2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Symmetries and privacy in control over the cloud: uncertainty sets and side knowledge*

Alimzhan Sultangazin[1] and Paulo Tabuada[1]

Abstract— **Control algorithms, like model predictive control, can be computationally expensive and may benefit from being executed over the cloud. This is especially the case for nodes at the edge of a network since they tend to have reduced computational capabilities. However, control over the cloud requires transmission of sensitive data (e.g., system dynamics, measurements) which undermines privacy of these nodes. When choosing a method to protect the privacy of these data, efficiency must be considered to the same extent as privacy guarantees to ensure adequate control performance. In this paper, we review a transformation-based method for protecting privacy, previously introduced by the authors, and quantify the level of privacy it provides. Moreover, we also consider the case of adversaries with side knowledge and quantify how much privacy is lost as a function of the side knowledge of the adversary.**

I. Introduction

A. Motivation

With an increasing connectivity of devices, there has been a marked growth in the use of cloud-based services, where a powerful server provides memory and computation to clients. In control, these ideas took form into control over the cloud — a technique where a controller is placed on the cloud, while both measurements and control inputs are exchanged by the plant and the cloud over a communication channel.

Control over the cloud provides numerous advantages, such as the opportunity to outsource expensive tasks to the cloud, easier installation and maintenance [1], and access to information from all of the cloud's clients for control decisions [2]. Several works [2], [3] have shown practical feasibility of model predictive control (MPC) over the cloud.

Notwithstanding the benefits of control over the cloud, the exposure of systems to the cloud can cause security vulnerabilities in a variety of applications [4], [5], including control of process plants and traffic infrastructure. One of the attacks exploiting these vulnerabilities is eavesdropping. In control over the cloud, an eavesdropping attack happens when an adversary listens in on the communication to capture information about the model, objective, and trajectory [6].

Traditionally, eavesdropping attacks are prevented with encryption - the client and the cloud establish a shared key with which they encrypt transmitted messages and decrypt the messages they receive. This technique, however, fails to protect the system if the privacy breach occurs within the cloud. Hence, there is a need for control-over-the cloud methods that do not require the incoming data to be decrypted by the cloud. While approaching this problem, one must surely keep in mind two other important concerns: efficiency and safety. We must not achieve privacy at the cost of degradation of control performance either due to delays in the feedback loop or inaccurate control inputs.

B. Related work

So far the problem of privacy in control over the cloud has been approached under the frameworks of homomorphic encryption, differential privacy, and algebraic transformations.

Homomorphic encryption techniques allow the cloud to perform the necessary computations on encrypted data without decrypting it [7]. Fully homomorphic encryption (FHE) was considered in [8]. Unfortunately, FHE is impractical for online optimization due to its execution time [7]. This has led to increased interest in partially homomorphic encryption (PHE), see [1], [6], [9], [10]. While improved, execution time remains a valid concern in PHE methods [6].

The notion of differential privacy was also recently applied to privacy in control (see [11], [12]). The main idea of these methods is to perturb the response to a data query with appropriate noise. 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.

Algebraic transformation methods, to which this work relates, were created to preserve privacy of optimization problems. The main idea of these methods is to provide the cloud with a different, but equivalent optimization problem. Then, the optimal solution to the original problem can be recovered from the optimal solution of this equivalent problem, provided by the cloud. These methods found applications in control due to their efficiency and guaranteed optimality of the solution [13]. For example, in [14] the authors propose a hybrid transformation-based method to preserve privacy of an MPC controller in networked control systems. In [15], transformation-based methods are used to provide privacy in a specific problem of AC optimal power flow.

C. Contributions

While efficiently enforcing privacy in control systems is difficult, some features of dynamical systems can be used to our advantage. We propose to use isomorphisms and symmetries of the dynamics as a source of transformations

*The work of the authors was partially supported by the NSF grants 1740047, 1705135; UC-NL grant LFR-18-548554; and 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 US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

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

978-1-7281-1398-2/19/$31.00 ©2019 IEEE

so as to keep not only the optimization problems, but also the resulting plant dynamics, equivalent. The advantages of this approach are increased computational efficiency, computation on encoded data and simplicity of design.

