

ABSTRACT

Title of dissertation: PRIVACY IN DISTRIBUTED MULTI-
-AGENT COLLABORATION:
CONSENSUS AND OPTIMIZATION

Nirupam Gupta
Doctor of Philosophy, 2018

Dissertation directed by: Associate Professor Nikhil Chopra
Department of Mechanical Engineering

Distributed multi-agent collaboration is an interactive algorithm that enables agents in a multi-agent system (MAS) to achieve pre-defined collaboration objective in a *distributed* manner, such as agreeing upon a common value (commonly referred as *distributed consensus*) or optimizing the aggregate cost of the MAS (commonly referred as *distributed optimization*).

Agents participating in a typical distributed multi-agent collaboration algorithm can lose privacy of their inputs (containing private information) to a passive adversary in two ways. The adversary can learn about agents' inputs either by corrupting some of the agents that are participating in the collaboration algorithm or by eavesdropping the communication links between the agents during an execution of the collaboration algorithm. Privacy of the agents' inputs in the former case is referred as internal privacy, and privacy of the agents' inputs in the latter case is referred as external privacy.

This dissertation proposes a protocol for preserving internal privacy in two particular distributed collaborations: distributed average consensus and distributed optimization. It is shown that the proposed protocol can preserve internal privacy of sufficiently *well connected* honest agents (agents that are not corrupted by the adversary) against adversarial agents (agents that are corrupted by the adversary), without affecting the collaboration objective.

This dissertation also investigates a model-based scheme, as an alternative to cryptographic encryptions, for external privacy in distributed collaboration algorithms that can be modeled as linear time-invariant networked control systems. It is demonstrated that the model-based scheme preserves external privacy, without affecting the collaboration objective, if the system parameters of the networked control system, that equivalently models the distributed collaboration algorithm, satisfy certain conditions. Unlike cryptographic encryptions, the model-based scheme does not rely on secure generation and distribution of keys amongst the agents for guaranteeing external privacy.

PRIVACY IN DISTRIBUTED MULTI-AGENT
COLLABORATION:
CONSENSUS AND OPTIMIZATION

by

Nirupam Gupta

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:

Prof. Nikhil Chopra (Chair/Advisor)

Prof. Balakumar Balachandran

Prof. Jonathan Katz (Dean's Representative)

Prof. Shapour Azarm

Prof. Yancy Diaz-Mercado

© Copyright by
Nirupam Gupta
2018

Dedication

To my parents.

&

Discoveries that await.

caturadhikaṃ śatamaṣṭagaṇaṃ dvāṣaṣṭistathā sahasrāṇām —
ayutadvaya viṣkambhasyāsanno vṛttapariṇāhaḥ —

- Āryabhaṭa (476 - 550 AD)

The circumference of a circle of diameter 20000,
is approximately equal to $(100 + 4) \times 8 + 62000$.

- Aryabhata (476 - 550 AD)

Acknowledgments

I would like to acknowledge all the people who helped me in the process of finishing this dissertation.

First and foremost, I am humbly grateful to my advisor Prof. Nikhil Chopra, for this dissertation is the result of his constant guidance and support. I am also grateful to Prof. Jonathan Katz, whose brilliant suggestions and guidance has been substantial in shaping the results presented in this dissertation. Special thanks to Prof. Balakumar Balachandran, for sharing some of his ideas on the topic and giving valuable advice that has helped in shaping the content of this dissertation. I am also thankful to Prof. Shapour Azarm and Prof. Yancy Diaz-Mercado for serving in my dissertation committee and providing their valuable advices for improving this dissertation.

I am greatly thankful to all the friends I made during my stay at University of Maryland, without them this journey would not have been so grand. I would like to thank Monil for helping me in editing the contents of this dissertation, and for helping me in several other ways throughout this period. I am thankful to Yimeng, with whom I have co-authored few good papers, for sharing his research ideas with me, and for inviting me on several occasions to his house for some delicious Chinese food. I am thankful to Gurtaj for discussions on many interesting ideas, and all those soccer matches we watched together. I am thankful to Prasad for his suggestions on my future ventures, and for inviting me on his many excursions to New York City. I am thankful to Bryan for discussing some interesting research ideas during my initial years. I am thankful to Jeffrey for recommending me for a post-doctoral research position at his workplace. I am thankful to Celeste for organizing such wonderful lab events, and for always inviting me. I am thankful to Gizem for helping me in the application process for my future endeavors, and those amazing guitar sessions. I am thankful to Vipin for his Krishna luncheons. I am thankful to Hesham for all the activities we did together. I am thankful to Joana and Paul for always inviting me to their celebrations. All these people, in some or the other way, have helped me during my PhD, and made it possible for me to write this dissertation.

I am grateful to my parents Sandhya Gupta and Pankaj Gupta, my brother Madhupam and my sister Ila. Their love and support has always given me courage in life.

The work presented in this dissertation was supported by the National Science Foundation under grant ECCS1711554 and the Naval Air Warfare Center Aircraft Division, Pax River, MD, under contract N00421132M022.

