# Symmetries and isomorphisms for privacy in control over the cloud*

Alimzhan Sultangazin[1], *Student Member, IEEE,* and Paulo Tabuada[1], *Fellow, IEEE,*

***Abstract*—Cloud computing platforms are being increasingly used for closing feedback control loops, especially when computationally expensive algorithms, such as model-predictive control, are used to optimize performance. Outsourcing of control algorithms entails an exchange of data between the control system and the cloud, and, naturally, raises concerns about the privacy of the control system's data (e.g., state trajectory, control objective). Moreover, any attempt at enforcing privacy needs to add minimal computational overhead to avoid degrading control performance. In this paper, we propose several transformation-based methods for enforcing data privacy. We also quantify the amount of provided privacy and discuss how much privacy is lost when the adversary has access to side knowledge. We address three different scenarios: a) the cloud has no knowledge about the system being controlled; b) the cloud knows what sensors and actuators the system employs but not the system dynamics; c) the cloud knows the system dynamics, its sensors, and actuators. In all of these three scenarios, the proposed methods allow for the control over the cloud without compromising private information (which information is considered private depends on the considered scenario).**

## I. Introduction

### A. Motivation

The recent advances in reliability and speed of communication have led to an increased use of cloud-based services, which provide computation and data storage capabilities to clients. Control over the cloud [1], [2], [3] has numerous advantages, which include easier installation and maintenance [4], and the availability of global information from all of the cloud's clients when making control decisions. However, the main advantage of control over the cloud is that it allows control systems to outsource expensive computational tasks to the cloud, thus potentially improving the speed of computation and freeing the local computational capabilities for other tasks.

An illustrative example of the benefits of outsourcing computing can be observed in Model Predictive Control (MPC). MPC is a conceptually simple, yet powerful scheme that was adopted in industry for multivariable control [5]. MPC inherently involves solving complex constrained optimization problems *on-line* (i.e., within one sampling interval). The work in [1] presents an experimental study that shows feasibility of MPC over the cloud for robot control. Another work (see [2]) considered the practicality and benefits of cloud-based MPC for a large-scale solar plant. The availability of global information provided by control over the cloud can have many practical benefits, as shown in [3]. There, the authors propose a solution to the problem of traffic flow estimation via the cloud.

However, relying on a third-party to perform computation is not without its dangers. Despite the benefits of control over the cloud, a number of studies have shown that exposing existing systems to connectivity may lead to security vulnerabilities in a vast variety of applications [6], [7], [8], [9], including control of process plants, traffic infrastructure, and smart meter systems. Cyber-security attacks vary based on the amount of resources the attacker possesses [10]. One of the most basic attacks that requires little resources is eavesdropping. It can often serve as a stepping stone in the implementation of more complex attacks [11]. In control over the cloud, eavesdropping involves the adversary listening in to the communication channel between sensors, controllers, and actuators to leak valuable information about the model, the controller, and trajectories [12]. The client is expected to disclose all of this sensitive information to the cloud if it intends to receive valid control inputs from it. For example, we would expect drivers to share their locations, final destinations and, perhaps, dynamics to successfully allow traffic control over the cloud.

Eavesdropping attacks are usually prevented with encryption - the plant and the cloud establish a shared key with which they encrypt transmitted messages and decrypt the received ones. However, if the adversary manages to undermine the security of the cloud (e.g., gain unauthorized access to its memory), this technique can no longer protect the system since the cloud accesses the decrypted data. As stated in [13], traditional IT security provides only a partial solution. Therefore, there is a pressing need for development of control-over-the-cloud methods that do not rely on decryption of the incoming data. Although much effort has been directed to this problem, a universally secure scheme for control over the cloud that could support any client functionality has not yet been created [14], [15]. When solving the problem of private control over the cloud, two other important concerns need to be accounted for: efficiency and safety. Privacy cannot come at the cost of degradation of control performance either due to delays in the feedback loop or inaccurate control inputs.

### B. Related work

The body of work on privacy in control over the cloud can be categorized into methods based on homomorphic encryption, differential privacy, and algebraic transformations.

When using homomorphic encryption techniques, the cloud is able to perform the computations on encrypted data without the need to decrypt it [16]. Homomorphic encyption can be

*The work of the authors was partially supported by the NSF grants 1740047, 1705135, by the UC-NL grant LFR-18-548554 and by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

[1]Alimzhan Sultangazin and Paulo Tabuada are with Department of Electrical and Computer Engineering, University of California - Los Angeles, CA 90095 USA (e-mail: {asultangazin, tabuada}@ucla.edu)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

classified into fully homomorphic encryption (FHE), which allows arbitrary computations on encypted data, and partially homomorphic encryption (PHE), which only allows for a subset of operations (e.g., modular multiplication) on encrypted data. Using PHE for control over the cloud with encrypted controllers was proposed in [17], [18]. In an effort to reduce communication with the cloud, in [19] the authors suggest using FHE for controller encryption. However, longer execution times of FHE [16] make it less practical than PHE when using optimization for control over the cloud. While PHE methods are shown to be feasible and are able to provide privacy guarantees [4], [12], [15], [20], [21], [22], the execution time, which grows disproportionally with an increase in key length [12], [15], remains a valid concern in these methods. A consequence of this is that using homomorphic encrypion may potentially lead to instability in the controlled system due to processing delays. To address this problem, some works (see [12]) have shown that encryption parameters can be chosen to ensure stability of the closed-loop performance, thus providing a natural trade-off between security and control performance. The practical feasibility of encrypted control systems has been validated in [23] by considering control of a DC motor in real time.

Inspired by studies in privacy of databases, the problem of privacy in control over the cloud has also been approached from the standpoint of differential privacy (see [24], [25]). This technique ensures that the risk of losing privacy of a single user's data by means of data queries is low. The main idea of these methods is to perturb the response to a data query with appropriate noise [26]. However, to achieve more privacy, the user must sacrifice accuracy (i.e., add more noise), which, in the context of control, degrades the control performance.

The ideas behind algebraic transformation methods have initially stemed from works on privacy in optimization. The idea is to use algebraic transformations to produce a different, but equivalent optimization problem. In other words, although the cloud does not know the original optimization problem, it can provide the client with an optimal solution to an equivalent optimization problem from which the client is able to recover the optimal solution to the original problem. Although initially these methods found application exclusively in linear programs [27], [28], several efforts have been directed to providing a unified framework and generalizing them to convex optimization problems (see [29], [30]). The work in [29] also shows one of the first attempls to define and *quantify* privacy of transformation-based methods. Algebraic transformation methods found applications in control due to their efficiency and guaranteed optimality of the solution [30]. For example, in [31] the authors propose a hybrid transformation-based method to preserve privacy of an MPC controller in networked control systems. In [32], transformation-based methods are used to provide privacy in a specific problem AC Optimal Power Flow.

### C. Contributions

This paper focuses on the use of transformation-based methods to preserve privacy of the system dynamics, control objective and constraints, and system trajectories. The contributions of this paper are fourfold:

1) we propose using isomorphisms and symmetries of control systems as a source of transformations so as to keep data private;
2) we quantify the privacy guaranteed by these methods via the dimension of the set that describes the uncertainty experienced by the adversary;
3) we quantify how much privacy is lost when the adversary is assumed to have access to side knowledge;
4) we show that the proposed method is computationally light as it only requires matrix multiplications.

The method proposed in this paper was initially introduced in [33]. In [34], it was extended to networked control systems with several agents requesting control input from a single cloud. In [35], the dimension of the set describing the uncertainty experienced by the adversary was proposed as a measure of privacy for this method and was evaluated for the special case of free group actions. This paper provides a unified presentation of the results in [33], [35] with simpler proofs and several new results, such as the bounds on privacy when the group action is not free and an exact quantification of privacy for prime systems.

While privacy quantification in optimization has been studied in [30], this work considers how much privacy is preserved in the more challenging context of control. Moreover, the measure of privacy proposed in this work has been chosen to be suitable for problems of optimization in control systems and, therefore, is different from any of those proposed in [30]. Although the application of transformation-based methods in control has been previously discussed in [31], the scheme proposed there only considers a special case, where the cloud optimizes the weighted sum of the norms of the input and state, and the state is taken to be the output of the system. Our algorithm can be applied to a wider class of problems as we allow for arbitrary quadratic costs, linear constraints and outputs different from the state.

The proposed results do not address the case where the adversary has some belief about the structure or the range of values of the system parameters. Addressing the adversary's beliefs is likely to be more natural in a probabilistic/information-theoretic setup that is outside of the scope of this paper, where we only employ deterministic techniques.

## II. Problem Definition

### A. Plant dynamics and control objective

We consider discrete-time affine plants, denoted by $\Sigma$, and described by:

$$\Sigma : \quad \begin{aligned} \bar{x}_{k+1} &= \bar{A}\bar{x}_k + \bar{B}u_k + \bar{c} \\ \bar{y}_k &= \bar{C}\bar{x}_k + \bar{d}, \end{aligned} \tag{II.1}$$

where $\bar{A} \in \mathbb{R}^{n\times n}$, $\bar{B} \in \mathbb{R}^{n\times m}$, $\bar{C} \in \mathbb{R}^{p\times n}$, $\bar{c} \in \mathbb{R}^n$, and $\bar{d} \in \mathbb{R}^p$ describe the dynamics of the system, and $\bar{x}_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $\bar{y}_k \in \mathbb{R}^p$ denote the state, input and output of the system at time $k$, respectively. We assume that system $\Sigma$ is controllable and observable. We also assume, without loss of generality, that $\ker \bar{B} = \{0\}$ and $\text{Im } \bar{C} = \mathbb{R}^p$, since we can always eliminate linearly dependent columns (resp. rows) from $\bar{B}$ (resp. $\bar{C}$).

