace noise sampled from $Lap(\frac{\Delta S \cdot d_{max}}{\epsilon})$ is added in d_{max} steps. At each step, the Laplace mechanism consumes the $\frac{\epsilon}{d_{max}}$ privacy budget; thus for each step, Algorithm 2 satisfies $\frac{\epsilon}{d_{max}}$-differential privacy. By using the sequential composition property, we can conclude that at a total of d_{max} steps, the Laplace mechanism consumes the $d_{max} \cdot \frac{\epsilon}{d_{max}} = \epsilon$ privacy budget, meaning that Algorithm 2 satisfies ϵ-differential privacy. By comparing Definition 5 with Definition 9, since the Laplace mechanism can guarantee the data record privacy of a dataset, it can also guarantee the node-privacy of a graph.

The exponential mechanism is used to redistribute probabilities on each agent's local graph. For a given node in a local graph, suppose the node has k adjacent edges. Then, the exponential mechanism will be used k times. If we

1. It is a common assumption in privacy-preserving multi-agent planning that agents are honest but curious about others' private information [3].

set privacy budget for this node to be $\frac{\epsilon}{k}$, based on the sequential composition property, the privacy consumption of this node is ϵ. Thus, the probability redistribution on the adjacent edges of this node satisfies ϵ-differential privacy. When this method is used on every node, based on the parallel composition property, the probability redistribution on this local graph satisfies ϵ-differential privacy.

Since the Laplace mechanism and the exponential mechanism are used by each agent, each agent is guaranteed ϵ-differential privacy. Although an environment may contain multiple agents, each agent maintains a local area, and these local areas are disjoint with each other. Since each agent is guaranteed ϵ-differential privacy, according to the parallel composition property, the proposed approach satisfies ϵ-differential privacy. □

Remark 1: In Algorithm 2, Laplace noise is used to randomize the node degree distribution. This implies that both the number of nodes and the number of edges in a local map will be perturbed. Since the topology of a map consists of nodes and edges, perturbing the numbers of nodes and edges incurs perturbation of the topology. Accordingly, as Algorithm 2 satisfies differential privacy, the perturbation of the topology of a map also satisfies differential privacy.

Corollary 1. *No agent is able to conclude anything about the existence of any subset of $\lceil \frac{\Delta S \cdot d_{max}}{\epsilon} \rceil$ nodes in another agent's map.*

Proof. In Algorithm 2, the Laplace noise is sampled from $Lap(\frac{\Delta S \cdot d_{max}}{\epsilon})$, meaning that the expected amount of noise is $\frac{\Delta S \cdot d_{max}}{\epsilon}$. As this noise is used to change the number of nodes in a map (recall Lines 5-6 in Algorithm 2), the expected number of nodes that will be changed is $\lceil \frac{\Delta S \cdot d_{max}}{\epsilon} \rceil$. Therefore, any subset of $\lceil \frac{\Delta S \cdot d_{max}}{\epsilon} \rceil$ nodes could be fake nodes. According to Definition 9 and Theorem 3, since Algorithm 2 can guarantee the node-privacy of a graph, an agent will be unable to distinguish real from fake statistical information between two neighboring graphs, e.g., the number of real nodes. This means that an agent cannot determine whether or not a node is fake. Hence, the existence of any subset of $\lceil \frac{\Delta S \cdot d_{max}}{\epsilon} \rceil$ nodes in an agent's map cannot be concluded by any other agents. □

Remark 2: From Corollary 1, in the Laplace mechanism in Algorithm 2, the value of ϵ controls the granularity of privacy, given that the values of ΔS and d_{max} have been fixed. A smaller ϵ implies a stronger privacy guarantee. However, a smaller ϵ also introduces a larger amount of noise. The increase of the amount of noise reduces the usability of a map. Thus, the value of ϵ should be carefully set.

