

On the Privacy of Optimization Approaches

P. C. Weeraddana, *Member, IEEE*, G. Athanasiou, *Member, IEEE*,
M. Jakobsson, *Student Member, IEEE*, C. Fischione, *Member, IEEE*,
and J. S. Baras, *Fellow, IEEE*

Abstract

Ensuring privacy of sensitive data is essential in many contexts, such as healthcare data, banks, e-commerce, wireless sensor networks, and social networks. It is common that different entities coordinate or want to rely on a third party to solve a specific problem. At the same time, no entity wants to publish its problem data during the solution procedure unless there is a privacy guarantee. Unlike cryptography and differential privacy based approaches, the methods based on optimization lack a quantification of the privacy they can provide. The main contribution of this paper is to provide a mechanism to quantify the privacy of a broad class of optimization approaches. In particular, we formally define a one-to-many relation, which relates a given adversarial observed message to an uncertainty set of the problem data. This relation quantifies the potential ambiguity on problem data due to the employed optimization approaches. The privacy definitions are then formalized based on the uncertainty sets. The properties of the proposed privacy measure is analyzed. The key ideas are illustrated with examples, including localization, average consensus, among others.

Index Terms

Privacy, distributed optimization, ADMM, secured multiparty computation.

I. INTRODUCTION

Privacy is central in many application domains, where parties jointly solve optimization problems. These interactions and collaborations are desirable to gain mutual benefits. For example, independent hospitals would like to coordinate for diagnostic decision making based on their existing patient records. Normally, optimization solvers require public data sharing, which substantially hinder the cooperation due to privacy concerns (e.g., privacy for patients' records). The challenge is how to *solve* problems among parties, while preserving privacy of their individual data.

P. C. Weeraddana, G. Athanasiou, M. Jakobsson, and C. Fischione with Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. J. S. Baras is with the Department of Electrical and Computer Engineering, University of Maryland, USA.

A. Problem Statement

Formally, an n -party design and decision making problem can be posed as follows:

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}; \mathbf{c}_1, \dots, \mathbf{c}_N) \\ & \text{subject to} && g(\mathbf{x}; \mathbf{c}_1, \dots, \mathbf{c}_N) \preceq \mathbf{0} \\ & && h(\mathbf{x}; \mathbf{c}_1, \dots, \mathbf{c}_N) = \mathbf{0} . \end{aligned} \tag{1}$$

Here, $f_0(\cdot)$ is a scalar-valued objective function of the decision variables $\mathbf{x} \in \mathbb{R}^M$, given the problem data $\{\mathbf{c}_i\}_{i=1:N}$,¹ where \mathbf{c}_i is the *private* data (e.g., matrices) associated with entity i . Moreover, $g(\cdot)$ and $h(\cdot)$ are vector-valued nonlinear and affine functions, respectively.

We now pose the questions: Q1) How to quantify the privacy of the problem data $\{\mathbf{c}_i\}_{i=1:N}$? Q2) How to *solve* problem (1), such that the “privacy” of $\{\mathbf{c}_i\}_{i=1:N}$ is preserved?

B. Existing Approaches

An ideal answer to Q1 is *perfect privacy*, where no information of the original problem data $\{\mathbf{c}_i\}_{i=1:N}$ can be extracted by a third party. However, answering Q2 with perfect privacy is generally impossible. There is an alternative fundamental theory for answering Q1, namely, cryptography [1], which includes a number of mechanisms to encrypt the original problem data $\{\mathbf{c}_i\}_{i=1:N}$. The underlying privacy definition is called *semantic security* [1, § 5]. Loosely speaking, the semantic privacy means that nothing can be *feasibly extracted* from the encrypted data. Given Q1 is handled by the semantic security, *secured multiparty computation* (SMC) [2] is a well established framework for answering Q2. In principle, SMC is applied to almost any problem, given that the required computations are written as a Boolean circuit, or as an arithmetic circuit. However, the general objective and constraints of problem (1) can be complicated functions, and therefore usually the associated solution approach become impractical [3], [4], [5, § 2.3.2].

An alternative line of research for answering Q1 considers the ϵ -*differential privacy* [6]. The definition allows a non-negligible information leakage, denoted by ϵ , as opposed to the cryptographic counterpart. Roughly speaking, the standard way of accomplishing ϵ -differential privacy for Q2 is based on a random perturbation mechanism, where the underlying subroutines of the associated solution method is appropriately mutated by using random noise with specific statistical properties, *cf.* [7]–[12]. As a consequence, there is a trade-off between the optimality

¹Dimensions of \mathbf{c}_i 's, can also be problem data, which we considered known, throughout to avoid cumbersome notations.

of the solution of problem (1) and the desired privacy level of the problem data, quantified by ϵ . In addition, the applicability of differential privacy based approaches for solving problem (1) can usually be limited. Note that specifying the noise properties, such as variances to achieve a pre-specified differential privacy is solely dependent on the *sensitivity* of the related functions (e.g., objective, constraint, subgradient functions) with respect to *perturbations of the problem data*. In general, quantifying such sensitivities in a *closed-form* is challenging. One can certainly compute some bounds. However, unless the bounds are tight, the noise variance computed based on the bounds will be over estimated, which in turn can degrade the optimality of the solution significantly [6], [7], [12].

As opposed to the cryptography and differential privacy based methods discussed above, there is another class of approaches, which answers Q2 [5], [13]–[20], yet answers to Q1 are rather intuitively explained. The key idea of the methods is to use algebraic manipulations to disguise the original problem into an equivalent problem so that the problem data $\{\mathbf{c}_i\}_{i=1:N}$ is somehow hidden. We refer to these approaches as algebraic transformation methods. Algebraic transformation methods are promising in the sense that they are efficient and usually guarantee the optimality of the solution of problem (1), though privacy of the methods is not quantified. The approaches are typically applied to a broader class of problems, as opposed to the cryptographic or differential privacy based approaches. Therefore it is still desirable to quantify their privacy as response to Q1.

C. Our Contributions

The main contribution of this paper is to quantify the privacy properties of a broader class of optimization approaches, where the algebraic transformation methods are particular cases. More specifically, the proposed privacy definition applies to 1) methods via standard objective/constraint transformations, 2) methods via standard variable transformations, 3) classical decomposition methods, e.g., primal and dual decomposition, and 4) state-of-the-art alternating direction method of multipliers method (ADMM).

To do this, the underlying disguise of problem data $\{\mathbf{c}_i\}_{i=1:N}$ due to those optimization approaches is formally represented as a one-to-many binary relation between the observed message of an adversary and an uncertainty set, in which the sensitive problem data $\{\mathbf{c}_i\}_{i=1:N}$ itself is a member. The privacy of the associated method can then be quantified by the properties

of the uncertainty set. The properties of the proposed privacy measure is analyzed. Given an optimization approach as response to Q2, our definition answers Q1. Implications of the proposed privacy measure is discussed. Our privacy definition allows comparing the privacy of different optimization approaches applied to a problem. It quantifies the *best* and the *worst* privacy for an specified problem data. Several examples are given to illustrate the key ideas.

In § II we present our new proposed definitions and other basic ones, useful for describing the privacy properties of optimization approaches. Application of the proposed privacy definitions for optimization approaches are illustrated in § III. Conclusions are given in § IV.

Notations: Boldface lower case and upper case letters represent vectors and matrices, respectively, and calligraphy letters represent sets. The Euclidean- n space and the positive integers are denoted by \mathbb{R}^n and \mathbb{N} , respectively. The superscript $(\cdot)^T$ is the transpose. We use the notation (\mathbf{a}, \mathbf{b}) to denote $[\mathbf{a}^T \mathbf{b}^T]^T$. The identity matrix, all zero vector, and all one vector, are denoted by \mathbf{I} , $\mathbf{0}$, and $\mathbf{1}$, respectively. The ℓ_2 -norm of \mathbf{x} is denote by $\|\mathbf{x}\|_2$. The power set of an arbitrary space Ω is denoted by $\mathcal{P}(\Omega)$. A ball with radius r and center \mathbf{b} is denoted by $\mathcal{B}(r, \mathbf{b})$. Finally, we denote by \succeq , the component-wise inequality.