To simplify notation, we lift every affine map $Wx + v$ to a linear map through the following construction:

$$Wx + v \mapsto \begin{bmatrix} W & v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}. \tag{II.2}$$

Applying (II.2) to (II.1):

$$\begin{aligned} x_{k+1} &\triangleq \begin{bmatrix} \bar{x}_{k+1} \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{c} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ 1 \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} u_k \\ &\triangleq Ax_k + Bu_k \\ y_k &\triangleq \begin{bmatrix} \bar{y}_k \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{C} & \bar{d} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ 1 \end{bmatrix} \triangleq Cx_k. \end{aligned} \tag{II.3}$$

In the remainder of the paper we suppress the inner structure for simplicity and represent all the systems in the linear form (II.3). However, the reader is advised to remember that we are dealing with affine maps. This is also true for the affine maps we will use to define isomorphisms.

We refer to system (II.3) as the triple $\Sigma = (A, B, C)$. We call a triple $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ a trajectory of $\Sigma$ if it satisfies (II.1) for all $k \in \mathbb{N}$.

Additionally, we define a cost function $J : \mathbb{R}^n \times (\mathbb{R}^m)^{N+1} \to \mathbb{R}$ for $N \in \mathbb{N} \cup \{+\infty\}$ that allows to compare trajectories and, thus, to formulate different control objectives. In alignment with the linear framework, we consider quadratic cost functions given by:

$$J(x, u) = \sum_{i=0}^{N} \Delta\eta_i^T M \Delta\eta_i, \tag{II.4}$$

where $\Delta\eta_i = \begin{bmatrix} x_i - x_i^* & u_i - u_i^* \end{bmatrix}^T$, $x = \{x_0, ..., x_N\}$ and $u = \{u_0, ..., u_N\}$. The sequences $x^* = \{x_0^*, ..., x_N^*\}$ and $u^* = \{u_0^*, ..., u_N^*\}$ denote the reference trajectories to be tracked. We define $M \in \mathbb{R}^{(n+m+1)\times(n+m+1)}$ to be a positive-definite matrix. Due to the lift (II.2), this cost includes not only quadratic, but also linear terms.

In addition to a cost, we also consider control objectives that require certain constraints to be satisfied at all times. These constraints are defined as:

$$D\eta_i \leq 0, \quad \forall i \in \{0, 1, ..., N\}, \tag{II.5}$$

where $\eta_i = \begin{bmatrix} x_i & u_i \end{bmatrix}^T$ and $D \in \mathbb{R}^{h\times(n+m+1)}$. Note that, despite appearing to be linear constraints, the constraints above are in fact affine, in view of the construction (II.2).

### B. Attack model and privacy objectives

The cloud is treated as a curious but honest adversary: the cloud adheres to the computations prescribed by an agreed-upon protocol, but may seek to extract and leak confidential information by keeping record of all computations and communicated messages.

The interaction between the plant and the cloud is performed in two steps. During the first step, called the handshaking, the plant provides the cloud with a suitably modified version of the plant model, cost, and constraints. In exchange, the cloud agrees to compute the input minimizing the provided cost, subject to the constraints and plant dynamics. During the second step, called plant execution, the plant repeatedly sends a suitably modified version of its measurements to the cloud. The cloud computes a new input based on the received measurements and sends it to the plant, where it is suitably modified before being applied to the plant.

In the previous paragraph we purposely used the vague expression "suitably modified". Making this expression more concrete requires that we first define the knowledge available to the plant. We consider the following three scenarios.

**Problem II.1** (Scenario 1)**.** *Assuming the cloud has no knowledge about the plant:*

1) *how to modify the plant $(A, B, C)$, cost $J$, and constraint matrix $D$ before sending them during the handshaking step,*
2) *how to modify the measurements sent to the plant, and*
3) *how to modify the inputs received from the plant,*

*so that the plant's trajectory minimizes cost $J$ in* (II.4)*, while preventing the cloud from learning the plant $(A, B, C)$, the cost $J$, the constraint matrix $D$, and the plant's trajectory $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$?*

**Problem II.2** (Scenario 2)**.** *Assuming the cloud has no knowledge about the plant except for knowing what are its sensors and actuators:*

1) *how to modify the plant $(A, B, C)$, cost $J$, and constraint matrix $D$ before sending them during the handshaking step;*
2) *how to modify the measurements sent to the plant, and*
3) *how to modify the inputs received from the plant,*

*so that the plant's trajectory minimizes cost $J$ in* (II.4)*, while preventing the cloud from learning the plant $(A, B, C)$, the cost $J$, the constraint matrix $D$, and the plant's trajectory $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$?*

**Problem II.3** (Scenario 3)**.** *Assuming the cloud has complete knowledge about the plant dynamics, including its sensors and actuators:*

1) *how to modify cost $J$, and constraint matrix $D$ before sending them alongside the plant $(A, B, C)$ during the handshaking step;*
2) *how to modify the measurements sent to the plant, and*
3) *how to modify the inputs received from the plant,*

*so that the plant's trajectory minimizes cost $J$ in* (II.4)*, while preventing the cloud from learning the cost $J$, the constraint matrix $D$, and the plant's trajectory $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$?*

These problems are solved in Section IV by utilizing isomorphisms and symmetries of control systems we define next in Section III.

## III. Isomorphisms and Symmetries of Control Systems

In this section, we introduce the notions of isomorphism and symmetry of control systems along with several technical results used in Section IV to provide a solution to the problems described in Section II.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

Let us denote by $\mathcal{S}_{n,m,p}$ the set of all controllable and observable linear control systems with state, input and output dimensions $n$, $m$, and $p$, respectively.

**Definition III.1.** An isomorphism of control systems in $\mathcal{S}_{n,m,p}$ is a quadruple $\psi = (P, F, G, S)$ consisting of a change of state coordinates $P : \mathbb{R}^n \to \mathbb{R}^n$, state feedback $F : \mathbb{R}^n \to \mathbb{R}^m$, a change of coordinates in the input space $G : \mathbb{R}^m \to \mathbb{R}^m$, and a change of coordinates in the output space $S : \mathbb{R}^p \to \mathbb{R}^p$. Transformations $P$ and $S$ are affine invertible maps, $F$ is an affine map and $G$ is a linear invertible map.

Recall that, to simplify notation, we lift the affine maps to linear maps using the transformation (II.2).

Let us also denote the set of isomorphisms of $\mathcal{S}_{n,m,p}$ described in Definition III.1 as $\mathcal{G}_{n,m,p}$. The set $\mathcal{G}_{n,m,p}$ forms a group under function composition as the group operation[1]. This allows us to define a group action of $\mathcal{G}_{n,m,p}$ on the set of linear control systems $\mathcal{S}_{n,m,p}$.

**Definition III.2.** Each element $\psi \in \mathcal{G}_{n,m,p}$ acts on $\Sigma \in \mathcal{S}_{n,m,p}$ to produce $\psi_*\Sigma$ given by:

$$\begin{aligned}\psi_*\Sigma &= (P, F, G, S)_*(A, B, C)\\ &= (P(A - BG^{-1}F)P^{-1}, PBG^{-1}, SCP^{-1}) \quad \text{(III.1)}\\ &\triangleq (\tilde{A}, \tilde{B}, \tilde{C}) \triangleq \tilde{\Sigma}.\end{aligned}$$

The map $\psi_*$ is called an isomorphism action. We also say that systems $\Sigma$ and $\tilde{\Sigma}$ are equivalent.

An isomorphism maps the state $x_k$, input $u_k$, and output $y_k$ of system $\Sigma$ to the state $\tilde{x}_k$, input $\tilde{u}_k$, and output $\tilde{y}_k$ of system $\tilde{\Sigma}$ as follows:

$$\tilde{x}_k = Px_k \quad \text{(III.2)}$$
$$\tilde{u}_k = Fx_k + Gu_k \quad \text{(III.3)}$$
$$\tilde{y}_k = Sy_k. \quad \text{(III.4)}$$

Similarly, an isomorphism induces transformation on the control objectives — i.e., the cost and constraints. The effect of $\psi$ on $\eta_k$ can be represented by:

$$\tilde{\eta}_k = \begin{bmatrix}\tilde{x}_k\\ \tilde{u}_k\end{bmatrix} = \begin{bmatrix}P & 0\\ F & G\end{bmatrix}\begin{bmatrix}x_k\\ u_k\end{bmatrix} \triangleq L\eta_k. \quad \text{(III.5)}$$

Therefore, the cost function $J$ can be expressed as a function of the sequence of modified states $\tilde{x} = \{\tilde{x}_0, ..., \tilde{x}_N\}$ and the sequence of modified inputs $\tilde{u} = \{u_0, ..., \tilde{u}_N\}$ as follows:

$$\tilde{J}(\tilde{x}, \tilde{u}) = \psi_* J(x, u) = \sum_{i=0}^{N} \Delta\tilde{\eta}_i^T \tilde{M} \Delta\tilde{\eta}_i, \quad \text{(III.6)}$$

where $\tilde{M} = L^{-T} M L^{-1}$. Applying the isomorphism action to the constraints in (II.5) yields:

$$\tilde{D}\tilde{\eta}_i \leq 0, \quad \forall i \in \{0, 1, ..., N\}, \quad \text{(III.7)}$$

where $\tilde{D} = \psi_* D = DL^{-1}$.

[1]A composition of two isomorphisms is given by $\psi_2 \circ \psi_1 = (P_2P_1, G_2F_1 + F_2P_1, G_2G_1, S_2S_1)$, the identity is $\psi_e = (I, 0, I, I)$ and the inverse is given by $\psi^{-1} = (P^{-1}, -G^{-1}FP, G^{-1}, S^{-1})$.

The effect of an isomorphism on the system, trajectory, cost and constraints will be used in Section IV to prevent the cloud from learning them.

For a given system $\Sigma$, there is a special subgroup of $\mathcal{G}_{n,m,p}$ called the symmetry group of $\Sigma$, which is defined by the following property.