The proposed method was initially introduced in [16]. As opposed to [14], it applies to a more general class of quadratic programs and provides encoding for the state and the output. In comparison to [15], our method is also more general since it is applied to a wider class of problems and considers the scenarios when the output is different from the state. In [17], this method was extended to networked control systems with several agents and a single cloud. In this work, we address two issues that were not discussed in [16], [17]:

1) we propose a measure of privacy and compute it for the different scenarios discussed in the paper;
2) we quantify how much privacy is lost when an adversary has access to side knowledge.

II. Problem Definition

A. Plant dynamics and control objective

We consider discrete-time *affine* plants, denoted by Σ, and described by:

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

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, and $\bar{x} \in \mathbb{R}^n$, $\bar{u} \in \mathbb{R}^m$ and $\bar{y} \in \mathbb{R}^p$ denote the state, input and output of the system, respectively. We assume that system Σ is controllable and observable. We also assume, without loss of generality, that $\ker \bar{B} = 0$ and $\operatorname{Im} \bar{C} = \mathbb{R}^p$, since we can always eliminate linearly independent columns (resp. rows) from $\bar{B}$ (resp. $\bar{C}$).

To simplify notation, we lift every affine map $Ax + c$ to a linear map as follows:

$$Ax + c \mapsto \begin{bmatrix} A & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}. \tag{II.2}$$

Applying (II.2) to (II.1), we obtain:

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

In what follows we conceal the inner structure for simplicity. Nevertheless, the reader should 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 Σ if it satisfies (II.1) for all $k \in \mathbb{N}$.

Additionally, we consider a cost function $J : \mathbb{R}^n \times (\mathbb{R}^m)^{N+1} \to \mathbb{R}$ for $N \in \mathbb{N} \cup \{+\infty\}$ that, by comparing trajectories, allows formulating different control objectives. We consider quadratic cost functions given by:

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

where $\Delta\eta_k = \begin{bmatrix} x_k - x_k^* & u_k - u_k^* \end{bmatrix}^T$, $x = \{x_0, ..., x_N\}$ and $u = \{u_0, ..., u_N\}$. Sequences $x^* = \{x_0^*, ..., x_N^*\}$ and $u^* = \{u_0^*, ..., u_N^*\}$ denote the desired setpoints. We define $M \in \mathbb{R}^{(n+m+1)\times(n+m+1)}$ to be a positive-definite matrix.

Along with the cost, control objectives require certain constraints to be satisfied at all times. We define them as:

$$D\eta_k \leq 0, \quad k = 0, ..., N, \tag{II.5}$$

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

We also assume that there are $n+1$ linearly independent constraints on the state x_k. This is a valid assumption since real-world systems usually have a bounded operational envelope (e.g., maximum velocity or maximum pressure).

B. Attack model and privacy objectives

The cloud is considered to be a curious but honest adversary: the cloud adheres to the agreed-upon protocol, but may try to extract confidential information by keeping record of all communicated messages.

The interaction between the plant and the cloud is performed in two steps:

1) *Handshaking*: the plant communicates to the cloud a suitably modified version of its model, cost, and constraints. In return, the cloud agrees to find the input minimizing the cost, subject to model and constraints;
2) *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.

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

1) the cloud has no knowledge about the plant;
2) the cloud has no knowledge about the plant, except for knowing what its sensors and actuators are;
3) the cloud has complete knowledge about the plant dynamics.

These scenarios dictate which modifications can be applied. For more details on the scenarios, refer to [16].

III. Main definitions and summary of previous results

In this section, we recap the results from [16] to provide the context for the main result of this work. The key notions that were discussed in [16] are those of isomorphism and symmetry of control systems.