II. PRIVACY QUANTIFICATIONS

Original definitions together with other basic definitions and assumptions for quantifying the privacy properties of the optimization approaches are given. The essential implications of the definitions are also discussed.

A. Problem Data and the Adversary

Definition 1 (Inputs, input tuples, and messages): Consider the problem (1). We call the *set of problem data* $\{\mathbf{c}_i\}_{i=1:N}$, denoted \mathcal{C} , the *inputs* of problem (1). Suppose $\mathbf{c} = \mathbf{f}((\mathbf{c}_i)_{i=1:N})$, where \mathbf{f} is a possibly vector valued function with non-constant components. Moreover, suppose that \mathbf{c} lies in a metric space $\mathcal{X}_{\mathbf{c}}$. We call \mathbf{c} , an *input tuple* of \mathcal{C} . Finally, we call a finite length data structure $S((\mathbf{c}_i)_{i=1:N}) = \left(S_1((\mathbf{c}_i)_{i=1:N}), \dots, S_M((\mathbf{c}_i)_{i=1:N}) \right)$, a *message* of \mathcal{C} , where $S_j((\mathbf{c}_i)_{i=1:N})$ represents any vector/matrix valued function. ◀

Note that elements of \mathbf{c} represents the central components what the problem data owner cares about. The key points of the definition is illustrated in Example 1. Throughout the paper we consider the following assumption on the input of a problem:

Assumption 1: The inputs of any optimization problem of the form (1) is deterministic, and therefore there is no associated statistical models. ◀

The considered adversarial model and its associated knowledge is defined next. The definition of the adversary is similar to the passive eavesdroppers considered in [1, § 5.1-5.3]. We consider the *passive adversary* model throughout this paper.

Definition 2 (Passive adversary): In a multi-party environment, a party involved in solving a problem of the form (1), or even a third party, is called a passive adversary if it taps the messages of input of (1) exchanged during different stages of the solution method, keeps a record of possibly all the messages exchanged, and tries to discover others' private data. ◀

Definition 3 (Adversarial knowledge): The set \mathcal{K} of information that an adversary might exploit to discover the input of problem (1) is called the adversarial knowledge. The knowledge can contain messages S of the problem input, which we call the *inevitable* knowledge. The rest of the knowledge $\mathcal{K} \setminus \{S\}$ is called the *auxiliary* knowledge. ◀

Note that the inevitable knowledge contains messages (e.g., S , cf. Definition 1) that must be revealed to a third party, such as transformed variants of inputs necessary for executing the solution method. The auxiliary knowledge can encompass components, such as eavesdropped measures of input elements, knowledge of optimality properties, etc., cf. Example 1. The following remark summarizes a property of \mathcal{K} that will be useful later.

Remark 1: For a given problem and its solution approach, the knowledge \mathcal{K} of a third party can be further improved and can never be decreased as time elapses. In other words, given time $t_0 \leq t_1 \leq t_2 \leq \dots$, we have $\mathcal{K} = \mathcal{K}_{t_0} \subseteq \mathcal{K}_{t_1} \subseteq \mathcal{K}_{t_2} \subseteq \dots$, where \mathcal{K}_{t_i} denotes the adversarial knowledge at time t_i . As a result, unless the underlying solution method is changed, it is always the case that $\mathcal{K}_{t_0} \subseteq \mathcal{K}_{t_1} \subseteq \mathcal{K}_{t_2} \subseteq \dots$. ◀

Note that *different* solution methods in general can yield different sets of knowledge, even if the problem is fixed. Finally, recall that the set \mathcal{C} of the inputs/outputs is deterministic, cf. Assumption 1. Therefore, the adversarial knowledge \mathcal{K} does not contain elements such as probability density functions or cumulative distribution functions to describe statistical models for $\mathbf{c}_i \in \mathcal{C}$.

Definition 4 (Obfuscation of the input and uncertainty sets): Let \mathcal{C} , \mathbf{c} , S , and \mathcal{K} denote the input, an input tuple, a message of problem (1), and the adversarial knowledge, respectively, cf. Definition 1. Suppose $S \subseteq \mathcal{K}$. We say $\mathbf{c} \in \mathcal{X}_{\mathbf{c}}$ is obfuscated if there exists a *one-to-many*

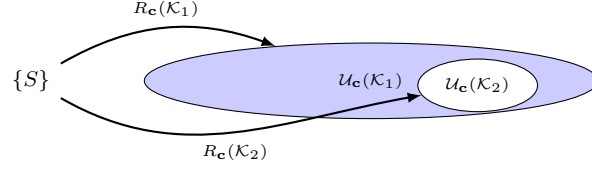


Fig. 1: Obfuscation of an input tuple \mathbf{c} , where $S \subseteq \mathcal{K}_1$, $S \subseteq \mathcal{K}_2$, $\mathcal{K}_1 \subseteq \mathcal{K}_2$, and the ellipsoids represent uncertainty sets of \mathbf{c} .

binary relation \mathcal{R}_c from the singleton $\{S\}$ to a nonempty set $\mathcal{U}_c \subseteq \mathcal{X}_c$, such that $(S, \mathbf{c}) \in \mathcal{R}_c$. We call the (possibly unbounded) set $\mathcal{U}_c = \{\mathbf{u} \mid (S, \mathbf{u}) \in \mathcal{R}_c\}$, the uncertainty set for \mathbf{c} . ◀

The *one-to-many* nature of such relations is the mechanism of achieving ambiguity in \mathbf{c} , cf. Figure 1. An example to clarify Definition 4 is given below:

Example 1: Consider a client-cloud environment, where the client has an optimization problem with problem input $\mathcal{C} = \{\mathbf{a}, \mathbf{b}\}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ are private data that the client cares about. The cloud (potential adversary) solves the problem for the client. Suppose that the employed deterministic solution methods for the problem is such that client has to outsource the two scalars s_1 and s_2 to the cloud, where $s_1 = \|\mathbf{a}\|_2^2$ and $s_2 = \mathbf{b}^T \mathbf{a}$, that is the message $S = (s_1, s_2)$. Suppose, in addition, cloud knows n (i.e., auxiliary knowledge) and wants to find input tuple \mathbf{a} . Then we have $\mathcal{R}_a = \{(S, \mathbf{a}) \mid s_1 = \|\mathbf{a}\|_2^2, s_2 = \mathbf{b}^T \mathbf{a}, \mathbf{b} \in \mathbb{R}^n, \mathbf{a} \in \mathbb{R}^n\}$.

Trivially, having more knowledge of an input tuple \mathbf{c} can only reduce the ambiguity in \mathbf{c} . This phenomenon is formally expressed in the following remark, cf. Figure 1.

Remark 2: Let $\mathcal{R}_c(\mathcal{K}_1)$ and $\mathcal{R}_c(\mathcal{K}_2)$ be relations defined as in Definition 4 associated with a problem, where the *inevitable* knowledge is identical in both cases, i.e., $\mathcal{K}_1 \supseteq S$ and $\mathcal{K}_2 \supseteq S$ for some message S . Then $\mathcal{K}_1 \subseteq \mathcal{K}_2 \Rightarrow \mathcal{U}_c(\mathcal{K}_1) \supseteq \mathcal{U}_c(\mathcal{K}_2)$. ◀

We note that, in many interesting cases, the considered optimization approaches in this paper can be used to realize a relation \mathcal{R}_c as defined in Definition 4. We capitalize on the properties of such relations resulted by optimization approaches, when quantifying the privacy.

