52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Per-se Privacy Preserving Solution Methods Based on Optimization

P. C. Weeraddana⋆, G. Athanasiou⋆, C. Fischione⋆, and J. S. Baras⋆⋆
⋆KTH Royal Institute of Technology, Stockholm, Sweden, ⋆⋆ University of Maryland, MD 20742, USA
{chatw, georgioa, carlofi}@kth.se, baras@umd.edu

Abstract— **Ensuring privacy is an essential requirement in various contexts, such as social networks, healthcare data, e-commerce, banks, and government services. Here, different entities coordinate to address specific problems where the *sensitive* problem data are distributed among the involved entities and no entity wants to publish its data during the solution procedure. Existing privacy preserving solution methods are mostly based on cryptographic procedures and thus have the drawback of substantial computational complexity. Surprisingly, little attention has been devoted thus far to exploit mathematical optimization techniques and their inherent properties for preserving privacy. Yet, optimization based approaches to privacy require much less computational effort compared to cryptographic variants, which is certainly desirable in practice. In this paper, a *unified framework* for transformation based optimization methods that ensure privacy is developed. A general definition for the privacy in the context of transformation methods is proposed. A number of examples are provided to illustrate the ideas. It is concluded that the theory is still in its infancy and that huge benefits can be achieved by a substantial development.**

I. INTRODUCTION

Privacy and security are central requirements in many real-world problems [1]–[12]. Several real-world optimization problems involve parties or nodes interacting via some networks that must collaborate to solve an optimization problem for mutual benefit. For example, independent hospitals would like to coordinate for diagnostic decision making based on their existing patient records. In the business sector, independent companies need to interact for completing a common business and thus have to work together to optimize their joint operations. Normally, optimization solvers require much public data sharing among the parties, which may substantially hinder the cooperation for optimization due to privacy concerns (e.g., privacy for patients' records). The fundamental question is how to solve optimization problems among parties that would receive much benefit by collaboration and yet are unwilling to share their data without preserving privacy.

Cryptography is the standard approach to preserve privacy in distributed optimization solvers [4]. Cryptographic primitives include secure multiparty computations [1], [3], [6], pseudo random generators [13], and homomorphic encryption [14]. These methods use cryptographic tools, such as zero-knowledge [15], oblivious transfer [16], oblivious evaluation of polynomials [17], secret sharing [18], and threshold cryptography [19]. In general, the area of cryptography-based privacy preserving optimization is well investigated [4]. In the context of optimization problems, cryptographic tools are used to securely perform iterations of well known *simplex* algorithm and *interior-point* algorithm so that sensitive data is not disclosed during the iterations to compute the solution, see [20]–[22] for secure simplex variants and [23] for secure variants of interior-point method. In terms of security, cryptographic methods are desirable, though they are unfavorable in terms of computational complexity and efficiency [23]. In particular, cryptography may introduce substantial overhead among the nodes due to the exchange of security information and coordination. Moreover, cryptography is prone to attacks by third parties who may inadvertently own the cryptographic keys.

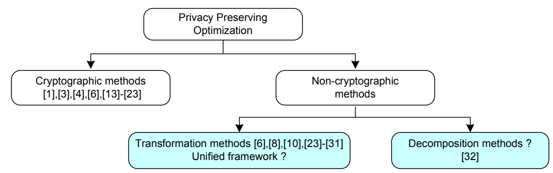

Fig. 1. Classification of privacy preserving methods

Non-cryptographic methods have attracted the interest of the research community [6], [8], [10], [23]–[31], which are essentially based on algebraic transformations. We refer to those approaches as *transformation methods* in general. The key idea of these methods is to use algebraic manipulations to disguise the original problem into an equivalent problem so that the private data of each are hidden. However, in these papers, only some specific problems have been considered and no attempts have been made to establish a systematic approach. As a result, even unintended mistakes have emerged in [28], [29] as pointed out by [26].

In this paper, we consider non-cryptographic approaches, as opposed to treatments based on well investigated cryptographic primitives. In particular, we investigate the transformation methods, see Fig. 1. We show that the transformation methods possess many appealing merits, which are desirable in practice, e.g., efficiency, scalability, natural (geographical) distribution of problem data. More importantly, they can be per-se privacy preserving without requiring any extra coordination or overhead. In addition to transformation methods, decomposition methods can also be used for distributed

This research was supported by EU projects Hycon2, Hydrobionets, VR project In network Optimization, the National Science Foundation (NSF) under grant award CNS-1035655, and the National Institute of Standards and Technology (NIST) grant award 70NANB11H148.

optimization with privacy, see Fig. 1. We proposed a detailed treatment of decomposition methods in [32]. In the following, we summarize more in detail the contribution of this paper.

A. Our Contributions

The main contributions of this paper can be summarized as follows:

1) We develop a unified framework where the *existing* privacy preserving solution approaches based on transformation methods can be derived. Absence of a systematic approach to design privacy preserving transformation based methods (e.g., [6], [8], [10], [23]–[31]) limits the scope of applicability of such methods and sometimes results in unintended mistakes (e.g., [26], [28], [29]). It is desirable to have a canonical framework where all existing approaches can be included, because this allows to develop standard proof techniques for proving the privacy properties.
2) We give a general definition for privacy, which allows the quantification of privacy of transformation based methods.
3) We present several examples to highlight the importance of our generalized transformation methods for privacy preserving optimization and to illustrate the proposed privacy definitions.