**Definition III.3.** Let $\Sigma \in \mathcal{S}_{n,m,p}$. An isomorphism $\psi \in \mathcal{G}_{n,m,p}$ is said to be a symmetry of $\Sigma$ if $\psi_*\Sigma = \Sigma$. The subgroup of symmetries of $\Sigma$ is denoted here as $\mathcal{K}_{n,m,p}(\Sigma)$.

The notion of isomorphism was crafted to preserve properties of control systems. Among these, trajectories have a special significance. A simple induction argument can be used to establish the following result.

**Lemma III.4.** *Let $\Sigma \in \mathcal{S}_{n,m,p}$ and $\psi \in \mathcal{G}_{n,m,p}$. If $\tilde{\Sigma} = \psi_*\Sigma$ and $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ is a trajectory of $\Sigma$, then $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$, as given by* (III.2) - (III.4)*, is a valid trajectory of $\tilde{\Sigma}$.*

This means that if the cloud receives $\tilde{\Sigma}$ during the handshaking step, then the received sequence of measurements $\tilde{y}$ and the produced sequence of control inputs $\tilde{u}$ in the subsequent execution step are compatible with the plant $\tilde{\Sigma}$. To elaborate, both the modified measurements $\tilde{y}$ and modified control inputs $\tilde{u}$ would be compatible with modified dynamics $\tilde{\Sigma}$.

Let us now define $\bar{\mathcal{S}}_{n,m,p}$ to be a set of quadruples $\Omega \triangleq \{\Sigma, J, D, \{x_k, y_k, u_k\}\}_{k\in\mathbb{N}}$ such that $\{x_k, y_k, u_k\}$ is a trajectory of a linear system $\Sigma \in \mathcal{S}_{n,m,p}$ minimizing cost function $J$ under constraints $D$.

**Lemma III.5.** *The set $\bar{\mathcal{S}}_{n,m,p}$ is a smooth manifold.*

*Proof.* We can see that $\bar{\mathcal{S}}_{n,m,p}$ is, in fact, the Cartesian product of $\mathcal{S}_{n,m,p}$ with the set of cost functions $\mathcal{M}^{++}(m+n+1, \mathbb{R})$, defined by positive-definite matrices, with the set of constraints $\mathcal{M}_d(h \times (m+n+1), \mathbb{R})$, defined by the set of full-rank matrices, where $d = \min\{h, m+n+1\}$. It is known that the product space is a smooth manifold if its constituents are smooth manifolds [36, p. 21]. It remains to show that these constinuents are indeed smooth manifolds.

Let us construct the map:

$$\begin{aligned}f_S :&\mathbb{R}^{n\times(n+1)} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times(n+1)} \to \mathbb{R}^2\\ &(A, B, C) \mapsto (\det \mathcal{C}, \det \mathcal{O}),\end{aligned} \quad \text{(III.8)}$$

where $\mathcal{C}$ and $\mathcal{O}$ are the controllability and observability matrices of the dynamics $(A, B, C)$. It can be seen that $\mathcal{S}_{n,m,p} = f_S^{-1}(\mathbb{R}^2 \backslash (0, 0))$. The function $f_S$ is continuous since each of its elements is defined by a polynomial function of the elements of $(A, B, C)$. Given that for continuous functions the preimage of every open set is an open set, we have that $\mathcal{S}_{n,m,p}$ is an open subset of the domain of $f_S$. Seeing that the domain of $f_S$ is a smooth manifold, $\mathcal{S}_{n,m,p}$ is a smooth manifold of dimension $n(n+1) + nm + p(n+1)$.

The set of positive-definite matrices $\mathcal{M}^{++}(m+n+1, \mathbb{R})$ is shown to be a smooth embedded submanifold of $\mathbb{R}^{(m+n+1)\times(m+n+1)}$ of dimension $(m+n+1)(m+n+2)/2$ in [37].

The set of full-rank matrices $\mathcal{M}_d(h \times (m+n+1), \mathbb{R})$ is a smooth manifold of dimension $h(m+n+1)$ [36, p. 19]. □

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

Similarly to $\mathcal{S}_{n,m,p}$, we can define a group action of $\mathcal{G}_{n,m,p}$ on $\bar{\mathcal{S}}_{n,m,p}$ in view of the previous discussion.

Therefore, we can use the isomorphism action of $\mathcal{G}_{n,m,p}$ to define an equivalence relation on $\bar{\mathcal{S}}_{n,m,p}$.

**Definition III.6.** Let $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}})$ and $\tilde{\Omega} = (\tilde{\Sigma}, \tilde{J}, \tilde{D}, \{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}})$ be elements of $\bar{\mathcal{S}}_{n,m,p}$. The equivalence relation $\sim_{\mathcal{G}}$ on $\bar{\mathcal{S}}_{n,m,p}$ denoted by:

$$\Omega \sim_{\mathcal{G}} \tilde{\Omega}, \tag{III.9}$$

is defined by the existence of $\psi \in \mathcal{G}_{n,m,p}$ such that:

$$\tilde{\Omega} = \psi_* \Omega; \tag{III.10}$$

i.e., $\tilde{\Sigma} = \psi_*\Sigma$, $\tilde{J} = \psi_* J$, $\tilde{D} = \psi_* D$, and $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$ is given in terms of $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ as in (III.2) - (III.4).

The equivalence relation $\sim_{\mathcal{G}}$, in turn, defines equivalence classes in $\bar{\mathcal{S}}_{n,m,p}$. The equivalence class of $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ defined by the action of $\mathcal{G}_{n,m,p}$ is the set:

$$\begin{aligned}[\Omega] &\triangleq \{\Omega' \in \bar{\mathcal{S}}_{n,m,p} | \exists \psi \in \mathcal{G}_{n,m,p} \text{ such that } \Omega' = \psi_* \Omega\} \\ &= \{\psi_* \Omega | \psi \in \mathcal{G}_{n,m,p}\}. \end{aligned} \tag{III.11}$$

This equivalence class is also called the orbit of $\Omega$ under action of $\mathcal{G}_{n,m,p}$.

To facilitate further results, let us show that $\mathcal{G}_{n,m,p}$ is a Lie group acting on $\bar{\mathcal{S}}_{n,m,p}$.

**Lemma III.7.** *The group $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1) + m(n+1) + m^2 + p(p+1)$ acting smoothly on $\bar{\mathcal{S}}_{n,m,p}$.*

*Proof.* It was previously established that $\mathcal{G}_{n,m,p}$ is a group. It is a Lie group because it is a Cartesian product of smooth manifolds (i.e., general linear groups and vector spaces of various dimensions) and its multiplication and inversion maps are smooth. Moreover, since the dimension of a product of smooth manifolds is equal to the sum of the factors' dimensions, the dimension of $\mathcal{G}_{n,m,p}$ is $n(n+1)+m(n+1)+m^2+p(p+1)$ [36, p. 21]. The group $\mathcal{G}_{n,m,p}$ acts smoothly on $\bar{\mathcal{S}}_{n,m,p}$ since its action involves matrix multiplication and matrix inversion: the former results in every element of the product being a polynomial function of the elements of the factors, while the latter is smooth by Cramer's rule [36]. □

The next result shows that when the cloud optimizes $\tilde{J}$ and the plant replaces each $y_k$ with output $\tilde{y}_k$, the resulting sequence of inputs $\tilde{u}$ can be used to reconstruct a sequence of inputs $u$ that optimizes $J$. Its proof amounts to using the change of variables (III.2)-(III.4).

**Lemma III.8.** *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ and $\psi \in \mathcal{G}_{n,m,p}$. Suppose the cloud solves the optimization problem:*

$$\begin{aligned} \min_{\tilde{u}} \quad & \tilde{J}(\tilde{x}, \tilde{u}) \\ \text{subject to} \quad & \tilde{D}\hat{\tilde{\eta}}_i \leq 0, \quad \forall i \in \{0, ..., N\}, \end{aligned}$$

*for the plant $\tilde{\Sigma} = \psi_*\Sigma$ and the sequence $\tilde{u}^*$ is a unique solution of this optimization problem. Then, the unique solution of the optimization problem:*

$$\begin{aligned} \min_{u} \quad & J(x, u) \\ \text{subject to} \quad & D\eta_i \leq 0, \quad \forall i \in \{0, ..., N\} \end{aligned}$$

*for the plant $\Sigma$ is the sequence $u^*$ such that $u_i^* = G^{-1}(\tilde{u}_i^* - Fx_i)$ for all $i \in \{0, ..., N\}$.*

## IV. Solving the control-over-the-cloud privacy problem

### A. Enforcing privacy

The main reason for using isomorphisms is to preclude the cloud from distinguishing between isomorphic systems. We now formalize the notion of indistinguishability.

**Definition IV.1.** A protocol renders two quadruples $\Omega$ and $\tilde{\Omega}$ indistinguishable by the cloud if the exchanged messages, when using the protocol between the cloud and the plant $\Omega$, and the exchanged messages, when using the protocol between the cloud and the plant $\tilde{\Omega}$, can be made the same.

The results from Section III allow us to construct a communication protocol between the plant and the cloud that, as will be further shown, solves Problems II.1-II.3. We start by detailing this protocol.