B. Privacy Index

Given a relation \mathcal{R} and its associated uncertainty set \mathcal{U} that lies in a metric space \mathcal{X} (cf. Definition 4). We consider the following metrics to quantify the size and the spread of \mathcal{U} .

- 1) Diameter of \mathcal{U} , denoted $\text{d}(\mathcal{U})$.
- 2) Counting measure of \mathcal{U} , denoted by $\mu(\mathcal{U})$ [21, p. 146].
- 3) Affine dimension of \mathcal{U} , denoted $\text{a}(\mathcal{U})$ [22, § 2.1.3].

The diameter gives a measure of the worst discrepancy between any two points in the uncertainty set. In particular, we define $\text{d}(\cdot)$ as follows:

Definition 5 (Diameter/center of an uncertainty set): Given an input tuple \mathbf{c} of the input \mathcal{C} and \mathbf{c} lies in a metric space $\mathcal{X}_{\mathbf{c}}$. Let $\mathcal{U}_{\mathbf{c}} \subseteq \mathcal{X}_{\mathbf{c}}$ denote the uncertainty set of \mathbf{c} . Then the diameter $\mathbf{d} : \mathcal{X}_{\mathbf{c}} \rightarrow \mathbb{R} \cup \infty$ is given by

$$\mathbf{d}(\mathcal{U}_{\mathbf{c}}) = \begin{cases} 2 \inf_{\mathbf{v}} \sup_{\mathbf{u} \in \mathcal{U}_{\mathbf{c}}} \|\mathbf{v} - \mathbf{u}\| & \mathcal{U}_{\mathbf{c}} \text{ is bounded} \\ \infty & \text{otherwise,} \end{cases} \quad (2)$$

where $\|\cdot\|$ is some norm on $\mathcal{X}_{\mathbf{c}}$. The argument \mathbf{v} that achieves $\mathbf{d}(\mathcal{U}_{\mathbf{c}})$ is called the *center* of $\mathcal{U}_{\mathbf{c}}$ and is denoted by $\mathbf{c}(\mathcal{U}_{\mathbf{c}})$. If the set $\mathcal{U}_{\mathbf{c}}$ is unbounded, we say any point in $\mathcal{X}_{\mathbf{c}}$ achieves $\mathbf{d}(\mathcal{U}_{\mathbf{c}})$, i.e., $\mathbf{c}(\mathcal{U}_{\mathbf{c}})$ is arbitrary. \blacktriangleleft

It is not difficult to see that the ball $\mathcal{B}(\mathbf{d}(\mathcal{U}_{\mathbf{c}})/2, \mathbf{c}(\mathcal{U}_{\mathbf{c}}))$, denoted $\mathcal{B}_{\mathcal{U}_{\mathbf{c}}}$, is the smallest volume ball, which encloses $\mathcal{U}_{\mathbf{c}}$. The metric $\mathbf{d}(\mathcal{U}_{\mathbf{c}})$ for measuring the size of the uncertainty set $\mathcal{U}_{\mathbf{c}}$ can be interpreted and motivated as follows. Suppose the set $\mathcal{U}_{\mathbf{c}}$ be bounded and an adversary wants to estimate \mathbf{c} . Because \mathbf{c} is deterministic (cf. Assumption 1), a reasonable criterion, from the view of the adversary, is to pick $\mathbf{c}_{\text{adv}}^*$ that minimizes the associated worst-case error $\|\mathbf{c}_{\text{adv}} - \mathbf{c}\|$ of a candidate guess \mathbf{c}_{adv} . These strategies are known as *worst-case robust designs*, cf. [22, § 6.4.2]. From (2), it follows that $\mathbf{c}_{\text{adv}}^* = \mathbf{c}(\mathcal{U}_{\mathbf{c}})$ and the adversary's worst discrepancy between its guess $\mathbf{c}_{\text{adv}}^*$ and \mathbf{c} is given by $(1/2)\mathbf{d}(\mathcal{U}_{\mathbf{c}})$. On the other hand, if the set $\mathcal{U}_{\mathbf{c}}$ is unbounded, a rational adversary has no preference for one point in $\mathcal{X}_{\mathbf{c}}$ over another. Therefore, the adversary's worst discrepancy is unbounded above, irrespective of its guess $\mathbf{c}_{\text{adv}}^*$.

The counting measure of an uncertainty set is an estimate of the number elements in it [21, p. 146], and is defined below.

Definition 6 (Counting measure of an uncertainty set): Given an input tuple \mathbf{c} of the input \mathcal{C} that lies in a metric space $\mathcal{X}_{\mathbf{c}}$. Let $\mathcal{U}_{\mathbf{c}} \subseteq \mathcal{X}_{\mathbf{c}}$ denote the uncertainty set of \mathbf{c} . The counting measure of $\mathcal{U}_{\mathbf{c}}$ is $\mu(\mathcal{U}_{\mathbf{c}})$, where $\mu: \mathcal{P}(\mathcal{X}_{\mathbf{c}}) \rightarrow \mathbb{N} \cup \infty$ is

$$\mu(\mathcal{A}) = \begin{cases} |\mathcal{A}| & \mathcal{A} \text{ is finite} \\ \infty & \text{otherwise,} \end{cases} \quad (3)$$

with $|\cdot|$ denoting the cardinality of a *finite* set. \blacktriangleleft

The metric $\mu(\mathcal{U}_{\mathbf{c}})$ for quantifying the uncertainty set $\mathcal{U}_{\mathbf{c}}$ can be motivated as follows. If $\mathcal{U}_{\mathbf{c}}$ is finite, the value $\mu(\mathcal{U}_{\mathbf{c}})$ can be used to quantify the adversary's *best* worst-case winning² probability, denoted $p_{\mathbf{w}-\mathbf{c}}^{\text{win}}$, and to characterizes the probability mass function (pmf) of adversary's

²Here the term *winning* refers to the event of guessing correctly \mathbf{c} .

guess, denoted $\mathbf{p}_{\text{adv}}^*$, to achieve $p_{\text{w-c}}^{\text{win}}$. Recall that \mathbf{c} is deterministic, cf. Assumption 1. That is, the adversary has no statistical models for describing the variation in \mathbf{c} . Therefore, a natural criterion for any rational adversary is to choose a probability distribution that maximizes the resulting winning probability $\Pr(\text{adversary wins})$, among all possible probability distributions for \mathbf{c} . From basic probability rules, we get $p_{\text{w-c}}^{\text{win}} = 1/\mu(\mathcal{U}_{\mathbf{c}})$ and $\mathbf{p}_{\text{adv}}^* = (1/\mu(\mathcal{U}_{\mathbf{c}}))\mathbf{1}$, as probabilistic intuition requires. If $\mathcal{U}_{\mathbf{c}}$ is infinite, as our probabilistic intuition suggests, $p_{\text{w-c}}^{\text{win}}$ ought to be 0. Thus, $\mu(\mathcal{U}_{\mathbf{c}})$ quantifies the “practical impossibility” of an *exact* guess.

The last metric used to quantify the properties of $\mathcal{U}_{\mathbf{c}}$ is its affine dimension, i.e., $a(\mathcal{U}_{\mathbf{c}})$. In particular, $a(\mathcal{U}_{\mathbf{c}})$ is the dimension of its affine hull [22, § 2.1.3]. Roughly speaking, $a(\mathcal{U}_{\mathbf{c}})$ quantifies the number of orthogonal directions, along which an adversary has to perform its search for locating \mathbf{c} .

Let us finally give a formal definition to quantify the privacy of optimization methods considered in this paper. To do this, we use the metrics $d(\cdot)$, $\mu(\cdot)$, and $a(\cdot)$ discussed above.