The rest of the paper is organized as follows. In Section II we present some basic definitions that are useful for describing the properties of privacy preserving optimization. A unified framework to model the transformation methods is presented in Section III. Conclusions are given in Section IV.

B. Notations

Boldface lower case and upper case letters represent vectors and matrices, respectively, and calligraphy letters represent sets. The set of real n-vectors is denoted by $\mathbb{R}^n$, the set of real $m \times n$ matrices is denoted by $\mathbb{R}^{m\times n}$. We denote by N the set of non-negative integers, i.e., $\mathbb{N} = \{0, 1, \ldots\}$. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$. The superscript $(\cdot)^{\mathrm{T}}$ stands for transpose. Vectors and matrices are delimited with square brackets, with the components separated by space. The ith submatrix of a matrix is denoted by a subscript. We use *parentheses* to construct column vectors from comma separated lists, e.g., $(\mathbf{a}, \mathbf{b}, \mathbf{c}) = [\mathbf{a}^{\mathrm{T}}\ \mathbf{b}^{\mathrm{T}}\ \mathbf{c}^{\mathrm{T}}]^{\mathrm{T}}$.

II. PER-SE PRIVACY PRESERVING DEFINITIONS

Many standard privacy/security conventional definitions are already adopted in cryptographic literature, see for example [6], [33]–[37]. However, such definitions cannot be directly applied or adopted for non-cryptographic approaches (in particular optimization based approaches). The reason is that the key mechanisms used for preserving the privacy in cryptographic protocols are entirely different from those that are to be used in optimization based approaches. Theoretical foundations for defining the privacy of optimization methods deserve attention in their own right. However, to highlight the appealing privacy preserving properties associated with optimization based approaches, and to provide a cohesive discussion, we give some *basic* definitions in the sequel.

Definition 1 (Optimization problem): We consider the following standard notation to describe the problem of finding a point $\mathbf{x}$ that minimizes the function $f_0(\mathbf{x})$ subject to a number of inequality and equality constraints:

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, q \\ & h_i(\mathbf{x}) = 0,\ i = 1, \ldots, p\ . \end{array} \tag{1}$$

We call (1) an optimization problem. Here $f_0 : \mathbb{R}^n \to \mathbb{R}$ is called the *objective function*, $f_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, q$ are called *inequality constraints* with the associated inequality constraint functions $f_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, q$, $h_i(\mathbf{x}) = 0,\ i = 1, \ldots, p$ are called *equality constraints* with the associated equality constraint functions $h_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, p$, and $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is called the *optimization variable* or the decision variable [38].

Definition 2 (Convex optimization problem): A *convex* optimization problem is

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, q \\ & \mathbf{C}\mathbf{x} - \mathbf{d} = \mathbf{0}\ , \end{array} \tag{2}$$

where the functions $f_i,\ i = 0, \ldots, q$ are *convex* and $h_i,\ i = , \ldots, p$ are affine, i.e., the equality constraint functions are given by $\mathbf{C} \in \mathbb{R}^{p\times n}$ with $\operatorname{rank}(\mathbf{C}) = p$ and $\mathbf{d} \in \mathbb{R}^p$. The optimization variable is $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Typically, we have $p < n$ in practice, otherwise the only *potential* solution, if it exists, is $\mathbf{C}^\dagger\mathbf{d}$, where $\mathbf{C}^\dagger$ is called the pseudo-inverse of $\mathbf{C}$ (see [38, § A.5.4]).

Definition 3 (K-party environment): A set of K parties is called a K-party environment.

Throughout this paper, we consider problems of the form (2), which are to be solved in a K-party environment via coordination among the parties. Note that the treatment can be generalized to problems of the form (1) in a straightforward manner, and therefore we restrict ourselves to problem (2) for clarity. The global variable $\mathbf{x}$, objective function f_0, inequality constraint functions $f_i\ i = 1, \ldots, q$, equality constraint functions $\mathbf{C}\mathbf{x} - \mathbf{d} = \mathbf{0}$, different subsets of components of $\mathbf{x}$, different subsets of functions, and/or function definitions themselves can be spread out among K parties. We use the terms *ownership* and *private data* to indicate the spreading of such functions, data, etc. among K parties. Note that the terms *party* and *entity* are often used interchangeably in this paper.

Example 1: Consider the case $f_0(\mathbf{x}) = \sum_{\mathbf{i=1}}^{\mathbf{K}} \mathbf{c}_{\mathbf{i}}^{\mathrm{T}}\mathbf{x}_{\mathbf{i}}$, where $\mathbf{c}_i \in \mathbb{R}^{n_i}$, $\mathbf{x}_i \in \mathbb{R}^{n_i}$ with $\sum_{i=1}^{K} n_i = n$, and $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$. Here, the objective function f_0 and the optimization variable $\mathbf{x}$ is spread out among K parties, such that $\mathbf{c}_i$ and $\mathbf{x}_i$ are owned by party i.

Definition 4 (Inputs and outputs of a convex problem): Consider the convex problem (2). We call the *set of problem parameters*, i.e., $\mathbf{C}, \mathbf{d}$, and those required to define the functions f_0, f_i themselves, as *inputs* of problem (2). Moreover, we call the solution and the optimal value of problem (2) as *outputs*.

