Location Privacy Preserving Cross-Layer Optimization in Multi-Hop Wireless Networks

***Abstract*—Integrating various protocol layers into a coherent framework, cross-layer optimization has been extensively studied to build a unified foundation for analyzing resource allocation problems in multi-hop wireless networks. Yet, the problem constraints and solutions reveal a lot about user/node locations. Protecting location privacy during cross-layer optimization is critical but has hardly been studied. In this paper, we propose a privacy-preserving scheme that leverages distributed computing to solve cross-layer optimization in multi-hop wireless networks. Inspired by Dantzig-Wolfe decomposition, each node locally constructs and solves its own subproblem based on local transmission/interference relations with its neighbors, but still enables solving the optimization problems. Our theoretical analysis and simulation validated the effectiveness and efficiency of our algorithms.**

***Index Terms*—Location privacy, cross-layer optimization, multi-hop wireless network, Dantzig-Wolfe decomposition**

I. Introduction

Pioneered by Alicherry [2], Chiang [9], Low [24], and many others, cross-layer optimization in multi-hop wireless networks [33], [21], [8], [18], [22], [14], [12] has been extensively studied for more than a decade. By integrating various protocol layers into a coherent framework, cross-layer optimization builds a unified foundation for the analysis of resource allocation problems. Typically, cross-layer design in the context of multi-hop wireless networks targets at multiple optimization objectives, including maximizing network throughput [2], [33], [21], minimizing the energy consumption [8], [9], [18], or generally, maximizing system utility [22], [14], [12].

Traditionally, to effectively carry out cross-layer optimization, a controller in a multi-hop wireless network (e.g., gateway in a wireless sensor network [1], [4], [3], access router in a wireless mesh network [5], [7], and secondary service provider in multi-hop cognitive radio networks [27], [14], [12]) has to collect, but not limited to, transmission/interference relations among all nodes for the link scheduling purpose. This also means the controller can utilize these relations and infer node locations easily. Moreover, some part of a cross-layer optimization solution, such as link scheduling and routing policies, can also disclose node locations. Section II-B will discuss these issues in more details.

Protecting node location privacy is a critical issue during cross-layer optimization in wireless networks. Yet, it has been neglected so far. Consider a coalition of wireless sensor network for battlefield surveillance [6]. Different countries, say, US and UK, from the same military alliance deploy the sensors to form a connected sensor network for fully covering a geographic area. But US may be reluctant to share locations of its sensors to UK, because revealing their locations makes it easy for any one to physically sabotage the sensors. As another example, consider a civil multi-hop wireless network, say a multi-hop cognitive radio network. Any secondary service provider who knows the secondary user locations can leverage this information to push location-based mobile advertisements for extra profit [16], [29]. In addition, any adversary which compromised the service provider will gain the full knowledge of user locations, and can easily infer commute routes, residence, habits, and preferences based on the location record of a user.

A question then arises. *Is it possible for the controller to correctly solve cross-layer optimization problems in multi-hop wireless networks, while wireless node locations are concealed before and after optimization?* In general, solving a cross-layer optimization problem without privacy is already time-consuming. Thus any proposed privacy-preserving scheme should not introduce extra heavy computation overhead.

For a typical cross-layer optimization problem, each node possesses part of the problem constraints and solutions, which contain rich information of its location. To protect location privacy, we aim to have all participants, including the controller and wireless nodes, jointly solve the cross-layer optimization problem, yet without inferring sensitive information regarding others' constraints and solutions. In this paper, we focus on dealing with a general cross-layer throughput maximization problem, which can be conventionally formulated as a mixed integer linear programming (MILP) problem.

There have been some existing works on privacy-preserving linear programming (LP). Some rely on interactive and relatively heavyweight cryptographic protocols [17], [30], [28]. They may be time-efficient for small-scale LP problems, but do not scale well for large-scale optimization problems or problems with combinational variables, which is the case for cross-layer optimization problems in wireless networks. Some rely on linear transformations to convert the original feasible region into another [13], [31]. This may work for some LP problems with variables ranging in $[\mathbf{0}, +\infty]$. However, cross-layer optimization problems in general take values from a known confined range for some variables. For example, some scheduling decision variables take binary values indicating if they are active in data transmission. We will illustrate in Section III-B that the privacy goal cannot be fulfilled by solely relying on existing linear transformation for such problems. Besides, solving MILP problems (let alone with privacy preserved) is more sophisticated than solving LP problems.

To achieve both privacy and computation efficiency, we develop a privacy-preserving scheme which heuristically solves the cross-layer optimization problems while protecting solution and constraint privacy from the controller. Rather than handling MILP problems directly, the algorithm first relaxes it into a set of LP problems. A privacy-preserving LP algorithm is first developed, inspired by the distributed computing property of Dantzig-Wolfe (D-W) decomposition method. D-W decomposes a large-scale LP problem into a set of separate small-scale subproblems. Each node locally constructs and solves its own subproblem with the knowledge of local transmission/interference relations. In such a way, the sensitive information regarding nodes' locations no longer need to report to the controller. Moreover, based on the intermediate results reported from nodes, the controller can still solve relaxed LP problems. Therefore, the controller does not have to acquire any node's location to formulate and solve the cross-layer optimization problem. To the best of our knowledge, this is the first work to study protecting node location privacy during cross-layer optimization in wireless networks.

The rest of this paper is organized as follows. Section II presents the problem statement. We describe our proposed privacy-preserving scheme in Section III. After that we conduct analysis on privacy and computational complexity in Section IV. Simulation results are discussed in Section V. Before concluding our paper, we give related works in Section VI.