Definition 7 (Privacy index): Let \mathcal{C} denote the input of problem (1) and \mathbf{c} denote any input tuple of \mathcal{C} that lies in a metric space $\mathcal{X}_{\mathbf{c}}$. Moreover, let $\mathcal{U}_{\mathbf{c}} \subseteq \mathcal{X}_{\mathbf{c}}$ denote the uncertainty set of \mathbf{c} . Then we call the 3-component tuple $(d(\mathcal{U}_{\mathbf{c}}), \nu(\mathcal{U}_{\mathbf{c}}), a(\mathcal{U}_{\mathbf{c}}))$ the privacy index of \mathbf{c} , where $\nu(\mathcal{U}_{\mathbf{c}}) = 1 - 1/\mu(\mathcal{U}_{\mathbf{c}})$. Moreover, the corresponding tuple is denoted by $\boldsymbol{\rho}(\mathcal{U}_{\mathbf{c}})$, i.e., $\boldsymbol{\rho}(\mathcal{U}_{\mathbf{c}}) = (d(\mathcal{U}_{\mathbf{c}}), \nu(\mathcal{U}_{\mathbf{c}}), a(\mathcal{U}_{\mathbf{c}}))$. ◀

Note that $\nu(\mathcal{U}_{\mathbf{c}}) \in [0, 1)$ is the adversary’s *best* worst-case losing probability, i.e., $\nu(\mathcal{U}_{\mathbf{c}}) = 1 - p_{\text{w-c}}^{\text{win}}$. Some basic properties of the $\boldsymbol{\rho}(\cdot)$ are proved in the following proposition.

Proposition 1: The privacy index $\boldsymbol{\rho}(\mathcal{U})$ has the following properties:

- a) $\mathbf{0} \preceq \boldsymbol{\rho}(\mathcal{U}) \preceq (\infty, 1, n)$ for all $\mathcal{U} \subseteq \mathbb{R}^n$.
- b) \mathcal{U} is a singleton $\Leftrightarrow \boldsymbol{\rho}(\mathcal{U}) = \mathbf{0}$.
- c) $\boldsymbol{\rho}(\mathbf{v} + \mathcal{U}) = \boldsymbol{\rho}(\mathcal{U})$, for all $\mathbf{v} \in \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^n$, where $\mathbf{v} + \mathcal{U} = \{\mathbf{v} + \mathbf{u} \mid \mathbf{u} \in \mathcal{U}\}$.
- d) $\boldsymbol{\rho}(\mathbf{P}\mathcal{U}) = \boldsymbol{\rho}(\mathcal{U})$, for all unitary matrices $\mathbf{P} \in \mathbb{R}^{n \times n}$ and $\mathcal{U} \subseteq \mathbb{R}^n$, where $\mathbf{P}\mathcal{U} = \{\mathbf{P}\mathbf{u} \mid \mathbf{u} \in \mathcal{U}\}$.
- e) $\mathcal{U} \subseteq \mathcal{V} \Rightarrow \boldsymbol{\rho}(\mathcal{U}) \preceq \boldsymbol{\rho}(\mathcal{V})$.
- f) Let $\mathcal{U}_{\mathbf{c}}, \mathcal{V}_{\mathbf{c}} \subseteq \mathbb{R}^n$ denote two uncertainty sets of \mathbf{c} . Then $\boldsymbol{\rho}(\mathcal{U}_{\mathbf{c}}) \preceq \boldsymbol{\rho}(\mathcal{V}_{\mathbf{c}}) \Rightarrow \exists \mathcal{U}^*, \mathcal{V}^* \in \mathbb{R}^n$, containing \mathbf{c} such that $\boldsymbol{\rho}(\mathcal{U}_{\mathbf{c}}) = \boldsymbol{\rho}(\mathcal{U}^*) \preceq \boldsymbol{\rho}(\mathcal{V}^*) = \boldsymbol{\rho}(\mathcal{V}_{\mathbf{c}})$, $\mathcal{B}_{\mathcal{U}^*} \subseteq \mathcal{B}_{\mathcal{V}^*}$. ◀

Proof:

a,b) The proof of these parts follow directly from the definition of the metrics d , μ , and a .

c,d) These parts follows from that translation and rotation do not change sets in \mathbb{R}^n .

e) Let us first show that $d(\mathcal{U}) \leq d(\mathcal{V})$. If at least one set of \mathcal{U} and \mathcal{V} is unbounded, then $d(\mathcal{U}) \leq d(\mathcal{V})$ holds trivially because for any unbounded \mathcal{W} , $d(\mathcal{W}) = \infty$. If \mathcal{U} and \mathcal{V} are bounded, for any $\mathbf{x} \in \mathbb{R}^n$, $g_{\mathcal{U}}(\mathbf{x}) = \sup_{\mathbf{u} \in \mathcal{U}} \|\mathbf{x} - \mathbf{u}\|$ is bounded, so is $g_{\mathcal{V}}(\mathbf{x}) = \sup_{\mathbf{v} \in \mathcal{V}} \|\mathbf{x} - \mathbf{v}\|$. Moreover, if $\mathcal{U} \subseteq \mathcal{V}$, then $g_{\mathcal{U}}(\mathbf{x}) \leq g_{\mathcal{V}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$. Therefore, $\inf_{\mathbf{x}} g_{\mathcal{U}}(\mathbf{x}) = d(\mathcal{U}) \leq \inf_{\mathbf{x}} g_{\mathcal{V}}(\mathbf{x}) = d(\mathcal{V})$ as required. The relation $\nu(\mathcal{U}) \leq \nu(\mathcal{V})$ follows from that $\mu(\mathcal{U}) \leq \mu(\mathcal{V})$. In particular, $(\mathbb{R}^n, \mathcal{P}(\mathbb{R}^n), \mu)$, μ being the counting measure on $\mathcal{P}(\mathbb{R}^n)$, is a measure space, cf. [21, p. 146]. Therefore, from [21, Theorem 5.1], $\mathcal{U} \subseteq \mathcal{V} \Rightarrow \mu(\mathcal{U}) \leq \mu(\mathcal{V})$, i.e., $\nu(\mathcal{U}) \leq \nu(\mathcal{V})$. Finally, $a(\mathcal{U}) \leq a(\mathcal{V})$ follows by definition.