Example 2: Consider the following linear program:

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \ , \end{array} \tag{3}$$

where the variable is $\mathbf{x} \in \mathbb{R}^n$ and problem data are $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, and $\mathbf{b} \in \mathbb{R}^m$. Suppose $\mathbf{x}^\star$ solves the problem. The parameters $\{\mathbf{c}, \mathbf{A}, \mathbf{b}\}$ are the inputs of the problem and $\{\mathbf{x}^\star, p^\star\}$ are the outputs of the problem, where $p^\star = \mathbf{c}^{\mathrm{T}}\mathbf{x}^\star$.

Definition 5 (Attacker model, Passive adversary): In a multi-party environment, an entity involved in solving a global optimization problem of the form (2), or even a third party is called a passive adversary, if it taps the communication lines of the multi-party environment to obtain messages exchanged during different stages of the solution method, keeps a record of all information it receives, and tries to discover others' private data.

The definition above is similar to the passive eavesdroppers considered in [39, § 5.1-5.3] and to the *good* and the *passive* adversary models given in [40] and [41], respectively. In a multi-party environment, there can be more than one adversaries. Unlike active adversaries (e.g., *Dolev-Yao* attackers [3]), as the name suggests, the passive adversaries are limited to eavesdropping and are not capable of taking any active actions, such as sending, deleting, or modifying fields in messages being sent as part of some solution method execution. We consider the *passive adversary* model throughout this paper to exemplify the privacy properties of non-cryptographic approaches.

Definition 6 (Adversarial knowledge): The set of information that an adversary might exploit to discover the input and/or the output of problem (2) is called the adversarial knowledge. This set can encompass eavesdropped measures of input/output elements, transformed variants of input/output elements, statements to infer the properties of the input/output, and others, see Example 3 for an illustration.

Let us finally give a formal definition to quantify the privacy of transformation methods. To do this, we consider a pair of parameters $(\xi, \eta) \in [0, 1) \times \mathbb{N}$. In particular, ξ quantifies the privacy of the input and the output of problem (2).[1] Roughly speaking, for a given adversarial knowledge, when $\xi = 0$, there is *no* protection against the adversary and when $\xi = 1$, there is *perfect* protection against the adversary.[2] Values of the parameter ξ between 0 and 1 correspond to moderate cases. On the other hand, for the same adversarial knowledge, the parameter η indicates a measure of the spread of the uncertainty of the private data. For example, the higher the value of η, the more effective the transformation used to disguise the private data. We refer to (ξ, η) as the *privacy index*.

As we have already remarked, the privacy index depends on the adversarial knowledge that compromises security of the transformation methods. Therefore, when the privacy index (ξ, η) is computed, it is always linked to a specified adversarial knowledge. The definitions considered in [23] for 2-party environments can be considered as particular cases of our following definition.

Definition 7 (Input privacy index, $(\xi^{\mathrm{in}}, \eta^{\mathrm{in}})$): Let $\mathcal{C}^{\mathrm{in}}$ denote the input of problem (2). Suppose the mechanism for solving the problem creates an obfuscation of the original element c of $\mathcal{C}^{\mathrm{in}}$ by using functions of the form $f_c^{\mathrm{in}} : \mathcal{C}^{\mathrm{in}} \to \mathcal{G}^{\mathrm{in}}$, where $\mathcal{G}^{\mathrm{in}} \subset \mathcal{K}$ and $\mathcal{K}$ denote the set of adversarial knowledge. Given the adversarial knowledge $\mathcal{K}$, let

$$\xi^{\mathrm{in}}(c) = 1 - 1/N_{\mathcal{K}}^{\mathrm{in}} \ , \tag{4}$$

where $N_{\mathcal{K}}^{\mathrm{in}} \geq 1$ is the cardinality of the uncertainty set

$$\mathcal{U}^{\mathrm{in}}(c) = \{c \,|\, f_c^{\mathrm{in}}(c) = k, \ f_c^{\mathrm{in}} \text{ is arbitrary}, \ \mathcal{K}\} \ . \tag{5}$$

Moreover, let $\eta^{\mathrm{in}}(c)$ be the *affine dimension* [38, § 2.1.3] of the set $\mathcal{U}^{\mathrm{in}}(c)$. We call $(\xi^{\mathrm{in}}(c), \eta^{\mathrm{in}}(c))$ the input privacy index of $c \in \mathcal{C}^{\mathrm{in}}$ in the presence of adversarial knowledge $\mathcal{K}$.

We use the convention that $N_{\mathcal{K}}^{\mathrm{in}}$ is infinity, whenever the set (5) is uncountable [43, Def. 2.4(d)]. Let us next give an example to illustrate the idea of input privacy.

Example 3: Consider once again the linear program (3), which is to be solved by Alice and Bob. The input ownership is as follows: Alice owns $\mathbf{c}$ and Bob owns $\{\mathbf{A}, \mathbf{b}\}$. Now suppose they use the following solution procedure: instead of the input $\mathbf{c}$, Alice uses $\hat{\mathbf{c}}$, where $\hat{\mathbf{c}} = \alpha\mathbf{c}$ and α is a positive scalar known by Alice only. Similarly, instead of the input $\{\mathbf{A}, \mathbf{b}\}$, Bob uses $\{\hat{\mathbf{A}}, \hat{\mathbf{b}}\}$, where $\hat{\mathbf{A}} = \beta\mathbf{A}$, $\hat{\mathbf{b}} = \beta\mathbf{b}$, and β is a positive scalar known by Bob only. The result is the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \hat{\mathbf{c}}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & \hat{\mathbf{A}}\mathbf{x} \leq \hat{\mathbf{b}} \\ & \mathbf{x} \geq \mathbf{0} \ , \end{array} \tag{6}$$

where the variable is $\mathbf{x}$. Problem (3) and (6) are clearly equivalent, because $\hat{\mathbf{A}}\mathbf{x} \leq \hat{\mathbf{b}} \Leftrightarrow \mathbf{A}\mathbf{x} \leq \mathbf{b}$ and minimizing $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ is identical to minimizing $\hat{\mathbf{c}}^{\mathrm{T}}\mathbf{x} = \alpha\mathbf{c}^{\mathrm{T}}\mathbf{x}$ with $\alpha > 0$. Either Alice or Bob can use any linear programming solver for solving the problem.