Table of Contents

Dedication	ii
Acknowledgements	iii
List of Figures	vi
List of Abbreviations	viii
1 INTRODUCTION	1
1.1 Distributed Multi-Agent Collaboration	3
1.2 Privacy in Distributed Collaboration	6
1.2.1 Internal Privacy	8
1.2.2 External Privacy	11
1.3 Dissertation Contribution	13
1.3.1 Internal Privacy in Distributed Average Consensus	14
1.3.2 Internal Privacy in Distributed Optimization	19
1.3.3 Model-Based Scheme For External Privacy	23
1.4 Notation and Preliminaries	26
1.4.1 Mathematical Model of the Communication Network	28
1.5 Dissertation Organization	30
2 INTERNAL PRIVACY IN DISTRIBUTED AVERAGE CONSENSUS	31
2.1 Formal Problem Description	33
2.1.1 Adversarial Model	34
2.1.2 Privacy Definition	34
2.1.3 Assumptions	35
2.2 Proposed Protocol	36
2.2.1 Bounded Integral Inputs	37
2.2.2 Bounded Real-Valued Inputs	38
2.2.3 Privacy Analysis	39
2.3 Illustration	46
2.3.1 Bounded Integral Inputs	46
2.3.2 Bounded Real-Valued Inputs	49

2.4	Extension	50
2.5	Concluding Remarks	51
2.5.1	Limitations	53
2.5.2	Future Research	54
3	INTERNAL PRIVACY IN DISTRIBUTED OPTIMIZATION	56
3.1	Formal Problem Description	57
3.1.1	Adversarial Model	58
3.1.2	Privacy Definition	59
3.1.3	Assumptions	60
3.2	Proposed Protocol	61
3.2.1	Privacy Analysis	63
3.2.2	Vertex Expansion and Privacy Strength	68
3.3	Illustration	69
3.4	Concluding Remarks	71
3.4.1	Limitations	73
3.4.2	Future Research	74
4	MODEL-BASED SCHEME FOR EXTERNAL PRIVACY IN DISTRIBUTED COLLABORATION ALGORITHMS	75
4.1	Formal Problem Description	78
4.1.1	Adversarial Model	79
4.1.2	Assumptions	79
4.2	Model-Based Privacy Scheme	80
4.2.1	Masking With System Kernel (MSK)	80
4.2.2	Privacy Definition	82
4.2.3	Privacy Analysis	89
4.3	Numerical Examples	102
4.3.1	Example I	102
4.3.2	Example II	104
4.4	Concluding Remarks	109
4.4.1	Limitations	109
4.4.2	Future Research	110
5	SUMMARY	111
5.1	Contribution: Internal Privacy in Distributed Collaboration	112
5.1.1	Limitations	113
5.1.2	Future Research	115
5.2	Contribution: External Privacy in Distributed Collaboration	116
5.2.1	Limitations	117
5.2.2	Future Research	118
	Bibliography	119

List of Figures

1.1	Examples of collaborations in multi-agent systems.	2
1.2	Illustrations on centralized and distributed multi-agent collaboration.	3
1.3	Three stages of a distributed multi-agent collaboration.	5
1.4	An example illustrating the different types of privacy, external and internal privacy, in distributed multi-agent collaboration.	7
1.5	In case of internal privacy, the view of an adversary is the collective information learned by all the agents that are corrupted by the adversary during an execution of the distributed collaboration algorithm.	10
1.6	In case of external privacy, the view of an adversary is the collective information contained in all the messages that are exchanged between the agents during an execution of the distributed collaboration algorithm.	13
1.7	Comparison of the proposed protocol with the existing protocols for internal privacy in distributed average consensus.	18
1.8	Comparison of the proposed protocol with the existing protocols for internal privacy in distributed optimization.	23
1.9	Comparison of the proposed model-based scheme with the existing solutions for external privacy in distributed collaboration algorithms.	26
2.1	Proposed protocol for internal privacy in distributed average consensus.	36
2.2	Illustration of the proposed distributed average consensus protocol. Arrows (in blue) indicate the flow of information over the communication links.	47
2.3	An example illustrating the internal privacy guarantee in the proposed distributed average consensus protocol.	51
3.1	Proposed protocol for internal privacy in distributed optimization.	62
3.2	Illustration of the proposed distributed optimization protocol. Arrows (in blue) indicate the flow of information over communication links.	70
4.1	An illustration of the modeling of a distributed state-consensus algorithm by a one-channel feedback networked control system.	76
4.2	Schematic of a networked control system with remote sensors for measuring states.	78
4.3	Schematic of the proposed model-based scheme, referred as masking with system kernel (MSK).	80

4.4	Illustration of the evolution of sets $\mathcal{K}_{t,0}(\mathbf{z}_t)$ with time t and the implication of condition C2 given in the definition of privacy of states.	88
4.5	An example for illustrating the proposed model-based scheme.	104
5.1	In the proposed protocol for internal privacy in distributed collaborations, the agents run a privacy mechanism on their inputs $\{\mathbf{l}_i\}$ before execute the distributed collaboration algorithm, to compute effective inputs $\{\tilde{\mathbf{l}}_i\}$ that satisfy $\text{Collab}_i(\{\mathbf{l}_k\}) = \text{Collab}_i(\{\tilde{\mathbf{l}}_k\})$	113