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 $\psi = (P, F, G, S)$ comprising a change of state coordinates $P : \mathbb{R}^n \to \mathbb{R}^n$, a 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 $S : \mathbb{R}^p \to \mathbb{R}^p$, where P and S are affine invertible maps, F is an affine map, and G is a linear invertible map.

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 with function composition as the group operation[1]. This allows us to define a group action of $\mathcal{G}_{n,m,p}$ on $\mathcal{S}_{n,m,p}$.

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:

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

This map is called an isomorphism action.

The isomorphism ψ maps the trajectory $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ of Σ to the trajectory $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$ of $\psi_*\Sigma$ as follows:

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

Similarly, the isomorphism ψ induces transformation on the control objectives (i.e., the cost and constraints) by modifying η_k as follows:

$$\tilde{\eta}_k = \begin{bmatrix}\tilde{x}_k\\ \tilde{u}_k\end{bmatrix} = \begin{bmatrix}P & 0\\ F & G\end{bmatrix}\begin{bmatrix}x_k\\ u_k\end{bmatrix} \triangleq L\eta_k. \quad \text{(III.3)}$$

Therefore, to express the cost function J and constraints in (II.5) in terms of modified states $\tilde{x}$ and inputs $\tilde{u}$, we need to use a modified cost and modified constraints:

$$\tilde{J}(\tilde{x}, \tilde{u}) = \psi_*J\,(x, u) = \sum_{k=0}^{N} \Delta\tilde{\eta}_k^T \tilde{M} \Delta\tilde{\eta}_k, \quad \text{(III.4)}$$

$$\tilde{D}\tilde{\eta}_k \leq 0, \quad k = 0, ..., N, \quad \text{(III.5)}$$

where $\tilde{M} = L^{-T}ML^{-1}$ and $\tilde{D} = \psi_*D = DL^{-1}$.

Let us now define $\bar{\mathcal{S}}_{n,m,p}$ to be a set of all quadruples $\big(\Sigma, J, D, \{x_k, y_k, u_k\}_{k\in\mathbb{N}}\big)$ such that $\{x_k, y_k, u_k\}_{k\in\mathbb{N}}$ is a trajectory of Σ minimizing cost function J under constraints D, where D contains $n+1$ linearly independent constraints on x_k. 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 action of $\mathcal{G}_{n,m,p}$ to define the equivalence relation on $\bar{\mathcal{S}}_{n,m,p}$.

Definition III.3. 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 $\Omega \sim_\mathcal{G} \tilde{\Omega}$, is defined by the existence of $\psi \in \mathcal{G}_{n,m,p}$ such that $\tilde{\Omega} = \psi_*\Omega$, — 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).

Algorithm 1 (Plant ($\mathcal{P}$)$\Longleftrightarrow$ Cloud ($\mathcal{C}$))

```
Input: $\mathcal{P}$: $\psi$, $\Sigma$, $J$, $D$, $y_k$, $\tilde{u}_k$;
       $\mathcal{C}$: $\tilde{y}_k$, $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$
Output: $\mathcal{P}$: $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$, $\tilde{y}_k$;
        $\mathcal{C}$: $\tilde{u}_k$
  Phase 1: Handshaking
  $\mathcal{P}$: Encode $\tilde{\Sigma} = \psi_*\Sigma$, $\tilde{J} = \psi_*J$ and $\tilde{D} = \psi_*D$;
  $\mathcal{P}$: Output $\tilde{\Sigma}$, $\tilde{J}$, and $\tilde{D}$ to the cloud;
  Phase 2: Execution
  $\mathcal{P}$: Encode measurements as $\tilde{y}_k = Sy_k$ and send to the cloud;
  $\mathcal{C}$: Use $\tilde{y}_k$ to estimate the state $\tilde{x}_k$ and compute the input $\tilde{u}_k$ minimizing $\tilde{J}$ subject to the constraints $\tilde{D}$ and the dynamics $\tilde{\Sigma}$;
  $\mathcal{C}$: Send $\tilde{u}_k$ to the plant;
  $\mathcal{P}$: Use $\psi$ to decode $\tilde{u}_k$ and produce $u_k$ using (III.2).
```