Remark 3: Similar to the Laplace mechanism, in the exponential mechanism, the value of ϵ has a huge impact on probability redistribution results. Given that a node has k adjacent edges and the probabilities of selecting the k edges are $u_1, ..., u_k$, if we set $\epsilon = 0$, the probability of selecting each edge will equally become $\frac{1}{k}$; if we set $\epsilon \to +\infty$, probability u_m becomes 1 and others become 0, where $u_m = max\{u_1, ..., u_k\}$. In addition to the two extreme situations, there is a median situation which is that the redistributed probabilities are identical to the original probabilities: $u'_1 = u_1, ..., u'_k = u_k$. Based on the computation method described in Section 5.2, each probability u'_i, $1 \leq i \leq k$, is computed as:

$$u'_i = \frac{exp(\frac{\epsilon u_i}{2\Delta u})}{\sum_{1 \leq j \leq k} exp(\frac{\epsilon u_j}{2\Delta u})}. \tag{5}$$

Let each $u'_i = u_i$, we have k equations.

$$\begin{cases} \frac{exp(\frac{\epsilon u_1}{2\Delta u})}{\sum_{1 \leq j \leq k} exp(\frac{\epsilon u_j}{2\Delta u})} = u_1, \\ ..., \\ \frac{exp(\frac{\epsilon u_k}{2\Delta u})}{\sum_{1 \leq j \leq k} exp(\frac{\epsilon u_j}{2\Delta u})} = u_k. \end{cases}$$

In our problem, $\Delta u = 1$. By solving the k equations, we have that

$$\epsilon_i = \frac{2(k \cdot ln(u_i) - \sum_{1 \leq j \leq k} ln(u_j))}{k \cdot u_i - \sum_{1 \leq j \leq k} u_j},$$

where $1 \leq i \leq k$. Thus, in applications, on one hand, these values of ϵ should be avoided, as they will make the redistributed probabilities identical to the original probabilities, which cannot offer any privacy preservation. On the other hand, the values of ϵ should be set close to these values to guarantee the usability of the redistributed probabilities.

Theorem 4. *The proposed planning approach can strongly preserve agents' privacy.*

Proof. As defined in Section 4.1, an agent's private information includes 1) the number of nodes in an agent's local area, 2) the number of edges in the local area, 3) the length of these edges, 4) the positions of any items in the local area and 5) the movements of any items in the local area. To prove this theorem, we only need to prove that the private information possessed by an agent cannot be inferred by another agent. First, according to Theorem 3 and Corollary 1, the proposed planning approach satisfies ϵ-differential privacy and guarantees the privacy of any subset of $\lceil \frac{\Delta S \cdot d_{max}}{\epsilon} \rceil$ nodes in an agent's local area. By properly setting the value of ϵ, the privacy of all nodes and edges in an agent's local area can be preserved. Therefore, the privacy of the number of nodes and edges of an agent's local area will also be preserved. Second, our approach dictates that the length information in a local area is replaced by probability distributions (recall Fig. 4). Also, these probabilities are redistributed using the exponential mechanism. Thus, the length information is strictly hidden. Therefore, an agent cannot infer the real length of any individual edge in another agent's local area. Third, since the privacy of any node or edge in an agent's local area has been preserved, the positions and movements of items have also been preserved. Based on the definition of strong privacy (Definition 2), the proposed approach can strongly preserve agents' privacy. □

6.4 Communication analysis

Let us suppose that there are m logistic centers. Each logistic center, i, has a capacity, lc_i, which is the maximum number of agents that can share the logistic center. Accordingly, we derive the following theorem:

Theorem 5. *In Step 1, the upper bound of the number of communication messages used to find a goal agent is* $\sum_{1\leq i\leq m} lc_i$.

Proof. In our approach, each agent is only aware of the existence of its own neighbors. This means that 1) each agent does not know how many neighbors any other agent has, and 2) each agent is not aware of how far away the goal agent is.

As the information regarding logistic centers is public, all agents know the capacity of each logistic center. Thus, to guarantee that the query message is able to reach the goal agent, an agent must assume that 1) each logistic center is using up its capacity, and 2) the goal agent is located in the most distant area. In this situation, the number of generated communication messages is $\sum_{1\leq i\leq m} lc_i$. □

Remark 4: Theorem 5 describes the communication overhead in the worst case. However, as time progresses, this communication overhead can be significantly reduced. This is because an agent memorizes the plans that it has previously created, meaning that an agent memorizes the routes to goal agents. Thus, in the future, an agent can simply exploit a route previously determined to reach a goal agent without the need for communication. Even if an agent decides to explore a new route, the communication overhead can be limited by setting the maximum number of query messages during the finding process. The maximum number of query messages is set to be identical to the number of messages used to find the same goal agent last time. Formally, we have the following corollary:

Corollary 2. *As time progresses, the communication overhead of each agent monotonically decreases.*