II. Problem Statement

A. Background

This work pertains to standard multi-hop wireless networks where a controller is in charge of transmission resource allocation based on the collected information regarding node transmission/interference relations, spectrum availability, channel status, etc. Suppose the network consists of $\mathcal{N} = \{1, \cdots, N\}$ nodes and a set of available spectrums $\mathcal{M} = \{1, \cdots, M\}$. There are also a set of $\mathcal{L} = \{1, \cdots, L\}$ multi-hop data sessions. We let s_l, d_l, and r_l be the source node, destination node, and traffic amount of session l, respectively. d_l could be multiple hops away from s_l. To wisely utilize transmission resources to carry out these data sessions, without loss of generality, we focus on a conventional throughput maximization problem. We summarize the formulations from [33], [21], [8], [18], [22], [14], [12] and construct a cross-layer optimization problem P_0. Its objective is to maximize $\sum_{l\in\mathcal{L}} r_l$, i.e., finding the maximal supportable traffic amount for all multi-hop sessions, with its constraints from both link and routing layers.

Maximize: $$\sum_{l\in\mathcal{L}} r_l$$

Subject to:

$$\sum_{\{j|i\in\mathcal{T}_j\}} s^m_{ji} + \sum_{q\in\mathcal{T}_i} s^m_{iq} \le 1 \qquad (1)$$

$$\sum_{\{p|q\in\mathcal{T}_p\}} s^m_{pq} + \sum_{j\in\mathcal{T}_i} s^m_{ij} \le 1 \quad (q\in\mathcal{I}_i) \qquad (2)$$

$$\sum_{l\in\mathcal{L}} f^l_{ij} = \sum_{m\in\mathcal{M}} 1\cdot s^m_{ij} \qquad (3)$$

$$\sum_{j\in\mathcal{N}} f^l_{js_l} = 0 \quad \text{and} \quad \sum_{j\in\mathcal{N}} f^l_{jd_l} = r_l \qquad (4)$$

$$\sum_{j\in\mathcal{N}} f^l_{d_lj} = 0 \qquad (5)$$

$$\sum_{j\in\mathcal{N}} f^l_{s_lj} = r_l \quad \text{and} \quad \sum_{j\in\mathcal{N}} f^l_{ij} = \sum_{p\in\mathcal{N}} f^l_{pi} \qquad (6)$$

$$s^m_{ij} \in \{0,1\}, \quad f^l_{ij} \ge 0, \quad r_l \ge 0$$

$$i,j \in [1,N], \quad m\in[1,M], \quad l\in[1,L].$$

Specifically, (1) and (2) are constraints from the link layer. Here, we adopt the widely used *protocol model*[1] to characterize interference relations among wireless nodes. A data transmission is successful only if the receiver is within the transmission range of its sender and out of interference ranges of all its neighbors. s^m_{ij} is a binary variable with $m\in\mathcal{M}$, $i,j\in\mathcal{N}$. We have $s^m_{ij} = 1$, if i transmits to j on spectrum band m. (1) requires that a node cannot use the same spectrum band for transmission and reception due to the "self-interference" constraint. $\mathcal{T}_i$ stands for the set of nodes which are within the transmission range of i. (2) is the constraint restricting interference among different links. $\mathcal{I}_i$ stands for the set of nodes which are within the interference range of i. (3) is the data rate constraint, where we consider unit data rate over any active link. (4) means that the incoming data at the source s_l and destination d_l is 0 and r_l, respectively. (5) means that the outgoing data rate at d_l is 0. The first part of (6) requires that the outgoing data rate at s_l is r_l. The second part says that the total incoming data of an intermediate node i for a specific session l should be equal to its total outgoing data.

Once the controller formulates P_0, it can apply its favorite algorithms to solve it optimally or heuristically. The solutions of s^m_{ij}'s, f^l_{ij}'s, and r_l's are then sent to nodes for transmission guidance purposes. We remark that the formulation of P_0 is obtained from existing research on cross-layer optimization in multi-hop wireless networks and is not our contribution.

B. Problem Statement

To formulate P_0, or even other general cross-layer optimization problems in multi-hop wireless networks, the controller has to be aware of transmission/interference relations among wireless nodes, which directly depend on the node locations.

Take constraint (2) as an illustration. For a specific node i, the controller needs to know all i's 1-hop ($j\in\mathcal{T}_i$, $q\in\mathcal{I}_i$) and 2-hop neighbors ($p|q\in\mathcal{T}_p$) to establish i's interference constraint, i.e., determine coefficients of variables s^m_{pq} and s^m_{ij} (to be either 0 or 1). In other words, if the coefficient of variable s^m_{pq} in (2) is 1, one can infer that 1) q is within p's transmission range; and 2) q is within i's interference range. We have a similar observation over s^m_{ij}'s coefficient in (1). We also notice that if the final solution contains $s^m_{ij} = 1$ or $f^l_{ij} > 0$, it implies that j is within the transmission range of i. With more information of such kinds, the controller can gradually narrow down and easily locate any node. Therefore, *coefficients of variables (s^m_{ij}'s in (1), (2) and (3) here) and their solutions (s^m_{ij}'s and f^l_{ij}'s here) can reveal network topology to the controller during optimization.*

[1] There are two existing models to characterize interference relations among wireless nodes, *protocol model*, adopted in this work, and *physical model*.

We target at protecting location privacy for nodes in multi-hop wireless networks, especially the ones that do not connect to the controller directly. For single-hop wireless networks, such as regular cellular networks, a node communicates with the base station/ controller in one hop. Hence, the controller can always locate its subscriber by analyzing the received signal strength during communications.