Suppose now that Bob is a passive adversary to Alice and vice-a-versa. Moreover, assume that Bob's knowledge $\mathcal{K}_{\mathrm{Bob}}$ of Alice's data $\mathbf{c}$ is

$$\mathcal{K}_{\mathrm{Bob}} = \big\{\hat{\mathbf{c}}, \{\exists\alpha > 0 \text{ s.t. } \hat{\mathbf{c}} = \alpha\mathbf{c}\}\big\} \tag{7}$$

Note that the first element of $\mathcal{K}_{\mathrm{Bob}}$ is the transformed version of $\mathbf{c}$ available to Bob. The second component can be considered as an statement learned by Bob. With this knowledge, the uncertainty set of $\mathbf{c}$ is given by

$$\mathcal{U}^{\mathrm{in}}(\mathbf{c}) = \{\mathbf{c} | \mathbf{c} = (1/\alpha)\hat{\mathbf{c}}, \alpha > 0\} \tag{8}$$

From Definition 7, we can conclude that the corresponding privacy index $(\xi^{\mathrm{in}}(\mathbf{c}), \eta^{\mathrm{in}}(\mathbf{c})) = (1, 1)$. The first element $\xi^{\mathrm{in}}(\mathbf{c}) = 1$ implies the cardinality of the uncertainty set $\mathcal{U}^{\mathrm{in}}(\mathbf{c})$ is uncountable, which is indeed desirable as far as the privacy of Alice data $\mathbf{c}$ is concerned. Nevertheless, $\eta^{\mathrm{in}}(\mathbf{c}) = 1$ means that the uncertainty set $\mathcal{U}^{\mathrm{in}}(\mathbf{c})$ is restricted to a line in the Euclidean space. Similarly we can show that $(\xi^{\mathrm{in}}(\mathbf{A}), \eta^{\mathrm{in}}(\mathbf{A})) = (\xi^{\mathrm{in}}(\mathbf{b}), \eta^{\mathrm{in}}(\mathbf{b})) = (1, 1)$.

Definition 8 (Output privacy index, $(\xi^{\mathrm{out}}, \eta^{\mathrm{out}})$): This is very similar to Definition 7, except that the output of problem (2) is considered instead of the input, see [32].

More examples will be discussed in section III. We note that a comprehensive treatment of a set of mathematical definitions is out of the scope of this paper. We believe that the basic definitions (5)-(8) given above are sufficient for investigating the appealing aspects and properties of privacy preserving optimization approaches presented in this paper.

[1] The *formal* definition of ξ is inspired by the *informal* definition of security considered in [6, p. 128].

[2] The perfect protection is usually impossible to be realized because adversaries coevolve, see [42, p. 48]

III. Transformation Based Methods for Privacy Preserving

Let (2) be the original problem to be solved in a privacy preserving manner. Now consider the situation where the entities who own the data of problem (2) rely on an untrusted party or parties during the solution method. The untrusted parties are assumed to be passive adversaries, see Definition 5. Each participating entity wants protection against passive adversaries. This is realized by disguising problem (2) into another, where the transformation methods play a key role.

Transformation methods are directly based on the notion of equivalence of optimization problems [38, § 4.1.3]. There are many general transformations that yield equivalent problems. In the sequel, we present *two* transformations, which we believe are the most important and useful for privacy preserving optimization to unify all special cases considered in [8], [10], [23]–[31]. To simplify the presentation, we denote by $\mathcal{D}$ the set of points for which the objective and all constraint functions are defined, or domain of problem (2), i.e., $\mathcal{D} = \bigcap_{i=0}^{q} \mathrm{dom}\, f_i \cap \mathbb{R}^n$. For many considered problem formulations in this paper, the domain $\mathcal{D} = \mathbb{R}^n$. Let $\mathbf{x}^\star$ denote the solution and $p^\star$ denote the optimal value of problem (2).

A. Transformation via Change of Variables

The following proposition establishes the equivalence between two problems by performing a transformation via change of variables.

Proposition 1: Let $\phi : \mathbb{R}^m \to \mathbb{R}^n$ be a function, with image *covering* the problem domain $\mathcal{D}$. Now consider the following change of variables:

$$\mathbf{x} = \phi(\mathbf{z}) \ . \tag{9}$$

The resulting problem is given by

$$\begin{array}{ll} \text{minimize} & f_0(\phi(\mathbf{z})) \\ \text{subject to} & f_i(\phi(\mathbf{z})) \le 0, \ i = 1, \dots, q \\ & \mathbf{C}\phi(\mathbf{z}) - \mathbf{d} = \mathbf{0} \ , \end{array} \tag{10}$$

where the variables are $\mathbf{z} \in \mathbb{R}^m$. Suppose $\mathbf{x}^\star$ solves problem (2). Then $\mathbf{z}^\star = \phi^{-1}(\mathbf{x}^\star)$ solves problem (10). Moreover, if $\mathbf{z}^\star$ solves problem (10), then $\mathbf{x}^\star = \phi(\mathbf{z}^\star)$ solves problem (2).