Proof. Every time an agent explores a new route to a goal agent, the maximum number of query messages is set to be equal to the number of messages used to find the same goal agent last time. As each agent memorizes only the shortest routes to goal agents, only routes that are shorter than these memorized routes will be taken by each agent. This means that the number of request messages currently being used must be fewer than or equal to the number used previously. Thus, the communication overhead of each agent monotonically decreases. □

In our approach, the setting of the communication budget C can be controlled by the privacy budget ϵ. In a multi-agent system, each agent k sets ϵ/C as their privacy budget and ceases to communicate when ϵ is used up. When $C > \sum_{1\leq i\leq m} lc_i$, the system can guarantee that all communication steps will be completed. However, a large amount of noise will be added to the system under these circumstances. When $C < \sum_{1\leq i\leq m} lc_i$, the system is likely to stop before finishing the communication steps. However, the noise added to the system will be limited. When $C = \sum_{1\leq i\leq m} lc_i$, the system will stop when all communication steps have been completed. Therefore, by adjusting the privacy budget ϵ and the communication budget C, the communication overhead of a multi-agent system can be controlled.

7 Application of Our Approach to Other Domains

This section illustrates how our approach can be applied to three other domains: networks, air travel, and rovers.

7.1 Packet routing in networks

In a network, nodes often transmit packets between each other. These nodes may belong to different areas, which are connected by routers or access points. In this domain, a router or access point can be thought of as similar to an agent, which manages a corresponding area. In a given area, the information possessed by each node, e.g., its load and performance, is private to the agent. Moreover, the number of nodes in an area and their communication links are also private to the agent. Thus, the agents expect that their privacy will be preserved.

As each node has only a limited range of communication, when a node transmits a packet to another node, the packet may be relayed multiple times by intermediate nodes before reaching its destination. Since it is highly desirable that nodes receive packets in a timely manner, the transmission must be efficient so that huge delays can be avoided. The proposed approach can be applied to create efficient plans for packet routing.

7.2 Airplane transport

The airplane transport problem consists of a set of planes and airports. Moreover, the travel map is partitioned into a set of areas. In the real world, each area can be thought of as a country. Therefore, the planes and airports located in a given area are private to the area air traffic controller. Clearly, each area controller wants to preserve information regarding the status and number of planes and airports in their area as private information.

The airports located on the boundary of two areas are public. The goal is to transport passengers between airports. In this problem, each area controller can be thought of as an agent. When a plane travels from one airport to another, as the plane has only limited fuel, passengers may be transferred multiple times on their way to their destination. Moreover, both area controllers and passengers would clearly prefer the plane to reach its destination as quickly as possible. Thus, an efficient privacy-preserving planning approach is required. The proposed approach can be applied to create efficient plans for passenger transport.

7.3 Rover exploration

This domain models Mars exploration rovers. Each rover can be thought of as an agent. The goal of these rovers is to collect samples. Each rover has its own private sets of targets and reachable locations. These targets and reachable locations can be thought of as private facts in our planning model, the privacy of which must be preserved.

Each rover collects samples in its reachable locations. When a rover needs to transmit the samples it has collected to another rover, these samples may have to be transmitted by intermediate rovers in the interim, as the number of locations reachable by each rover is limited. Since samples may decay as time progresses, it is desirable for the rovers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3017497, IEEE Transactions on Dependable and Secure Computing

to transmit the samples to the destination as quickly as possible. Hence, an efficient privacy-preserving planning approach is required. The proposed approach can be applied to create efficient plans for sample transmission.

In summary, our approach can be applied to all of the planning problems, in which each party has private information and local plans can be created by each party using reinforcement learning techniques. Moreover, reinforcement learning has a broad range of applications, including task scheduling in cloud computing [43], traffic light control [44], and robot coordination [45]. Since most of these applications may also have privacy requirements, our method has the potential to be applied to these real-world scheduling and coordination problems as well.

8 Experiments

8.1 Experimental setup

The experiments in the present research are conducted based on two scenarios: logistics and packet routing, which are typical logistic-like problems. In the logistics scenario, as described in Section 3, each military base has a set of packages to transport to other military bases. These military bases may be located in different areas and managed by different military units. The information pertaining to each military base is private to the managing military unit.

The packet routing scenario is similar to the logistics scenario, in that each node in an ad hoc network houses a set of packets to be sent to other nodes. Nodes may belong to different groups and are served by different access points. The information of each node is private to the serving access point. The key difference between these two scenarios is that in the packet routing scenario, new nodes may dynamically join the network and existing nodes may leave the network at any time, while this is not the case for the logistics scenario. These experiments have also been conducted on the air travel and rover scenarios. As the results present a similar trend to logistics, they are not discussed here.