We summarize the design objectives below:

- We aim to protect node location privacy from the controller. As discussed, the controller can infer the node location while performing a cross-layer optimization.
- Second, a node cannot infer the location of any others except its 1- or 2-hop neighbors. This is because one can always analyze the overheard signal strength and estimate their locations when they are transmitting data.
- The formulated optimization problem is a mixed integer linear programming (MILP) problem, solving which is NP-complete [11]. Therefore, we need develop a scheme that can solve the problem efficiently while achieving the aforementioned privacy protection goals.

III. Our Proposed Scheme

A. An Observation

We first define decision variable vectors $\mathbf{s}_i$'s and $\mathbf{f}_i$'s as $\mathbf{s}_i = \{s_{ij}^m | 1 \leq j \leq N, 1 \leq m \leq M\}$ and $\mathbf{f}_i = \{f_{ij}^l | 1 \leq j \leq N, 1 \leq l \leq L\}$. $\mathbf{s}_i$ and $\mathbf{f}_i$ are considered "owned" by node i. P_0 can be abstracted as P_1

$$
\begin{aligned}
\textbf{Maximize:} \quad & \mathbf{1}^\top \mathbf{r} \\
\textbf{Subject to:} \quad & \mathbf{D}_1\mathbf{s}_1 + \ldots + \mathbf{D}_N\mathbf{s}_N + \\
& \mathbf{E}_1\mathbf{f}_1 + \ldots + \mathbf{E}_N\mathbf{f}_N + \mathbf{H}\mathbf{r} \lesseqgtr \mathbf{b}_0 \qquad (7) \\
& \mathbf{F}_1\mathbf{s}_1 + \mathbf{G}_1\mathbf{f}_1 \lesseqgtr \mathbf{b}_1 \\
& \qquad \vdots \\
& \mathbf{F}_N\mathbf{s}_N + \mathbf{G}_N\mathbf{f}_N \lesseqgtr \mathbf{b}_N \\
& \mathbf{r} \in \mathbb{R}_{\geq 0}^{L\times 1}, \mathbf{s}_i \in \{0,1\}^{NM\times 1}, \mathbf{f}_i \in \mathbb{R}_{\geq 0}^{NL\times 1} \\
& \forall i \in [1, N]
\end{aligned}
$$

where (7) is the general form of constraints (1), (2), (4), and (6), and i's variables $\mathbf{s}_i$ and $\mathbf{f}_i$ couple with those from the rest nodes and $\mathbf{r}$. Meanwhile, $\mathbf{F}_i\mathbf{s}_i + \mathbf{G}_i\mathbf{f}_i \lesseqgtr \mathbf{b}_i$ $(1 \leq i \leq N)$ are abstracted from (3) and (5) where $\mathbf{s}_i$ and $\mathbf{f}_i$ are independent from the rest variables. Note that P_1 has a typical structure of general cross-layer optimization problems, with a number of global constraints and local constraints from each node.

Observing constraint $\mathbf{F}_i\mathbf{s}_i + \mathbf{G}_i\mathbf{f}_i \lesseqgtr \mathbf{b}_i$ in P_1, its coefficients $\mathbf{F}_i$ and $\mathbf{G}_i$ can be readily derived at node i with local information. Take (3) as an example. Since i has full knowledge of its 1-hop neighbors, it puts the coefficient of s_{ij}^m $(j \in \mathcal{T}_i)$ as 1 and that of s_{ij}^m $(j \notin \mathcal{T}_i)$ as 0, that is, $\mathbf{F}_i$ is available at i. This is the same case with $\mathbf{G}_i$. Moreover, i is capable of deriving $\mathbf{D}_i$ and $\mathbf{E}_i$ in (7) as well. Take the relatively complicated constraint (2) as an illustration. Since i is aware of all its 2-hop neighbors, it notifies all p's to set the coefficients of their decision variables s_{pq}^m's, which satisfy $q \in \mathcal{I}_i$, to "1" in this constraint[2]. In such a way, $\mathbf{D}_p$ is available at node p. Besides, as explained above, i sets the coefficient of s_{ij}^m $(j \in \mathcal{T}_i)$ as 1 and that of s_{ij}^m $(j \notin \mathcal{T}_i)$ as 0, according to its 1-hop neighbor information.

To sum up, each node i is able to obtain its coefficients $\mathbf{D}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$, and $\mathbf{G}_i$ in P_1 just based on local information. This observation will play an important role in our privacy-preserving scheme design, which will be made clear soon.

B. A Basic Technique

As discussed, each node i possesses part of the constraint coefficients ($\mathbf{D}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$, and $\mathbf{G}_i$) of P_1. Among these data, coefficients $\mathbf{D}_i$ and $\mathbf{F}_i$ together with solutions of variables $\mathbf{s}_i$ and $\mathbf{f}_i$, can reveal node locations. Hence, to perform cross-layer optimization without compromising node location privacy, we propose to have all participants, including the controller and nodes, jointly solve P_1 without inferring others' $\mathbf{D}_i$'s, $\mathbf{F}_i$'s, $\mathbf{s}_i$'s, and $\mathbf{f}_i$'s.

For this purpose, we first apply the affine mapping [13], [31] to P_1. Ideally, if we can arbitrarily transform the feasible region of P_1 from one vector space to another and keep mapping parameters as secret keys, there is no way for the controller to learn the exact original feasible area information and output, including $\{\mathbf{D}_i\}$, $\{\mathbf{F}_i\}$, $\{\mathbf{s}_i\}$, and $\{\mathbf{f}_i\}$. Regarding $\mathbf{H}$, it is the coefficient matrix for vector $\mathbf{r}$ (traffic amounts). Since it does not contain any information regarding node locations, it is public information. This is the similar case for $\{\mathbf{E}_i\}$, $\{\mathbf{G}_i\}$ and $\{\mathbf{b}_i\}$.

