Towards the use of symmetries to ensure privacy in control over-the-cloud

Alim Sultangazin and P. Tabuada

Department of Electrical and Computer Engineering
University of California, Los Angeles

57th IEEE Conference on Decision and Control,
Miami Beach, FL, USA

December 19th, 2018

Optimization and privacy

- Optimization has numerous applications in control (e.g., MPC, minimum-energy state estimation).
- In many problem instances, the optimization is performed repeatedly once new measurements arrive.

Optimization and privacy

- Optimization has numerous applications in control (e.g., MPC, minimum-energy state estimation).
- In many problem instances, the optimization is performed repeatedly once new measurements arrive.
- There are at least two reasons to perform optimization over the cloud:
  - when local compute power is insufficient;
  - when data is distributed.

Optimization and privacy

- Optimization has numerous applications in control (e.g., MPC, minimum-energy state estimation).
- In many problem instances, the optimization is performed repeatedly once new measurements arrive.
- There are at least two reasons to perform optimization over the cloud:
  - when local compute power is insufficient;
  - when data is distributed.
- Protecting data privacy is paramount to enable a wider acceptance of optimization over the cloud.
- In the context of control (e.g., MPC) we need to provide the cloud with:
  - plant (e.g., am I driving a car or a motorbike today);
  - cost function (e.g., am I optimizing for safety or speed?);
  - and measurements (e.g., am I violating speed limits? Where did I sleep last night?).

Optimization and privacy

Objectives

- How to:
  - leverage the compute power of the cloud;
  - keep data private;
  - do so in a computationally efficient manner so as not to degrade control performance?

Optimization and privacy

Objectives

- How to:
  - leverage the compute power of the cloud;
  - keep data private;
  - do so in a computationally efficient manner so as not to degrade control performance?
- Answer: leverage isomorphisms and symmetries of control systems.

Related work

- Data encryption
  - Partial or full homomorphic encryption [Y. Shoukry et al. ‘16]
  - Data obfuscation [C. Wang, K. Ren, and J. Wang ‘11]
  - Multi-party computation [W. Du and M. J. Atallah ‘01]
- Data perturbation
  - cloud receives perturbed data of a collection of systems (e.g. differential privacy).
  - [J. Cortés et al. ‘16], [F. Koufogiannis and G. J. Pappas ‘17]

Drawbacks

- large computational overhead (HE has exponential complexity)
- only studied for linear programs, does not handle dynamics
- requires several clients
- methods require adding noise, which reduces estimation performance; noise might accumulate with time

Problem Formulation: dynamics

- Linear system $\Sigma = (A, B, C)$, which we refer to as a plant, is described by:

$$x[k+1] = Ax[k] + Bu[k] \qquad y[k] = Cx[k], \tag{3.1}$$

where x, u and y are the state, input and output of the system, respectively.

Problem Formulation: dynamics

- Linear system $\Sigma = (A, B, C)$, which we refer to as a plant, is described by:

$$x[k+1] = Ax[k] + Bu[k] \qquad y[k] = Cx[k], \tag{3.1}$$

  where x, u and y are the state, input and output of the system, respectively.
- The triple $\{x[k], u[k], y[k]\}_{k \in \mathbb{N}}$ is called a trajectory if it satisfies (3.1) for all $k \in \mathbb{N}$.

Problem Formulation: cost function

- Moreover, each plant has a cost function that defines the control objective and constraints. We consider quadratic cost functions and affine constraints:

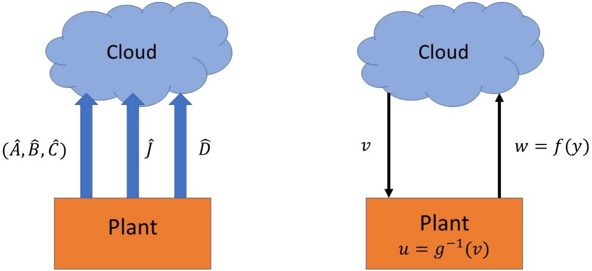
$$J(x,u) = \sum_{k=0}^{N} \Delta\eta^T[k] M \Delta\eta[k] \qquad D\eta[k] \leq 0,$$

where $\eta[k] = \begin{bmatrix} x[k] & u[k] \end{bmatrix}^T$, $\Delta\eta[k] = \begin{bmatrix} x[k] - x^*[k] & u[k] - u^*[k] \end{bmatrix}^T$ and x^*, u^* are desired state and input, respectively.