Three evaluation metrics are used in the two scenarios:

1) average route length: the average length of the routes from initial states to goal states;

2) average communication overhead: the average number of communication messages used to make a plan;

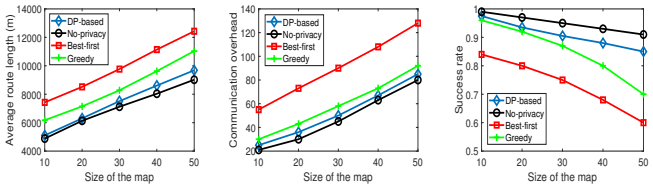
3) success rate: the ratio of the number of the successfully transmitted packages/packets to the total number of packages/packets.

In both scenarios, the map shape or network topology is similar to that in Fig. 1. The size of the maps/neworks varies from 10 logistic centers/access points to 50 logistic centers/access points; correspondingly the number of military bases/network nodes varies from 50 to 250 [2]. The probability of a package/packet being generated on each military base/node is set to 0.2. The communication budget of each agent varies from $C = 40$ to $C = 80$ depending on variations in the map/network size. The privacy budget of each agent is set to $\epsilon = 0.5$. Moreover, in the packet routing scenario, during the route finding process, there is a probability of 0.1 that an existing node will leave the network and a probability of 0.1 that a new node will join the network. The parameter values in the proposed algorithms are chosen experimentally, and set to $\alpha = 0.1$, $\gamma = 0.9$ and $\zeta = 0.95$.

The proposed planning approach, denoted as *DP-based*, is evaluated in comparison with three closely related approaches. The first approach, denoted as *No-privacy*, is also developed by us. The major features of *No-privacy* are the same as *DP-based*, but the privacy-preserving mechanism has been removed. Although *No-privacy* is not applicable to privacy-preserving planning, it can be used to evaluate how the privacy-preserving mechanism impacts the performance of our *DP-based* approach. The second approach is based on best-first forward search, denoted as *Best-first*, and has been used in [14], [7], [12]. In the *Best-first* approach, when an agent transmits a package/packet to a logistic center/access point, the agent broadcasts this state to all the other agents. The nearest agent takes the package/packet based on this state and transmits it to the next logistic center/access point. This process continues until the goal agent is reached. The third approach is GPPP (greedy privacy-preserving planner), denoted as *Greedy*, which was developed in [13]. The *Greedy* approach consists of two phases: global planning and local planning. In the global planning phase, all agents collaboratively devise a global plan using a best-first search method. Next, in the local planning phase, each agent creates a local plan by executing a single-agent planning procedure.

8.2 Experimental results

8.2.1 The logistics scenario



(a) Average route length (b) Average communication overhead (c) Success rate

Fig. 7. Performance of the four approaches on the logistics scenario with variation of the map size

Fig. 7 demonstrates the performance of the four approaches on the logistics scenario with variation of the map size. As the map size grows larger, for all four approaches, the average route length and the average communication overhead progressively increase, while the success rate gradually decreases.

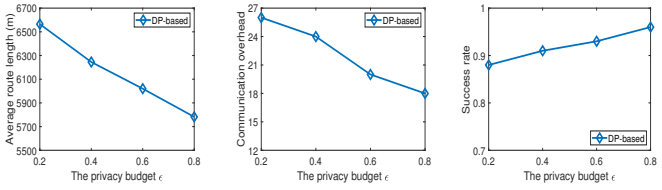
As the map size increases, the distance between an original agent and a destination agent may be enlarged accordingly. Therefore, the average route length increases. Moreover, when this occurs, the number of intermediate agents also increases. Thus, the average communication overhead rises as well. Due to this increase in the average communication overhead, the communication budget of

2. The topologies of maps/networks are created by simulation, as most real-world graph datasets [46], [47], [48] do not contain distance information and thus cannot be used in our experiments. We leave the experiments with real-world datasets as one of our future studies.

some agents may be used up before a plan is made. Hence, the success rate reduces.