Let $\boldsymbol{\alpha}_i$ and $\boldsymbol{\gamma}_i$ be random $MN \times MN$ non-singular matrices. Let $\boldsymbol{\beta}_i$ and $\boldsymbol{\eta}_i$ be random $MN \times 1$ vectors which are chosen by node i. The affine mapping defined by $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$ transforms $\mathbf{s}_i$ into $\bar{\mathbf{s}}_i = \boldsymbol{\alpha}_i^{-1}(\mathbf{s}_i + \boldsymbol{\beta}_i)$. Similarly, $\mathbf{f}_i$ is transformed into $\bar{\mathbf{f}}_i = \boldsymbol{\gamma}_i^{-1}(\mathbf{f}_i + \boldsymbol{\eta}_i)$. Since this is a one-to-one mapping, P_1 can be expressed as the following problem (P_2) with decision variables $\{\bar{\mathbf{s}}_i\}$, $\{\bar{\mathbf{f}}_i\}$, and $\mathbf{r}$,

$$
\begin{aligned}
\textbf{Maximize:} \quad & \mathbf{1}^\top \mathbf{r} \\
\textbf{Subject to:} \quad & \sum_{1\leq i\leq N} \Big(\mathbf{D}_i\boldsymbol{\alpha}_i\bar{\mathbf{s}}_i + \mathbf{E}_i\boldsymbol{\gamma}_i\bar{\mathbf{f}}_i\Big) + \mathbf{H}\mathbf{r} \lesseqgtr \bar{\mathbf{b}}_0 \\
& \mathbf{F}_i\boldsymbol{\alpha}_i\bar{\mathbf{s}}_i + \mathbf{G}_i\boldsymbol{\gamma}_i\bar{\mathbf{f}}_i \lesseqgtr \bar{\mathbf{b}}_i \\
& \bar{\mathbf{s}}_i \in \left\{\boldsymbol{\alpha}_i^{-1}\boldsymbol{\beta}_i, \boldsymbol{\alpha}_i^{-1}(\mathbf{1} + \boldsymbol{\beta}_i)\right\} \qquad (8) \\
& \bar{\mathbf{f}}_i \in \left[\boldsymbol{\gamma}_i^{-1}\boldsymbol{\eta}_i, +\infty\right] \qquad (9) \\
& \mathbf{r} \in \mathbb{R}_{\geq 0}^{L\times 1}, \quad \forall i \in [1, N]
\end{aligned}
$$

where $\bar{\mathbf{b}}_0 = \mathbf{b}_0 + \sum_{1\leq i\leq N} \big(\mathbf{D}_i\boldsymbol{\beta}_i + \mathbf{E}_i\boldsymbol{\eta}_i\big)$ and $\bar{\mathbf{b}}_i = \mathbf{b}_i + \mathbf{F}_i\boldsymbol{\beta}_i + \mathbf{G}_i\boldsymbol{\eta}_i$. Node i keeps $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\eta}_i$ as its secret key for affine mapping. It then uploads the masked coefficients, (8) and (9), to the controller. $\mathbf{D}_i\boldsymbol{\beta}_i + \mathbf{E}_i\boldsymbol{\eta}_i$ and $\mathbf{F}_i\boldsymbol{\beta}_i + \mathbf{G}_i\boldsymbol{\eta}_i$ are also updated to construct $\bar{\mathbf{b}}_0$ and $\bar{\mathbf{b}}_i$, respectively. Without i's secret key, it is impossible to derive the sensitive data $\mathbf{D}_i$ and $\mathbf{F}_i$. Similarly, once P_2 is solved, the solution of $\bar{\mathbf{s}}_i$ and $\bar{\mathbf{f}}_i$ cannot reveal that of $\mathbf{s}_i$ and $\mathbf{f}_i$, as they are also masked by i's secret key.

[2] This data transmission from i to p can be easily carried via local scheduling. Besides, it is only conducted once for the entire optimization and the payload amount is small, so communication overhead here is limited.

Nonetheless, node i's secret key is not protected well with this basic technique. The controller can formulate $\frac{\boldsymbol{\alpha}_i^{-1}\boldsymbol{\beta}_i}{\boldsymbol{\alpha}_i^{-1}(1+\boldsymbol{\beta}_i)} = \frac{\boldsymbol{\beta}_i}{1+\boldsymbol{\beta}_i}$ and obtain $\boldsymbol{\beta}_i$ directly. $\boldsymbol{\alpha}_i$ can also be derived. Actually, for optimization problems that involve variables taking values from a known confined range (e.g., $\mathbf{s}_i \in [\mathbf{0}, \mathbf{1}]$ in our problem), this kind of leakage exists in applying affine mapping for coefficient and/or solution privacy.

C. Enhanced Technique via Dantzig-Wolfe Decomposition

To address the privacy leakage issue just explained, our idea is to hide the constraint (8) from the controller. *A challenge then arises: without the knowledge of (8), can the controller still solve P$_2$?* Observing (8), it is available at each node. If we can develop a distributed algorithm that keeps (8) at each node locally but still enable the platform and all nodes to collaboratively solve P$_2$, we can then resolve the challenge.

For this purpose, we develop an enhanced technique that is inspired by Dantzig-Wolfe (D-W) decomposition, i.e., a large-scale LP problem can be divided into a set of separate small-scale subproblems. We use D-W to decompose P$_2$ into a set of subproblems. Each node locally constructs its own subproblem with its own (8) and solves it. With the intermediate results reported from the nodes, the controller can still solve P$_2$. In such a way, the controller does not have to acquire (8) in solving P$_2$. As long as the intermediate results do not reveal any information of $\{\boldsymbol{\alpha}_i\}$ and $\{\boldsymbol{\beta}_i\}$, the privacy issue in the basic technique no longer exists.