The equivalence class of $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ via equivalence relation $\sim_\mathcal{G}$ is the set:

$$[\Omega] \triangleq \{\Omega' \in \bar{\mathcal{S}}_{n,m,p} | \exists\psi \in \mathcal{G}_{n,m,p} \text{ such that } \Omega' = \psi_*\Omega\}$$

For a given system Σ, there is also a special subgroup in $\mathcal{G}_{n,m,p}$ called the subgroup of symmetries.

Definition III.4. Let $\Sigma \in \mathcal{S}_{n,m,p}$. An isomorphism ψ of Σ is a symmetry of Σ if $\psi_*\Sigma = \Sigma$. The subgroup of symmetries of Σ is denoted here by $\mathcal{K}_{n,m,p}(\Sigma)$.

In [16], we have proposed to use Algorithm 1 to preserve privacy of information communicated to the cloud. We have shown that, by using this scheme, information communicated to the cloud remains *consistent* (i.e., after the modification, the resulting trajectory remains a valid trajectory of the modified dynamics) and the plant is able to *perfectly reconstruct* the desired input (i.e., the optimal solution of the original optimization problem) from the cloud's input.

We use isomorphisms to prevent the cloud from distinguishing between isomorphic systems and, thus, keep the communicated system private from the cloud. We now formalize the notion of indistinguishability.

Definition III.5. A protocol renders two quadruples Ω and $\hat{\Omega}$ indistinguishable by the cloud if the exchanged messages, when using the protocol between the cloud and plant Ω, and the exchanged messages, when using the protocol between the cloud and plant $\hat{\Omega}$, can be made the same.

Theorem III.6 (Theorem III.6 in [16]). *Algorithm 1 renders isomorphic systems* $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}})$ *and* $\tilde{\Omega} = (\tilde{\Sigma}, \tilde{J}, \tilde{D}, \{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}})$ *by the cloud.*

The result described in Theorem III.6 states that the cloud cannot differentiate between any two plants, costs, constraints or trajectories contained in the same-equivalence

[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 $\psi^{-1} = (P^{-1}, -G^{-1}FP, G^{-1}, S^{-1})$.

class, thereby protecting the privacy of the system. In what follows, we quantify the amount of privacy provided.

IV. Quantifying privacy

Our scheme ensures privacy by preventing the cloud from knowing which quadruple Ω in its equivalence class $[\Omega]$ it is interacting with. Clearly, the larger the equivalence class, the more privacy is guaranteed. Since each equivalence class has infinitely many elements, we cannot use cardinality 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.

We begin by considering the stabilizer subgroup of $\mathcal{K}_{n,m,p}(\Omega) \subset \mathcal{G}_{n,m,p}$ for some $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ defined by:

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

We note that $\mathcal{K}_{n,m,p}(\Omega) \subset \mathcal{K}_{n,m,p}(\Sigma)$ since the stabilizer subgroup must preserve dynamics. To gain insight about the stabilizer subgroup $\mathcal{K}_{n,m,p}(\Omega)$, let us consider the subgroup of symmetries $\mathcal{K}_{n,m,p}(\Sigma)$ in more detail.

In [18], Respondek gives a characterization of symmetries of (A, B). For pairs (A, B), the output is no longer relevant and the isomorphisms degenerate into the form $\phi = (P, F, G)$. This result can be interpreted to state that the symmetry ϕ is uniquely determined by its P. We use the results from [18], [19] to characterize the symmetry subgroup of a controllable system (A, B), denoted by $\mathcal{K}_{n,m}(A, B)$.

