Protecting the privacy of networked multi-agent systems controlled over the cloud

Alimzhan Sultangazin[1], Suhas Diggavi[1], and Paulo Tabuada[1]

***Abstract*— The vision of an Internet-of-Things calls for combining the increasing connectivity of devices at the edge with the ability to compute either at the edge or on more powerful servers in the network. There is great interest in exploring the feasibility of these ideas when devices such as quadcopters or ground robots at the edge are controlled over the cloud, i.e., by leveraging computational power available elsewhere in the network. One of the main difficulties, especially in the context of the *Internet-of-Battlefield-Things* is the need to keep the data private. In this paper we propose a solution to this problem by extending previous results by the authors from a single system controlled over the cloud to networks of systems that are controlled and coordinated over the cloud. We propose a non-cryptographic lightweight encoding scheme that ensures the privacy of the data exchanged by all the participating parties.**

I. Introduction

The vision of an Internet-of-Things requires devices at the edge to communicate seamlessly and offload more intensive computing jobs to servers in the network. This vision has recently inspired the concept of *Internet-of-Battlefield-Things* (IoBT) [1] [2]. In addition to the challenges present in the Internet-of-Things, IoBT requires re-thinking its architecture to provide resiliency in the presence of malicious agents and dynamically changing contested environments.

Among the manifold computational problems to be solved in IoBT stands the control of agents at the edge. In many situations human soldiers are aided by quadcopters, ground robots, or other types of agents that need to be controlled and coordinated. Although the dynamics of these agents is decoupled, their mission requires meeting objectives that couple their behavior resulting in complex control and optimization problems that need to be solved in real-time. As the number of agents increases, so does the computational load and it is, therefore, natural to offload this computation to the cloud. However, as IoBTs need to operate in contested environments, protecting the privacy of all the exchanged data is paramount. This need becomes even more pressing since some of the systems being controlled may be passively or actively under the control of an adversary.

*The work of the authors was partially supported by the NSF awards 1740047, 1705135, by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196, and by the UC-NL grant LFR-18-548554. 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] Department of Electrical and Computer Engineering, University of California - Los Angeles, USA {asultangazin, suhasdiggavi, tabuada}@ucla.edu

The current literature addressing the issue of privacy in control over the cloud mainly focuses on two approaches: data encryption and data perturbation.

The data encryption approach comprises methods, such as full or partial homomorphic encryption [3], [4], [5], data obfuscation [6], and multi-party computation [7]. Typically, the main drawback when using data encryption methods on real systems is their large computational overhead, susceptibility of some methods to attacks during the key distribution phase, and the need to disclose partial or complete information about the system because the cloud typically performs computations on the original model. As shown in [8], [9], this information would render systems susceptible to adversaries, who gain unauthorized access. Given that control performance is negatively impacted by delays and jitter, it is not feasible to rely on encryption techniques for real-time control.

Data perturbation, a method which has its origins in the theory of database security, has as of late found its application in control theory (see [10], [11]). This method requires that the cloud receives perturbed aggregate data of a collection of systems, while preventing any knowledge about an individual system to be inferred from this data. Rather than using a usual binary metric of privacy (i.e., private or not private), data perturbation allows privacy to be measured continuously, ranging from complete privacy to its total absence. Moreover, data perturbation methods, to be able to provide privacy guarantees, typically need to perturb the data by adding enough noise, which in turn greatly reduces control performance. Moreover, since we are dealing with time-series and not a single database query, the added noise may accumulate over time and further deteriorate control performance.

In this paper we address the challenge of protecting privacy when controlling and coordinating a collection of agents over the cloud. Rather than leveraging cryptographic or information-theoretic techniques, we extend previous results by the authors on lightweight encoding techniques based on the symmetries of control systems [12]. While our previous work focused on controlling a single system over the cloud, in this paper we address the control of multiple agents. This requires new protocols that guarantee that no participating agent nor the cloud is able to learn private information exchanged by the participating agents.

II. Problem Definition

In this paper, the problem of protecting privacy of a group of agents, which are controlled over the cloud, is addressed. We start by briefly recalling the results in [12] addressing the case of a single agent controlled over the cloud. We then introduce the problem of ensuring privacy when two agents are controlled over the cloud and show how to extend the results from a single agent to multiple agents by carefully routing information among the agents before being sent to the cloud and, conversely, by carefully disseminating information coming from the cloud through the participating agents.

A. Single agent controlled over the cloud

Consider a discrete-time linear system, denoted by Σ, and described by:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k], \end{aligned} \tag{II.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ describe the dynamics of the system, and $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ denote the state, input and output of the system, respectively. As a shorthand, we may refer to (II.1) as $\Sigma = (A, B, C)$ and call it the *plant*.

We define a triple $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$ to be a trajectory of Σ if it satisfies (II.1) for all $k \in \mathbb{N}$.