Problem Formulation: attack model and privacy objectives

The cloud is an honest but curious adversary (i.e. it will follow the protocol all parties agree upon, but may attempt to extract and leak private info).

Problem Formulation: Algorithm



Communication algorithm:

1. Handshake: plant transmits suitably modified versions of the plant model, cost and constraints.
2. Plant operation: plant sends 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.

Problem Formulation: Scenarios

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.

Problem Formulation: Scenarios

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.

Scenarios considered:

1. The cloud has no knowledge about the plant;

Problem Formulation: Scenarios

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.

Scenarios considered:

1. The cloud has no knowledge about the plant;
2. The cloud has no knowledge about the plant except knowing its sensors and actuators (e.g., the plant is a house and the cloud knows it receives current and voltage measurements);

Problem Formulation: Scenarios

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.
Scenarios considered:

1. The cloud has no knowledge about the plant;
2. The cloud has no knowledge about the plant except knowing its sensors and actuators (e.g., the plant is a house and the cloud knows it receives current and voltage measurements);
3. The cloud has complete knowledge about plant dynamics including its sensors and actuators (e.g., the plant is an autonomous car controlled over the manufacturer's cloud).

CyPhyLab

Problem Formulation: Objectives

Objectives:

- Modify plant (except in 3), cost, constraints and measurements to prevent the cloud from inferring them.
- Construct input from the data provided by the cloud so that controlling the plant with such input results in a trajectory minimizing the cost J.

Results

Definition

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, G and S are invertible linear maps and F is a linear map such that:

$$\hat{\Sigma} = \psi_* \Sigma = (P(A - BG^{-1}F)P^{-1}, PBG^{-1}, SCP^{-1}).$$

- We can interpret an isomorphism $\psi = (P, F, G, S)$ as a change of coordinates in the states, a change of coordinates in the inputs with feedback, and a change of coordinates in the outputs:

$$z = Px, \qquad v = Fx + Gu \qquad w = Sy.$$

- These changes of coordinates also induce a new cost $\hat{J}$ and new constraints $\hat{D}$.

Results

Let us define a quadruple of the dynamics, cost, constraints and the trajectory as:

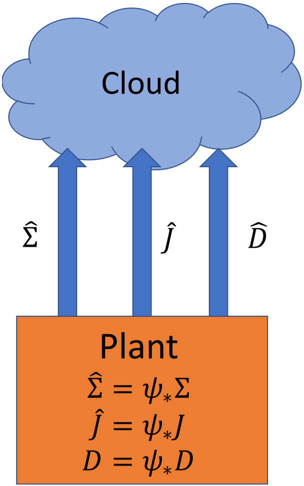
$$\Omega = \{\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}}\}. \tag{4.1}$$

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 Ω.

Definition

Let $\mathcal{G}$ be a subgroup of the group of all isomorphisms of Σ. Two quadruples Ω and $\hat{\Omega}$ are called $\sim_{\mathcal{G}}$-equivalent if there exists an isomorphism $\psi \in \mathcal{G}$ such that $\psi_*\Sigma = \hat{\Sigma}$, $\hat{J} = \psi_* J$, $\hat{D} = \psi_* D$ and system variables transformation equations hold for every $k \in \mathbb{N}$.