**Algorithm 1** Secure communication

**Input:** Plant: $\psi$, $\Sigma$, $J$, $D$, $\tilde{u}_k$;
Cloud: $\tilde{y}_k$, $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$
**Output:** Plant: $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$, $\tilde{y}_k$;
Cloud: $\tilde{u}_k$

***Phase 1: Handshaking***:

1: Plant: Encode $\Sigma$, $J$, $D$ into $\tilde{\Sigma} = \psi_*\Sigma$, $\tilde{J} = \psi_* J$ and $\tilde{D} = \psi_* D$;
2: Plant: Send $\tilde{\Sigma}$, $\tilde{J}$, and $\tilde{D}$ to the cloud;

***Phase 2: Execution***:

3: Plant: Encode measurement $y_k$ into $\tilde{y}_k = Sy_k$ and send $\tilde{y}_k$ to the cloud;
4: Cloud: Use the received $\tilde{y}_k$ to estimate $\tilde{x}_k$ and compute $\tilde{u}_k$ minimizing $\tilde{J}$ subject to the constraints $\tilde{D}$ and the dynamics $\tilde{\Sigma}$;
5: Cloud: Send $\tilde{u}_k$ to the plant;
6: Plant: Use the isomorphism $\psi$ to decode $\tilde{u}_k$ and produce $u_k$ using (III.3);
7: Plant: Apply $u_k$ to the actuators.

From Lemma III.8, we see that Algorithm 1 provides the plant with the inputs $u_k$ that satisfy the original control objective — i.e., the plant's trajectory minimizes cost $J$ under affine constraints $D$.

Let us note how all the required computations in this algorithm are matrix multiplications, which means that both handshaking and execution can be performed in $O(k^3)$ time, where $k = \max\{n, m, p\}$. However, performing matrix multiplications of constant matrices (e.g., $G^{-1}F$) in advance would reduce the complexity of the execution to $O(k^2)$. Both of these complexities were calculated only for the client side (i.e., Plant) of the algorithm.

Let us now show that applying this protocol indeed makes any two systems in the same equivalence class indistinguishable from each other.

**Theorem IV.2.** *Algorithm 1 renders isomorphic systems $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}})$ and*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

$\tilde{\Omega} = (\tilde{\Sigma}, \tilde{J}, \tilde{D}, \{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}})$ *indistinguishable by the cloud.*

*Proof.* Since $\Omega$ and $\tilde{\Omega}$ are isomorphic, there exists an isomorphism $\psi$ such that $\psi_*\Sigma = \tilde{\Sigma}$, $\psi_* J = \tilde{J}$, and $\psi_* D = \tilde{D}$. Indistinguishibility of $\Omega$ and $\tilde{\Omega}$ will be shown by running two instances of Algorithm 1: one with $\Omega$ and $\psi$ as inputs, the other - with $\tilde{\Omega}$ and the identity isomorphism $\psi_e$. Let us denote the communication algorithm described in Algorithm 1 applied to $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ with the selected isomorphism $\psi \in \mathcal{G}_{n,m,p}$ by $\text{Alg}(\Omega, \psi)$. During handshaking:

- when $\text{Alg}(\Omega, \psi)$ is executed, the plant sends $\psi_*\Sigma$, $\psi_* J$, and $\psi_* D$;
- when $\text{Alg}(\tilde{\Omega}, \psi_e)$ is executed ($\psi_e$ is the identity of $\mathcal{G}_{n,m,p}$), the plant sends $\tilde{\Sigma}$, $\tilde{J}$, and matrix $\tilde{D}$ unprotected.

Thus, the communicated dynamics and optimization problems are the same. During execution:

- when $\text{Alg}(\Omega, \psi)$ is executed, $\psi$ takes trajectories $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ of $\Sigma$ to trajectories $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$ of $\psi_*\Sigma$;
- when $\text{Alg}(\tilde{\Omega}, \psi_e)$ is executed, the trajectories are $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$.

Therefore, the cloud receives the same measurements from both plants. In response, since both plants communicated the same optimization problem, the cloud sends the same control inputs to both plant $\Omega$ and $\tilde{\Omega}$. □

The result described in Theorem IV.2 states that the cloud cannot differentiate between any two plants, costs, constraints or trajectories contained in the same equivalence class of the $\sim_{\mathcal{G}}$-equivalence relation, thereby protecting the privacy of the system. In the next section, we quantify the amount of privacy provided by Algorithm 1.

### B. Quantifying privacy

Privacy is created by preventing the cloud from knowing which quadruple $\Omega$ in its equivalence class $[\Omega]$ it is interacting with. Clearly, the larger the equivalence class, the more privacy is ensured. Since each equivalence class has infinitely many elements, cardinality cannot be used as a measure of privacy. In this section, we show that each equivalence class is a smooth manifold and we quantify privacy using the dimension of this manifold.

*1) Preliminaries: stabilizer subgroups and their dimensions:* The stabilizer subgroup of $\mathcal{G}_{n,m,p}$ for any $\Omega \in \bar{S}_{n,m,p}$, denoted by $\mathcal{K}_{n,m,p}(\Omega)$, is defined by:

$$\mathcal{K}_{n,m,p}(\Omega) = \{\psi \in \mathcal{G}_{n,m,p} | \psi_*\Omega = \Omega\}. \tag{IV.1}$$

The subgroup $\mathcal{K}_{n,m,p}(\Omega)$ must be a subset of the symmetry subgroup $\mathcal{K}_{n,m,p}(\Sigma)$ since it must preserve the dynamics.

In [38], Respondek gives a characterization of the symmetries of controllable pairs $(A, B)$. Since when considering pairs $(A, B)$ the output is not relevant, the isomorphisms of $(A, B)$ degenerate into the form $\phi = (P, F, G)$, where the matrices $P$, $F$ and $G$ are defined to be the same as their counterparts in Definition III.1. We denote the group of these isomorphisms by $\mathcal{G}_{n,m}$. The group action of $\mathcal{G}_{n,m}$ is given by:

$$\phi_*(A, B) = (P(A - BG^{-1}F)P^{-1}, PBG^{-1}). \tag{IV.2}$$

Let us define the symmetry subgroup of controllable systems $(A, B)$ as:

$$\mathcal{K}_{n,m}(A, B) = \{\phi \in \mathcal{G}_{n,m} | \phi_*(A, B) = (A, B)\}. \tag{IV.3}$$

The next proposition uses the results from [39] and the notion of controllability indices (see [40] for a definition) to estimate the dimension of $\mathcal{K}_{n,m}(A, B)$:

**Proposition IV.3.** *Let (A,B) be a controllable pair. Then:*

$$m(n+1) - s \leq \dim \mathcal{K}_{n,m}(A, B) \leq n(m+1) - s,$$

*where:*

$$\begin{aligned} s &= \sum_{i=2}^{m} r_{i-1} r_i, \\ r_1 &= \text{rank } B, \\ r_i &= \text{rank } S_{i-1}(A, B) - \text{rank } S_{i-2}(A, B), \ \ i = 2, ..., m, \\ S_j(A, B) &= \begin{bmatrix} B & AB & ... & A^j B \end{bmatrix}, \ \ j = 1, ..., m-1. \end{aligned}$$

*and $\{\kappa_i\}_{i=1}^m$ are controllability indices of $(A, B)$.*

*Proof.* The symmetry subgroup $\mathcal{K}_{n,m}(A, B)$ consists of solutions to the following system of equations:

$$\begin{cases} A = P(A - BG^{-1}F)P^{-1} \\ B = PBG^{-1}, \end{cases} \tag{IV.4}$$

which is equivalent to:

$$\begin{cases} AP + BF = PA \\ BG = PB. \end{cases} \tag{IV.5}$$

Recall that elements of the pair $(A, B)$ and transformations $(P, F, G)$ are, in fact, affine maps. If we express (IV.5) using the inner structure of the maps, we get:

$$\begin{cases} \bar{A}\bar{P} + \bar{B}\bar{F} = \bar{P}\bar{A} \\ \bar{B}G = \bar{P}\bar{B} \\ (\bar{A} - I)\bar{p} + \bar{B}\bar{f} = \bar{P}\bar{c} - \bar{c}, \end{cases} \tag{IV.6}$$

where $P = \begin{bmatrix} \bar{P} & \bar{p} \\ 0 & 1 \end{bmatrix}$ and $F = \begin{bmatrix} \bar{F} & \bar{f} \end{bmatrix}$. Finding elements of $\mathcal{K}_{n,m}(A, B)$ is equivalent to finding $(\bar{P}, \bar{p}, \bar{F}, \bar{f}, G)$. According to Theorem 2.2 in [39], the dimension of solution space $S$ of $(\bar{P}, \bar{F}, G)$ satisfying the first and second equations in (IV.6) is equal to:

$$\begin{aligned} \dim S &= m(n+m) - \sum_{i=1}^{m} r_{i-1} r_i \\ &= m(n+m) - r_0 r_1 - \sum_{i=2}^{m} r_{i-1} r_i \\ &= mn - \sum_{i=2}^{m} r_{i-1} r_i, \end{aligned} \tag{IV.7}$$

because $r_0 = r_1 = m$, $\kappa_1 = m$ and $(A, B)$ is a controllable pair. Fixing $(\bar{P}, \bar{F}, G)$, one can find the dimension of the solution space of the third equation in (IV.6). It can be observed that the dimension of the solution space is equal to dim ker $\begin{bmatrix} \bar{A} - I & \bar{B} \end{bmatrix}$. Since rank $\bar{B} = m$, it follows that:

$$m \leq \dim \ker \begin{bmatrix} \bar{A} - I & \bar{B} \end{bmatrix} \leq n. \tag{IV.8}$$

The result then follows from (IV.7) and (IV.8). □

This result can be used to estimate the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$. If $\Sigma = (A,B,C)$, then, from Proposition IV.3, we know the dimension of $\mathcal{K}_{n,m}(A,B)$ and that any $\phi \in \mathcal{K}_{n,m}(A,B)$ satisfies $\phi_*(A,B) = (A,B)$. Given $\phi = (P,F,G) \in \mathcal{K}_{n,m}(A,B)$, finding a corresponding $\psi = (P,F,G,S) \in \mathcal{K}_{n,m,p}(\Sigma)$ requires finding $S$ such that $C = SCP^{-1}$. Since we assume $C$ has linearly independent rows, for a given $P$, this equation has at most one solution. A solution exists if and only if $\text{Im } C^T \subset \text{Im } P^{-T}C^T$ [41]. Let $\mathcal{Q}(A,B,C)$ be the subset of $\mathcal{K}_{n,m}(A,B)$ defined by the elements $(P,F,G)$ for which a unique solution to $C = SCP^{-1}$ exists. It can be seen that there is a one-to-one correspondence between $\mathcal{Q}(A,B,C)$ and $\mathcal{K}_{n,m,p}(\Sigma)$. Since $\mathcal{Q}(A,B,C) \subset \mathcal{K}_{n,m}(A,B)$, this gives an upper bound on the dimension of the symmetry subgroup:

$$\dim \mathcal{K}_{n,m,p}(\Sigma) \leq \dim \mathcal{K}_{n,m}(A,B). \tag{IV.9}$$

**Lemma IV.4.** *For any* $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}}) \in \bar{\mathcal{S}}_{n,m,p}$,

$$\textit{dim } \mathcal{K}_{n,m,p}(\Omega) \leq \textit{dim } \mathcal{K}_{n,m,p}(\Sigma) \leq \textit{dim } \mathcal{K}_{n,m}(A,B),$$

*where dim* $\mathcal{K}_{n,m,p}(A,B)$ *is given by Proposition IV.3.*

Let us consider a special case, in which the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$ can be computed exactly.