Proposition IV.1. *Let (A,B) be a controllable linear system. Then, dim $\mathcal{K}_{n,m}(A, B) = m(n+1) - \sum_{i=2}^{k_1} r_{i-1} r_i$, where:*

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

Using this result, we can estimate the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$. To go from keeping (A, B) invariant to keeping (A, B, C) invariant, we need to find S that keeps C invariant. In other words, assuming that we have found (P, F, G) that preserves (A, B), we need to additionally find S such that $C = SCP^{-1}$. Since we assume C is surjective, this equation has at most one solution. This gives an upper bound on the dimension of the subgroup of symmetries dim $\mathcal{K}_{n,m,p}(\Sigma) \leq m(n+1) - \sum_{i=2}^{k_1} r_{i-1} r_i$. In future work, we plan to further investigate the symmetry subgroup of Σ in order to find what exactly dim $\mathcal{K}_{n,m,p}(\Sigma)$ is equal to.

To find a transformation P that keeps Ω invariant, let us use the assumption that we have $n+1$ linearly independent constraints on the state x_k expressed by the constraint matrix D. Therefore, any $\psi \in \mathcal{K}_{n,m,p}(\Omega)$ must satisfy:

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

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 Ω invariant is $\psi = \psi_e = (I, 0, I, I)$. Therefore, the only element of $\mathcal{K}_{n,m,p}(\Omega)$ is $\phi_e = (I, 0, I, I)$.

Let us now define the orbit map:

$$\begin{aligned} f_\Omega : \mathcal{G}_{n,m,p} &\to \bar{\mathcal{S}}_{n,m,p} \\ \psi &\mapsto \psi_* \Omega. \end{aligned} \tag{IV.2}$$

Since $\mathcal{K}_{n,m,p}(\Omega)$ is trivial, we can show that f_Ω is injective.

The result of the discussion above can be formalized in the following statement.

Lemma IV.2. *Let $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k \in \mathbb{N}})$ be an arbitrary system in $\bar{\mathcal{S}}_{n,m,p}$. Then, $f_\Omega : \mathcal{G}_{n,m,p} \to \bar{\mathcal{S}}_{n,m,p}$, mapping ψ to $\psi_* \Omega$, is injective.*

To facilitate further results, we show that $\bar{\mathcal{S}}_{n,m,p}$ is a smooth manifold and $\mathcal{G}_{n,m,p}$ is a Lie group. The proofs are omitted to conserve space, but provided in [20].

Lemma IV.3. *Let $\bar{\mathcal{S}}_{n,m,p}$ denote the set of controllable and observable systems along with costs and constraints. Then, $\bar{\mathcal{S}}_{n,m,p}$ is a smooth manifold.*

Lemma IV.4. *Let $\mathcal{G}_{n,m,p}$ be the isomorphism group. Then, $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1) + m(n+1) + m^2 + p(p+1)$ acting smoothly, freely, and properly on $\bar{\mathcal{S}}_{n,m,p}$.*

Consider the first scenario from Section II, in which the cloud does not know anything about the system. In this scenario, the plant encodes Ω using any isomorphism $\psi \in \mathcal{G}_{n,m,p}$ that can be regarded as a private key used to encode and decode the information exchanged with the cloud. Using the previous lemmas, we prove the following result.

Proposition IV.5. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Assuming that the cloud has no knowledge about the plant, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_\mathcal{G}$ of dimension equal to:*

$$\text{dim } \mathcal{G}_{n,m,p} = n(n+1) + m(n+1) + m^2 + p(p+1),$$

if Algorithm 1 is used.

Proof. In Theorem III.6 we have shown that Algorithm 1 makes isomorphic systems indistinguishable by the cloud. This means that, for the cloud, the uncertainty set is the set of systems isomorphic to Ω, i.e. the equivalence class $[\Omega]_\mathcal{G}$.