The proposed *DP-based* approach achieves much better performance than the *Best-first* and *Greedy* approaches. The reinforcement learning algorithm in the *DP-based* approach can find shorter routes than the other two approaches. Moreover, in the *DP-based* approach, agents are allowed to communicate only with neighbors, and a privacy budget is adopted to control communication overhead. Thus, the *DP-based* approach uses less communication overhead than the other two approaches. In addition, the *DP-based* approach successfully makes more plans than the other two approaches before the communication budget is used up. Overall, the performance of *No-privacy* approach is slightly better than the *DP-based* approach. As privacy is not taken into account in the *No-privacy* approach, the information shared between agents is accurate, and agents can make accurate plans based on this accurate information. However, the private information of each agent is entirely disclosed to other agents under this approach, a situation that should be avoided in real-world applications. More specifically, in Fig. 7(a), the average route length in the *DP-based* approach is only about 2% longer than for the *No-privacy* approach. This is because in the *DP-based* approach, a plan is made up of a set of local plans created by the initial agent and the intermediate agents. Each of these local plans is created by an individual agent with reference to its private but accurate information. Since most of the information used to create a plan is accurate, the introduction of our privacy-preserving mechanism does not substantially impact the average route length.

The *Best-first* approach achieves the worst performance out of the four approaches. In the *Best-first* approach, a package is transmitted to the nearest agent. However, in large and complex maps, the nearest agent may not always be the best choice. Moreover, always choosing the nearest agent may result in a transmission loop; if this situation arises, packages will never reach their destinations. In comparison, the performance of *Greedy* approach is better than the *Best-first* approach, as the *Greedy* approach features a global planning phase that involves selecting the appropriate logistic centers to create a high-level route, which conserves communication overhead.

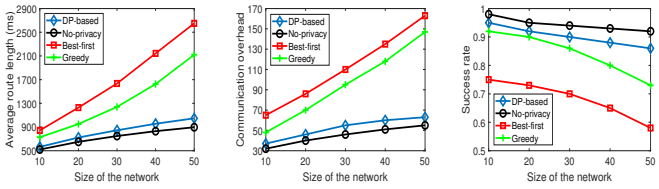


(a) Average route length (b) Average communication overhead (c) Success rate

Fig. 8. Performance of the *DP-based* approach on the logistics scenario with variation of the privacy budget value

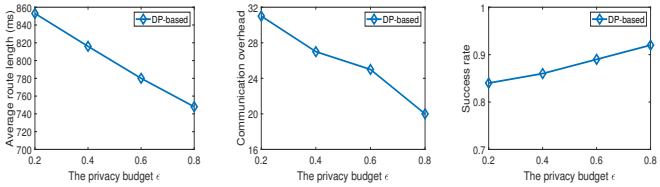
Fig. 8 demonstrates the performance of the *DP-based* approach on the logistics scenario with variation of the privacy budget ϵ value from 0.2 to 0.8. The number of logistic centers is fixed at 10. It can be seen that with the increase of the privacy budget ϵ value, the performance of the *DP-based* approach improves, namely it achieves a shorter average route length (Fig. 8(a)), lower average communication overhead (Fig. 8(b)), and higher success rate (Fig. 8(c)). According to the Laplace mechanism, when the ϵ value is small, the noise, added to the map, is large. A large noise value will significantly affect the agents planning. For example, agent a has two neighbors b and c. Now, suppose that 1) agent a wants to send a package to d, and 2) delegating the package to b is a better choice than c. However, when agents b and c obfuscate their maps, due to the large noise, the obfuscation results may make c appear to be a better choice than b. Thus, agent a may make a sub-optimal plan. This situation is alleviated when the ϵ value increases.

8.2.2 The packet routing scenario



(a) Average route length (b) Average communication overhead (c) Success rate

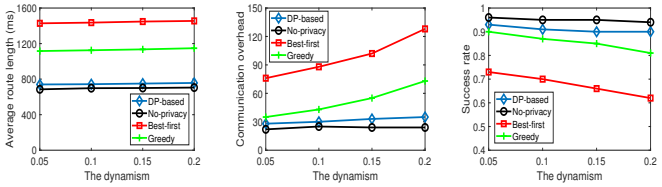
Fig. 9. Performance of the four approaches on the packet routing scenario with variation of the network size



(a) Average route length (b) Average communication overhead (c) Success rate

Fig. 10. Performance of the *DP-based* approach on the packet routing scenario with variation of the privacy budget

Fig. 9 illustrates the performance of the four approaches on the packet routing scenario with variation of the network size, while Fig. 10 depicts the performance of the *DP-based* approach on the packet routing scenario with variation of the privacy budget ϵ. After comparing Fig. 7 to Fig. 9 and Fig. 8 to Fig. 10, it can be concluded that these approaches exhibit similar trends in terms of their results on the two scenarios, but that the performance of these approaches is worse on the packet routing scenario than on the logistic scenario. This is mainly due to the dynamism of the packet routing scenario. When a node leaves the network, the routes involving that node are broken. Thus, agents have to re-find routes. This incurs extra communication overhead and reduces success rates to some extent.