**Definition IV.5.** A system $\Sigma \in \mathcal{S}_{n,m,p}$ is said to be a prime system if it is $\sim_{\mathcal{G}}$-equivalent to the system of the form:

$$\Sigma: \quad \begin{cases} x_{k+1}^{(i,1)} = x_k^{(i,2)}, \\ \vdots \\ x_{k+1}^{(i,\kappa_i)} = u_k^{(i)}, \\ y_k^{(i)} = x_k^{(i,1)}, \ \ 1 \leq i \leq m, \end{cases} \tag{IV.10}$$

where $x_k = \left[x_k^{(1,1)}, ..., x_k^{(1,\kappa_1)}, ..., x_k^{(m,1)}, ..., x_k^{(m,\kappa_m)}\right]^T \in \mathbb{R}^n$ and $\{\kappa_i\}_{i=1}^m$ are controllability indices of $(A,B)$.

For prime systems we have the following characterization of the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$.

**Lemma IV.6.** *Let* $\Sigma \in \mathcal{S}_{n,m,p}$ *be a prime system. Then,*

$$\sum_{i=1}^{m} r_{\kappa_i} + m \leq \textit{dim } \mathcal{K}_{n,m,p}(\Sigma) \leq \sum_{i=1}^{m} r_{\kappa_i} + n, \tag{IV.11}$$

*where*

$$r_1 = \textit{rank } B,$$
$$r_i = \textit{rank } S_{i-1}(A,B) - \textit{rank } S_{i-2}(A,B), \ \ i = 2, ..., m,$$
$$S_j(A,B) = \begin{bmatrix} B & AB & ... & A^jB \end{bmatrix}, \ \ j = 1, ..., m-1,$$

*and* $\{\kappa_i\}_{i=1}^m$ *are controllability indices of* $(A,B)$.

*Proof.* Without loss of generality, let us consider a prime system of the form (IV.10). From Proposition 2 in [38], we can see that if a system is prime, a symmetry $\psi = (P,F,G,S)$ is uniquely defined by a transformation on its outputs (i.e., by transformation $S$).

We want to show that, in order to define a symmetry, transformation $S$ needs to be constructed in such a way that each transformed output $\tilde{y}_k^{(i)}$ is an affine function of outputs $y_k^{(j)}$ with relative degrees greater or equal than that of $y_k^{(i)}$. To simplify notation, we prove this claim for the example with controllability indices $\kappa_1 = \kappa_2 = 2$, $\kappa_3 = 1$, although the employed arguments apply to any prime system:

$$\begin{aligned} x_{k+1}^{(1,1)} &= x_k^{(1,2)} \qquad x_{k+1}^{(2,1)} = x_k^{(2,2)} \qquad x_{k+1}^{(3,1)} = u_k^{(3)} \\ x_{k+1}^{(1,2)} &= u_k^{(1)} \qquad x_{k+1}^{(2,2)} = u_k^{(2)} \\ y_k^{(1)} &= x_k^{(1,1)} \qquad y_k^{(2)} = x_k^{(2,1)} \qquad y_k^{(3)} = x_k^{(3,1)}. \end{aligned} \tag{IV.12}$$

We will show, by contradiction, that if $S$ produces a transformed output based on outputs of a smaller relative degree, then $S$ cannot be part of a symmetry. In other words, there exist no matrices $P$, $F$, and $G$ such that the quadruple $(P,F,G,S)$ satisfies the equations:

$$A = P(A - BG^{-1}F)P^{-1} \tag{IV.13}$$
$$B = PBG^{-1} \tag{IV.14}$$
$$C = SCP^{-1}. \tag{IV.15}$$

Assume that (IV.13)-(IV.15) are satisfied and that $S$ contains non-zero elements $S_{ij}$ if $\kappa_i > \kappa_j$ (i.e., the transformed output uses outputs of a smaller relative degree). From (IV.15), we have that:

$$SCA^qB = CPA^qB, \quad \forall\ 0 \leq q < \kappa_1. \tag{IV.16}$$

By using (IV.13) and (IV.14), the following relation can be shown:

$$PA = AP + PBG^{-1}F = AP + BF. \tag{IV.17}$$

Recursively substituting (IV.17) into (IV.16) results in:

$$\begin{aligned} SCA^qB &= C(PA)A^{q-1}B = C(AP+BF)A^{q-1}B \\ &= CBFA^{q-1}B + CAPA^{q-1}B \\ &= CBFA^{q-1}B + CA(PA)A^{q-2}B \\ &= \dots \\ &= \sum_{l=0}^{q-1} CA^lBFA^{q-l-1}B + CA^qPB. \end{aligned}$$

Equation (IV.14) implies that $PB = BG$ and, thus, leads to:

$$SCA^qB = \sum_{l=0}^{q-1} CA^lBFA^{q-l-1}B + CA^qBG. \tag{IV.18}$$

Note that $CA^lB$ is a diagonal matrix such that:

$$[CA^lB]_{ii} = \begin{cases} 1, \text{ if } \kappa_i = l+1 \\ 0, \text{ otherwise.} \end{cases} \tag{IV.19}$$

In other words, this diagonal matrix marks the indices corresponding to the outputs of equal relative degree. In addition, the expression $FA^{q-l-1}B$ is an $m \times m$ matrix composed out of elements of $F$ (recall that $A$ and $B$ are in the form (IV.10)).

The left-hand side of (IV.18) selects the columns of $S$ corresponding to the outputs of relative degree $\kappa_i = q+1$. For the example in (IV.12), taking $q = 0$ gives:

$$SCB = \begin{bmatrix} 0 & 0 & S_{13} \\ 0 & 0 & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix}. \tag{IV.20}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

The right-hand side of (IV.18) fills the rows corresponding to the outputs of relative degree smaller or equal than $\kappa_i = q+1$ with values from $G$. In case of example in (IV.12), the right-hand side, given $q=0$, is:

$$CBG = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \times & \times & \times \end{bmatrix}. \tag{IV.21}$$

Thus, the equality in (IV.18), which was derived using the definition of symmetry, forces $S_{ij}$ to zero if $\kappa_i > \kappa_j$. In the example in (IV.12), this leads to $S_{13} = S_{23} = 0$. This contradicts the assumption that $S$ produces a transformed output based on outputs of a smaller relative degree.

This idea can be generalized to any prime system and, therefore, each transformed output $\tilde{y}_k^{(i)}$ can only be an affine function of outputs $y_k^{(j)}$ with relative degrees greater or equal than that of $y_k^{(i)}$.

The number of outputs $y_k^{(j)}$ with a relative degree greater or equal to that of $y_k^{(i)}$ (i.e., greater or equal than $k_i$) is equal to $r_{k_i}$ [39]. Therefore, each modified output $y_k^{(i)}$ is an affine function with $r_{k_i}$ arguments. The constant terms of transformations $P$, $F$, and $S$, denoted by $\bar{p}$, $\bar{f}$, and $\bar{s}$, respectively, need to satisfy the following equalities:

$$\begin{cases} (\bar{A}-I)\bar{p} + \bar{B}\bar{f} = \bar{P}\bar{c} - \bar{c} \\ \bar{s} = \bar{C}\bar{p} + \bar{d} - \bar{S}\bar{d}, \end{cases}$$

where $P = \begin{bmatrix} \bar{P} & \bar{p} \\ 0 & 1 \end{bmatrix}$, $F = \begin{bmatrix} \bar{F} & \bar{f} \end{bmatrix}$, and $S = \begin{bmatrix} \bar{S} & \bar{s} \\ 0 & 1 \end{bmatrix}$. Similarly to the proof of Proposition IV.3, the dimension of the solution space of this system is given by the dimension of the kernel of the linear map defining the left-hand side of the system of equations as:

$$m \leq \dim \ker \begin{bmatrix} \bar{A}-I & \bar{B} & 0 \\ 0 & 0 & I \end{bmatrix} \leq n, \tag{IV.22}$$

thereby leading to the result of this lemma.

□

*2) Main results:* Consider the scenario from Problem II.1, in which the cloud does not know anything about the system. In this scenario, the plant encodes $\Omega$ using an isomorphism $\psi = (P, F, G, S)$ that can be regarded as a private key used to encode and decode the information exchanged with the cloud. This isomorphism $\psi$ is chosen from $\mathcal{G}_{n,m,p}$, the group of all isomorphisms.