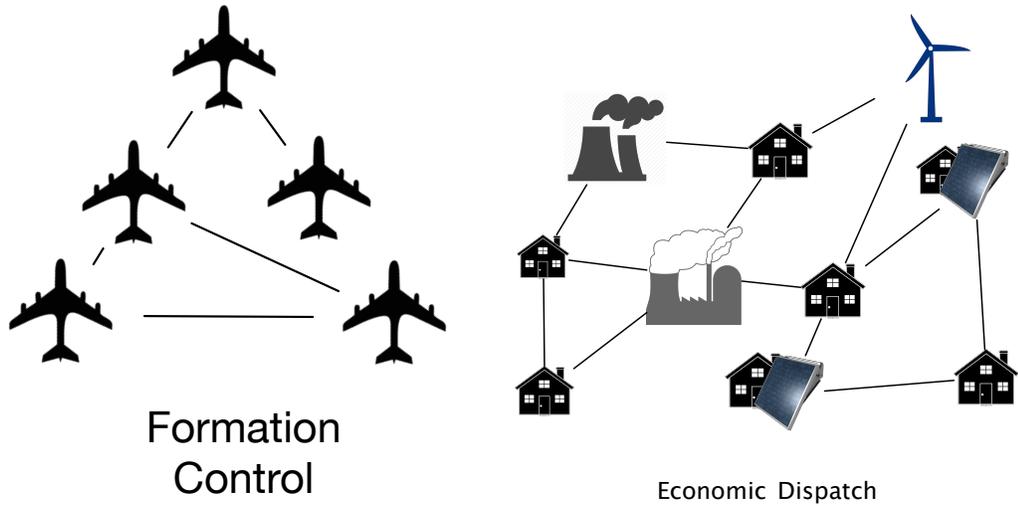
List of Abbreviations

AT	Algebraic Transformations
CST	Control System Theoretic
DP	Differential Privacy
KL	Kullback-Leibler
LTI	Linear Time-Invariant
MAS	Multi-agent System
MPC	Secure Multiparty Computation
MSK	Masking with System Kernel
NCS	Networked Control System
NIST	National Institute of Standards and Technology
RSA	Rivest-Shamir-Adleman
SCADA	Supervisory Control And Data Acquisition

CHAPTER 1: INTRODUCTION

For the purpose of this dissertation, Multi-agent system (MAS) refers to a network of multiple agents, wherein each agent is capable of communicating, processing and storing information (or signals). Depending on the type of MAS, an agent could either be a machine or a human. For example, social network is a MAS with humans assuming the role of agents. Whereas, internet-of-things is a MAS with agents being electronic embedded devices. In a typical MAS, the agents collaborate with each other to fulfill a desired collaboration objective. Collaboration of agents in a MAS, which is also commonly referred to as multi-agent collaboration, relies on an interactive algorithm or protocol¹ that defines a set of instructions for each and every agent in order for the MAS to fulfill the desired collaboration objective. For example, consider formation control of autonomous vehicles (refer Fig. 1.1a). In this case, a collaboration algorithm prescribes control laws for all the vehicles in order for them to maintain a desired geometric formation. Consider the economic dispatch in a power grid (refer Fig. 1.1b). Here, a collaboration algorithm enables power generation plants and power consumers in minimizing the cost of power generation while satisfying the power demands.

¹Throughout this dissertation, the terms algorithm and protocol are used interchangeably.

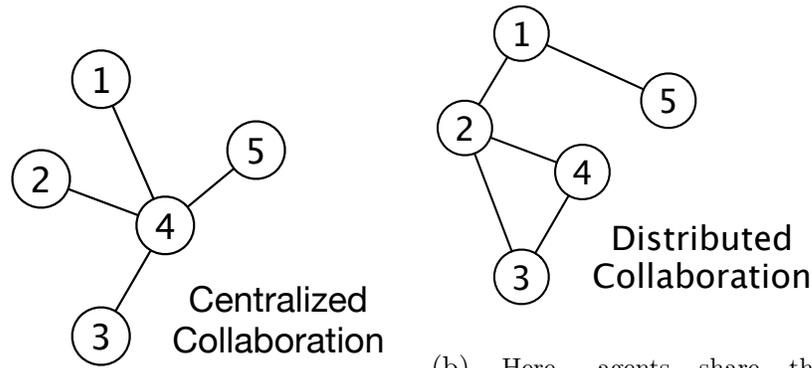


- (a) The objective is to maintain a spatial geometric formation of vehicles while they navigate in autonomous manner.
- (b) The objective is to minimize the cost of energy generation while satisfying the energy consumption requirements.

Figure 1.1: Examples of collaborations in multi-agent systems.

Multi-agent collaboration can either be *centralized* or *distributed*. In centralized collaboration (as shown in Fig. 1.2a), one of the agents acts as a leader. All the other agents communicate the information held by them with the leader, which then processes the received information in a prescribed manner in order to fulfill the collaboration objective (eg. cloud-based services). In distributed collaboration (as shown in Fig. 1.2b), every agent communicates information held by it with its neighboring agents in the MAS. Then, every agent processes the received information in a prescribed manner to fulfill the collaboration objective (often requires multiple rounds of communication). Therefore, unlike centralized collaboration in distributed collaboration all the agents share the responsibility of collaboration and every agent is only required to be aware of its local network topology. This dissertation mainly

deals with distributed multi-agent collaborations.



(a) Here, agent 4 acts as the leader for the entire multi-agent system. All other agents communicate with the leader, who in turn processes the received information to achieve the desired objective.

(b) Here, agents share the responsibility of collaboration. Each agent communicates with its neighbors and every agent processes the received information for achieving the desired collaboration objective.

Figure 1.2: Illustrations on centralized and distributed multi-agent collaboration.

1.1 Distributed Multi-Agent Collaboration

As mentioned above, in distributed multi-agent collaboration the agents communicate with (only) their neighbors in the network in order to achieve the desired collaboration objective. Therefore, in general, distributed collaboration algorithms are much more scalable than their centralized counterpart and are easier to set up as adding a new agent in the network only requires it to be linked with any one of the agents already present. Moreover, a distributed collaboration algorithm imposes uniform computation and communication costs on the agents, unlike centralized col-

laboration wherein the leader bears most of the computation and communication costs. This dissertation focuses only on distributed collaboration algorithms due to the fact that a centralized collaboration algorithm can always be treated as a distributed collaboration algorithm with a very particular (star-like) communication network topology between the agents.