Additionally, each plant has a certain cost function $J : \mathbb{R}^n \times (\mathbb{R}^m)^{N+1} \to \mathbb{R}$ that defines the control objective. To be consistent with the linear framework, we consider quadratic cost functions described by:

$$J(x, u) = \sum_{k=0}^{N} \Delta\eta^T[k] M \Delta\eta[k], \tag{II.2}$$

where $\Delta\eta[k] = \begin{bmatrix} x[k] - x^* \\ u[k] - u^* \end{bmatrix}$. The state x^* and input u^* denote the desired objective induced by the cost function. In other words, the objective is to force u to converge to u^* and x to converge to x^* with the "distance" between the state and input from their desired values x^* and u^* being measured by J. In addition, the mission may require certain constraints to be satisfied. These constraints are defined by:

$$D\eta[k] \leq 0, \tag{II.3}$$

where $\eta[k] = \begin{bmatrix} x[k] & u[k] \end{bmatrix}^T$, $D \in \mathbb{R}^{\ell\times(n+m)}$, and ℓ is the number of constraints. The inequality in (II.3) is interpreted element-wise, i.e., $D\eta[k] \leq 0$ represents the conjunction of ℓ inequalities, one per row of the matrix D.

The communication between the plant and the cloud is to be performed in two steps: handshake and plant operation. During handshake, the plant transmits suitably modified versions of the plant model, cost and constraints. In exchange, the cloud agrees to faithfully compute the input minimizing the cost function, subject to constraints. During plant operation, the plant sends a suitably modified version of its measurements to the cloud. The cloud computes a new input based on the received measurements and minimization of the cost and sends it to the plant, where it is suitably modified before being applied to the plant.

The modifications applied to the plant model, cost and constraints depend on the knowledge available to the cloud and privacy guarantees that we aim to provide. We now provide a brief description of three possible scenarios. For more details, the reader is referred to [12].

In the first scenario, we assume that the cloud has no knowledge about the plant. The first objective is to find a way to modify the plant dynamics, the cost, the constraints, and the measurements (this is all the data that is sent to the cloud) to prevent the cloud from inferring the plant (A, B, C), the cost J, the constraint matrix D, and the plant trajectory $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$. The second objective is to construct an input from the data provided by the plant so that controlling the plant with such input results in a trajectory minimizing the cost J.

In the second scenario, we assume that the cloud has no knowledge about the plant except for knowing its sensors and actuators. This occurs, e.g., if the cloud knows that it is controlling ground vehicles. Although it may not know the specific model for the ground vehicles, it knows that position and velocity will be measured. The first objective is to find a way to modify the plant dynamics, the cost, the constraints, and the measurements (this is all the data that is sent to the cloud) to prevent the cloud from inferring the plant (A, B, C), the cost J, the constraint matrix D, and the plant trajectory $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$. The second objective is to construct an input from the data provided by the plant so that controlling the plant with such input results in a trajectory minimizing the cost J. Given the cloud's knowledge of existing sensors and actuators, the class of transformations used for encoding the exchanged data will be smaller as detailed in [12].

In the third scenario, we assume that the cloud has complete knowledge about plant dynamics, including its sensors and actuators. Hence, we can no longer modify the plant model but we can still modify the cost, the constraints, and the measurements to prevent the cloud from inferring the cost J, the constraint matrix D, and the plant trajectory $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$. The second objective is to construct an input from the data provided by the plant so that controlling the plant with such input results in a trajectory minimizing the cost J. In this scenario, the class of employed transformations is even smaller given that the cloud has even more knowledge as explained in [12].

B. Two agents controlled over the cloud

Expanding on the aforementioned problem formulation, we now discuss the problem of ensuring privacy when two agents are controlled over the cloud. Consider two discrete-time linear systems $\Sigma_1 = (A_1, B_1, C_1)$ and $\Sigma_2 = (A_2, B_2, C_2)$ of the form (II.1). Although the dynamics of each agent is decoupled, the mission objectives require coordinated motion. This is captured by a single quadratic

cost function:

$$J(x_1, x_2, u_1, u_2) = \sum_{k=0}^{N} \Delta\eta^T[k] M \Delta\eta[k], \tag{II.4}$$

where $\Delta\eta[k] = \begin{bmatrix}\Delta x_1[k] & \Delta x_2[k] & \Delta u_1[k] & \Delta u_2[k]\end{bmatrix}^T$, $\Delta x_i[k] = x_i[k] - x_i^*$, $\Delta u_i[k] = u_i[k] - u_i^*$ for all $k \in \{1, 2\}$. The variables x_1, u_1, x_2 and u_2 correspond to the state and input of Σ_1 and the state and input of Σ_2, respectively. The states x_1^*, x_2^* and inputs u_1^*, u_2^* denote the desired objective induced by the cost function.