**Proposition IV.7.** *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem II.1, the cloud cannot distinguish between $\Omega$ and any other system in the uncertainty set $[\Omega]_{\mathcal{G}}$ (i.e., the equivalence class of $\Omega$ defined by the action of $\mathcal{G}_{n,m,p}$) of dimension:*

$$\textit{dim } \mathcal{G}_{n,m,p} - \textit{dim } \mathcal{K}_{n,m,p}(\Omega), \tag{IV.23}$$

*if Algorithm 1 is used.*

*This implies that the dimension of $[\Omega]_{\mathcal{G}}$ is greater or equal than:*

$$n^2 + m(m+1) + p(p+1) + \sum_{i=2}^{m} r_{i-1} r_i, \tag{IV.24}$$

*where $r_i$ is given in Lemma IV.3.*

*For $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ such that its corresponding $\Sigma \in \mathcal{S}_{n,m,p}$ is prime, this implies that the dimension of $[\Omega]_{\mathcal{G}}$ is greater or equal to:*

$$n^2 + m(n+1) + m^2 + p(p+1) - \sum_{i=1}^{m} r_{\kappa_i}, \tag{IV.25}$$

*where $r_{\kappa_i}$ is given in Lemma IV.6.*

*Proof.* From Theorem IV.2, we know that Algorithm 1 renders isomorphic systems indistinguishable by the cloud. Therefore, the uncertainty set is the set of systems isomorphic to $[\Omega]_{\mathcal{G}}$ - namely, the equivalence class of $\Omega$ defined by the action of $\mathcal{G}_{n,m,p}$.

Let us define a map:

$$\begin{aligned} \theta_\Omega : \mathcal{G}_{n,m,p} &\to \bar{\mathcal{S}}_{n,m,p} \\ \psi &\mapsto \psi_* \Omega. \end{aligned}$$

Here, $\theta_\Omega$ is smooth because, as shown in Lemma III.7, $\mathcal{G}_{n,m,p}$ acts smoothly on $\bar{\mathcal{S}}_{n,m,p}$. The stabilizer set can be defined by:

$$\mathcal{K}_{n,m,p}(\Omega) = (\theta_\Omega)^{-1}(\Omega) = \{\psi | \psi_* \Omega = \Omega\}.$$

Since $\theta_\Omega$ and its inverse are smooth and, therefore, continuous, the subgroup $\mathcal{K}_{n,m,p}(\Omega)$ is closed.

By Theorem 21.17 in [36], the quotient space $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is a smooth manifold of dimension $\dim \mathcal{G}_{n,m,p} - \dim \mathcal{K}_{n,m,p}(\Omega)$ such that the quotient map $\pi : \mathcal{G}_{n,m,p} \to \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is a smooth submersion.

Now, let us define a map:

$$\begin{aligned} \Theta_\Omega : \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega) &\to \bar{\mathcal{S}}_{n,m,p} \\ \psi \mathcal{K}_{n,m,p}(\Omega) &\mapsto \psi_* \Omega, \end{aligned}$$

where $\psi\mathcal{K}_{n,m,p}(\Omega)$ is a left coset of $\mathcal{K}_{n,m,p}(\Omega)$. It can be shown that $\Theta_\Omega$ is well-defined.

By Theorem 4.29 in [36], $\Theta_\Omega$ is smooth because $\theta_\Omega = \Theta_\Omega \circ \pi$ is smooth and $\pi$ is a smooth submersion.

It can be shown that the map $\Theta_\Omega$ is equivariant (see [36, p. 164]) and, therefore, by the equivariant rank theorem [36, p. 165], we have that $\Theta_\Omega$ has a constant rank.

Let us show that $\Theta_\Omega$ is injective. If $\Theta_\Omega(\psi_1 \mathcal{K}_{n,m,p}(\Omega)) = \Theta_\Omega(\psi_2 \mathcal{K}_{n,m,p}(\Omega))$, then $(\psi_1)_*\Omega = (\psi_2)_*\Omega$. This implies that $(\psi_1)^{-1}\psi_2 \in \mathcal{K}_{n,m,p}(\Omega)$ and, therefore, $\psi_1 \mathcal{K}_{n,m,p}(\Omega) = \psi_2 \mathcal{K}_{n,m,p}(\Omega)$. Therefore, $\Theta_\Omega$ is a smooth immersion.

By Proposition 5.18 in [36], the image of $\Theta_\Omega$ (i.e., the equivalence class $[\Omega]_{\mathcal{G}}$) is an immersed submanifold such that $\Theta_\Omega : \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega) \to [\Omega]_{\mathcal{G}}$ is a diffeomorphism and, therefore, the dimension of $[\Omega]_{\mathcal{G}}$ is equal to the dimension of $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$.

A more concrete quantification of privacy can be given for various special cases. Using the results of Proposition IV.3 and Lemma IV.4, we have that, for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$, the uncertainty sets under the scenario described in Problem II.1 are smooth manifolds of dimension greater or equal to the value in (IV.24)

The dimension of the uncertainty sets for prime systems can be shown to be greater or equal to the value in (IV.25) using Lemma IV.6. □

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

We can determine the knowledge the cloud can extract about the plant by considering what properties remain invariant under isomorphisms. Since controllability, observability, and the relative degree remain invariant, the cloud will not learn anything else beyond knowing that the plant is controllable, observable, and has a certain relative degree.

**Example IV.8.** To illustrate how different the systems produced by the proposed encoding scheme can be, consider a system with the following dynamics:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We arbitrarily choose two sets of isomorphisms $\psi_1, \psi_2 \in \mathcal{G}_{n,m,p}$ such that the elements of their constituent matrices are between 0 and 1 (i.e., we pick isomorphisms from a bounded set of $\mathcal{G}_{n,m,p}$). We will not be explicitly writing these isomorphisms here due to space limitations. Applying these isomorphisms to the system above, we arrive at completely different systems $\tilde{\Sigma}_1 = \psi_{1*}\Sigma$ and $\tilde{\Sigma}_2 = \psi_{2*}\Sigma$:

$$\tilde{A}_1 = \begin{bmatrix} 35 & 9.4 & -40 \\ -3 & 0.1 & 2.9 \\ 28 & 8.3 & -33 \end{bmatrix}, \quad \tilde{A}_2 = \begin{bmatrix} -4.06 & 5.35 & 1.48 \\ 4.87 & -4.0 & -2.33 \\ 0.68 & 2.40 & -0.88 \end{bmatrix},$$

$$\tilde{B}_1 = \begin{bmatrix} 16 & -7.7 \\ -2.2 & 1.5 \\ 13 & -6.1 \end{bmatrix}, \quad \tilde{B}_2 = \begin{bmatrix} 0.16 & 0.95 \\ 1.03 & -1.27 \\ 0.70 & -0.31 \end{bmatrix},$$

$$\tilde{C}_1 = \begin{bmatrix} 2.4 & 0.02 & -1.8 \\ 1.5 & 0.01 & -1.1 \end{bmatrix}, \quad \tilde{C}_2 = \begin{bmatrix} -0.33 & -1.68 & 1.56 \\ 3.39 & -4.58 & -0.97 \end{bmatrix}.$$

Proposition IV.7 can be used to quantify privacy of other scenarios presented in Section II.

Consider the scenario in Problem II.2, where the cloud does not know the dynamics but knows which sensors and actuators will be used. An arbitrary isomorphism can no longer be used for encoding since it could lead to inputs and outputs that are inconsistent with existing sensors and actuators. This inconsistency would signal the cloud that the plant is being dishonest about its measurements and provide the cloud with an opportunity to exploit this fact to gather additional knowledge. Therefore, we need to restrict the group of isomorphisms used for encoding. These isomorphisms are given by any composition of $\psi_1 = (P, 0, I, I)$ for any $P \in GL(n, \mathbb{R})$ and $\psi_2 \in \mathcal{K}_{n,m,p}(\Sigma)$. It can be shown that this set of isomorphisms forms a subgroup that we denote by $\mathcal{H}_{n,m,p}(\Sigma) \subset \mathcal{G}_{n,m,p}$.

**Corollary IV.9.** *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem II.2, the cloud cannot distinguish between $\Omega$ and any other system in the uncertainty set $[\Omega]_{\mathcal{H}}$ (i.e., the equivalence class of $\Omega$ defined by the action of $\mathcal{H}_{n,m,p}$) of dimension:*

$$dim\ \mathcal{H}_{n,m,p}(\Sigma) - dim\ \mathcal{K}_{n,m,p}(\Omega), \tag{IV.26}$$

*if Algorithm 1 is used. This implies that the dimension of $[\Omega]_{\mathcal{H}}$ is greater or equal to $n(n+1)$.*

*Proof.* From Theorem IV.2, we know that Algorithm 1 renders isomorphic systems indistinguishable by the cloud. However, the uncertainty set is no longer the equivalence class under the entire group of isomorphisms $\mathcal{G}_{n,m,p}$, but the equivalence class under a smaller group $\mathcal{H}_{n,m,p}(\Sigma)$ denoted by $[\Omega]_{\mathcal{H}}$.

It can be shown that $\mathcal{H}_{n,m,p}(\Sigma)$ is a Lie subgroup of $\mathcal{G}_{n,m,p}$. This subgroup $\mathcal{H}_{n,m,p}(\Sigma)$ can be thought of as a product manifold of $\mathcal{K}_{n,m,p}(\Sigma)$ and a space of invertible affine maps. Since the dimension of a product manifold is a sum of its factors' dimensions, we have:

$$\text{dim } \mathcal{H}_{n,m,p}(\Sigma) = \text{dim } \mathcal{K}_{n,m,p}(\Sigma) + n(n+1).$$

The result follows by applying Proposition IV.7 to $\mathcal{H}_{n,m,p}(\Sigma)$. Using the result from Lemma IV.4, we can see that the dimension of the uncertainty set for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is greater or equal to $n(n+1)$. □

Since in this scenario the plant can no longer change the input, the cloud will learn the transfer function, but not the particular realization of the plant. The cloud would still be unable to learn the trajectory of the state.