A distributed collaboration algorithm consists of three stages: initial stage, intermediate stage and final stage (refer Fig. 1.3). In the initial stage, each agent $i \in \{1, \dots, n\}$ (n being the total number of agents) collects or holds information: In_i , which is referred to as input hereafter. The inputs $\{\text{In}_i\}$ are private and known only to the respective agents. In the intermediate stage, each agent i communicates with its neighbors by transmitting messages $\{m_{ij}\}$ (here, index j represents the neighbors of i) and generates an internal state: State_i , which is referred to as state hereafter. The states $\{\text{State}_i\}$ and the messages generated during the intermediate stage are part of the collaboration algorithm and allow agents to achieve the desired collaboration objective. Ultimately, in the final stage the agents achieve respective outputs: $\{\text{Out}_i\}$ that depend entirely upon the inputs $\{\text{In}_i\}$, and are completely independent of the internal states $\{\text{State}_i\}$ or messages generated during the intermediate stage². Specifically,

$$\text{Out}_i = \text{Collab}_i(\{\text{In}_k\})$$

where Collab_i is a function that dictates the output of agent i in the distributed collaboration owing to the desired collaboration objective. For example, if the inputs

²Therefore, it is possible to have multiple distributed collaboration algorithms to achieve a collaboration objective.

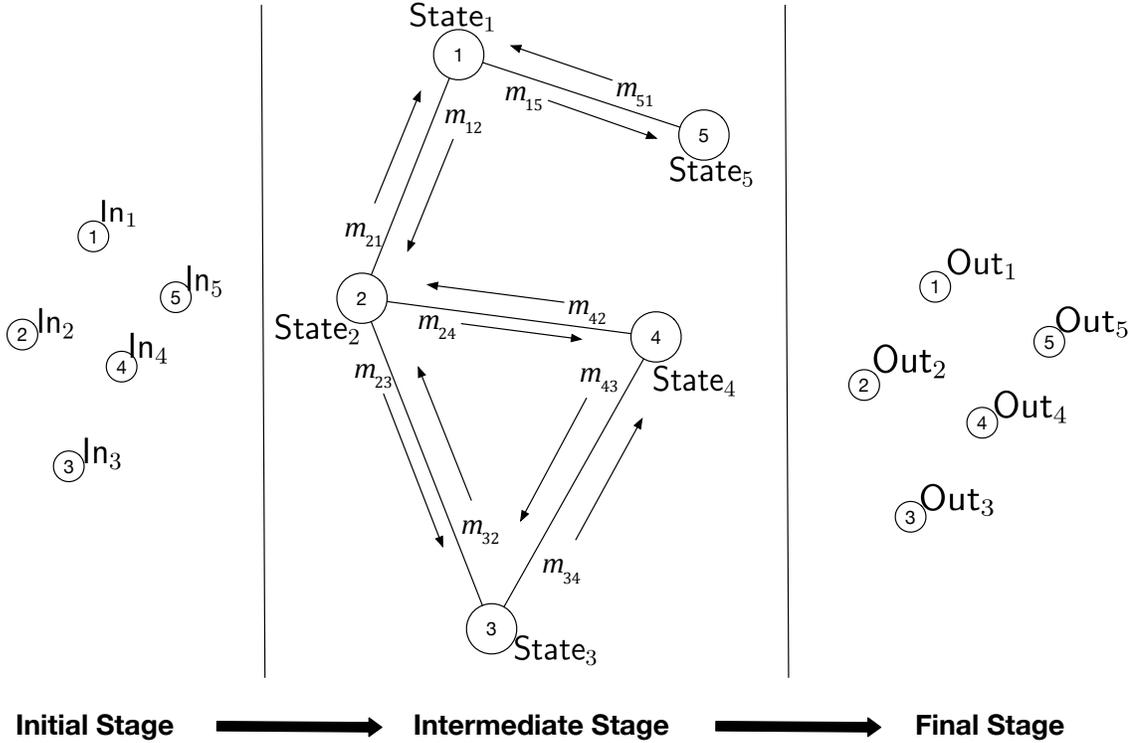


Figure 1.3: Three stages of a distributed multi-agent collaboration.

belong to a set of numbers and the collaboration objective for each agent is to compute the sum of all the inputs then $\text{Collab}_i(\{\text{In}_k\}_{k \in \{1, \dots, n\}}) = \sum_k \text{In}_k, \forall i$.

The communication network (or network topology) of the agents during the intermediate stage is mathematically represented by an undirected simple graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of agents and \mathcal{E} is the set of communication links between agents. As \mathcal{G} is undirected, each element of \mathcal{E} is given by an unordered pair $\{i, j\}$, where i and j are the agents at the terminal ends of the link. \mathcal{N}_i represents the set of neighbors of an agent i . (As the graph is simple, therefore $i \notin \mathcal{N}_i$.)

The subsequent section introduces the issue of privacy in distributed collaboration along with some discussion on privacy definitions that are essential for un-

derstanding the content of this dissertation.

1.2 Privacy in Distributed Collaboration

As mentioned above, in distributed collaboration, every agent's state gets shared with multiple (neighboring) agents. Due to this distributed sharing of states the privacy of an agent's inputs is at risk not only against a passive adversary that could be listening to the communication links between agents (also known as eavesdropper), but also against a passive adversary that corrupts a group of agents [1–3]. An adversary is passive if it does not disrupt the distributed collaboration algorithm, but uses the information learned during an execution of the distributed collaboration algorithm to infer something about the inputs of the agents participating in the collaboration. Needless to say, when agents' inputs hold sensitive information, privacy in distributed collaboration algorithms becomes even more critical and almost an indispensable requirement.