In addition, coordination may also be imposed by constraints that couple both plants. Let D denote a joint constraint matrix for both Σ_1 and Σ_2, respectively. Then, the constraints are given by:

$$D\eta[k] \leq 0, \tag{II.5}$$

where $\eta[k] = \begin{bmatrix}x_1[k] & x_2[k] & u_1[k] & u_2[k]\end{bmatrix}^T$. Note that while two systems are decoupled in terms of dynamics, they are coupled in terms of the cost function and constraints.

To illustrate the problem formulation we consider a simple example where two ground robots are to be controlled to move on a convoy.

Example II.1. We define the control task to be the regulation of the convoy speed to the desired value v^* as well as the regulation of the distance between the two ground robots to the desired value d^*. For simplicity, we model the ground robots as point masses moving along a line. The model is then given by:

$$\dot{l} = v \tag{II.6}$$

$$\dot{v} = \frac{1}{m} F, \tag{II.7}$$

where l is the position of the ground robot, v is its velocity, and F is the force applied to move the robot that we treat as the control input. In matrix notation, this model becomes:

$$\begin{bmatrix}\dot{l} \\ \dot{v}\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix} \begin{bmatrix}l \\ v\end{bmatrix} + \begin{bmatrix}0 \\ \frac{1}{m}\end{bmatrix} F. \tag{II.8}$$

Let us denote the state of each robot by $x_i = \begin{bmatrix}l_i & v_i\end{bmatrix}^T$ and the input to each robot as $u_i = F_i$ for $i \in \{1, 2\}$. The dynamics of each robot would be given by matrices A_i and B_i defined as:

$$A_i = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix} \tag{II.9}$$

$$B_i = \begin{bmatrix}0 \\ \frac{1}{m_i}\end{bmatrix}, \tag{II.10}$$

for $i \in \{1, 2\}$.

The requirement that the robots move in a convoy is expressed by the cost function:

$$J(x_1, x_2) = \sum_{k=0}^{N} \|l_1[k] - l_2[k] - d^*\|^2 + \|v_1[k] - v^*\|^2 + \|v_2[k] - v^*\|^2.$$

This cost function is a special case of the general quadratic form in (II.4).

Suppose that we also have some constraints on the velocity and force that can be applied to both robots:

$$|u_i| \leq u_{max} \tag{II.11}$$

$$|v_i| \leq v_{max}. \tag{II.12}$$

These constraints can be easily represented in terms of constraint matrix D in (II.5).

Each robot can control the applied force F_i by commanding the motors that force the wheels to rotate. When they are controlled over the cloud, this force is computed by the cloud based on the minimization of the cost function J evaluated on position and velocity information measured by each robot and provided to the cloud. If the robots are transporting important cargo and an adversary intercepts position and velocity information sent to the cloud, then it could disrupt the mission. Keeping this information private is thus an extremely important problem and one possible solution is provided in this paper.

C. Attack model and privacy objectives

The cloud is treated as an honest but curious adversary. That is, the cloud will follow the protocol all parties agree upon, but it may be interested in extracting and leaking private information. Similarly, each agent is also regarded as an honest but curious adversary since it is possible that one or more of the agents have been compromised by an adversary.

The communication between plant Σ_1, plant Σ_2, and the cloud is performed in two phases: handshaking and plant operation. During the first phase, plant Σ_1 performs a handshake with plant Σ_2, wherein the former sends the latter a suitably modified version of the plant model, the joint cost, and constraints. We assume that only plant Σ_1 knows the joint cost and joint constraints. Next, plant Σ_2 performs a handshake with the cloud, wherein the former sends the latter a suitably modified version of a joint plant model of both its dynamics and the modified dynamics of Σ_1, a further modified joint cost function, and suitably modified constraints. In exchange, the cloud agrees to calculate the input minimizing the provided cost, subject to constraints. During the second phase, plant Σ_1 sends a suitably modified version of its measurements to plant Σ_2. Then, plant Σ_2 appends its own measurements to the measurements received from Σ_1 and sends a suitably modified version of these measurements to the cloud. The cloud computes a new input from the received measurements by minimizing the cost it received, and sends it to the plant Σ_2. The latter suitably modifies the received input, applies a portion of the received input to its system and sends the remaining portion to Σ_1, where it is suitably modified before being applied to the plant.

We intentionally used the expression "suitably modified" because the nature of the modification will entirely depend on the knowledge of the cloud about both plants, knowledge of

each plant about the other plants and the privacy guarantees that we intend to provide.

III. Summary of previous results for a single agent

In this section, we summarize the theoretical results from [12] since they provide the tools upon which we will build a solution to the multiple agent problem. The key notion that was discussed in [12] is the notion of isomorphism of control systems.