Proof: See [32]. ■

Interestingly, such an equivalence can be exploited to ensure privacy, as we see next.

Privacy Properties

If the function ϕ is chosen appropriately, the transformation via change of variables can be used to achieve *input privacy* (see Definition 7) for many inputs, except for $\mathbf{d}$, via the function compositions:

$$\hat{f}_i(\mathbf{z}) = f_i(\phi(\mathbf{z})) \ , \ \mathrm{dom}\, \hat{f}_i = \{\mathbf{z} \in \mathrm{dom}\, \phi \mid \phi(\mathbf{z}) \in \mathrm{dom}\, f_i\} \ ,$$

$$\hat{h}_i(\mathbf{z}) = \mathbf{C}\phi(\mathbf{z}) - \mathbf{d} \ , \ \mathrm{dom}\, \hat{h}_i = \{\mathbf{z} \in \mathrm{dom}\, \phi \mid \phi(\mathbf{z}) \in \mathbb{R}^n\} \ .$$

The *output privacy* (see Definition 8) for the optimal solution is attained by the definition of ϕ, see (9). In the sequel, we highlight these privacy properties by some examples. First, let's see some choices for the function ϕ.

Example 4:

- **Scaling:** Here, we simply use the change of variable $\mathbf{x} = \phi(\mathbf{z}) = a\mathbf{z}$, where a is a scalar.
- **Translation:** Here, we use the following change of variable, $\mathbf{x} = \phi(\mathbf{z}) = \mathbf{z} + \mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^n$.
- **Affine transformation:** This is a generalization of both scaling and translation. Specifically, we use the following change of variable, $\mathbf{x} = \phi(\mathbf{z}) = \mathbf{B}\mathbf{z} + \mathbf{a}$, where $\mathbf{B} \in \mathbb{R}^{n \times m}$ is a *full* rank matrix with $\mathrm{rank}(B) = n$ and $\mathbf{a} \in \mathbb{R}^n$. Thus, a particular inverse transformation $\phi^{-1} : \mathbb{R}^n \to \mathbb{R}^m$ is given by

$$\mathbf{z} = \phi^{-1}(\mathbf{x}) = \mathbf{B}^\dagger \mathbf{x} - \mathbf{B}^\dagger \mathbf{a} \ ,$$

  where $\mathbf{B}^\dagger = \mathbf{B}^{\mathrm{T}}(\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}$, which is typically known as pseudo-inverse or Moore-Penrose inverse.
- **Nonlinear transformations:** One example is as follows, $x_i = \phi_i(z_i) = a_i \exp(z_i)$, where $a_i > 0$.

We see that all the approaches [8], [23]–[31] have used *change of variables* (affine transformations) as one of the mechanism for preserving privacy in their proposed solution methods. To illustrate this, we consider some original examples, as well as few key examples from the literature.

Example 5 (Computation outsourcing in the cloud [27]): Here we have a 2-party environment. Suppose a cloud customer (party 1) wants to outsource to the cloud (party 2) his linear program

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{B}\mathbf{x} \ge \mathbf{0} \ , \end{array} \tag{11}$$

where the variable is $\mathbf{x} \in \mathbb{R}^n$ and the problem data are $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, nonsingular $\mathbf{B} \in \mathbb{R}^{n \times n}$, and $\mathbf{b} \in \mathbb{R}^m$. The cloud it self is assumed to be the passive adversary. The customer does not want to reveal problem data $\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{B}$ and the solution $\mathbf{x}^\star$ of the problem to the cloud, i.e., input privacy for $\{\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{B}\}$ and output privacy for $\mathbf{x}^\star$ is the requirement.

The cloud customer then uses the affine transformation [27, § III-C]

$$\mathbf{x} = \phi(\mathbf{z}) = \mathbf{M}\mathbf{z} - \mathbf{r} \ , \tag{12}$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is a nonsingular matrix and $\mathbf{r} \in \mathbb{R}^n$ is a vector, both known by the customer only. The equivalent problem outsourced by the customer to the cloud is given by

$$\begin{array}{ll} \text{minimize} & \hat{\mathbf{c}}^{\mathrm{T}}\mathbf{z} \\ \text{subject to} & \hat{\mathbf{A}}\mathbf{z} = \hat{\mathbf{b}} \\ & \hat{\mathbf{B}}\mathbf{z} \ge \mathbf{0} \ , \end{array} \tag{13}$$

where the variable is $\mathbf{z} \in \mathbb{R}^m$ and the problem input is $\hat{\mathbf{c}} = \mathbf{M}^{\mathrm{T}}\mathbf{c}$, $\hat{\mathbf{A}} = \mathbf{A}\mathbf{M}$, $\hat{\mathbf{b}} = \mathbf{b} + \mathbf{A}\mathbf{r}$, and $\hat{\mathbf{B}} = \mathbf{B}\mathbf{M}$. The cloud computes the optimal solution of problem (13), which we denote by $\mathbf{z}^\star$.

The sensitive inputs of problem (11), $\{\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{B}\}$, cannot be recovered by a potential adversary or the cloud because the matrix $\mathbf{M}$ and the column vector $\mathbf{r}$ are not known to the cloud. For the same reasons, the cloud cannot construct the sensitive output $\mathbf{x}^\star$ by using $\mathbf{z}^\star$. Thus, the solution procedure yields both input privacy and output privacy.