(a) Average route length (b) Average communication overhead (c) Success rate

Fig. 11. Performance of the three approaches on the packet routing scenario with variation of the dynamism

Fig. 11 illustrates the performance of the four approaches on the packet routing scenario with variation of the dynamism, such that the probability of a node leaving or joining the network varies from 0.05 to 0.2 and the network size is fixed at 10 access points. From Fig. 11, it can be seen that an increase in the dynamism negatively affects the *Best-first* and *Greedy* approaches in terms of their average communication overhead and success rates, but does not significantly impact the *DP-based* and *No-privacy* approaches.

As the dynamism increases, the frequency with which nodes leave or join the network also increases. Thus, the number of affected routes increases as well. In the *Best-first* and *Greedy* approaches, when a route is broken, a new finding process is launched. This may not significantly affect the average route length (Fig. 11(a)), as route length depends on the positions of nodes rather than the number of nodes. However, launching a new finding process results in additional communication overhead, and may thus reduce success rates due to depletion of the communication budget. By contrast, the *DP-based* and *No-privacy* approaches do not require a new finding process when a route is broken. In the *DP-based* approach, routes are found by using reinforcement learning on obfuscated local network topologies. These obfuscated local network topologies are obtained using differential privacy. Differential privacy can guarantee that a node being brought in or out of a local network will have minimum effect on the statistical information. Therefore, when a node leaves or joins a local network, the serving access point does not need to re-obfuscate the new network or to communicate with the original access point about the change in the network. Hence, the communication budget can be conserved, and the success rate is preserved.

8.3 Summary

According to the experimental results, the proposed *DP-based* approach achieves better results than the *Best-first* and *Greedy* approaches in all experimental situations considered here. The average length of routes found by the *DP-based* approach is about 25% and 15% shorter, respectively, than those found using the *Best-first* and *Greedy* approaches. The *DP-based* approach also uses about 20% and 10% less communication overhead than the *Best-first* and *Greedy* approaches, respectively. Moreover, the *DP-based* approach achieves about 10% and 5% higher success rates than the *Best-first* and *Greedy* approaches, respectively.

Regarding performance, the *DP-based* approach is slightly worse than the *No-privacy* approach by a factor of about 3% in terms of average route length, 2% in communication overhead and 2% in success rate. The *DP-based* approach, however, strongly protects the privacy of agents, which is entirely disregarded in the *No-privacy* approach. Therefore, based on the experimental results, the efficiency of our *DP-based* approach can be proven.

9 Conclusion and Future Work

This paper proposes a novel strong privacy-preserving planning approach for logistic-like problems. In this approach, an agent creates a complete plan by using obfuscated private information from each intermediate agent, where this obfuscation is achieved by adopting the differential privacy technique. Due to the advantages of differential privacy, following obfuscation, an agent's private information cannot be deduced by other agents regardless of their reasoning power. This approach is the first in existence to achieve strong privacy, completeness and efficiency simultaneously by taking advantage of differential privacy. Moreover, this approach is communication-efficient. Compared to the benchmark approaches, our approach achieves better performance in various aspects.

In the future, we intend to extend our approach by introducing malicious agents. Existing approaches commonly assume that agents are honest but curious. Introducing malicious agents, which provide false information to others, may be a challenging and interesting addition to the field of multi-agent planning. Also, as described in the experimental part (Section 8), we will continue to search usable real-world datasets and evaluate our approach with them.

References

[1] M. E. desJardins, E. H. Durfee, C. L. Ortiz, and M. J. Wolverton, "A Survey of Research in Distributiond Continual Planning," *AI Magazine*, vol. 20, pp. 13–22, 1999.

[2] D. Ye, M. Zhang, and A. V. Vasilakos, "A Survey of Self-organisation Mechanisms in Multi-Agent Systems," *IEEE Trans. on Syst., Man and Cyber.: Syst.*, vol. 47, no. 3, pp. 441–461, 2017.