Definition III.1. Let $\Sigma = (A, B, C)$ and $\hat{\Sigma} = (\hat{A}, \hat{B}, \hat{C})$ be linear control systems. The quadruple $\psi = (P, F, G, S)$ is an isomorphism from Σ to $\hat{\Sigma}$, denoted by $\psi_*\Sigma = \hat{\Sigma}$, if $P : \mathbb{R}^n \to \mathbb{R}^n$, $G : \mathbb{R}^m \to \mathbb{R}^m$, and $S : \mathbb{R}^p \to \mathbb{R}^p$ are invertible linear maps, $F : \mathbb{R}^n \to \mathbb{R}^m$ is a linear map and:

$$\begin{aligned}\hat{\Sigma} = \psi_*\Sigma &= (P(A - BG^{-1}F)P^{-1}, PBG^{-1}, SCP^{-1}) \\ &\equiv (\hat{A}, \hat{B}, \hat{C}). \end{aligned} \tag{III.1}$$

The isomorphism modifies the state x, input u and output y of system Σ to state z, input v and output y of system $\hat{\Sigma}$ as follows:

$$z[k] = Px[k] \tag{III.2}$$

$$v[k] = Fx[k] + Gu[k] \tag{III.3}$$

$$w[k] = Sy[k]. \tag{III.4}$$

Isomorphisms also have the effect of transforming the cost and constraints as follows:

$$\Delta\hat{\eta}_k = \begin{bmatrix}\Delta z_k \\ \Delta v_k\end{bmatrix} = \begin{bmatrix}P & 0 \\ F & G\end{bmatrix}\begin{bmatrix}\Delta x_k \\ \Delta u_k\end{bmatrix} \equiv L\Delta\eta_k \tag{III.5}$$

$$\hat{J} = \psi_* J(x, u) = \sum_{k=0}^{N} \Delta\hat{\eta}_k^T \hat{M} \Delta\hat{\eta}_k \tag{III.6}$$

$$\hat{D}\hat{\eta}_k \leq 0, \tag{III.7}$$

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

It can be observed that the set of isomorphisms of a given system Σ, with function composition as a group operation, forms a group. Hence, we can define an equivalence relation between the quadruples $\{\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}}\}$.

Definition III.2. Let $\mathcal{G}$ be a subgroup of the group of all isomorphisms of Σ. Two quadruples $(\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}})$ and $(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}})$ are called $\sim_{\mathcal{G}}$-equivalent if there exists an isomorphism $\psi \in \mathcal{G}$ such that $\psi_*\Sigma_1 = \Sigma_2$, $\hat{J} = \psi_* J$, $\hat{D} = \psi_* D$ and (III.2)-(III.4) hold for every $k \in \mathbb{N}$.

The notion of isomorphism is used in [12] to establish the following technical results that will serve as a foundation for the contributions in this paper. The reader is referred to [12] for the proofs of these results.

Lemma III.3. *If $\hat{\Sigma} = \psi_*\Sigma$ and $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$ is a trajectory of Σ, then $\{Px[k], Fx[k] + Gu[k], Sy[k]\}_{k\in\mathbb{N}}$ is a trajectory of $\hat{\Sigma}$.*

This lemma states that the trajectory given by $\{Px[k], Fx[k] + Gu[k], Sy[k]\}_{k\in\mathbb{N}}$ is a valid trajectory of the modified system $\hat{\Sigma}$. Hence, if an agent with dynamics Σ provides the cloud with $\hat{\Sigma}$, then it can transform its own trajectory $\{x[k], u[k], y[k]\}_{k\in\mathbb{N}}$ to $\{Px[k], Fx[k]+Gu[k], Sy[k]\}_{k\in\mathbb{N}}$ and send this transformed trajectory to the cloud. Since this is a valid trajectory for $\hat{\Sigma}$, the cloud has no way of knowing that it is working with a transformed plant and transformed trajectory rather than with the original ones. However, this raises the question: how do we construct the input minimizing the real cost for the real plant from the input computed by the plant based on transformed plant, transformed cost, and transformed trajectory? The answer is provided by the next result.

Lemma III.4. *Suppose the cloud solves the optimization problem:*

$$\begin{aligned} \min_{v} \quad & \hat{J}(Px, v) \\ \text{subject to} \quad & \hat{D}\hat{\eta}_k \leq 0, \end{aligned}$$

for the plant $\hat{\Sigma} = \psi_*\Sigma$ and this optimization problem has the unique solution v^o. Then, the unique solution of the optimization problem:

$$\begin{aligned} \min_{u} \quad & J(x, u) \\ \text{subject to} \quad & D\eta_k \leq 0, \end{aligned}$$

for the plant Σ is given by $u^o = G^{-1}(v^o - Fx)$.

The main privacy guarantee provided by the proposed scheme is described in the next result.

Theorem III.5. *Any two quadruples:*

$$\begin{aligned}&(\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}}) \\ &(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}}),\end{aligned}$$

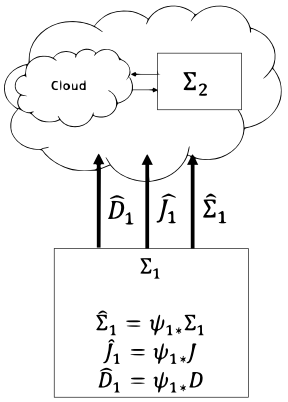
of plants, costs, constraints and trajectories, which are $\sim_{\mathcal{G}}$-equivalent, are indistinguishable by the cloud.