To see the changes in privacy indexes due to the changes in adversarial knowledge, let us consider the variable transformation (12) used in Example 5. In particular, we compute the input privacy index of the sensitive input $\mathbf{A}$ with different adversarial knowledge. Similar computations can be carried out to quantify the privacy of other inputs and outputs of problem (11) as well.

Scenario 1: Suppose that the clouds' knowledge $\mathcal{K}_{\text{cloud}}$ of the cloud customer's data $\mathbf{A} \in \mathbb{R}^{m \times n}$ is

$$\mathcal{K}_{\text{cloud}} = \left\{ \hat{\mathbf{A}}, \{\exists\ \mathbf{M} \in \mathbb{R}^{n \times n} \text{ s.t } \hat{\mathbf{A}} = \mathbf{A}\mathbf{M}\} \right\} \ . \tag{14}$$

With this knowledge, the uncertainty set of $\mathbf{A}$ is given by

$$\mathcal{U}^{\rm in}(\mathbf{A}) = \{\mathbf{A}|\mathbf{A}\mathbf{M} = \hat{\mathbf{A}}, \mathbf{M} \in \mathbb{R}^{n\times n}\} \ .$$

We denote by $(\xi_0^{\rm in}(\mathbf{A}), \eta_0^{\rm in}(\mathbf{A}))$ the corresponding input privacy index. From Definition 7, we conclude that $(\xi_0^{\rm in}(\mathbf{A}), \eta_0^{\rm in}(\mathbf{A})) = (1, nm)$, where $\xi_0^{\rm in}(\mathbf{A}) = 1$ follows from the fact that the cardinality of the uncertainty set $\mathcal{U}^{\rm in}(\mathbf{A})$ is uncountable and $\eta_0^{\rm in}(\mathbf{A}) = nm$ follows from the fact that the matrix $\mathbf{A}$, as seen by the cloud, is arbitrary, and therefore the affine dimension of $\mathcal{U}^{\rm in}(\mathbf{A})$ is simply the number of elements in $\mathbf{A}$, i.e., nm.

Scenario 2: Next consider the case where the cloud has more knowledge, in addition to $\mathcal{K}_{\rm cloud}$ given in (14). In particular, we consider the scenario where the cloud knows the first column of $\mathbf{M}$. Let $(\xi_1^{\rm in}(\mathbf{A}), \eta_1^{\rm in}(\mathbf{A}))$ denote the corresponding input privacy index. After some calculations, we get the corresponding privacy index $(\xi_1^{\rm in}(\mathbf{A}), \eta_1^{\rm in}(\mathbf{A})) = (1, (n-1)m)$ [32].

Roughly speaking, the higher the adversarial knowledge, the smaller the privacy against the adversary. It is interesting to note that the privacy definitions (7)-(8) can be used to quantify these variations in privacy as desired, e.g., $(\xi_1^{\rm in}(\mathbf{A}), \eta_1^{\rm in}(\mathbf{A})) \preceq (\xi_0^{\rm in}(\mathbf{A}), \eta_0^{\rm in}(\mathbf{A}))$, where the notation "$\preceq$" here means the componentwise inequality. One can easily show that if the adversary knows all n columns of $\mathbf{M}$, then the input privacy index $(\xi_n^{\rm in}(\mathbf{A}), \eta_n^{\rm in}(\mathbf{A}))$ of $\mathbf{A}$ is $(0,0)$, provided columns of $\mathbf{M}$ are independent. As desired privacy index $(0,0)$ essentially means that there is no privacy against the adversary.

As listed in Example 4, it is also possible to use nonlinear change of variables for developing privacy preserving solution methods. To see this let us consider the following example:

Example 6 (Nonlinear transformation, a 2-party situation): Alice wants to outsource to an untrusted party the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n (\alpha_i / x_i) \\ \text{subject to} & \sum_{i=1}^n \beta_i x_i^2 \leq \gamma \\ & \mathbf{x} \geq \mathbf{0} \ , \end{array} \tag{15}$$

where the variable is $\mathbf{x}$. Here, the problem data are $\alpha_i > 0$, $\beta_i > 0$, and $\gamma > 0$. Suppose Alice wants input privacy for problem data $\{\alpha_i, \beta_i\}_{i=1,\dots,n}, \gamma$. By using the nonlinear change of variable given in Example 4, we have $x_i = \phi_i(z_i) = \alpha_i \exp(z_i)$. Next Alice can obtain the equivalent problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \exp(-z_i) \\ \text{subject to} & \sum_{i=1}^n \lambda_i \exp(2z_i) \leq 1 \ , \end{array} \tag{16}$$

where the variable is $\mathbf{z} = (z_1, \dots, z_n)$ and problem parameter $\lambda_i = (\beta_i \alpha_i^2 / \gamma)$. Now Alice can outsource problem (16) to the untrusted party. We can see that this solution procedure clearly yields input privacy for the sensitive input $\{\{\alpha_i, \beta_i\}_{i=1,\dots,n}, \gamma\}$. The solution $\mathbf{x}^\star$ of the original problem is simply obtained by $x_i^\star = \alpha_i \exp(z_i^\star)$, where $z_i^\star$ is the solution of problem (16).

To quantify the input privacy, as well as the output privacy of the transformations used in Example 6, we can follow similar arguments as we presented in the case of Example 5.