For example, consider the case of computing net loss of energy in a distributed power grid shown in Fig. 1.4. To achieve this objective, the agents (power plants and consumers) can implement a distributed average consensus³ to compute the net sum of all agents' power consumption (or generation) over time to get the net loss of energy in the power grid. Suppose that the red colored dotted edges are insecure communication links, that is messages that are exchanged over these links can be

³Distributed average consensus is a special distributed collaboration algorithm that enables agents to compute the average of their inputs in a distributed manner, that is $\text{Collab}_i(\{\ln_k\}) = 1/n \sum_k \ln_k, \forall i$.

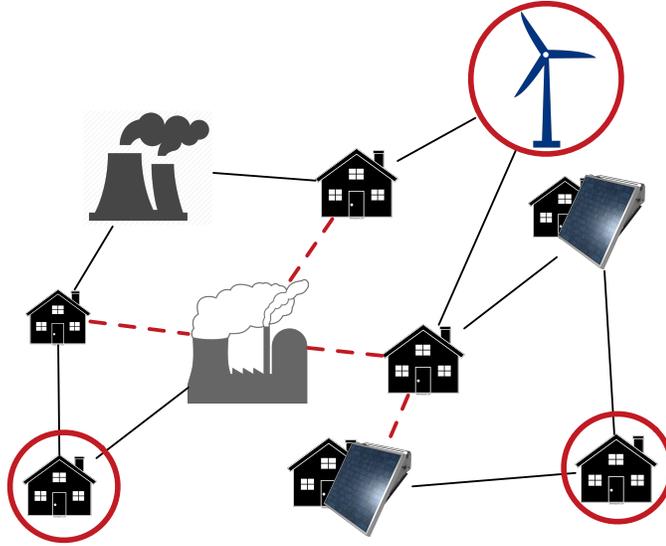


Figure 1.4: An example illustrating the different types of privacy, external and internal privacy, in distributed multi-agent collaboration.

read by an adversary. Further, suppose that the agents circled in red are corrupted by the same adversary. In this case, if the agents use a traditional distributed average consensus algorithm (refer. [4–6]) then the energy consumption (or generation) of an agent gets revealed to the adversary either if one of the communication links emerging from the agent is insecure or if any one of the agent’s neighbors is corrupted by the adversary.

Now, depending upon the means through which an adversary learns information about agents’ inputs in a distributed collaboration algorithm, privacy in distributed multi-agent collaboration can be classified into two types:

1. *Internal Privacy*: refers to the privacy of agents’ inputs against a passive adversary that corrupts some of the agents that are participating in the distributed collaboration.

2. *External Privacy*: refers to the privacy of agents' inputs against a passive adversary reading messages that are exchanged between agents in the intermediate stage of the distributed collaboration.

Note: In this dissertation, the adversary is always assumed passive. That is, the adversary does not disrupt the distributed collaboration algorithm, but may use the information learned (or viewed) by it during an execution of the distributed collaboration algorithm to infer something about the inputs of the agents participating in the collaboration.

The following subsections provide formal descriptions on internal and external privacy.

1.2.1 Internal Privacy

Internal privacy is concerned with privacy of agents' inputs against a passive adversary that corrupts some of the agents participating in the collaboration. The agents that are corrupted by the passive adversary are also commonly referred as passively adversarial or semi-honest agents. Formally, a distributed collaboration algorithm preserves internal privacy if the entire *view* of the adversary — information learned by the adversary or corrupted agents during an execution of the distributed collaboration algorithm — does not leak any (or significant) additional information (in statistical sense) about the remaining honest agents' inputs apart from what is already revealed from the outputs of the corrupted agents [7]. The latter is unavoidable as the proposed privacy protocol should not be perturbing the outputs of any

agent. If $\mathcal{C} \subset \mathcal{V}$ denotes the set of corrupted agents and $\mathcal{E}_c \subseteq \mathcal{E}$ denotes the set of communication links incident to \mathcal{C} , then the entire view of the adversary is given as

$$\text{view of } \mathcal{C} = \{\text{In}_i, \text{State}_i, \text{Out}_i\}_{i \in \mathcal{C}} + \{m_{ij}\}_{\{i,j\} \in \mathcal{E}_c}$$

For the purpose of rigorous mathematical analysis of internal privacy, we define $\text{View}_{\mathcal{C}}(\{\text{In}_i\})$ as the random variable denoting the information learned⁴ by agents in \mathcal{C} (or the view of \mathcal{C}) during an execution of the distributed collaboration algorithm with agents holding inputs: $\{\text{In}_i\}$. If the distribution of $\text{View}_{\mathcal{C}}(\{\text{In}_i\})$ is equivalent to the distribution of $\text{View}_{\mathcal{C}}(\{\text{In}'_i\})$ for all inputs $\{\text{In}_i\}, \{\text{In}'_i\}$ that satisfy

$$\text{In}_i = \text{In}'_i, \forall i \in \mathcal{C}$$

$$\text{Collab}_i(\{\text{In}_k\}) = \text{Collab}_i(\{\text{In}'_k\}), \forall i$$

then the distributed collaboration algorithm preserves *perfect* internal privacy of honest agents $\mathcal{H} = \mathcal{V} \setminus \mathcal{C}$ against \mathcal{C} as an adversary that corrupts agents in \mathcal{C} does not get *any* information about the inputs of the honest agents, other than the collaboration outputs and inputs of the agents in \mathcal{C} . However, perfect internal privacy is not always feasible, especially in cases where the domain of the inputs is unbounded. For such cases we can use other similarity measures, such as relative entropy or KL-Divergence (ref. [8]), to measure similarities between the distributions of $\text{View}_{\mathcal{C}}(\{\text{In}_i\})$ for different inputs $\{\text{In}_i\}$ and quantify the strength of internal privacy of agents in presence of corrupted agents \mathcal{C} .