[3] A. Torreno, E. Onaindia, A. Komenda, and M. Stolba, "Cooperative Multi-Agent Planning: A Survey," *ACM Computing Surveys*, vol. 50, no. 6, pp. 84:1–84:32, 2017.

[4] G. Shani, "Advances and Challenges in Privacy Preserving Planning," in *IJCAI*, Stockholm, Sweden, 2018, pp. 5719–5723.

[5] S. Shekhar and R. I. Brafman, "Representing and Planning with Interacting Actions and Privacy," *Artificial Intelligence*, vol. 278, pp. 103 200:1–19, 2020.

[6] J. Tozicka, M. Stolba, and A. Komenda, "The Limits of Strong Privacy Preserving Multi-Agent Planning," in *ICAPS*, 2017, pp. 297–305.

[7] R. I. Brafman, "A Privacy Preserving Algorithm for Multi-Agent Planning and Search," in *IJCAI*, 2015, pp. 1530–1536.

[8] C. Dwork, "DIfferential Privacy," in *Proc. of ICALP*, Venice, Italy, 2006, pp. 1–12.

[9] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.

[10] A. Torreno, E. Onaindia, and O. Sapena, "FMAP: Distributed Cooperative Multi-Agent Planning," *Applied Intelligence*, vol. 41, pp. 606–626, 2014.

[11] A. Torreno, E. Onaindia, and O. Sapena, "Global Heuristics for Distributed Cooperative Multi-Agent Planning," in *ICAPS*, 2015, pp. 225–233.

[12] M. Stolba and A. Komenda, "The MADLA Planner: Multi-Agent Planning by Combination of Distributed and Local Heuristic Search," *Artificial Intelligence*, vol. 252, pp. 175–210, 2017.