From Lemmas IV.3 and IV.4, we know that $\bar{\mathcal{S}}_{n,m,p}$ is a smooth manifold, and $\mathcal{G}_{n,m,p}$ is a Lie group acting smoothly, freely and properly on $\bar{\mathcal{S}}_{n,m,p}$. Hence, by quotient manifold theorem [21], we have that the orbit space $\bar{\mathcal{S}}_{n,m,p}/\mathcal{G}_{n,m,p}$ is a smooth manifold of dimension dim $\bar{\mathcal{S}}_{n,m,p} -$dim $\mathcal{G}_{n,m,p}$ and the quotient map $\pi : \bar{\mathcal{S}}_{n,m,p} \to \bar{\mathcal{S}}_{n,m,p}/\mathcal{G}_{n,m,p}$ is a smooth submersion. Using submersion level set theorem [21], we can further show that, the orbit $[\Omega]_\mathcal{G} = \pi^{-1}(\omega)$, where ω is a representative element of this orbit in the orbit space, is a submanifold with dimension dim $\mathcal{G}_{n,m,p}$. □

Proposition IV.5 is used to quantify privacy of other scenarios presented in Section II.

Consider the second scenario, where the cloud does not know the plant but knows which sensors and actuators are used. We can no longer pick an arbitrary isomorphism since it could lead to inputs and outputs inconsistent with existing sensors and actuators. This inconsistency would signal the cloud that the plant is being dishonest about its measurements and undermine the trust of their communication. Therefore, we need to restrict the group of isomorphisms used for encoding to those that keep the inputs and outputs unchanged. 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.6. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, if we assume that the cloud knows the sensors and actuators used by the plant, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_{\mathcal{H}}$ of dimension dim $\mathcal{H}_{n,m,p}(\Sigma)$, where*

$$n(n+1) \leq \textit{dim } \mathcal{H}_{n,m,p}(\Sigma) \leq n(n+1)+m(n+1)-\sum_{i=2}^{k_1} r_{i-1}r_i, \tag{IV.3}$$

if Algorithm 1 is used.

Proof. From Theorem III.6, we know that Algorithm 1 makes isomorphic systems indistinguishable by the cloud. However, since the cloud knows the sensors and actuators, 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 $\mathcal{H}_{n,m,p}(\Sigma)$ is 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 and we know bounds on $\mathcal{K}_{n,m,p}(\Sigma)$ (see the discussion after Proposition IV.1), the bounds on dimension of $\mathcal{H}_{n,m,p}(\Sigma)$ are given by (IV.3). The result follows by applying Proposition IV.5 for $\mathcal{H}_{n,m,p}(\Sigma)$. □

In the third scenario, where the cloud possesses the complete knowledge of dynamics, we are free to use only the isomorphisms from the symmetry subgroup $\psi \in \mathcal{K}_{n,m,p}(\Sigma)$.

Corollary IV.7. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Assuming that the cloud has the complete knowledge of dynamics, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_{\mathcal{K}}$ (i.e., the equivalence class of Ω defined by the action of $\mathcal{K}_{n,m,p}(\Sigma)$) of dimension dim $\mathcal{K}_{n,m,p}(\Sigma) \leq m(n+1) - \sum_{i=2}^{k_1} r_{i-1}r_i$.*

Proof. The proof of this statement is similar to that of Corollary IV.6. The dimension of $\mathcal{K}_{n,m,p}(\Sigma)$ is given by Proposition IV.1. □

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

Example IV.8. Consider a drone with linearized dynamics given in [22] and a bounded operational envelope (i.e., constraints on the extreme values of its state). From linear model in [22], we observe that $n = 12$, $m = 4$, and $p = 4$ and r-numbers are the following: $r_1 = 4$, $r_2 = 4$, $r_3 = 2$, $r_4 = 2$. 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.5, we estimate the dimension of the uncertainty set to be 240.

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.6 and estimate the dimension of the uncertainty set to be between 156 and 180.

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.7, we estimate the dimension of the uncertainty set to be less or equal than 24. 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 finding $\mathcal{K}_{n,m,p}(\Sigma)$ for a given Σ.