f) Let $\bar{\mathcal{U}} = \{\mathbf{P}_{\mathcal{U}}(\mathbf{u} + \mathbf{t}_{\mathcal{U}}) \mid \mathbf{u} \in \mathcal{U}_c\}$ and $\bar{\mathcal{V}} = \{\mathbf{P}_{\mathcal{V}}(\mathbf{v} + \mathbf{t}_{\mathcal{V}}) \mid \mathbf{v} \in \mathcal{V}_c\}$ for some $\mathbf{t}_{\mathcal{U}}, \mathbf{t}_{\mathcal{V}} \in \mathbb{R}^n$ and unitary matrices $\mathbf{P}_{\mathcal{U}}, \mathbf{P}_{\mathcal{V}} \in \mathbb{R}^{n \times n}$. Then from the hypothesis and from part c), d), it follows that $\rho(\mathcal{U}_c) = \rho(\bar{\mathcal{U}}) \preceq \rho(\bar{\mathcal{V}}) = \rho(\mathcal{V}_c)$. Let us now find a particular choice $\mathbf{t}_{\mathcal{U}}^*, \mathbf{t}_{\mathcal{V}}^*, \mathbf{P}_{\mathcal{U}}^*$, and $\mathbf{P}_{\mathcal{V}}^*$ to compute \mathcal{U}^* and \mathcal{V}^* . If both $\mathcal{U}_c, \mathcal{V}_c$ are unbounded, then choose $\mathbf{t}_{\mathcal{U}}^* = 0, \mathbf{t}_{\mathcal{V}}^* = 0, \mathbf{P}_{\mathcal{U}}^* = \mathbf{P}_{\mathcal{V}}^* = \mathbf{I}$. The result follows immediately because $\mathcal{B}_{\mathcal{U}^*} = \mathcal{B}_{\mathcal{V}^*} = \mathbb{R}^n \cup \infty$. If only \mathcal{V}_c is unbounded, the same choice of $\mathbf{t}_{\mathcal{U}}^*, \mathbf{t}_{\mathcal{V}}^*, \mathbf{P}_{\mathcal{U}}^*$, and $\mathbf{P}_{\mathcal{V}}^*$ yields the result, because $\mathcal{B}_{\mathcal{V}^*} = \mathbb{R}^n \cup \infty$. However, if both $\mathcal{U}_c, \mathcal{V}_c$ are bounded, we have to be more specific. Recall that $\mathcal{B}_{\mathcal{U}_c}$ is the smallest volume (bounded) ball containing \mathcal{U}_c , cf. Definition 5. Therefore, \exists a limit point $\mathbf{l}_{\mathcal{U}}$ at the boundary of $\mathcal{B}_{\mathcal{U}_c}$ such that $\forall \epsilon \succ 0, \mathcal{B}(\epsilon, \mathbf{l}_{\mathcal{U}}) \cap \mathcal{U}_c$ is nonempty, cf. Figure 2. Similarly, \exists a limit point $\mathbf{l}_{\mathcal{V}}$ at the boundary of $\mathcal{B}_{\mathcal{V}_c}$ such that $\forall \epsilon \succ 0, \mathcal{B}(\epsilon, \mathbf{l}_{\mathcal{V}}) \cap \mathcal{V}_c$ is nonempty. Then by construction, $\mathcal{U}^* = \{\mathbf{u} + (\mathbf{c} - \mathbf{l}_{\mathcal{U}}) \mid \mathbf{u} \in \mathcal{U}_c\}$ and $\mathcal{V}^* = \{\mathbf{P}\mathbf{v} + (\mathbf{c} - \mathbf{P}\mathbf{l}_{\mathcal{V}}) \mid \mathbf{v} \in \mathcal{V}_c\}$ satisfy the required necessary conditions, where \mathbf{P} is a unitary matrix such that $(\mathbf{a}_{\mathcal{U}} - \mathbf{l}_{\mathcal{U}}) / \|\mathbf{a}_{\mathcal{U}} - \mathbf{l}_{\mathcal{U}}\|_2 = \mathbf{P}(\mathbf{a}_{\mathcal{V}} - \mathbf{l}_{\mathcal{V}}) / \|\mathbf{a}_{\mathcal{V}} - \mathbf{l}_{\mathcal{V}}\|_2$, $\mathbf{a}_{\mathcal{U}} = c(\mathcal{U}_c)$, and $\mathbf{a}_{\mathcal{V}} = c(\mathcal{V}_c)$, cf. Figure 2. To see this, note that the sets $\mathcal{U}^*, \mathcal{V}^*$ are based on rotations and translations of the original sets $\mathcal{U}_c, \mathcal{V}_c$. In particular, $\mathbf{t}_{\mathcal{U}}^* = \mathbf{c} - \mathbf{l}_{\mathcal{U}}, \mathbf{t}_{\mathcal{V}}^* = \mathbf{c} - \mathbf{P}\mathbf{l}_{\mathcal{V}}, \mathbf{P}_{\mathcal{U}}^* = \mathbf{I}, \mathbf{P}_{\mathcal{V}}^* = \mathbf{P}$. When $\mathbf{u} = \mathbf{l}_{\mathcal{U}}$, we conclude that $\mathbf{c} \in \mathcal{U}^*$ and, so is $\mathbf{c} \in \mathcal{V}^*$. Moreover, the rotation induced by \mathbf{P} ensures $\mathcal{B}_{\mathcal{U}^*} \subseteq \mathcal{B}_{\mathcal{V}^*}$, cf. Figure 2.

This completes the proof of the Proposition. ■

As we will see next, Proposition 1 is useful for explaining interesting implications of the privacy index. Especially, the parts a), b) e), and f) of Proposition 1 are more relevant.

C. Implications of the Privacy Index

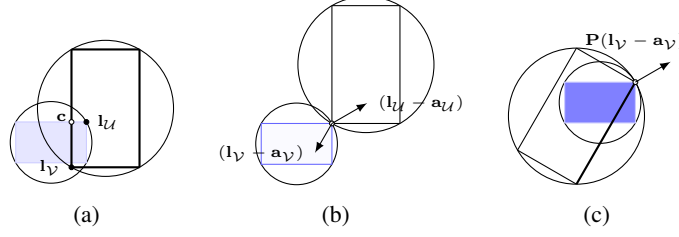


Fig. 2: Proposition 1-f): (a) Original sets \mathcal{U}_c (shaded box) and \mathcal{V}_c (solid rectangular wire frame); (b) A representation of intermediate steps, where the \mathcal{U}_c and \mathcal{V}_c are shifted, yet they still contains \mathbf{c} ; (c) Sets \mathcal{U}^* (shaded box) and \mathcal{V}^* (solid rectangular wire frame).

By definition, the privacy index $\boldsymbol{\rho}$ lies in the extended \mathbb{R}^3 and is nonnegative, *cf.* Proposition 1-a). Moreover, Proposition 1-a) provides the *smallest* and the *largest* possible values of $\boldsymbol{\rho}$ for an input-tuple in \mathbb{R}^n . Note that if an uncertainty set \mathcal{U}_c is a singleton, then it contains \mathbf{c} only, *cf.* Definition 4. Therefore, from Proposition 1-b), $\boldsymbol{\rho}(\mathcal{U}_c) = \mathbf{0} \Leftrightarrow \mathcal{U}_c = \{\mathbf{c}\}$. Thus, $\boldsymbol{\rho} = \mathbf{0}$ signifies zero privacy, where an adversary can exactly determine the associated input tuple.

Proposition 1-e) and f) provide a mechanism to compare different $\boldsymbol{\rho}$ vectors, wherefore the privacy of different solution methods can be compared. Proposition 1-e) and Remark 1 ensure that the optimization based approaches lacks robustness to adversarial attacks, such as those involving side information. In addition, Proposition 1-e) is useful to compute lower bounds for $\boldsymbol{\rho}$ of a complicated uncertainty set. The converse of Proposition 1-e) does not holds in general.

Proposition 1-f) is a weaker form of e): given a nonincreasing (with respect to \mathbb{R}_+^3) sequence $\{\boldsymbol{\rho}(\mathcal{U}_c^{(n)})\}_n^N$, \exists a sequence $\{\bar{\mathcal{U}}_c^{(n)}\}_n^N$ of sets, each containing \mathbf{c} and $\mathcal{B}_{\bar{\mathcal{U}}_c^{(1)}} \supseteq \cdots \supseteq \mathcal{B}_{\bar{\mathcal{U}}_c^{(N)}} \supseteq \{\mathbf{c}\}$. Thus, Proposition 1-f) gives a way to compare the privacy of two algorithms given the privacy indexes, irrespective of the underlying solution machinery. In addition, Proposition 1-f) suggests a natural ordering of privacy indexes in general, (Figure 3) providing meanings to the *smallest* (i.e., $\mathbf{0}$) and the *largest* (i.e., $(\infty, 1, n)$) privacy for an *input tuple* $\mathbf{c} \in \mathbb{R}^n$, *cf.* Proposition 1-a).

III. OPTIMIZATION APPROACHES AND THEIR PRIVACY

In this section, standard optimization approaches are concisely presented, *cf.* Q2 § I-A. Examples are given to quantify the privacy of these optimization approaches, *cf.* Q1. All the examples are designed specifically to illustrate our point.