This theorem shows that there exists a group of plants that will be indistinguishable by the cloud. That is, the cloud is not able to pinpoint the original plant, trajectory, and cost among all the different modified versions in the corresponding equivalence class.

The following algorithm leverages these results to ensure privacy for control over the cloud of a single agent.

Algorithm III.6. *(Plant $\Longleftrightarrow$ Cloud)*

1) *Phase 1: Handshaking*
   The plant encodes its dynamics, cost function and constraint matrix into $\hat{\Sigma} = \psi_*\Sigma$, $\hat{J}(z, v) = \psi_* J(x, u)$ and $\hat{D} = \psi_* D$ and sends them to the cloud.
2) *Phase 2: Plant operation*
   Encoding: The plant periodically measures $y[k]$, encodes it into $w[k] = Sy[k]$ and sends it to the cloud.
   Optimization: The cloud uses the received encoded measurement $w[k]$, estimates the plant state $z[k]$, computes the input $v[k]$ minimizing $\hat{J}$ subject to the constraint



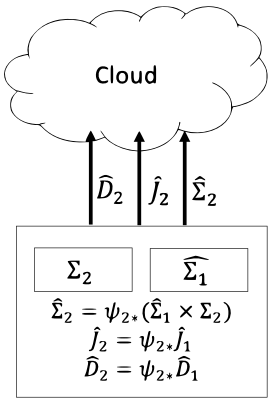
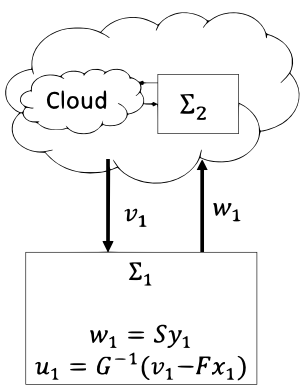


Fig. 1. Handshakes between Σ_1 and Σ_2 (left) and between Σ_2 and the cloud (right)



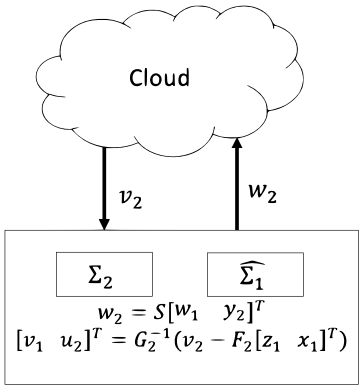


Fig. 2. Plant operation between Σ_1 and Σ_2 (left) and between Σ_2 and the cloud (right)

$\hat{D}\eta_k \leq 0$ and the dynamics $\hat{\Sigma}$, and sends $v[k]$ to the plant.
Decoding: The plant decodes $v[k]$ to produce $u[k]$, using (III.3)*, and sends $u[k]$ to the actuators.*

This algorithm can be used for any of the scenarios described in Section II-A. However, depending on the scenario we choose a different group of isomorphisms. The more knowledge the cloud has about the plant, the smaller is the group of isomorphisms and thus less information can be kept private. Readers interested in a detailed description of these isomorphism groups can consult [12] for further details.

IV. Solving the control-over-the-cloud privacy problem for multiple agents

The results from Section III are now extended to the case where multiple agents are controlled over the cloud. To simplify the presentation, we consider first the case of 2 agents and then describe the modifications needed to handle finitely many agents. In addition, we restrict attention to the scenario where the cloud has no knowledge about the agents being controlled. The two other scenarios described in Section II-A can be similarly treated by suitably modifying the symmetry groups as described in [12].

We describe the communication protocol by breaking it into two parts. The first one is concerned with the communication between plant Σ_1 and plant Σ_2. It is useful to think about this step as the communication between plant Σ_1 and a "super-cloud" consisting of the aggregation of the cloud and plant Σ_2 (see Figure 1). Hence, plant $\Sigma_1 = (A_1, B_1, C_1)$ encodes its dynamics, the shared cost function J, and the constraint matrix D using isomorphism $\psi_1 = \{P_1, F_1, G_1, S_1\}$ resulting in the encoded plant $\hat{\Sigma_1} = (\hat{A}_1, \hat{B}_1, \hat{C}_1)$, encoded cost function $\hat{J}_1$, and encoded constraint matrix $\hat{D}_1$ that are then sent to the "super-cloud".