Yet, D-W decomposition can only be directly applied to LP problems, while P$_2$ is an MILP problem. In the following, we first develop a privacy-preserving LP algorithm based on D-W decomposition to solve a relaxed P$_2$. Based on that, our privacy-preserving scheme to solve P$_2$ can be readily derived.

1) Privacy-Preserving LP Algorithm: We first relax the 0-1 binary constraint over each $\bar{s}_{ij}^m$ to $0 \le \bar{s}_{ij}^m \le 1$ in P$_2$ and obtain P$_3$

$$\begin{aligned}
\textbf{Maximize:} \quad & \mathbf{1}^\top \mathbf{r} \\
\textbf{Subject to:} \quad & \sum_{1 \le i \le N} \left(\mathbf{D}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i + \mathbf{E}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i \right) + \mathbf{H}\mathbf{r} \lesseqqgtr \bar{\mathbf{b}}_0 \\
& \mathbf{F}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i + \mathbf{G}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i \lesseqqgtr \bar{\mathbf{b}}_i \qquad (10) \\
& \bar{\mathbf{s}}_i \in \left[\boldsymbol{\alpha}_i^{-1} \boldsymbol{\beta}_i, \boldsymbol{\alpha}_i^{-1}(\mathbf{1} + \boldsymbol{\beta}_i) \right] \qquad (11) \\
& \bar{\mathbf{f}}_i \in \left[\boldsymbol{\gamma}_i^{-1} \boldsymbol{\eta}_i, +\boldsymbol{\infty} \right] \qquad (12) \\
& \mathbf{r} \in \mathbb{R}_{\ge 0}^{L \times 1}, \quad \forall i \in [1, N].
\end{aligned}$$

Assume that its constraints define a bounded polyhedron[3]. According to Minkowski's representation theorem [26], a variable belong to a bounded polyhedron can be described by a convex combination of its extreme points. Let $\{\bar{\mathbf{s}}_i^j | j \in P_i\}$ be a complete set of extreme points of $\bar{\mathbf{s}}_i$, where P_i stands for their index set. $\bar{\mathbf{s}}_i^j$ itself is also a vector. It follows that $\bar{\mathbf{s}}_i = \sum_{j \in P_i} \lambda_i^j \bar{\mathbf{s}}_i^j$ where coefficients λ_i^j are nonnegative and satisfy $\sum_{j \in P_i} \lambda_i^j = 1$. Similarly, $\bar{\mathbf{f}}_i = \sum_{j \in P_i} \lambda_i^j \bar{\mathbf{f}}_i^j$, where $\{\bar{\mathbf{f}}_i^j | j \in P_i\}$ is a complete set of extreme points of $\bar{\mathbf{f}}_i$. P$_3$ is then equivalently reformulated as a *master problem* (MP)

$$\begin{aligned}
\textbf{Maximize:} \quad & \mathbf{1}^\top \mathbf{r} \\
\textbf{Subject to:} \quad & \sum_{1 \le i \le N} \Bigg(\mathbf{D}_i \boldsymbol{\alpha}_i \sum_{j \in P_i} \lambda_i^j \bar{\mathbf{s}}_i^j + \\
& \qquad \mathbf{E}_i \boldsymbol{\gamma}_i \sum_{j \in P_i} \lambda_i^j \bar{\mathbf{f}}_i^j \Bigg) + \mathbf{H}\mathbf{r} \lesseqqgtr \bar{\mathbf{b}}_0 \\
& \sum_{j \in P_i} \lambda_i^j = 1, \mathbf{r} \in \mathbb{R}_{\ge 0}^{L \times 1}, \lambda_i^j \ge 0, \forall i \in [1, N]
\end{aligned}$$

with decision variables $\{\lambda_i^j\}$ and $\mathbf{r}$. The MP does not specify constraints (10)-(12) as they are represented by $\{\bar{\mathbf{s}}_i^j\}$ and $\{\bar{\mathbf{f}}_i^j\}$.

The privacy-preserving LP algorithm starts by the controller constructing a *restricted master problem* (RMP). Compared with MP that includes complete extreme points, RMP only contains an initial set of extreme points $\bar{\mathbf{s}}_i^{j_0}$ and $\bar{\mathbf{f}}_i^{j_0}$. They satisfy *pricing problem*'s constraints that will be introduced later. To formulate the RMP, each node i sends to the controller coefficients $\mathbf{D}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i^{j_0}$ and $\mathbf{E}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i^{j_0}$, and $\mathbf{D}_i \boldsymbol{\beta}_i + \mathbf{E}_i \boldsymbol{\eta}_i$ (to construct $\bar{\boldsymbol{b}}_0$). Recall that $\mathbf{D}_i$, $\boldsymbol{\alpha}_i$, $\mathbf{E}_i$ and $\boldsymbol{\gamma}_i$ are all available at i. $\bar{\mathbf{s}}_i^{j_0}$ and $\bar{\mathbf{f}}_i^{j_0}$ are the solution of the *pricing problem* to be discussed soon.