In future work, we plan to give a better quantification of the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$, which will directly affect statements of Corollaries IV.6 and IV.7 by changing the estimates for the dimension of the equivalence classes.

V. Side knowledge

The guarantees provided in Section IV no longer hold if the adversary has partial knowledge about the encoding isomorphism. The cloud can obtain this through some external channels or through some prior knowledge about the system.

Recall that by Lemma IV.4, $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1) + m(n+1) + m^2 + p(p+1)$. Suppose the cloud has partial knowledge about the encoding isomorphism. We represent the partial knowledge available to the cloud by 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 III.6, however the privacy guaranteed by the scheme changes. It is obvious that the size of the uncertainty set constructed by isomorphisms that satisfy (V.1) is no greater and, in general, smaller than the one constructed with no side knowledge. Moreover, the uncertainty set is no longer an equivalence class because the preimage of ρ 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 dimension $n(n+1) + m(n+1) + m^2 + p(p+1) - k$.*

Let us now consider the map f_Ω defined earlier in (IV.2). It was shown in Lemma IV.2 that f_Ω is injective. The image of $f_\Omega(\rho^{-1}(l))$ constitutes the uncertainty set, between the elements of which the cloud cannot distinguish. Therefore, the goal of this section is to find the dimension of $f_\Omega(\rho^{-1}(l))$.

Proposition V.2. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Suppose that Algorithm 1 is used and the cloud has the side knowledge about the selected isomorphism $\psi \in \mathcal{G}_{n,m,p}$:*

$$\rho(\psi) = l \in \mathbb{R}^k, \tag{V.2}$$

where $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ is a surjective map of constant rank k. Then, assuming the cloud has no knowledge about the plant and given f_Ω is the orbit map for action of $\mathcal{G}_{n,m,p}$, the cloud cannot distinguish between Ω and any other system in the uncertainty set $\mathcal{U} = f_\Omega(\rho^{-1}(l))$ of dimension $n(n+1) + m(n+1) + m^2 + p(p+1) - k$.

The proofs for Lemma V.1 and Proposition V.2 are omitted to conserve space, but provided in [20].

Proposition V.2 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 using similar proofs.

To illustrate, consider again Example IV.8 and assume the cloud has no knowledge about the plant. If we suppose that the adversary possesses side knowledge about 40 elements of the chosen isomorphism, then the dimension of the uncertainty set is bound to decrease from 240 to 200.

VI. Conclusion

In this paper, we have extended the results of the transformation-based privacy algorithm we introduced in [16]. The proposed algorithm has benefits over existing solutions due to its computational efficiency at the client, conceptual simplicity and connection to the properties of dynamical systems. We have, for the first time, provided a criterion for measuring the amount of privacy provided by the proposed algorithm. Moreover, we have considered the implications of the adversary having side channels, other than its direct communication with the client, from which it is able to learn information about the system. In the future, we want to give a better quantification of the privacy in scenarios, where the cloud has some knowledge about plant dynamics, by studying the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$. Furthermore, we wish to determine how this algorithm performs in practice - in particular, we would like to see how privacy is affected if we only have a finite number of keys (i.e., $\mathcal{G}_{n,m,p}$ is no longer infinite) because this accurately models what happens in real computer-based systems.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. Schulze Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo, “Towards encrypted mpc for linear constrained systems,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 195–200, April 2018.

[11] 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.

[12] 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.

[13] P. Weeraddana and C. Fischione, “On the privacy of optimization,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9502 – 9508, 2017, 20th IFAC World Congress.

[14] 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*, ser. CPS-SPC ’15, 2015, pp. 31–42.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] A. Sultangazin and P. Tabuada, “Symmetries and isomorphisms for privacy in control over the cloud,” *arXiv e-prints*, p. arXiv:1906.07460, Jun 2019.

[21] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. Springer-Verlag New York, 2003.

[22] F. Sabatino, “Quadrotor control: modeling, nonlinear control design, and simulation,” Master’s thesis, KTH Electrical Engineering, 6 2015.