While the way in which the dynamics are encoded remains identical to the way it is done in the single plant case, it is important to consider how the cost function given by (II.4) is being transformed. The isomorphism ψ_1 affects only the state and input of Σ_1 and, therefore, a transformation matrix L needs to be constructed appropriately. Consider how ψ_1 affects $\eta[k]$:

$$\hat{\eta}_1[k] = \begin{bmatrix} Px_1[k] \\ x_2[k] \\ Fx_1[k] + Gu_1[k] \\ u_2[k] \end{bmatrix} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ F_1 & 0 & G_1 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \\ u_1[k] \\ u_2[k] \end{bmatrix} \equiv L_1\eta[k]. \tag{IV.1}$$

Therefore, it can be shown that:

$$\hat{J}_1(z_1, x_2, v_1, u_2) = \psi_{1*}J(x_1, x_2, u_1, u_2) = \sum_{k=0}^{N} \Delta\hat{\eta}_1^T[k]\hat{M}_1\Delta\hat{\eta}_1^T[k] \tag{IV.2}$$

$$\hat{D}_1 = \psi_{1*}D = DL_1^{-1}, \tag{IV.3}$$

where $\hat{M}_1 = L_1^{-T}ML_1^{-1}$. The isomorphism takes the state x_1, input u_1 and output y_1 of system Σ_1 to the state z_1, input v_1 and output w_1 of system Σ_1 as follows:

$$z_1[k] = P_1x_1[k] \tag{IV.4}$$

$$v_1[k] = F_1x_1[k] + G_1u_1[k] \tag{IV.5}$$

$$w_1[k] = S_1y_1[k]. \tag{IV.6}$$

These expressions are used to encode the output of plant Σ_1 before sending it to plant Σ_2 and to decode the input received from plant Σ_2 (see Figure 2).

As shown in Theorem III.5, the "super-cloud" cannot access the plant, the cost and constraints of agent Σ_1. As the "super-cloud" consists of the cloud and agent Σ_2 we conclude that neither the cloud neither Σ_2 can access the plant, the cost and constraints of agent Σ_1.

In the second part of the communication protocol, we consider Σ_2 and $\hat{\Sigma}_1$ to be a single plant, which wants to communicate with the cloud. We define the Cartesian product $\times$ between two plants $\Sigma_1 = (A_1, B_1, C_1)$ and $\Sigma_2 = (A_2, B_2, C_2)$ to be:

$$\Sigma_c = \Sigma_1 \times \Sigma_2 = \left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \right). \tag{IV.7}$$

Now, we can define a composite plant appearing to communicate with the cloud to be:

$$\Sigma_c = \hat{\Sigma}_1 \times \Sigma_2 \equiv (A_c, B_c, C_c). \tag{IV.8}$$

Plant Σ_c, the shared cost function $\hat{J}_1$, and the constraint matrix $\hat{D}_1$ are encoded using isomorphism $\psi_2 = \{P_2, F_2, G_2, S_2\}$ and, as a result, $\hat{\Sigma}_2$, the modified cost function $\hat{J}$, and the modified constraint matrix $\hat{D}_2$ are sent to the cloud.

The encoding expression for the dynamics is given by:

$$\begin{aligned}\hat{\Sigma}_2 &= \psi_{2*}\Sigma_c \\ &= (P_2(A_c - B_c G_2^{-1} F_2)P_2^{-1}, P_2 B_c G_2^{-1}, S_2 C_c P_2^{-1}).\end{aligned} \tag{IV.9}$$

To derive the expression for the cost function and constraints, consider how ψ_2 affects $\hat{\eta}_1[k]$:

$$\begin{aligned}\hat{\eta}_2 &= \begin{bmatrix} P_2 x_c \\ F_2 x_c + G_2 u_c \end{bmatrix} = \begin{bmatrix} P_2 & 0 \\ F_2 & G_2 \end{bmatrix} \begin{bmatrix} z_1[k] \\ x_2[k] \\ v_1[k] \\ u_2[k] \end{bmatrix} \\ &\equiv L_2 \hat{\eta}_2.\end{aligned} \tag{IV.10}$$

Therefore, it can be shown that:

$$\begin{aligned}\hat{J}_2(z_2, v_2) &= \psi_{2*} J(z_1, x_2, v_1, u_2) \\ &= \sum_{k=0}^{N} \Delta\hat{\eta}_2^T[k] \hat{M}_2 \Delta\hat{\eta}_2^T[k] \qquad \text{(IV.11)} \\ \hat{D}_2 &= \psi_{2*}\hat{D}_1 = \hat{D}_1 L_2^{-1}, \qquad \text{(IV.12)}\end{aligned}$$

where $\hat{M}_2 = L_2^{-T} \hat{M}_1 L_2^{-1}$.

The isomorphism takes the state $x_c = \begin{bmatrix} z_1 & x_2 \end{bmatrix}^T$, input $u_c = \begin{bmatrix} v_1 & u_2 \end{bmatrix}^2$ and output $y_c = \begin{bmatrix} w_1 & y_2 \end{bmatrix}^T$ of system Σ_c to the state z_2, input v_2 and output w_2 of system Σ_1 as follows:

$$\begin{aligned} z_2[k] &= P_2 x_c[k] \qquad \text{(IV.13)} \\ v_2[k] &= F_2 x_c[k] + G_2 u_c[k] \qquad \text{(IV.14)} \\ w_2[k] &= S_2 y_c[k]. \qquad \text{(IV.15)}\end{aligned}$$

These expressions are used to encode the output of plant Σ_c before sending it to the cloud and to decode the input received from the cloud (see Figure 2).