[13] S. Maliah, G. Shani, and R. Stern, "Collaborative Privacy Preserving Multi-Agent Planning," *Autonomous Agents and Multi-Agent Systems*, vol. 31, pp. 493–530, 2017.
[14] R. Nissim and R. I. Brafman, "Distributed Heuristic Forward Search for Multi-Agent Planning," *Journal of AI Research*, vol. 51, pp. 292–332, 2014.
[15] M. Stolba, J. Tozicka, and A. Komenda, "Secure Multi-Agent Planning Algorithms," in *ECAI*, 2016, pp. 1714–1715.
[16] M. Stolba, J. Tozicka, and A. Komenda, "Secure Multi-Agent Planning," in *PrAISe*, 2016, p. Article No. 11.
[17] M. Stolba, J. Tozicka, and A. Komenda, "Quantifying Privacy Leakage in Multi-Agent Planning," *ACM Transactions on Internet Technology*, vol. 18, no. 3, pp. 28:1–28:21, 2018.
[18] C. Braun, K. Chatzikokolakis, and C. Palamidessi, "Quantitative Notions of Leakage for One-try Attacks," *Electrical Notes in Theoretical Computer Science*, vol. 249, pp. 75–91, 2009.
[19] G. Smith, "On the Foundations of Quantitative Information Flow," in *The 12th International Conference on Foundations of Software Science and Computational Structures*, York, UK, 2009, pp. 288–302.
[20] F. Fioretto, T. Mak, and P. V. Hentenryck, "Privacy-Preserving Obfuscation of Critical Infrastructure Networks," in *Proc. of IJCAI*, 2019, pp. 1086–1092.
[21] W. Yeoh and M. Yokoo, "Distributed Problem Solving," *Artificial Intelligence Magazine*, vol. 33, no. 3, pp. 53–65, 2012.
[22] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing Graphs with Node Differential Privacy," in *Proc. of TCC*, 2013, pp. 457–476.
[23] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially Private Data Analysis of Social Networks via Restricted Sensitivity," in *Proc. of ITCS*, 2013, pp. 87–96.
[24] D. Proserpio, S. Goldberg, and F. McSherry, "Calibrating Data to Sensitivity in Private Data Analysis," in *Proc. of VLDB*, 2014, pp. 637–648.
[25] F. Fioretto, E. Pontelli, and W. Yeoh, "Distributed Constraint Optimization Problems and Applications: A Survey," *Journal of Artificial Intelligence Research*, vol. 61, pp. 623–698, 2018.
[26] T. Grinshpoun and T. Tassa, "A Privacy-Preserving Algorithm for Distributed Constraint Optimization," in *AAMAS*, 2014, pp. 909–916.
[27] T. Tassa, T. Grinshpoun, and A. Yanai, "A Privacy Preserving Collusion Secure DCOP Algorithm," in *IJCAI*, 2019, pp. 4774–4780.
[28] A. Ben-Efraim and E. Omri, "Concrete Efficiency Improvements for Multiparty Garbling with an Honest Majority," in *Proc. of LATINCRYPT*, 2017.
[29] C. Niu, Z. Zheng, F. Wu, S. Tang, X. Gao, and G. Chen, "Unlocking the Value of Privacy: Trading Aggregate Statistics over Private Correlated Data," in *SIGKDD*, London, UK, 2018, pp. 2031–2040.
[30] R. I. Brafman and C. Domshlak, "From One to Many: Planning for Loosely Coupled Multi-Agent Systems," in *Proc. of the 18th International Conference on Autonomated Planning and Scheduling (ICAPS)*, Sydney, Australia, 2008, pp. 28–35.
[31] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C. Gao, H. Li, and Y. Tan, "Secure Multi-Party Computation: Theory, Practice and Applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.
[32] T. Zhu, G. Li, P. Xiong, and W. Zhou, "Answering differentially private queries for continual datasets release," *Future Generation Computer Systems*, vol. 87, pp. 816–827, 2018.
[33] D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Differentially Private Malicious Agent Avoidance in Multiagent Advising Learning," *IEEE Trans. on Cyber.*, p. DOI: 10.1109/TCYB.2019.2906574, 2019.
[34] T. Zhu and P. S. Yu, "Applying Differential Privacy Mechanism in Artificial Intelligence," in *Proc. of the IEEE 39th Interna. Conf. on Distributed Computing Systems*, 2019, pp. 1601–1609.
[35] T. Zhu, P. Xiong, G. Li, W. Zhou, and P. S. Yu, "Differentially private model publishing in cyber physical systems," *Future Generation Computer Systems*, 2018.
[36] T. Zhang, T. Zhu, P. Xiong, H. Huo, Z. Tari, and W. Zhou, "Correlated Differential Privacy: Feature Selection in Machine Learning," *IEEE Trans. on Indust. Inform.*, vol. 16, no. 3, pp. 2115–2124, 2020.
[37] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.
[38] S. Raskhodnikova and A. Smith, *Differentially Private Analysis of Graphs*. Springer, 2015, pp. 543–547.
[39] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, "Systematic Topology Analysis and Generation Using Degree Correlations," in *Proc. of SIGCOMM*, Pisa, Italy, 2006, pp. 135–146.
[40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, pp. 269–271, 1959.
[41] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 12:1–19, 2019.
[42] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 94–103. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2007.41
[43] Z. Peng, D. Cui, J. Zuo, Q. Li, B. Xu, and W. Lin, "Random task scheduling scheme based on reinforcement learning in cloud computing," *Cluster Computing*, vol. 18, no. 4, pp. 1595–1607, 2015.
[44] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Trans. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.
[45] J. Kober, J. A. D. Bagnell, and J. Peters, "Reinforcement Learning in Robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
[46] [Online]. Available: snap.stanford.edu/data/roadNet-CA.html
[47] [Online]. Available: snap.stanford.edu/data/roadNet-PA.html
[48] [Online]. Available: snap.stanford.edu/data/roadNet-TX.html

Dayong Ye received his MSc and PhD degrees both from University of Wollongong, Australia, in 2009 and 2013, respectively. Now, he is a research fellow of Cyber-security at University of Technology, Sydney, Australia. His research interests focus on differential privacy, privacy preserving, and multi-agent systems.

Tianqing Zhu received the BEng and MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and the PhD degree in computer science from Deakin University, Australia, in 2014. She is currently an associate professor at the School of Computer Science in University of Technology Sydney, Australia. Her research interests include privacy preserving and network security.

Sheng Shen is pursuing his PhD at the School of Computer Science, University of Technology Sydney. His current research interests include data privacy preserving, differential privacy and federated learning.

Prof. Wanlei Zhou received the PhD degree from The Australian National University, Canberra, Australia in 1991 in Computer Science and Engineering. He is currently the Head of School of School of Computer Science in University of Technology Sydney, Australia. His research interests include distributed systems, network security, and privacy preserving.

Prof. Philip S. Yu received the Ph.D. degree in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago. His research interest is on big data, including data mining, data stream, database and privacy.