Finally, in the scenario described in Problem II.3, where the cloud possesses the complete knowledge of dynamics, only the isomorphisms from the symmetry subgroup $\psi \in \mathcal{K}_{n,m,p}(\Sigma)$ can be used. To provide privacy guarantees for this scenario, let us assume that we have $n+1$ linearly independent constraints on the state $x_k$ expressed by the constraint matrix $D$. This is a reasonable assumption because systems often have an operational envelope bounding the states. Therefore, any $\psi \in \mathcal{K}_{n,m,p}(\Omega)$ must satisfy:

$$\begin{aligned} DL^{-1} = D &\iff DL = D \\ &\iff \begin{bmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} P & 0 \\ F & G \end{bmatrix} = \begin{bmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{bmatrix} \\ &\implies D_{11}P = D_{11}. \end{aligned}$$

Given that $D_{11} \in \mathbb{R}^{h_1 \times (n+1)}$ is injective, the last equality is satisfied if and only if $P = I$. Since $P$ uniquely defines $F$, $G$ and $S$, we also have that the only isomorphism that keeps $(A, B, C, D_{11})$ invariant is $\psi = \psi_e = (I, 0, I, I)$ . Therefore, the only element of $\mathcal{K}_{n,m,p}(\Omega)$ is $\phi_e = (I, 0, I, I)$ and dim $\mathcal{K}_{n,m,p}(\Omega) = 0$.

**Corollary IV.10.** *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem II.3, the cloud cannot distinguish between $\Omega$ and any other system in the uncertainty set $[\Omega]_{\mathcal{K}}$ (i.e., the equivalence class of $\Omega$ defined by the action of $\mathcal{K}_{n,m,p}(\Sigma)$) of dimension:*

$$dim\ \mathcal{K}_{n,m,p}(\Sigma) - dim\ \mathcal{K}_{n,m,p}(\Omega), \tag{IV.27}$$

*if Algorithm 1 is used.*

*When the constraint matrix $D$ contains $n+1$ linearly independent constraints on the state, the dimension of the uncertainty set is equal to dim $\mathcal{K}_{n,m,p}(\Sigma)$, which is less or equal to:*

$$n(m+1) - \sum_{i=2}^{m} r_{i-1} r_i,$$

*where $r_i$ is given in Lemma IV.3.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2982611, IEEE Transactions on Automatic Control

*Moreover, for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ such that its corresponding $\Sigma \in \mathcal{S}_{n,m,p}$ is prime, the dimension of $[\Omega]_{\mathcal{K}}$ is greater or equal to*

$$\sum_{i=1}^{m} r_{k_i} + m$$

.

*Proof.* The proof of this statement is similar to that of Corollary IV.9. The dimensions of equivalence classes for prime and general systems were evaluated using results of Proposition IV.3 and Lemma IV.6. □

In this scenario, by applying Algorithm 1, the plant would be able to conceal the state trajectory from the cloud.

To illustrate the main results of this section, consider the following example.

**Example IV.11.** Consider a drone with linearized dynamics given in [42] and a bounded operational envelope (i.e., constraints on the extreme values of its state). From the linear model in [42] we observe that $n = 12$, $m = 4$, $p = 4$ and $r_1 = 4$, $r_2 = 4$, $r_3 = 2$, $r_4 = 2$. Suppose we decide to offload the control of this drone to the cloud. Let us evaluate the privacy guarantees Algorithm 1 can provide in each of the scenarios described in Section II.

In the first scenario, when the cloud has no prior knowledge about the drone, we can choose any $\psi \in \mathcal{G}_{n,m,p}$. Therefore, using Propositon IV.7, we estimate the dimension of the uncertainty set to be greater than 212.

In the second scenario, when the cloud knows what sensors and actuators the drone has, we must choose an isomorphism $\psi \in \mathcal{H}_{n,m,p}(\Sigma)$ to keep inputs and outputs consistent. A practical example of this could be if the cloud was owned by a company that provides computations specifically for drones. In this case, we use Corollary IV.9 and estimate the dimension of the uncertainty set to be greater than 156.

Finally, when the cloud has complete knowledge about the plant, we are forced to choose a symmetry $\psi \in \mathcal{K}_{n,m,p}(\Sigma)$ to keep the dynamics unchanged. This scenario could, for example, occur if the cloud belongs to the drone's manufacturer. Using Corollary IV.10, we estimate the dimension of the uncertainty set to be less or equal than 32. Unfortunately, we generally cannot provide a guarantee for the lower bound in this scenario. The dimension of the uncertainty set, however, can be found exactly by determining $\mathcal{K}_{n,m,p}(\Sigma)$ for a given $\Sigma$.

## V. Side knowledge

The privacy guarantees derived in Section IV are compromised when the adversary has partial information about the encoding isomorphism. In our problem formulation, we assume that the cloud may have learned those through some external channels or through some prior knowledge about the system.

Recall that by Lemma III.7, $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1) + m(n+1) + m^2 + p(p+1)$. In this section, we assume that the constraint matrix $D$ has $n+1$ linearly independent constraints on the state and, therefore, as shown in the previous section, $\mathcal{K}_{n,m,p}(\Omega) = \{\psi_e\}$, where $\psi_e$ is the identity element of $\mathcal{G}_{n,m,p}$.

Suppose the cloud has partial knowledge about the encoding isomorphism. We shall represent the partial knowledge available to the cloud as a projection from $\mathcal{G}_{n,m,p}$ onto a $k$-dimensional vector space. Let us define $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ to be a surjective map of constant rank $k$, providing side knowledge about the encoding isomorphism. Then, we can say that the cloud knows some vector $l \in \mathbb{R}^k$, where:

$$l = \rho(P, F, G, S). \tag{V.1}$$

Note that this map is not known to us, and the results that follow do not require the knowledge of this map.

Side knowledge does not change the result of Theorem IV.2, however the privacy guaranteed by the scheme changes. It is obvious that the size of the uncertainty set defined by isomorphisms that satisfy (V.1) is no greater and, in general, smaller than if no side knowledge is available. Moreover, the uncertainty set is no longer neither an orbit nor an equivalence class because the preimage of $\rho$ does not necessarily have a group structure.

Let us show that the object defined by (V.1) on $\mathcal{G}_{n,m,p}$ is still a manifold.

**Lemma V.1.** *Let $\mathcal{G}_{n,m,p}$ be the group of all isomorphisms, $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ be a surjective map of constant rank $k$ and assume the cloud knows that $l = \rho(P, F, G, S)$. Then, $\rho^{-1}(l)$, representing the possible encoding isomorphisms used by the client, is a properly embedded submanifold of $\mathcal{G}_{n,m,p}$. Its dimension is dim $\mathcal{G}_{n,m,p} - k$.*

*Proof.* By the global rank theorem [36, p. 83], since $\rho$ is a surjective map of constant rank $k$, it is a smooth submersion. From the submersion level set theorem [36, p. 105], since both $\mathcal{G}_{n,m,p}$ and $\mathbb{R}^k$ are smooth manifolds and $\rho$ is a smooth submersion, we have that $\rho^{-1}(l)$ is a properly embedded submanifold of dimension dim $\mathcal{G}_{n,m,p}$ − dim $\mathbb{R}^k = n(n+1) + m(n+1) + m^2 + p(p+1) - k$. □

Let us now consider the map $\Theta_\Omega$ defined earlier in Proposition IV.7. Since $\mathcal{K}_{n,m,p}(\Omega) = \psi_e$, we have that $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is equivalent to $\mathcal{G}_{n,m,p}$. Therefore, the map $\Theta_\Omega$ is equivalent to the orbit map $\theta_\Omega$. It was shown in Proposition IV.7 that $\Theta_\Omega$ is injective. The image of $\Theta_\Omega(\rho^{-1}(l))$ constitutes the uncertainty set, between the elements of which the cloud is not be able to distinguish. Therefore, the main result of this section requires finding the dimension of $\Theta_\Omega(\rho^{-1}(l))$.

**Proposition V.2.** *Assume $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is such that the constraint matrix $D$ has $n+1$ linearly independent constraints on the state. Suppose that Algorithm 1 is used and the cloud has the following side knowledge about the selected isomorphism $\psi$:*

$$\rho(P, F, G, S) = l \in \mathbb{R}^k,$$

*where $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ is a surjective map of constant rank $k$. Then, under the scenario described in Problem II.1, the*

*cloud cannot distinguish between $\Omega$ and any other system in the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ of dimension:*

$$dim\ \mathcal{G}_{n,m,p} - k = n(n+1) + m(n+1) + m^2 + p(p+1) - k. \tag{V.2}$$

*Proof.* By Theorem IV.2, Algorithm 1 renders isomorphic systems indistinguishable by the cloud. However, the cloud knows that we use an isomorphism $\psi \in \rho^{-1}(l)$ and, therefore, the uncertainty set is no longer the equivalence class under the entire group of isomorphisms $\mathcal{G}_{n,m,p}$, but the subset of this equivalence class $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$.

By the property of the orbit map [36, p. 166], for each $\Omega$, the orbit map $\Theta_\Omega$ is smooth and has constant rank. Since $\Theta_\Omega$ is also injective, we have, by the Global Rank Theorem, that it is a smooth immersion [36, p. 83]. As it was shown in Lemma V.1, the set $\rho^{-1}(l)$ is an embedded submanifold of $\mathcal{G}_{n,m,p}$ and, therefore, the inclusion map $i : \rho^{-1}(l) \to \mathcal{G}_{n,m,p}$ is a smooth embedding.

The map $\Theta_\Omega \circ i$ is a smooth immersion because it is a composition of smooth immersions. Since images of smooth immersions are smooth immersed submanifolds (by Proposition 5.18 from [36]), the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ is a smooth immersed submanifold of $\bar{\mathcal{S}}_{n,m,p}$ diffeomorphic to $\rho^{-1}(l)$ and, hence, has the same dimension (refer to Lemma V.1).