Results: Algorithm

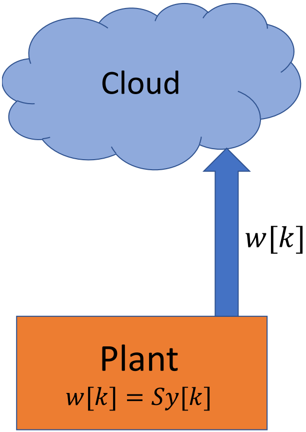


Algorithm (Plant $\Longleftrightarrow$ Cloud)

1. Phase 1: Handshaking

   The plant encodes its dynamics, cost function and constraint matrix and sends them to the cloud.

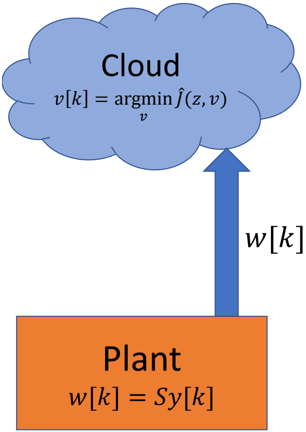
Results: Algorithm



Algorithm (Plant $\Longleftrightarrow$ Cloud)

1. Phase 1: Handshaking
   The plant encodes its dynamics, cost function and constraint matrix and sends them to the cloud.
2. Phase 2: Plant operation (repeated)
   Encoding: The plant measures $y[k]$, encodes it into $w[k] = Sy[k]$ and sends it to the cloud.

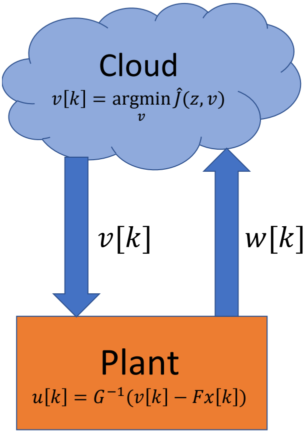
Results: Algorithm



Algorithm (Plant $\Longleftrightarrow$ Cloud)

1. **Phase 1: Handshaking**
   The plant encodes its dynamics, cost function and constraint matrix and sends them to the cloud.
2. **Phase 2: Plant operation** (repeated)
   Encoding: The plant measures $y[k]$, encodes it into $w[k] = Sy[k]$ and sends it to the cloud.
   Optimization: The cloud uses $w[k]$, estimates the plant state $z[k]$, computes the input $v[k]$ minimizing $\hat{J}$ subject to the constraint $\hat{D}\eta_k \leq 0$ and the dynamics $\hat{\Sigma}$, and sends $v[k]$ to the plant.

Results: Algorithm



Algorithm (Plant $\Longleftrightarrow$ Cloud)

1. **Phase 1: Handshaking**
   The plant encodes its dynamics, cost function and constraint matrix and sends them to the cloud.
2. **Phase 2: Plant operation** (repeated)
   Encoding: The plant measures $y[k]$, encodes it into $w[k] = Sy[k]$ and sends it to the cloud.
   Optimization: The cloud uses $w[k]$, estimates the plant state $z[k]$, computes the input $v[k]$ minimizing $\hat{J}$ subject to the constraint $\hat{D}\eta_k \leq 0$ and the dynamics $\hat{\Sigma}$, and sends $v[k]$ to the plant.
   Decoding: The plant decodes $v[k]$ to produce $u[k]$ and sends $u[k]$ to the actuators.

Results: Main theorems - Compatibility

Lemma

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

- If the cloud receives $\hat{\Sigma}$, then the received measurements Sy and produced control inputs are compatible with the plant $\hat{\Sigma}$.

Results: Main theorems - Correctness

Lemma (On the utility of a modified optimization problem)

Suppose the cloud solves the optimization problem:

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

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:

$$\min_{u} \quad J(x, u) \qquad \text{subject to} \quad D\eta_k \leq 0,$$

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

- By applying the "decoded" input, $u^o = G^{-1}(v^o - Fx)$, we control the plant optimally.