A. Transformation of Objective and Constraint Functions

Suppose $\psi_0 : \mathbb{D}_0 \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is monotonically increasing and $\mathbb{D}_0 \supseteq \text{range}(f_0)$. Moreover,

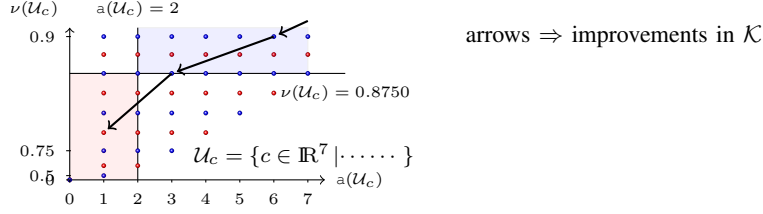


Fig. 3: Feasible $\rho(\mathcal{U}_c)$ for a fixed $d(\mathcal{U}_c)$. The privacy corresponds to the points in the upper-right shaded area are better than that of the points in lower-left shaded area. Rest of the points are incomparable.

suppose $\psi_i : \mathbb{D}_i \subseteq \mathbb{R} \rightarrow \mathbb{R}$, with $\mathbb{D}_i \supseteq \text{range}(g_i)$, is such that $\psi_i(z) \leq 0 \Leftrightarrow z \leq 0$ and $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}$ is such that $\varphi_i(z) = 0 \Leftrightarrow z = 0$. Let \mathbb{P} denote the problem formed by using $\psi_0(f_0(\mathbf{x}))$, $\psi_i(g_i(\mathbf{x}))$, and $\varphi_i(h_i(\mathbf{x}))$, instead of $f_0(\mathbf{x})$, $g_i(\mathbf{x})$, and $h_i(\mathbf{x})$, respectively. Here g_i and h_i are i th component of g and h , respectively. Then \mathbf{x}^* solves \mathbb{P} if and only if it solves problem (1). With this method, potential obfuscation of problem data (cf. Definition 4) is directly linked to the function compositions, $\psi_0(f_0(\cdot))$, $\psi_i(g_i(\cdot))$, and $\varphi_i(h_i(\cdot))$.

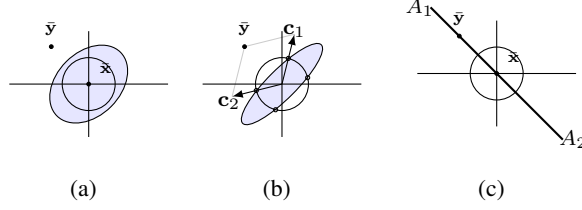
Scaling used in [5], [18]–[20] is a particular case of the objective and constrained transformation. In [15], authors handle their affine constraint functions, denoted $\mathbf{Ax} - \mathbf{b} = 0$, by using $\varphi(\mathbf{z}) = \mathbf{Bz}$, where \mathbf{B} is full column rank with the desired properties $\mathbf{Ax} - \mathbf{b} = 0 \Leftrightarrow \mathbf{B}(\mathbf{Ax} - \mathbf{b}) = 0$. Now, we present a new, yet an important example to convey the idea.

Example 2 (Localization from range measurements): Suppose the area $\mathcal{A} \in \mathbb{R}^2$ with $d(\mathcal{A}) = D$ contains N military deployed beacons, denoted $i = 1, \dots, N$, that serve civilian boats to locate their positions. Let $\bar{\mathbf{x}} \in \mathbb{R}^2$ denote the position of boat B. The range measurements, denoted $y_i \in \mathbb{R}$ and $\mathbf{a}_i \in \mathbb{R}^2$, $i = 1, \dots, N$, from the beacons are received by B to compute its location \mathbf{x} . In particular, y_i and \mathbf{a}_i are the distance and the unit vector, respectively, from B to beacon i . Assuming a linear measurement model with unknown noise, an estimate $\hat{\mathbf{x}}$ of \mathbf{x} is given by the solution of the problem, minimize $\|\mathbf{Ax} - \mathbf{y}\|_2$, where the variable is $\mathbf{x} \in \mathbb{R}^2$, and the problem data are $\mathbf{A} = -[\mathbf{a}_1, \dots, \mathbf{a}_N]^T$ and $\mathbf{y} = (y_1, \dots, y_N)$. However, due to security reasons, military beacons do not want to send their range measurements, which are simply the polar coordinates of their positions, with respect to B. Now the idea is to simply use the objective transformation $\psi_0(z) = z^2$, and let B solve

$$\text{minimize}_{\mathbf{x}} \mathbf{x}^T (\sum_{i=1}^N \mathbf{a}_i \mathbf{a}_i^T) \mathbf{x} + 2(\sum_{i=1}^N y_i \mathbf{a}_i^T) \mathbf{x} \triangleq \mathbf{x}^T \bar{\mathbf{A}} \mathbf{x} + 2\bar{\mathbf{y}}^T \mathbf{x}.$$

The beacons communicate with each other to compute $\bar{\mathbf{A}}$ and $\bar{\mathbf{y}}$, which is transmitted to B, i.e., the message $S = (\bar{\mathbf{A}}, \bar{\mathbf{y}})$. Let us now explore the privacy of this method on the input tuple $\mathbf{c} = y_1 \mathbf{a}_1$, the location of beacon 1. Without loss of generality, let $\bar{\mathbf{x}} = (0, 0)$. Suppose B knows by definition $\|\mathbf{a}_1\| = 1$, $\bar{\mathbf{A}} = \mathbf{a}_1 \mathbf{a}_1^T + \sum_{i=2}^N (\mathbf{a}_i \mathbf{a}_i^T)$, 3) $\bar{\mathbf{y}} = y_1 \mathbf{a}_1 + \sum_{i=2}^N y_i \mathbf{a}_i$, $y_i > 0$, and, of course, $N = N_0$, i.e., the auxiliary knowledge. Thus, $\mathcal{U}_c = \{\mathbf{c} = y_1 \mathbf{a}_1 \mid \|\mathbf{a}_1\| = 1, \mathbf{a}_i \in \mathcal{E}(\bar{\mathbf{A}}), y_1 \mathbf{a}_1 = (\bar{\mathbf{y}} - \sum_{i=2}^{N_0} y_i \mathbf{a}_i), \|\mathbf{a}_i\| = 1 \forall i = 2, \dots, N_0, y_i \leq D \forall i = 1, \dots, N_0\}$, where $\mathcal{E}(\bar{\mathbf{A}}) = \{\mathbf{a} \mid \mathbf{a}^T \bar{\mathbf{A}}^{-1} \mathbf{a} \leq 1\}$. Note that $\mathbf{a}_i \in \mathcal{E}(\bar{\mathbf{A}}) \forall i$ is a necessary condition, by definition of $\bar{\mathbf{A}}$. Let us next compute $\rho(\mathcal{U}_c)$.

We first build some intuition of \mathcal{U}_c . The figure above illustrates all possible cases, the circle and the shaded area represent the unit circle \mathcal{O} and the ellipse \mathcal{E} , respectively. So, $\mathbf{a}_1 \in \mathcal{O} \cap \mathcal{E}$. Note that y_i is simply the



magnitude of some vector in \mathbb{R}^2 , which is spanned by vectors in $\mathcal{O} \cap \mathcal{E}$. Now consider scenario 1, where $N_0 > 2$. The scenario 1-cases (a),(b) yield $\rho_{1,a,b}(\mathcal{U}_c) = (D, 1, 2)$, where the subscript of ρ denotes the scenario and the cases. The scenario 1-case (c) yields $\rho_{1,c}(\mathcal{U}_c) = (D, 1, 1)$. Now consider scenario 2, where $N_0 = 2$. Clearly, the scenario 2-case (a) doesn't exist. If scenario 2-case (b), $\mathcal{O} \cap \mathcal{E}$ is a set with 4 points, i.e., \mathbf{a}_1 is on one of the 4 points, cf. the open circles in (b). Moreover, because B knows \bar{y} , $\mathbf{c} = y_1 \mathbf{a}_1$ is either at \mathbf{c}_1 or \mathbf{c}_2 , i.e., $\mathcal{U}_c = \{\mathbf{c}_1, \mathbf{c}_2\}$, cf. figure (b). Thus, $\rho_{2,b}(\mathcal{U}_c) = (\|\mathbf{c}_1 - \mathbf{c}_2\|, 0.5, 1)$. If scenario 2-case (c), we similarly get $\mathcal{U}_c = A_1 A_2$, the line segment of length D , cf. figure (c), i.e., $\rho_{2,c}(\mathcal{U}_c) = (D, 1, 1)$. The ρ when $N_0 = 1$ is similarly computed.