Using Lemma III.7, the dimension of the uncertainty set is evaluated to be:

$$n(n+1) + m(n+1) + m^2 + p(p+1) - k.$$

□

Remark: although Proposition V.2 was proved under the assumption that $D$ has $n+1$ linearly independent constraints on the state, this assumption can be dropped if we assume the intersection of $\rho^{-1}(l)$ and the left cosets of $\mathcal{K}_{n,m,p}(\Omega)$ in $\mathcal{G}$ is well-behaved.

This result shows that the proposed scheme degrades gracefully with side knowledge — i.e., side knowledge allows the cloud to reduce the dimension of the uncertainty set only by the amount of side knowledge and not more. Moreover, this result can be generalized for other scenarios considered in Section IV-B2 using similar proofs.

**Corollary V.3.** *Assume $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is such that the constraint matrix $D$ has $n+1$ linearly independent constraints on the state. Suppose that Algorithm 1 is used and the cloud has the following side knowledge $l \in \mathbb{R}^k$ about the selected isomorphism $\psi$:*

$$l = \rho(P, F, G, S),$$

*where $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ is a surjective map of constant rank $k$. Then, under the scenario described in Problem II.2, the cloud cannot distinguish between $\Omega$ and any other system in the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ of dimension:*

$$dim\ \mathcal{H}_{n,m,p}(\Sigma) - k. \tag{V.3}$$

*Under the scenario described in Problem II.3, the dimension of the uncertainty set is:*

$$dim\ \mathcal{K}_{n,m,p}(\Sigma) - k. \tag{V.4}$$

## VI. Conclusion

In this paper, we proposed a transformation-based method to preserve privacy in control over the cloud. In addition to its low computational overhead, we have formally shown that this method precludes the adversary from inferring the private data by eavesdopping on the messages exchanged between the plant and the cloud. We quantified the guaranteed privacy via the dimension of the set that describes the uncertainty experienced by the adversary. The problem of computing the dimension of the stabilizer set $\mathcal{K}_{n,m,p}(\Omega)$ remains open, and its solution requires a detailed analysis of system-theoretic properties. The authors are currently investigating other measures of privacy that may lead to a deeper insight into the proposed method. As part of the future work, we aim to perform a feasibility study of this encryption scheme by implementing it on a physical testbed.

## References


[1] A. Vick, J. Guhl, and J. Kruger, "Model predictive control as a service - concept and architecture for use in cloud-based robot control," in *the 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Aug 2016, pp. 607–612.

[2] T. Hegazy and M. Hefeeda, "Industrial automation as a cloud service," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2750–2763, Oct 2015.

[3] B. Hoh, T. Iwuchukwu, Q. Jacobson, D. Work, A. M. Bayen, R. Herring, J. C. Herrera, M. Gruteser, M. Annavaram, and J. Ban, "Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 849–864, May 2012.

[4] Y. Lin, F. Farokhi, I. Shames, and D. Nezic, "Secure control of nonlinear systems using semi-homomorphic encryption," in *the 57th IEEE Conference on Decision and Control*, 2018, pp. 5002–5007.

[5] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, pp. 2967–2986, 2014.

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11, 2011, pp. 6–6.

[7] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, and A. Winnicki, "Cyber-physical systems security: Experimental analysis of a vinyl acetate monomer plant," in *the 1st ACM Workshop on Cyber-Physical System Security*, 2015, pp. 1–12.

[8] F. M. Cleveland, "Cyber security issues for advanced metering infrasttructure (ami)," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, July 2008, pp. 1–5.

[9] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green lights forever: Analyzing the security of traffic infrastructure," in *Proceedings of the 8th USENIX Conference on Offensive Technologies*, 2014, pp. 7–7.

[10] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, "Secure control systems: A quantitative risk management approach," *IEEE Control Systems*, vol. 35, no. 1, pp. 24–45, 2015.

[11] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, Oct 2009, pp. 911–918.

[12] F. Farokhi, I. Shames, and N. Batterham, "Secure and private control using semi-homomorphic encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, Oct 2017.

[13] H. Sandberg, S. Amin, and K. H. Johansson, "Cyberphysical security in networked control systems: An introduction to the issue," *IEEE Control Systems*, vol. 35, no. 1, pp. 20–23, Feb 2015.

[14] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in *HotSec'10 Proceedings of the 5th USENIX conference on Hot topics in security*. Berkeley, CA, USA: USENIX Association, July 2010, pp. 1–8.

[15] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-based MPC with Encrypted Data," *ArXiv e-prints*, 2018.

[16] F. Armknecht, C. Boyd, C. Carr, K. Gjosteen, A. Jaeschke, C. A. Reuter, and M. Strand, "A guide to fully homomorphic encryption," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1192, 2015.
[17] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6836–6843.
[18] T. Fujita, K. Kogiso, K. Sawada, and S. Shin, "Security enhancements of networked control systems using rsa public-key cryptosystem," in *2015 10th Asian Control Conference (ASCC)*, May 2015, pp. 1–6.
[19] J. Kim, C. Lee, H. Shim, J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Encrypting controller using fully homomorphic encryption for security of cyber-physical systems," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 175 – 180, 2016, 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.
[20] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 163 – 168, 2016, 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems.
[21] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 5053–5058.
[22] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based Quadratic Optimization with Partially Homomorphic Encryption," *arXiv e-prints*, Sep. 2018.
[23] K. Kogiso, R. Baba, and M. Kusaka, "Development and examination of encrypted control systems," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2018, pp. 1338–1343.
[24] J. Cortes, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *2016 IEEE 55th Conference on Decision and Control*, Dec 2016, pp. 4252–4272.
[25] F. Koufogiannis and G. J. Pappas, "Differential privacy for dynamical sensitive data," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 1118–1125.
[26] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.
[27] O. L. Mangasarian, "Privacy-preserving linear programming," *Opt. Letters*, vol. 5, no. 1, pp. 165–172, Feb 2011.
[28] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 820–828.
[29] P. C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras, "Per-se privacy preserving solution methods based on optimization," in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, Dec 2013, pp. 206–211.
[30] P. Weeraddana and C. Fischione, "On the privacy of optimization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9502 – 9508, 2017, 20th IFAC World Congress.
[31] Z. Xu and Q. Zhu, "Secure and resilient control design for cloud enabled networked control systems," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, 2015, pp. 31–42.
[32] D. Wu, B. C. Lesieutre, P. Ramanathan, and B. Kakunoori, "Preserving privacy of AC optimal power flow models in multi-party electric grids," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2050–2060, July 2016.
[33] A. Sultangazin and P. Tabuada, "Towards the use of symmetries to ensure privacy in control over the cloud," in *2018 IEEE 57th Conference on Decision and Control*, Dec 2018, pp. 5008–5–13.
[34] A. Sultangazin, S. Diggavi, and P. Tabuada, "Protecting the privacy of networked multi-agent systems controlled over the cloud," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, July 2018, pp. 1–7.
[35] A. Sultangazin and P. Tabuada, "Symmetries and privacy in control over the cloud: uncertainty sets and side knowledge," University of California, Los Angeles, Tech. Rep., 05 2019. [Online]. Available: http://www.cyphylab.ee.ucla.edu/Home/publications/UCLA-CyPhyLab-2019-01.pdf?attredirects=0
[36] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. Springer-Verlag New York, 2003.
[37] B. Vandereycken, P. A. Absil, and S. Vandewalle, "Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank," in *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, Aug 2009, pp. 389–392.
[38] W. Respondek, "Symmetries and minimal flat outputs of nonlinear control systems," in *New Trends in Nonlinear Dynamics and Control and their Applications*, W. Kang, C. Borges, and M. Xiao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 65–86.
[39] M. A. Beitia, J. M. Gracia, and I. de Hoyos, "A linear matrix equation: a criterion for block similarity," *Linear and Multilinear Algebra*, vol. 31, pp. 93–118, 1992.
[40] P. Antsaklis and A. Michel, *Linear Systems*. Birkhäuser Boston, 2005.
[41] A. J. Laub, *Matrix Analysis For Scientists And Engineers*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004.
[42] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," Master's thesis, KTH Electrical Engineering, 6 2015.

**Paulo Tabuada** was born in Lisbon, Portugal, one year after the Carnation Revolution. He received his "Licenciatura" degree in Aerospace Engineering from Instituto Superior Tecnico, Lisbon, Portugal in 1998 and his Ph.D. degree in Electrical and Computer Engineering in 2002 from the Institute for Systems and Robotics, a private research institute associated with Instituto Superior Tecnico. Between January 2002 and July 2003 he was a postdoctoral researcher at the University of Pennsylvania. After spending three years at the University of Notre Dame, as an Assistant Professor, he joined the Electrical and Computer Engineering Department at the University of California, Los Angeles, where he currently is the Vijay K. Dhir Professor of Engineering.

Paulo Tabuada's contributions to cyber-physical systems have been recognized by multiple awards including the NSF CAREER award in 2005, the Donald P. Eckman award in 2009, the George S. Axelby award in 2011, the Antonio Ruberti Prize in 2015, and the grade of fellow awarded by IEEE in 2017. He has been program chair and general chair for several conferences in the areas of control and of cyber-physical systems such as NecSys, HSCC, and ICCPS. He currently serves as the chair of HSCC's steering committee and he served on the editorial board of the IEEE Embedded Systems Letters and the IEEE Transactions on Automatic Control.

**Alimzhan Sultangazin** was born in Karagandy, Kazakhstan. He received the B.Eng. degree in electrical and electronic engineering from the Nazarbayev University, Astana, Kazakhstan, in 2017 and the M.S. degree in electrical and computer engineering from the University of California, Los Angeles (UCLA) in 2019. From 2019, he is pursuing his Ph.D. degree in electrical and computer engineering at UCLA. Currently, his main research interest is security and privacy in cyber-physical systems.