Algorithm 1 Privacy-Preserving LP Algorithm

```
Output: $C^*$, $\bar{\mathbf{s}}_i^*$ and $\bar{\mathbf{f}}_i^*$
1:  Each node $i$ ($i \in [1, N]$) sends to controller $\mathbf{D}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i^{j_0}$, $\mathbf{E}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i^{j_0}$, and $\mathbf{D}_i \boldsymbol{\beta}_i + \mathbf{E}_i \boldsymbol{\eta}_i$;
2:  Controller formulates RMP with current extreme points, optimally solves it, and obtains $\mathbf{w}$, $\{z_i\}$, and $C^*$;
3:  Controller sends $\mathbf{w}$ and $z_i$ to each node $i$;
4:  for $1 \le i \le N$ do
5:      Node $i$ formulates PP, optimally solves it, and obtains $\bar{\mathbf{s}}_i^*$ and $\bar{\mathbf{f}}_i^*$;
6:      if $\phi^* > 0$ then
7:          Report to controller $\mathbf{D}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i^*$ and $\mathbf{E}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i^*$.
8:      else
9:          Report nothing;
10:     end if
11: end for
12: if $\{\mathbf{D}_i \boldsymbol{\alpha}_i \bar{\mathbf{s}}_i^*, \mathbf{E}_i \boldsymbol{\gamma}_i \bar{\mathbf{f}}_i^*\} \ne \emptyset$ then
13:     Go to line 2;
14: else
15:     Output $C^*$;
16: end if
```

Since RMP only contains a subset of complete extreme points, its optimal result serves as a lower bound of P$_3$'s optimal result. Follow the idea of D-W decomposition, once RMP is solved, we need to determine if any new extreme points can be added to further improve the result. In particular, each node i formulates a *pricing problem* (PP) examining if it produces a positive reduced cost[4]. According to the definition [10], the

[3] This assumption is not necessary in D-W decomposition. But it does hold in the LP problem formulated in this study and simplifies the mathematical description herein.

[4] If MP is a minimization problem, then a negative reduced cost indicates the result of RMP can be further decreased.

reduced cost of variable λ_i^j in our problem is computed as $-\mathbf{w}^\top(\mathbf{D}_i\boldsymbol{\alpha}_i\overline{\mathbf{s}}_i + \mathbf{E}_i\boldsymbol{\gamma}_i\overline{\mathbf{f}}_i) - z_i$, where $\mathbf{w}$ and z_i are the optimal dual solutions associated with RMP's constraints

$$\sum_{1\leq i\leq N}\left(\mathbf{D}_i\boldsymbol{\alpha}_i\sum_{j\in P_i'}\lambda_i^j\overline{\mathbf{s}}_i^j + \mathbf{E}_i\boldsymbol{\gamma}_i\sum_{j\in P_i'}\lambda_i^j\overline{\mathbf{f}}_i^j\right) + \mathbf{Hr} \lesseqqgtr \overline{\mathbf{b}}_0 \quad (13)$$

and

$$\sum_{j\in P_i'}\lambda_i^j = 1,$$

respectively. We denote by P_i' ($P_i' \subset P_i$) the index of the subset of complete extreme points. Each node i formulates its PP as

$$\begin{aligned} &\textbf{Maximize:} && \phi = -\mathbf{w}^\top\left(\mathbf{D}_i\boldsymbol{\alpha}_i\overline{\mathbf{s}}_i + \mathbf{E}_i\boldsymbol{\gamma}_i\overline{\mathbf{f}}_i\right) - z_i \\ &\textbf{Subject to:} && \mathbf{F}_i\boldsymbol{\alpha}_i\overline{\mathbf{s}}_i + \mathbf{G}_i\boldsymbol{\gamma}_i\overline{\mathbf{f}}_i \lesseqqgtr \overline{\mathbf{b}}_i \\ & && \overline{\mathbf{s}}_i \in \left[{\boldsymbol{\alpha}_i}^{-1}\boldsymbol{\beta}_i, {\boldsymbol{\alpha}_i}^{-1}(\mathbf{1}+\boldsymbol{\beta}_i)\right] \qquad (14) \\ & && \overline{\mathbf{f}}_i \in \left[\boldsymbol{\gamma}_i^{-1}\boldsymbol{\eta}_i, +\boldsymbol{\infty}\right] \end{aligned}$$

where $\overline{\mathbf{s}}_i$ and $\overline{\mathbf{f}}_i$ are decision variables. To formulate PP, $\mathbf{w}$ and z_i are received from the controller, and all the parameters, $\mathbf{D}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$, $\mathbf{G}_i$, $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$, $\boldsymbol{\gamma}_i$, $\boldsymbol{\eta}_i$, and $\overline{\mathbf{b}}_i$ are available at node i.

As PP is an LP problem, node i can optimally solve it without difficulty. The optimal solution $\overline{\mathbf{s}}_i^*$ and $\overline{\mathbf{f}}_i^*$ that produces the most positive reduced cost ϕ^* becomes the entering extreme point that will be added back to RMP to further improve its result. Specifically, node i sends $\mathbf{D}_i\boldsymbol{\alpha}_i\overline{\mathbf{s}}_i^* + \mathbf{E}_i\boldsymbol{\gamma}_i\overline{\mathbf{f}}_i^*$ to the controller. It serves as the coefficient of a new added variable λ_i^{j+1} to (13). In such a way, RMP is updated. The controller then solves the new RMP and obtains a new set of $\mathbf{w}$ and z_i which are sent to node i for constructing a new PP. We denote by C^* be the optimum result of RMP, which is calculated at the controller. The iteration continues until none of the nodes reports a positive reduced cost, i.e., the current C^* is already optimum to MP. There is still one remaining issue, i.e., how to obtain the initial set of extreme points $\overline{\mathbf{s}}_i^{j_0}$ and $\overline{\mathbf{f}}_i^{j_0}$ for RMP? For this purpose, node i just needs to find a solution of $\overline{\mathbf{s}}_i$ and $\overline{\mathbf{f}}_i$ that satisfy the constraints of PP. We summarize our proposed privacy-preserving LP algorithm for optimally solving P_3 in Algorithm 1.

Since P_3 is distributively computed with sensitive constraint (14) kept locally at each node, the controller cannot infer i's secret key $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$. Besides, i is capable of obtaining the optimum solution $\overline{\mathbf{s}}_i^* = \sum_{j\in P_i'}\lambda_i^j\overline{\mathbf{s}}_i^{j*}$ and $\overline{\mathbf{f}}_i^* = \sum_{j\in P_i'}\lambda_i^j\overline{\mathbf{f}}_i^{j*}$.