B. Transformation of Objective and Constraint Functions

The equivalence between two problems, by performing transformation of objective and constraint functions, is established by the following proposition:

Proposition 2: Suppose $\psi_0 : \mathbb{D}_0 \subseteq \mathbb{R} \to \mathbb{R}$ is monotonically increasing, with domain covering the image of f_0, i.e., $\mathbb{D}_0 \supseteq$ image f_0. Moreover, suppose that for $i = 1, \dots, q$, $\psi_i : \mathbb{D}_i \subseteq \mathbb{R} \to \mathbb{R}$, with $\mathbb{D}_i \supseteq$ image f_i, $\psi_i(z) \leq 0$ if and only if $z \leq 0$ and $\psi : \mathbb{R}^p \to \mathbb{R}^m$ satisfies $\psi(\mathbf{z}) = \mathbf{0}$ if and only if $\mathbf{z} = \mathbf{0}$. Then if $\mathbf{x}^\star$ solves the problem

$$\begin{array}{ll} \text{minimize} & \psi_0(f_0(\mathbf{x})) \\ \text{subject to} & \psi_i(f_i(\mathbf{x})) \leq 0, \ i = 1, \dots, q \\ & \psi(\mathbf{C}\mathbf{x} - \mathbf{d}) = \mathbf{0} \ , \end{array} \tag{17}$$

where the variable is $\mathbf{x} \in \mathbb{R}^n$, the solution must also solve problem (2) and vice versa. Moreover, the optimal value of problem (2), $p^\star$, and that of problem (17), $q^\star$, are related by

$$\psi_0(p^\star) = q^\star \ . \tag{18}$$

Proof: See [32]. ■

Let us next provide couple of examples which give insights into Proposition 2 in the context of privacy.

Example 7 (Scaling): The idea of scaling was presented in Example 3. Scaling is used in part to develop privacy preserving solution methods in references such as [23], [25]–[29]. Generally speaking, here all the functions $\psi_i, \ i = 0, \dots, q$ and ψ are linear (see (17)), i.e.,

$$\psi_i(z) = a_i z \ , \ i = 0, \dots, q \qquad \text{and } \psi(\mathbf{z}) = \mathbf{B}\mathbf{z} \ , \tag{19}$$

where $\{a_i\}_{i=0,\dots,q}$ are positive scalars and $\mathbf{B} \in \mathbb{R}^{p\times p}$ is a diagonal matrix with nonzero diagonal entries, which are unknown to any passive adversary. Specifically, the generalized permutation matrix used in [26, § 4.1], [23, p. 69], the scalar γ used in [27, III-B-3], and the positive diagonal matrix S used in [25, § III] are identical to some scaling of the form (19).

Example 8 (Horizontally Partitioned Linear Programs [10]): The considered set up is in a multi-party environment. The method is built on the following equality constraint transformation:

$$\psi(\mathbf{z}) = \mathbf{B}\mathbf{z} \ , \tag{20}$$

where the $\mathbf{B} \in \mathbb{R}^{m\times p}$ with $m \geq p$ and $\text{rank}(\mathbf{B}) = p$. Thus, the following equivalence holds:

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0} \Leftrightarrow \psi(\mathbf{A}\mathbf{x} - \mathbf{b}) \Leftrightarrow \mathbf{B}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0} \ , \tag{21}$$

where $\mathbf{A} \in \mathbb{R}^{p\times n}$ and $\mathbf{b} \in \mathbb{R}^p$. We can easily show (21) by using the pseudo-inverse of $\mathbf{B}$, which is given by $\mathbf{B}^\dagger = (\mathbf{B}^{\rm T}\mathbf{B})^{-1}\mathbf{B}^{\rm T}$. In reference [10], the authors exploit the partitioning of matrices $\mathbf{A}, \mathbf{b}$ together with carefully chosen partitioned matrix $\mathbf{B}$ to develop a solution method that yields input privacy for sensitive input (see [10, § 2] for details).

Due to page limitations, we omit the computations of input privacy index and the output privacy index associated with the transformations used in Examples 7-8. Similar arguments as in the case of Example 5 can be applied straightforwardly.

One can readily apply hybrid variants of the *transformation via change of variables* (see § III-A) and *transformation of objective and constraints* (see § III-B). Such hybrid variants can be found in [25], [27]–[29]. For example, in [25, § III], the authors have first used the change of variable $\mathbf{x} = \mathbf{Q}\mathbf{z} - \mathbf{Q}\mathbf{r}$, where $\mathbf{Q}$ is a generalized permutation matrix [26, § 4.1]. Then, for some constraints in the resulting equivalent problem, they performed scaling by $\mathbf{Q}^{-1}$ and a positive diagonal matrix $\mathbf{S}$.

IV. Conclusions

In this paper we proposed a unified framework that encompasses transformation methods, which is a non-cryptographic privacy preserving solution approach. The proposed framework is general and allows to avoid potential mistakes and limitations that can arise when developing transformation methods. More importantly, the framework plays a central role when developing generalized standard proof techniques to guarantee the privacy properties associated with transformation methods. A new general definition for the privacy of transformation methods was proposed. The proposed definitions gracefully quantify the privacy of transformation methods, in the presence of passive adversaries with different knowledge levels. The key idea of transformation methods, their applicability, and computation of their privacy was exemplified by a number of examples. We believe that the theory of non-cryptographic approaches, such as transformation methods for privacy preserving optimization is at a very early stage, and therefore substantial extensions are required. We started working toward such direction in [32].

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," in *Proc. IEEE Symp. Found. of Comp. Science*, Chicago, USA, Nov. 1982, pp. 160–164.