Results: Main theorems - Privacy

Theorem (On the privacy of quadruples)

Any two quadruples:

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

related by an isomorphism (in other words, $\sim_{\mathcal{G}}$ equivalent) are indistinguishable by the cloud, i.e., the exchanged messages between the cloud and plant are the same.

- The cloud knows the quadruple $(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}})$ belongs to an equivalence class but cannot pinpoint which member of the equivalence class it is.

Results: Main theorems - Privacy

Theorem

Any two quadruples $(\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}})$ *and* $(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}})$ *related by an isomorphism are indistinguishable by the cloud, i.e., the exchanged messages between the cloud and plant are the same.*

- When the cloud has no knowledge about the plant, inputs, or outputs, we use the full isomorphism group.

Results: Main theorems - Privacy

Theorem

Any two quadruples $(\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}})$ *and* $(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}})$ *related by an isomorphism are indistinguishable by the cloud, i.e., the exchanged messages between the cloud and plant are the same.*

- When the cloud has no knowledge about the plant, inputs, or outputs, we use the full isomorphism group.
- When the cloud knows the sensors and actuators but not the plant model, we use the subgroup of isomorphisms that leaves the inputs and outputs invariant.
  - The cloud learns the transfer function but not the plant realization neither the state trajectory.

Results: Main theorems - Privacy

Theorem

Any two quadruples $(\Sigma, J, D, \{x[k], u[k], y[k]\}_{k\in\mathbb{N}})$ *and* $(\hat{\Sigma}, \hat{J}, \hat{D}, \{z[k], v[k], w[k]\}_{k\in\mathbb{N}})$ *related by an isomorphism are indistinguishable by the cloud, i.e., the exchanged messages between the cloud and plant are the same.*

- When the cloud has no knowledge about the plant, inputs, or outputs, we use the full isomorphism group.
- When the cloud knows the sensors and actuators but not the plant model, we use the subgroup of isomorphisms that leaves the inputs and outputs invariant.
  - The cloud learns the transfer function but not the plant realization neither the state trajectory.
- When the cloud has full knowledge, we use the subgroup of isomorphisms that leaves the inputs, outputs, and plant model invariant.
  - The state trajectory remains private.

Results: Analysis of the algorithm

- The cloud does not require any new protocol since it remains oblivious to the fact that "encryption" is being used.
- At the client side, the algorithm only involves matrix multiplications. This results in a lightweight encoding scheme.

Conclusion

- In this paper, the problem of ensuring privacy was addressed by using isomorphisms and symmetries of control systems.
- We showed how isomorphisms of control systems can be used to obtain a lightweight encoding scheme that protects privacy of the exchanged data.

Ongoing work

- How to quantify privacy?
  - The number of elements in each equivalence class is infinite.
  - Manifold dimension is a possible quantification of privacy.
  - More detailed description is possible in certain cases: for controllable and observable systems that are prime, the cloud only learns the controllability indices (=observability indices).

Ongoing work

- How to quantify privacy?
  - The number of elements in each equivalence class is infinite.
  - Manifold dimension is a possible quantification of privacy.
  - More detailed description is possible in certain cases: for controllable and observable systems that are prime, the cloud only learns the controllability indices (=observability indices).
- How about side knowledge?
  - If side knowledge is modeled by a surjective linear map π of the isomorphism (secret key), the dimension of the manifold of "uncertainty" is reduced by the dimension of the image of π.

Ongoing work

- How to quantify privacy?
  - The number of elements in each equivalence class is infinite.
  - Manifold dimension is a possible quantification of privacy.
  - More detailed description is possible in certain cases: for controllable and observable systems that are prime, the cloud only learns the controllability indices (=observability indices).
- How about side knowledge?
  - If side knowledge is modeled by a surjective linear map π of the isomorphism (secret key), the dimension of the manifold of "uncertainty" is reduced by the dimension of the image of π.
- Experimental validation is ongoing.

Reference list

- 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.
- 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.
- 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
- 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 (CDC), Dec 2016, pp. 4252– 4272.
- 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.