Again, it follows from Theorem III.5 that the cloud would not be able to learn the plant, the cost and the constraints of Σ_c and, therefore, those of Σ_1 and Σ_2. In view of this, we can construct the following algorithm that preserves privacy of Σ_1 and Σ_2.

Algorithm IV.1. *(Plant Σ_1 $\Longleftrightarrow$ Plant Σ_2 $\Longleftrightarrow$ Cloud)*

1) *Phase 1: Handshaking*
   a) *Plant Σ_1 encodes its dynamics, cost function, and constraint matrix into $\hat{\Sigma}_1 = \psi_{1*}\Sigma_1$, $\hat{J}_1(z_1, x_2, v_1, u_2) = \psi_{1*}J(x_1, x_2, u_1, u_2)$, and $\hat{D}_1 = \psi_{1*}D$ and sends them to plant Σ_2.*
   b) *Plant Σ_2 encodes the dynamics of Σ_c, cost function $\hat{J}_1$, and constraint matrix $\hat{D}_1$ into $\hat{\Sigma}_2 = \psi_{2*}\Sigma_c$, $\hat{J}_2(z_2, v_2, u_2) = \psi_{2*}\hat{J}_1(z_1, x_2, v_1, u_2)$, and $\hat{D}_2 = \psi_{2*}\hat{D}_1$ and sends them to the cloud.*
2) *Phase 2: Execution*
   Encoding:
   a) *Plant Σ_1 periodically measures y_1, encodes it into $w_1 = S_1 y_1$ and sends it to plant Σ_2.*
   b) *Plant Σ_2 periodically measures y_2, appends it to w_1 to form $y_c = [w_1, y_2]^T$, encodes it into $w_2 = S_2 y_c$ and sends it to the cloud.*

   Optimization:
   The cloud uses the received encoded measurement w_2, estimates the plant state z_2, computes the input v_2 minimizing $\hat{J}_2$ subject to the constraint $\hat{D}_2\hat{\eta}_2 \leq 0$ and the dynamics $\hat{\Sigma}_2$, and sends v_2 to plant Σ_2.
   Decoding:
   a) *Plant Σ_2 decodes v_2 to produce u_c, using* (IV.14)*, applies u_2 to the actuators and sends v_1 to plant Σ_1.*
   b) *Plant Σ_1 decodes v_1 to produce u_1, using* (IV.5)*, and applies u_1 to the actuators.*

The main results of this section are summarized in the following formal statement that can be seen as a corollary of Lemmas III.3 and III.4 and Theorem III.5.

Corollary IV.2. *Using the protocol described in Algorithm V.1 and any isomorphisms $\psi_1 \in \mathcal{G}_1$ and $\psi_2 \in \mathcal{G}_2$, where $\mathcal{G}_1$ and $\mathcal{G}_2$ are the groups of all isomorphisms of control systems Σ_1 and $\hat{\Sigma}_1 \times \Sigma_2$, respectively, the following holds:*

1) *the trajectory of the plants Σ_1 and Σ_2 in closed loop with the cloud optimizes the cost J subject to the constraints $D\eta_k \leq 0$;*
2) *the cloud is not able to distinguish between $(\Sigma_c, \hat{J}_1, \hat{D}_1, \{x_c[k], u_c[k], y_c[k]\}_{k\in\mathbb{N}})$ and any other quadruple $(\hat{\Sigma}_2, \hat{J}_2, \hat{D}_2, \{z_2[k], v_2[k], w_2[k]\}_{k\in\mathbb{N}})$ in the same equivalence class of the $\sim_{\mathcal{G}_2}$-equivalence relation.*
3) *the plant Σ_2 and is not able to distinguish between $(\Sigma_1, J, D, \{x_1[k], u_1[k], y_1[k]\}_{k\in\mathbb{N}})$ and any other quadruple $(\hat{\Sigma}_1, \hat{J}_1, \hat{D}_1, \{z_1[k], v_1[k], w_1[k]\}_{k\in\mathbb{N}})$ in the same equivalence class of the $\sim_{\mathcal{G}_1}$-equivalence relation.*
4) *the plant Σ_1 gains no knowledge about plant Σ_2 other than what it can infer from the joint cost J and constraints D.*

V. Privacy encoding and communication topology

In the previous section, we solved the privacy problem when controlling two agents over the cloud. This solution can be generalized to multiple agents. Consider the simplest solution, wherein we form a line graph connecting all the agents back to back. Algorithm V.1 applied recursively would produce the desired result. In other words, we could have agent 1 forwarding its encoded information to agent 2 who appends the received information to its own, encodes it, and forwards it to agent 3 and repeat this process until the last

agent who will communicate with the cloud. This can be succinctly presented in the form of an algorithm.