2) Our Final Scheme: We have developed a privacy-preserving LP algorithm to optimally solve P_3, an LP relaxation of P_2. Recall that our ultimate goal is to develop a privacy-preserving scheme to solve P_0, an MILP problem. However, the computation complexity for optimally solving an MILP problem is already NP-complete [11] even without considering privacy. Thus, it is desirable to devise a scheme to solve P_0 in polynomial time, in addition to preserve privacy.

Our main idea is to fix the values of $\overline{s}_{ij}^m$'s in P_2 sequentially through a series of relaxed LP problems, each of which is then solved by the privacy-preserving LP algorithm.

Specifically, as the first step, we relax the 0-1 binary constraint over each $\overline{s}_{ij}^m$ to $0 \leq \overline{s}_{ij}^m \leq 1$ and relax P_2 to P_3. Then, we apply the proposed privacy-preserving LP algorithm to optimally solve it. According to the discussion above, the optimal solution $\overline{\mathbf{s}}_i^*$ and $\overline{\mathbf{f}}_i^*$ of P_3 are available at node i, who then derives $\mathbf{s}_i^*$ and $\mathbf{f}_i^*$ by $\mathbf{s}_i^* = \boldsymbol{\alpha}_i\overline{\mathbf{s}}_i^* - \boldsymbol{\beta}_i$ and $\mathbf{f}_i^* = \boldsymbol{\gamma}_i\overline{\mathbf{f}}_i^* - \boldsymbol{\eta}_i$. Among $\mathbf{s}_i^*$, node i picks $\tilde{s}_{ij}^m = \max_{j,m}\{\mathbf{s}_i^*\}$, i.e., the maximal value from vector $\mathbf{s}_i^*$. $\tilde{s}_{ij}^m$ together with a one-time pseudonym pid_i are reported to the controller. Note that only the value of $\tilde{s}_{ij}^m$ is revealed, while the information of i, j, and m is hidden. The controller, who collects $\{\tilde{s}_{ij}^m, pid_i\}$ from all nodes, determines $\max_i\{\tilde{s}_{ij}^m\}$ and broadcasts its corresponding pid_i. If there is a tie, the controller randomly chooses one from them. Node i with $\max\{\tilde{s}_{ij}^m\}$ then sets its $\tilde{s}_{ij}^m$ as 1 and rest s_{iq}^m's ($q \in \mathcal{N}$, $q \neq j$, $m \in \mathcal{M}$) to 0's. Hence, i's link layer decision variables $\mathbf{s}_i$ have been fixed. Based on that, node i further fixes $\mathbf{f}_i$ with (3). Up to now, all decision variables "owned" by i are determined in the first iteration. Then another round of iteration is adopted to fix another set of variables $\mathbf{s}_j$'s and $\mathbf{f}_j$'s "owned" by j following the same procedure above. The iteration continues until all $\mathbf{s}_i$'s and $\mathbf{f}_i$'s are fixed. The corresponding results of $\mathbf{r}$ derived in the last round of iteration together with $\mathbf{s}_i$'s and $\mathbf{f}_i$'s serve as the final solution of P_0.

Remark. Once P_0 is solved via our privacy-preserving scheme, the solution of $\mathbf{r}$ and the result $\sum_{l\in\mathcal{L}} r_l$ are available at the controller, while solutions of $\mathbf{s}_i$ and $\mathbf{f}_i$ are only available at node i. It means that node i has full knowledge of link scheduling and routing strategy to guide its data transmission. Since $\mathbf{s}_i$'s and $\mathbf{f}_i$'s are kept locally, the controller cannot use them to analyze network topology. The reason we do not hide $\mathbf{r}$ and $\sum_{l\in\mathcal{L}} r_l$ is twofold. First, they do not contain any sensitive information about node locations. Second, they are indispensable for billing or regulation purposes.

IV. Related Work

Here, we review representative studies of privacy-preserving linear programming and their demerits with details.

To prevent the computing party learning any information regarding an LP problem, Li and Atallah [17] proposed to use additive-split of the constraint matrix between two involved parties, followed by a series of interactive cryptographic protocols collaboratively executed in each iteration step of the simplex algorithm. Toft [28] later pointed out that there are some technique flaws in the work [17] and also proposed a secure simplex algorithm based on secret sharing with slightly better protocol complexity. Vaidya [30] presented a revised simplex algorithm based on secure scalar product and secure comparison protocols. These works heavily rely on relatively heavyweight crypto techniques. They may fit be time-efficient for small-scale LP problems. Yet, they do not scale well for large-scale optimization problems or problems with combinational variables, which are the general case for cross-layer optimization problems in wireless networks.

An approach to improve the computation efficiency in the literature [13], [31] is to first have the source disguise the constraint matrix by multiplying a random matrix. To further hide problem solutions, an affine mapping is used to transform the original feasible region into another. Then the computing party conducts linear programming on the disguised problem. Since all the transformation is linear, the original solution can still be correctly obtained via reverse operations at the source. This may work for some LP problems with variables ranging in $[\mathbf{0}, +\infty]$. However, cross-layer optimization problems in general take values from a known confined range for some variables. Section III-B illustrated that the privacy goal cannot be fulfilled by solely relying on existing linear transformation.