[2] Mengran Xue and S. Roy, "Characterization of security levels for the dynamics of autonomous vehicle networks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3916–3921.

[3] A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[4] O. Goldreich, *The Foundations of Cryptography*, vol. 2, Cambridge University Press, Cambridge, UK, 2004.

[5] J. Le Ny and G.J. Pappas, "Differentially private filtering," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3398–3403.

[6] W. Du and Z. Zhan, "A practical approach to solve secure multi-party computation problems," in *Proc. New Security Paradigms Works.*, Virginia beach, Virginia, USA, Sept. 23–26 2002, pp. 127–135.

[7] Eleni Stai, John S. Baras, and Symeon Papavassiliou, "Social networks over wireless networks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 2696–2703.

[8] O. L. Mangasarian, "Privacy-preserving linear and nonlinear approximation via linear programming," *Opt. Methods and Software*, pp. 1–10, Oct. 2011.

[9] M. Alizadeh, Tsung-Hui Chang, and A. Scaglione, "Grid integration of distributed renewables through coordinated demand response," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3666–3671.

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

[11] F. Pasqualetti, F. Dorfler, and F. Bullo, "Cyber-physical security via geometric control: Distributed monitoring and malicious attacks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3418–3425.

[12] H. Fawzi, P. Tabuada, and S. Diggavi, "Security for control systems under sensor and actuator attacks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3412–3417.

[13] J. Biskup and U. Flegel, "On pseudonymization of audit data for intrusion detection," in *Workshop on Design Issues in Anonymity and Unobservability*, NY, USA, Nov. 2000, pp. 161–180.

[14] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[15] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley and Sons, Inc. 1996.

[16] M. Rabin, "How to exchange secrets by oblivious transfer," in *Technical Report Tech. Memo TR-81, Aiken Computation Laboratory*, 1981.

[17] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," in *31th ACM Symposium on Theory of Computing*, Atlanta, GA, USA, May 1999, pp. 245–254.

[18] A. Shamir, "How to share a secret," *Communication of the ACM*, vol. 2, no. 11, pp. 612–613, Nov. 1979.

[19] Y. Desmedt, "Some recent research aspects of threshold cryptography," in *1st International Workshop on Information Security*, Ishikawa, Japan, Sept. 1997, pp. 158–173.

[20] O. Catrina and S. de Hoogh, "Secure multiparty linear programming using fixedpoint arithmetic," in *ESORICS*, Athens, Greece, Sept. 2010, pp. 134–150.

[21] J. Li and M.J. Atallah, "Secure and private collaborative linear programming," in *International Conference on Collaborative Computing: Netowrking, Applications and Worksharing*, Atlanta, GA, USA, Nov. 2006, pp. 1–8.

[22] T. Toft, "Solving linear programs using multiparty computation," *Financial Cryptography and Data Security. LNCS*, pp. 90–107, 2009.

[23] 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.

[24] O. L. Mangasarian, "Privacy-preserving linear programming," *Opt. Letters*, vol. 5, no. 1, pp. 165–172, Feb. 2011.

[25] 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.

[26] 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.

[27] 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.

[28] J. Vaidya, "Privacy-preserving linear programming," in *Proc. ACM Symp. on App. Comp.*, Honolulu, Hawaii, USA, Mar. 2009, pp. 2002–2007.

[29] W. Du, *A Study of Several Specific Secure Two-Party Computation Problems*, Ph.D. thesis, Purdue University, 2001.

[30] 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.

[31] 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.

[32] P. C. Weeraddana, G. Athanasiou, M. Jakobsson, C. Fischione, and J. S. Baras, "Per-se privacy preserving distributed optimization," *arXiv, Cornell University Library*, 2013, [Online]. Available: `http://arxiv.org/abs/1210.3283`.

[33] R. Canetti, "Security and composition of multi-party cryptographic protocols," *J. Crypto.*, vol. 13, no. 1, pp. 143–202, 2000.

[34] R. Canetti, *Studies in Secure Multiparty Computation and Applications*, Ph.D. thesis, J. Weizmann Institute, Israel, 1995.

[35] S. Micali and P. Rogaway, "Secure computation," in *Advances in Cryptology - CRYPTO '91*, J. Feigenbaum, Ed. 1991, vol. 576 of *Lecture Notes in Computer Science*, pp. 392–404, Springer-Verlag.

[36] S. Goldwasser and L. Levin, "Fair computation of general functions in presence of immoral majority," in *Advances in Cryptology - CRYPTO '90*, A. J. Menezes and S. A. Vanstone, Eds. 1990, vol. 537 of *Lecture Notes in Computer Science*, pp. 77–93, Springer-Verlag.

[37] D. Beaver, "Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority," *J. Crypto.*, vol. 4, no. 2, pp. 75–122, 1991.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[39] O. Goldreich, *The Foundations of Cryptography*, vol. 2, chapter 7, Cambridge University Press, Cambridge, UK, 2004.

[40] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. IEEE Symp. Found. of Comp. Science*, LA, USA, Oct. 1987, pp. 427–438.

[41] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," in *Proc. ACM Symp. on Theory of Comp.*, NY, USA, May 1987, pp. 218–229.

[42] F. B. Schneider, "Blueprint for a science of cybersecurity," in *Developing a Blueprint for a Science of Cybersecurity*, Robert Meushaw, Ed., vol. 19, pp. 47–57. The Next Wave; The National Security Agency's review of emerging technologies, 2012.

[43] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, Inc., USA, 3 edition, 1976.