⁴The proposed privacy protocol introduces randomness in the collaboration algorithm to preserve privacy of agents' inputs.

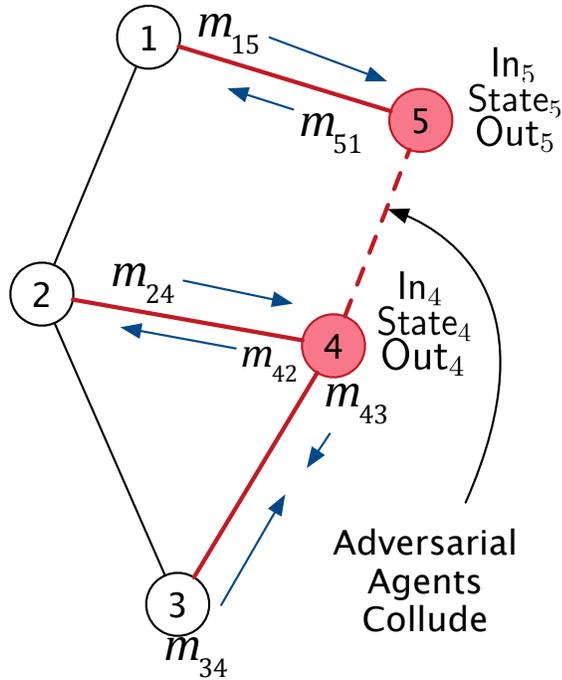


Figure 1.5: In case of internal privacy, the view of an adversary is the collective information learned by all the agents that are corrupted by the adversary during an execution of the distributed collaboration algorithm.

For example, consider the example shown in Fig. 1.5. Here, agents 4 and 5 are adversarial or semi-honest agents, that is $\mathcal{C} = \{4, 5\}$. As shown, view of \mathcal{C} consists of all the messages agents 4 and 5 exchange with the honest neighbors during the execution of the distributed collaboration algorithm: $\{m_{15}, m_{51}, m_{24}, m_{42}, m_{34}, m_{43}\}$ and the information held by them: $\{In_4, In_5, State_4, State_5, Out_4, Out_5\}$. In this particular case, a distributed collaboration algorithm preserves *perfect* internal privacy of honest agents 1, 2 and 3 if distribution of $View_{\mathcal{C}}(\{In_i\})$ is equivalent to the distribution of $View_{\mathcal{C}}(\{In'_i\})$ for all $(\{In_i\}, \{In'_i\})$ that satisfy the constraints: $In_4 = In'_4$, $In_5 = In'_5$ and the outputs being the same.

1.2.2 External Privacy

External privacy is concerned with privacy of all the agents' inputs against a passive adversary that is capable of reading messages that are exchanged between the agents during the intermediate stage of a distributed collaboration algorithm. Such an adversary is commonly referred to as *eavesdropper*. Formally, a distributed collaboration algorithm preserves external privacy of agents' inputs if the entire *view* of the eavesdropper does not leak any (or significant) information about the honest agents' inputs. Here, *view* of the eavesdropper is the information contained in the messages that are exchanged between the agents during an execution of the distributed collaboration algorithm, that is

$$\text{view of an eavesdropper} = \{m_{ij}\}_{\{i,j\} \in \mathcal{E}}$$

Similar to the case of internal privacy, for the purpose of rigorous mathematical analysis of external privacy, we define $\text{View}(\{\text{In}_i\})$ as the random variable denoting the information learned from the messages by the eavesdropper (or the view of the eavesdropper) during an execution of the distributed collaboration algorithm with agents holding inputs: $\{\text{In}_i\}$. Now, if distribution of $\text{View}(\{\text{In}_i\})$ is same for all the inputs in the entire input domain then the distributed collaboration algorithm preserves external privacy *perfectly*. However, perfect external privacy is only possible if the messages that are transmitted by the agents are encrypted using one-time pad as is shown rigorously by Shannon [9]. It is well known that one-time pad is pragmatically infeasible and therefore, most of the modern cryptographic encryptions rely

on certain hard problems⁵ for their security (that is, the privacy of the messages being encrypted). This dissertation investigates information-theoretic guarantee for external privacy that holds regardless of the adversary's computation power. Thus, in this dissertation an alternate (relaxed) definition of external privacy⁶ is used, drawing inspiration from Shannon's perfect secrecy, which requires distribution of $\text{View}(\{\text{In}_i\})$ to be same for all the inputs $\{\text{In}_i\}$ belonging to a *subset* (that can be made to be as large as desirable) of the input domain, instead of the entire input domain. The size of this subset gives a measure on the strength of external privacy and should be tunable for the sake of desirable privacy strength.

For example, consider the example shown in Fig. 1.5. The view of an eavesdropper consists of all the messages exchanged between the agents during an execution of the distributed collaboration algorithm. Let $\{\text{In}_i\} \in \mathcal{D}_i, \forall i \in \{1, \dots, 5\}$. If,

$$\text{View}(\{\text{In}_i\}) = \text{View}(\{\text{In}'_i\})$$

for all $(\{\text{In}_i\}, \{\text{In}'_i\})$ in $\mathcal{D}_1 \times \dots \times \mathcal{D}_5$ (entire inputs domain) then the distributed collaboration algorithm has *perfect* external privacy. Alternately, information-theoretic external privacy can still be guaranteed if there exists a subset $\mathcal{SD}_i \subset \mathcal{D}_i, \forall i$ such that

$$\text{View}(\{\text{In}_i\}) = \text{View}(\{\text{In}'_i\})$$

for all $(\{\text{In}_i\}, \{\text{In}'_i\})$ in $\mathcal{SD}_1 \times \dots \times \mathcal{SD}_5$ and the size of \mathcal{SD}_i could be made as large

⁵Problems that can not be solved efficiently using any existing computing hardware, such as the known decisional composite residuosity problem or the RSA problem.