More importantly, most of the above works [17], [30], [13], [31] deal with two-party computation scenario, i.e., two parties want to jointly solve an LP problem when each of them owns part of it [17], [30], [13], or all the information regarding an LP problem belongs to one party while the computation is conducted at another party [31]. However, in our case, constraint coefficients and solutions of cross-layer optimization problems contain location information from different wireless nodes, which requires a more sophisticated design for preserving privacy. Mangasarian [25] discussed protecting constraint privacy in LP problems considering multiple parties. The idea is to have each participant mask its constraint by multiplying a secret random matrix before uploading it to the computing party. Li *et al.* [20] extended this scheme considering inequality constraints. Apparently, this class of approaches only work for LP problems with block structure, i.e., the variables from one party is independent of the ones from others. Nonetheless, in cross-layer optimization, link scheduling and routing decision variables of one user intertwine with those from its neighboring nodes. Thus, these approaches cannot be applied here either.

V. Conclusions and Future Work

We discuss the problem of protecting node location privacy during cross-layer optimization in multi-hop wireless networks. A novel privacy-preserving scheme is developed to heuristically solve the optimization problem without revealing any useful information within problem constraint and solution, which contains critical knowledge about node locations. The result indicates that our proposed scheme only introduces light extra computation and communication overheads when compared with the algorithm without privacy. Note that our privacy-preserving scheme is for cross-layer optimization problems which adopt *protocol model* as the interference model. As future work, we plan to extend our scheme to cross-layer optimization problems that adopt *physical model*, another widely used, yet, more complicated interference model.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 38, no. 4, pp. 393–422, March 2002.

[2] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, no. 11, pp. 1960–1971, November 2006.

[3] S. J. Baek, G. de Veciana, and X. Su, "Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 22, no. 6, pp. 1130–1140, August 2004.

[4] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, March 2003.

[5] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of Mobile Computing and Networking (MobiCom'05)*, Cologne, Germany, August 2005.

[6] T. Bokareva, W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten, and S. Jha, "Wireless sensor networks for battlefield surveillance," in *Proceedings of the Land Warfare Conference*, 2006.

[7] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, March 2005.

[8] M. Cao, X. Wang, S.-J. Kim, and M. Madihian, "Multi-hop wireless backhaul networks: A cross-layer design paradigm," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 25, no. 4, pp. 738–748, 2007.

[9] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 23, no. 1, pp. 104–116, 2005.

[10] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, 1963.

[11] R. Diestel, *Graph Theory*. Springer, 2005.

[12] L. Ding, T. Melodia, S. N. Batalama, and J. D. Matyjas, "Distributed routing, relay selection, and spectrum allocation in cognitive and cooperative ad hoc networks," in *Proceedings of IEEE International Conference on Sensing, Communication and Networking (SECON'10)*, Boston, Massachusetts, June 2010.

[13] W. Du, "A study of several specific secure two-party computation problems," Ph.D. dissertation, Purdue University, 2001.

[14] Z. Feng and Y. Yang, "Joint transport, routing and spectrum sharing optimization for wireless networks with frequency-agile radios," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM'09)*, Rio de Janeiro, Brazil, April 2009.

[15] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, September 2003.

[16] B. Kölmel and S. Alexakis, "Location based advertising," in *Proceedings of the First International Conference on Mobile Business, Athens, Greece*, 2002, pp. 1–7.

[17] J. Li and M. J. Atallah, "Secure and private collaborative linear programming," in *Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Atlanta, Georgia, November 2006.

[18] M. Li, P. Li, X. Huang, Y. Fang, and S. Glisic, "Energy consumption optimization for multihop cognitive cellular networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 358–372, 2015.

[19] P. Li, X. Huang, and Y. Fang, "Capacity scaling of multihop cellular networks," in *Proceeding of the IEEE International Conference on Computer Communications (INFOCOM'11)*, Shanghai, China, April 2011.

[20] W. Li, H. Li, and C. Deng, "Privacy-preserving horizontally partitioned linear programs with inequality constraints," *Optimization Letters*, vol. 7, no. 1, pp. 137–144, 2013.

[21] X. Lin and S. B. Rasool, "Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1874–1887, 2009.

[22] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings of IEEE Conference on Decision and Control (CDC'04)*, Paradise Island, Bahamas, December 2004.

[23] Y. Lin and Y. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'00)*, Tel Aviv, Israel, March 2000.

[24] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions On Networking*, vol. 11, no. 4, pp. 525–536, 2003.

[25] O. L. Mangasarian, "Privacy-preserving horizontally partitioned linear programs," *Optimization Letters*, vol. 6, no. 3, pp. 431–436, 2012.

[26] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley, 1988.

[27] J. Tang, S. Misra, and G. Xue, “Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks,” *Computer Networks (Elsevier) Journal*, vol. 52, no. 11, pp. 2148–2158, August 2008.

[28] T. Toft, “Solving linear programs using multiparty computation,” in *Financial Cryptography and Data Security, 13th International Conference, FC 2009, Accra Beach, Barbados, February 23-26, 2009. Revised Selected Papers*, 2009, pp. 90–107.

[29] R. Unni and R. Harmon, “Perceived effectiveness of push vs. pull mobile location based advertising,” *Journal of Interactive advertising*, vol. 7, no. 2, pp. 28–40, 2007.

[30] J. Vaidya, “A secure revised simplex algorithm for privacy-preserving linear programming,” in *Proceedings of the 23rd IEEE International Conference on Advanced Information Networking and Applications*, Bradford, United Kingdom, May 2009.

[31] C. Wang, K. Ren, and J. Wang, “Secure and practical outsourcing of linear programming in cloud computing,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM’10)*, San Diego, California, USA, March 2010.

[32] Y. Ye, “An $o(n^3l)$ potential reduction algorithm for linear programming,” *Math. Programming*, vol. 50, no. 4, pp. 239–258, 1991.

[33] H. Zhai and Y. Fang, “Impact of routing metrics on path capacity in multirate and multihop wireless ad hoc networks,” in *Proceedings of the IEEE International Conference on Network Protocols (ICNP’06)*, Santa Barbara, CA, November 2006.