Based on our privacy index, $\rho_{1,a,b} \geq \rho_{1,c} = \rho_{2,c} \geq \rho_{2,b}$. The first inequality clearly agrees with our intuition. The second equality agrees with our intuition in the sense that the ambiguity in c is identical in the associated two settings. The last inequality can be intuitively explained as follows: scenario 1,2-case (c) corresponds to a setting where the beacon measurements are linearly dependent, and therefore the beacon 1 can be any point along the line $A_1 A_2$, as opposed to scenario 2-case (b), where the beacon measurements are not linearly dependent.

B. Transformation via Change of Variables

Suppose the decision variable \mathbf{x} in problem (1) is in \mathbb{R}^n . Let $\phi : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a function such that $\text{range}(\phi) \supseteq \bigcap_i \text{dom}(g_i)$. Now consider the problem, denoted \mathcal{Q} , achieved by the change of variables $\mathbf{x} = \phi(\mathbf{z})$. Then \mathcal{Q} is equivalent to problem (1) in the sense that if \mathbf{z}^* solves \mathcal{Q} , then $\mathbf{x}^* = \phi(\mathbf{z}^*)$ solves problem (1). Note again that any potential obfuscation of problem data (cf. Definition 4) is directly linked to the function compositions, $f_0(\phi(\cdot))$, $g(\phi(\cdot))$, and $h(\phi(\cdot))$.

All the approaches in [5], [13], [14], [16]–[20] can be obtained as a particular case of the *change of variables*. In particular, those methods use an affine ϕ . Let us now give an example for a case where ϕ is not necessarily affine.

Example 3 (Resource allocation): Suppose an ammunition supply depot (ASD) wants to transfer N -types of nuclear-ammunition to the war field. Let β_i denote the radiation per unit volume of i th ammunition type. The overall radiation should be no greater than a specified value γ . The containers for each type i should be cubic of length x_i . The criterion for computing the volume of type i is determined by the penalty function α_i/x_i , where α_i denote the priority levels of the i th type. Suppose the ASD relies on a third party to compute the desired container box lengths $\mathbf{x}^* = (x_i^*)_{i=1:N}$, by minimizing $\sum_{i=1}^N (\alpha_i/x_i)$ subject to $\sum_{i=1}^N \beta_i x_i^3 \leq \gamma$ and $\mathbf{x} \geq \mathbf{0}$. The problem data is $\alpha_i > 0$, $\beta_i > 0$, and $\gamma > 0$. On the other hand, suppose the ASD does not want to reveal the input tuples (α_i, β_i) , $i = 1, \dots, N$.

What if ASD uses the change of variable $x_i = \phi_i(z_i) = \alpha_i \exp(z_i)$ for all i . Then it is sufficient to ask the third party (say B) to minimize $\sum_{i=1}^N \exp(-z_i)$ subject to $\sum_{i=1}^N s_i \exp(3z_i) \leq 1$, with variable $\mathbf{z} = (z_1, \dots, z_N)$ and problem parameters $s_i = (\beta_i \alpha_i^3 / \gamma)$. This method requires the transmission of the message $S = (s_i)_{i=1:N}$ to B. Suppose B knows the dependence of s_i on $\alpha_i, \beta_i, \gamma$ and the positivity of $\alpha_i, \beta_i, \gamma$. Therefore, $\mathcal{U}_{(\alpha_i, \beta_i)} = \{(\alpha_i, \beta_i) \mid s_i = \beta_i \alpha_i^3 / \gamma, \beta_i, \alpha_i, \gamma > 0\}$, which corresponds to $\rho(\mathcal{U}_{(\alpha_i, \beta_i)}) = (\infty, 1, 2)$. This is the highest possible privacy for an input tuple in \mathbb{R}^2 , cf. Proposition 1-(a).

One can readily apply hybrid variants of the *transformation via change of variables* and *transformation of objective and constraints*, cf. [18], [20].

C. Decomposition Methods

In this section, we highlight important aspects of decomposition methods for preserving privacy of problem data. In particular, the dual decomposition method is considered. Arguments, in the case of primal decomposition methods and ADMM method are similar.

We start by noting that the *decomposition* methods are applied whenever the problem structure inherently possesses some separability properties, where the original problem can be solved by coordinating smaller subproblems, one for each subsystem. Thus, instead of problem (1), the general problem we consider in this section is given by

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^N f_i(\mathbf{x}_i, \mathbf{y}_i; \mathbf{c}_i) \\ & \text{subject to} && (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{G}_i(\mathbf{c}_i), \quad i = 1, \dots, N \\ & && \mathbf{y}_i = \mathbf{E}_i \mathbf{z}, \quad i = 1, \dots, N, \end{aligned} \quad (4)$$

where the variables are $(\mathbf{x}_i, \mathbf{y}_i)_{i=1:N}$ and \mathbf{z} , and i denotes the subsystem index. Typically, \mathbf{x}_i , \mathbf{y}_i and \mathbf{z} are called local, interface, and global variables, respectively. The matrix \mathbf{E}_i is a 0-1 matrix that maps \mathbf{z} into the interface variables \mathbf{y}_i . This mapping accounts for the coupling of the problem and ensures consensus between subsystem interface variables and the global variables. The set $\mathcal{G}_i(\mathbf{c}_i)$ is the constraints at the subsystem i and is dependent on the private problem data \mathbf{c}_i . Let us now summarize the standard dual decomposition algorithm and see how it preserves privacy for private data \mathbf{c}_i .

Algorithm 1: Dual Decomposition

Given $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_N)$ such that $\mathbf{E}^T \boldsymbol{\lambda} = \mathbf{0}$. Set $k = 1$.

while (stopping criterion)

1. Subsystem i minimizes $f_i(\mathbf{x}_i, \mathbf{y}_i; \mathbf{c}_i) + \boldsymbol{\lambda}_i^T \mathbf{y}_i$ subject to $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{G}_i(\mathbf{c}_i)$, with $(\mathbf{x}_i, \mathbf{y}_i)$. Let $(\mathbf{x}_i^{(k)}(\mathbf{c}_i), \mathbf{y}_i^{(k)}(\mathbf{c}_i))$ denote the solution.
 2. Subsystems coordinate to update global \mathbf{z} , i.e., $\mathbf{z} := (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T (\mathbf{y}_1^{(k)}(\mathbf{c}_1), \dots, \mathbf{y}_N^{(k)}(\mathbf{c}_N))$.
 3. Subsystem i updates $\boldsymbol{\lambda}_i$ as, $\boldsymbol{\lambda}_i := \boldsymbol{\lambda}_i + \alpha_k (\mathbf{y}_i^{(k)}(\mathbf{c}_i) - \mathbf{E}_i \mathbf{z})$. Set $k := k + 1$.
-

In the algorithm above, α_k is an appropriate step size [23]. As far as the privacy of the problem data is concerned, the crucial step of the algorithm is step 2. This is because the coordination requires exchange of messages, which are dependent on the problem data. In particular, the i th subsystem requires revealing message, denoted S_i^{dual} , where $S_i^{\text{dual}} = (\mathbf{y}_i^{(k)}(\mathbf{c}_i))_{k=1:T}$ and T is the number of iteration before the algorithm termination. Note that $\mathbf{y}_i^{(k)}(\mathbf{c}_i)$ is, in fact, the subgradients of dual master objective function [23]. Therefore, any potential obfuscation of problem data of subsystem i is directly linked to the computed subgradients until the algorithm termination, i.e., S_i^{dual} . An example is given below.