⁶The details of the definition are contained in Chapter 4.

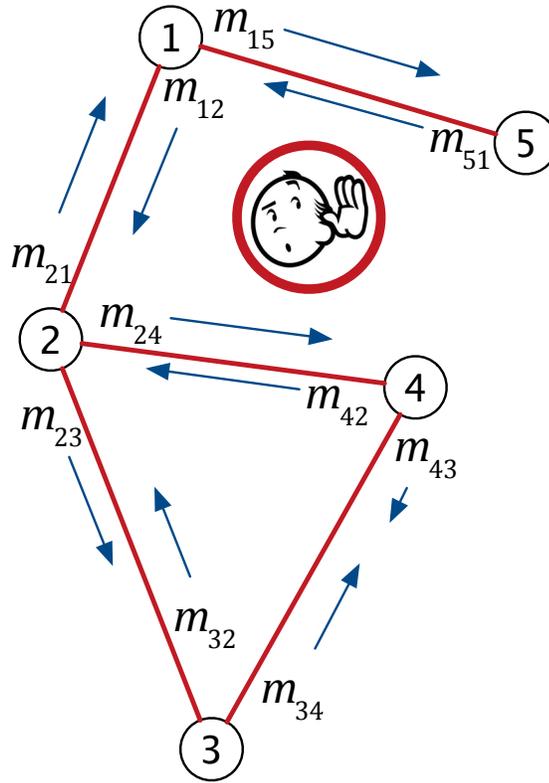


Figure 1.6: In case of external privacy, the view of an adversary is the collective information contained in all the messages that are exchanged between the agents during an execution of the distributed collaboration algorithm.

as desirable for each i .

1.3 Dissertation Contribution

This dissertation proposes and studies a privacy protocol for preserving internal privacy in two particular distributed collaborations: distributed average consensus and distributed optimization. The proposed privacy protocol guarantee internal privacy (in the formal sense as described above) if certain conditions on the communication network are satisfied. The detailed description of the privacy protocol for

the case of distributed average consensus and optimization is given in Chapters 2 and 3, respectively.

Further, this dissertation also proposes and studies a model-based scheme for (information-theoretic) external privacy of agents in certain distributed collaboration algorithms that can be modeled as linear time-invariant (LTI) networked control systems (NCS). The proposed model-based scheme guarantees external privacy (in the formal sense as described above) if certain conditions on the parameters are satisfied. The details of the proposed model-based scheme is given in Chapter 4.

It should be noted that the privacy schemes or protocols proposed in this dissertation do not affect the accuracy of distributed collaboration algorithms and add minimal overhead communication and computation costs on the agents. Detailed comparison of the proposed privacy protocols with existing methods is presented in the following subsections.

1.3.1 Internal Privacy in Distributed Average Consensus

Distributed average consensus is a distributed collaboration between agents in a MAS wherein the inputs of the agents belong to a set of numbers and the collaboration objective for each agent is to compute the average of all the agents' inputs [4–6]. Specifically,

$$\text{Collab}_i(\{\text{In}_k\}) = \frac{1}{n} \sum_k \text{In}_k, \forall i$$

Apart from the obvious application of computing the average or sum of the inputs, other interesting known applications of distributed average consensus include

cooperative control of dynamic systems [10, 11], sensor fusion for distributed state estimation [12], solving economic dispatch problems in power grids [13], distributed support vector machine [14] and anonymous voting in social networks.

Result: The proposed privacy protocol can guarantee *perfect* internal privacy of honest agents' inputs (with input domain of finite size) as long as the adversarial (semi-honest) agents do not constitute a vertex cut⁷ of the communication network.

Comparison with Existing Solutions: While internal privacy in distributed average consensus can often be achieved by relying on generic completeness theorems for (information-theoretic) *secure multi-party computation* (MPC) [7, 15, 16], those results assume a complete network topology with a dedicated communication channel between each pair of agents. In contrast, this dissertation deals with distributed average consensus algorithms that can be used regardless of the network topology. Garay et al. [17] studied secure computation in incomplete networks, and showed that arbitrary functions (including the average) can be computed with information-theoretic privacy against t colluding semi-honest agents so long as the communication network has $(t+1)$ -connectivity⁸. However, their work relies on protocols for secure message transmission [18] that emulate pairwise channels between every pair of agents over an incomplete network. In addition to incurring a significant cost in terms of round and message-complexity, relying on secure message transmission also requires the agents to have complete knowledge of the network topology. The protocol proposed in this dissertation adds minimal cost to existing

⁷Refer Section 1.4 for the definition of vertex cut.

⁸Refer Section 1.4 for the definition of network connectivity.

distributed average consensus protocols, and requires agents to be aware of only their neighboring agents in the MAS. It is nevertheless interesting to observe that the privacy protocol proposed in this dissertation also requires the network to have $(t + 1)$ -connectivity in order to guarantee internal privacy of honest agents against at most t arbitrary adversarial agents.

There have been several proposals [2,19] for achieving *differential privacy*(DP) in distributed average consensus by having agents add local noise to internal states computed during the intermediate stage of a specific distributed average consensus algorithm. This added noise induces a loss in accuracy [19,20]; that is, the agents are only able to compute an *approximation* to the true average (rather than the exact average), where there is an inherent trade-off between privacy and the achievable accuracy. The privacy protocol proposed in this dissertation allows the agents to compute the average of their inputs accurately.