Algorithm V.1. *Agent $i-1$ (or Σ_{i-1})$\Longleftrightarrow$ Agent i (or Σ_i) $\Longleftrightarrow$ Agent $i+1$*

1) *Phase 1: Handshaking*
   a) *Plant Σ_{i-1} encodes its dynamics, cost function, and constraint matrix into $\hat{\Sigma}_{i-1} = \psi_{(i-1)*}\Sigma_{i-1}$, $\hat{J}_{i-1}(z_{i-1}, x_i, v_{i-1}, u_i) = \psi_{1*}J(x_{i-1}, x_i, u_{i-1}, u_i)$, and $\hat{D}_{i-1} = \psi_{(i-1)*}D$ and sends them to agent i (i.e. Σ_i). Note that $\psi_i = (P_i, F_i, G_i, S_i)$ is an isomorphism of Σ_i.*
   b) *Plant Σ_i encodes the dynamics of $\hat{\Sigma}_{i-1} \times \Sigma_i$, cost function $\hat{J}_i$, and constraint matrix $\hat{D}_i$ into $\hat{\Sigma}_i = \psi_{i*}\Sigma_c$, $\hat{J}_i(z_i, v_i, u_i) = \psi_{i*}\hat{J}_{i-1}(z_{i-1}, x_i, v_{i-1}, u_i)$, and $\hat{D}_i = \psi_{i*}\hat{D}_{i-1}$ and sends them to agent $i+1$.*
2) *Phase 2: Execution*

   Encoding:
   a) *Plant Σ_{i-1} periodically measures y_{i-1}, encodes it into $w_{i-1} = S_{i-1}y_{i-1}$ and sends it to plant Σ_i.*
   b) *Plant Σ_i periodically measures y_i, appends it to w_{i-1} to form y_c, encodes it into $w_i = S_i y_c$ and sends it to the cloud.*

   Optimization:
   The cloud uses the received encoded measurement w_i, estimates the plant state z_i, computes the input v_i minimizing $\hat{J}_i$ subject to the constraint $\hat{D}_i\hat{\eta}_i \leq 0$ and the dynamics $\hat{\Sigma}_i$, and sends v_i to plant Σ_i.

   Decoding:
   a) *Plant Σ_i decodes v_i to produce u_c, using* (IV.14)*, applies u_i to the actuators and sends v_{i-1} to plant Σ_{i-1}.*
   b) *Plant Σ_{i-1} decodes v_{i-1} to produce u_{i-1}, using* (IV.5)*, and applies u_{i-1} to the actuators.*

VI. Conclusion

In this paper, the problem of ensuring privacy when controlling a single agent over the cloud was extended to the case of multiple agents. We showed how isomorphisms of control systems can be used to obtain a lightweight encoding scheme that protects privacy of the exchanged data. This result generalizes to the case of finitely many plants controlled by a single cloud.

References

[1] N. Suri, M. Tortonesi, J. Michaelis, P. Budulas, G. Benincasa, S. Russell, C. Stefanelli, and R. Winkler, "Analyzing the applicability of internet of things to the battlefield environment," in *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, May 2016.

[2] T. Abdelzaher, N. Ayanian, T. Basar, S. Diggavi, J. Diesner, D. Ganesan, R. Govindan, S. Jha, T. Lepoint, B. Marlin, K. Nahrstedt, D. Nicol, R. Rajkumar, S. Russell, S. Seshia, F. Sha, P. Shenoy, M. Srivastava, G. Saukhatme, A. Swami, P. Tabuada, D. Towsley, N. Vaidya, and V. Veeravalli, "Will distributed computing revolutionize peace? the emergence of battlefield iot," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2018.

[3] 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)*, Dec 2015, pp. 6836–6843.

[4] 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 NECSYS 2016.

[5] 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.

[6] 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.

[7] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: A review and open problems," in *Proceedings of the 2001 Workshop on New Security Paradigms*, ser. NSPW '01. New York, NY, USA: ACM, 2001, pp. 13–22. [Online]. Available: http://doi.acm.org/10.1145/508171.508174

[8] 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.

[9] T. Tanaka, M. Skoglund, H. Sandberg, and K. H. Johansson, "Directed Information as Privacy Measure in Cloud-based Control," *ArXiv e-prints*, May 2017.

[10] J. Corts, 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 (CDC)*, Dec 2016, pp. 4252–4272.

[11] 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.

[12] A. Sultangazin and P. Tabuada, "Towards the use of symmetries to ensure privacy in control over the cloud," University of California, Los Angeles, Tech. Rep., 01 2018. [Online]. Available: http://www.cyphylab.ee.ucla.edu/Home/publications/UCLA-CyPhyLab-2018-01.pdf?attredirects=0