Example 4 (Average consensus): Consider the computation of the arithmetic average \bar{c} of $N > 2$ scalars, denoted c_1, \dots, c_N , where each c_i is private to subsystem i . Note that \bar{c} is simply the argument y that minimizes $\sum_{i=1}^N (y - c_i)^2$. This minimization can be equivalently posed as

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^N (y_i - c_i)^2 \\ & \text{subject to} && y_i = z, \quad i = 1, \dots, N, \end{aligned} \quad (5)$$

with variables $(y_i)_{i=1:N}$ and z . Let us next show, without loss of generality, how Algorithm 1 achieves privacy for input tuple c_1 of subsystem 1, given adversary is N .

Note that the step 1 of Algorithm 1 is $y_i^{(k)}(c_i) := (c_i - \lambda_i/2)$. (6)

The step 2 is simply the average $z := (1/N) \sum_{i=1}^N (c_i - \lambda_i/2)$ and step 3 is $\lambda_i := \lambda_i + \alpha_k (y_i^{(k)}(c_i) - z)$. Thus, the message exposed by subsystem 1 to subsystem N is $S_1 = (y_1^{(k)}(c_1))_{k=1:T}$, where T is the total iterations.

Without explicitly describing the relation \mathcal{R}_{c_1} (or equivalently the uncertainty set \mathcal{U}_{c_1}), we use an alternative method, where the associated $\rho(\mathcal{U}_{c_1})$ is computed by construction. Now note that if for $i = 1, \dots, N-1$, c_i and λ_i satisfy (6) for all $k = 1, \dots, T$, so is $\hat{c}_i(\theta_i) = (c_i - \theta_i, \lambda_i - 2\theta_i)$, where $\theta = (\theta_i)_{i=1:N-1}$ is chosen such that $\mathbf{1}^T \theta = 0$. Indeed, θ_1 here parameterizes the uncertainty set \mathcal{U}_{c_1} . In particular, $\mathcal{U}_{c_1} = \{c_1 - \theta_1 \mid \mathbf{1}^T \theta = 0\}$, which yields $\rho(\mathcal{U}_{c_1}) = (\infty, 1, 1)$. This is the highest possible privacy for an input tuple in \mathbb{R} , cf. Proposition 1-a).

Even though, we assume that subsystem N is a passive adversary (cf. Definition 2), it is easily seen that, even if r number of subsystems collude, $\rho(\mathcal{U}_{c_1}) = (\infty, 1, 1)$ remains intact, given $r < N-1$. In other words, as long as subsystem 1 has at least one reliable subsystem j , $j \neq 1$, $\rho(\mathcal{U}_{c_1}) = (\infty, 1, 1)$.

IV. CONCLUSIONS

In this paper, a mechanism to quantify the privacy of a broader class of optimization approaches was proposed. We defined a one-to-many relation, which relates a given adversarial observed message to an uncertainty set of the problem data. The privacy measure was then designed based on the uncertainty sets. The properties of the proposed privacy measure was analyzed. From the definition it is clear that optimization based approaches lacks robustness to adversarial attacks, such as those involving side information. Our privacy definition clearly model those

intrinsic properties of optimization approaches. The key ideas were illustrated by examples. An important possible extensions is to explore ways of designing optimization approaches to guarantee a specified privacy level.

REFERENCES

- [1] O. Goldreich, *The Foundations of Cryptography*, vol. 2, Cambridge University Press, Cambridge, UK, 2004.
- [2] A. C. Yao, “Protocols for secure computations,” in *Proc. IEEE Symp. Found. of Comp. Science*, Chicago, USA, Nov. 1982, pp. 160–164.
- [3] I. Damgård, “Theory and practice of multiparty computation,” in *Int. Conf. in Sec. and Crypt. Netw.*, Maiori, Italy, Sept. 2006, pp. 360–364.
- [4] T. Toft, “Solving linear programs using multiparty computation,” *Financ. Crypt. and Data Sec. LNCS*, pp. 90–107, 2009.
- [5] A. Bednarz, *Methods for Two-Party Privacy-Preserving Linear Programming*, Ph.D. thesis, Discipline of Applied Mathematics, School of Mathematical Sciences, The University of Adelaide, 2012.
- [6] C. Dwork, F. McSherry, K. Nissim, and A. Smith, *Theory of Cryptography*, chapter Calibrating Noise to Sensitivity in Private Data Analysis, pp. 265–284, Lec. Notes in Comp. Sci. Springer, 2006.
- [7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization,” *J. of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011.
- [8] D. Kifer, A. Smith, and A. Thakurta, “Private convex empirical risk minimization and high-dimensional regression,” *JMLR: Workshop and Conf. Proceedings*, vol. 23, pp. 25.1–25.40, 2012.
- [9] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, “Differentially private approximation algorithms,” in *Int. Symp. on Discrete Alg.*, Austin, Texas, Jan. 2010.
- [10] J. L. Ny and G. J. Pappas, “Differentially private filtering,” in *IEEE Conf. on Dec. and Control*, Grand Wailea, Maui, Hawaii, Dec. 10–13 2012, pp. 3398–3403.
- [11] J. L. Ny and G. J. Pappas, “Differentially private kalman filtering,” in *Commun., Control, and Comp. (Allerton), 2012 50th Annual Allerton Conference on*, University of Illinois Urbana-Champaign, IL, USA, Oct. 1–5 2012, pp. 1618–1625.
- [12] Z. Huang, S. Mitra, and N. Vaidya, “Differentially private distributed optimization, submitted,” 2014, [Online]. Available: <http://arxiv.org/abs/1401.2596>.
- [13] O. L. Mangasarian, “Privacy-preserving linear programming,” *Opt. Let.*, vol. 5, no. 1, pp. 165–172, Feb. 2011.
- [14] O. L. Mangasarian, “Privacy-preserving linear and nonlinear approximation via linear programming,” *Opt. Meth. SW.*, pp. 1–10, Oct. 2011.
- [15] O. L. Mangasarian, “Privacy-preserving horizontally partitioned linear programs,” *Opt. Let.*, vol. 6, no. 3, pp. 431–436, Mar. 2012.
- [16] O. L. Mangasarian and E. W. Wild, “Privacy-preserving classification of horizontally partitioned data via random kernels,” in *Proc. Int. Conf. on Dat. Mining*, Las Vegas, USA, July 2008, pp. 473–479.
- [17] O. L. Mangasarian, E. W. Wild, and G. M. Fung, “Privacy-preserving classification of vertically partitioned data via random kernels,” *ACM Trans. on Knowl. Discov. from Data*, vol. 2, no. 12, Oct. 2008.
- [18] J. Dreier and F. Kerschbaum, “Practical privacy-preserving multiparty linear programming based on problem transformation,” in *Proc. IEEE Int. Conf. on Info. Privacy, Secu., Risk and Trust*, Boston, USA, Oct. 2011, pp. 916–924.
- [19] A. Bednarz, N. Bean, and M. Roughan, “Hiccups on the road to privacy-preserving linear programming,” in *Proc. ACM Works. on Privacy in the Electron. Society*, Chicago, IL, USA, Nov. 2009, pp. 117–120.
- [20] C. Wang, K. Ren, and J. Wang, “Secure and practical outsourcing of linear programming in cloud computing,” in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 820–828.
- [21] J. N. McDonald and N. A. Weiss, *A Course in Real Analysis*, Elsevier, Waltham, MA, USA, second edition, 2013.
- [22] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [23] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Ath. Sci., Belmont, MA, 2nd edition, 1997.