Other works have aimed at ensuring privacy in distributed average consensus without sacrificing accuracy [21,22,24–26]. The protocol proposed in [22] can not preserve internal privacy of an honest agent if any one of its neighbor is adversarial. The protocols in [21,25] can preserve internal privacy of an honest agent only if that agent has an honest neighbor which is not a neighbor of an adversarial agent. The privacy protocol proposed in this dissertation can preserve internal privacy of a group of honest agents if that group of agents is not cut⁹ by the adversarial agents in the communication network. Therefore, an honest agent loses its privacy only if all its neighbors are adversarial, otherwise the proposed privacy protocol is able to

⁹Defined in Section 1.4.

protect its privacy up to a certain extent, as described formally in Section 2.4 of Chapter 2. Unlike the protocols in [21, 25], the proposed privacy protocol requires the agents to mask their inputs, with correlated random values shared with their neighbors, only once before executing a distributed average consensus protocol. Further, the privacy analysis in this dissertation provides a sufficient condition ($t + 1$ network connectivity) for designing the communication network topology wherein internal privacy of all the honest agents can be preserved against at most t arbitrary adversarial agents.

Note that while protocols based on homomorphic encryption [24, 26] can achieve the strong guarantees for internal privacy, they are computationally less efficient than the proposed approach since they rely on complex public-key encryption techniques. Homomorphic encryption based privacy protocols rely on intractability (hardness) of certain decision problems for guaranteeing privacy assuming bounded computation power of the adversary. However, this dissertation investigates information-theoretic (refer. [7]) guarantee for internal privacy which holds regardless of the computation power of the adversary.

It should be noted that some of the above privacy protocols [2, 21, 24, 25] consider synchronicity between agents, whereas the proposed protocol in this dissertation does not require synchronicity between agents. Synchronicity between agents in a distributed collaboration algorithm requires additional effort and is not easy to achieve. Also, note that all of these existing solutions only consider either integral inputs or real-valued inputs, and it should be noted that the input domain plays a critical role in the resultant privacy guarantees of any privacy protocol. For in-

stance, secure multiparty computation and homomorphic-encryption based privacy schemes [17, 24, 26] are only suitable for integral inputs, whereas differential privacy schemes [2, 19, 25] are only suitable for real-valued inputs (trade-off worsens for the case of quantized or integral inputs [23]). Whereas, the proposed privacy protocol in this dissertation can guarantee internal privacy in case of both integral and real-valued agents' inputs.

The proposed privacy protocol is very closely related to the protocol proposed by Emmanuel et al. [27]. However, authors in [27] assume that the communication network topology is complete¹⁰ and only deal with integral inputs. This dissertation considers a more general network topology and also handles the case of real-valued inputs.

	Topology	Accuracy	Comm. Cost	Synchronous	Inputs
Proposed	Not Complete	Preserved	$\Theta(\mathcal{N}_i)$	No	Reals & Integers
MPC	Not Complete	Preserved	$\Theta(t \cdot n)$	No	Integers
DP	Not Complete	Not preserved	$\Theta(\mathcal{N}_i)$	Yes	Reals
CST	Not Complete	Preserved	$\Theta(\mathcal{N}_i)$	Yes	Reals
Emmanuel et al. 2012	Complete	Preserved	$\Theta(n)$	No	Integers

Figure 1.7: Comparison of the proposed protocol with the existing protocols for internal privacy in distributed average consensus.

The comparison of the proposed privacy protocol for internal privacy in distributed average consensus with existing privacy protocols is summarized by the

¹⁰A communication network topology is complete if there exists a direct communication link between every pair of agents.

chart in Fig. 1.7. In the chart above, CST stands for control system theoretic and collectively represents the exiting solutions (ref. [21, 25]) that rely on the notion of unobservability in control system theory for preserving internal privacy by re-designing a particular distributed average consensus protocol that is mathematically equivalent to a linear time-invariant control system. The communication cost is given for a single agent i using the standard ‘big-theta’ notation $\Theta(\cdot)$, where $|\cdot|$ denotes the cardinality of a set, \mathcal{N}_i denotes the set of neighbors of agent i , t is the assumed upper-bound on the total number of corrupted agents and n is the total number of agents in the MAS or network participating in distributed collaboration (distributed average consensus in this case). Note that a distributed collaboration protocol is synchronous if it requires all the agents to operate in a synchronous manner.

1.3.2 Internal Privacy in Distributed Optimization

Distributed optimization is a distributed collaboration of agents in a MAS wherein the agents’ inputs belong to a class of real-valued multi-variate functions and the collaboration objective for each agent is to compute the minima of the aggregate of all the agents’ costs [28–30]. Specifically, $l_n : \mathbb{R}^m \rightarrow \mathbb{R}, \forall i$ and

$$\text{Collab}_i(\{l_n\}) = \arg \min \sum_k l_n, \forall i$$

Some of the interesting known applications of distributed optimization include distributed statistical learning [31, 32], distributed state estimation [33] and formation control of autonomous vehicles [34].

Result: The proposed privacy protocol can guarantee internal privacy of the *affine* parts of honest agents' costs as long as the adversarial (semi-honest) agents do not constitute a vertex cut¹¹ of the communication network.

Comparison with Existing Solutions: Existing methods for preserving privacy in distributed optimization that are based on secure multiparty computation protocols [35–37] require the communication network to be complete. Otherwise, these methods rely on secure message transmission (cf. Dolev, et al. [18]) between agents to emulate a complete network on an incomplete network [17] and require every agent to have prior knowledge of the paths to other agents (for setting up a communication channel between every pair of agents) in the network. The privacy protocol proposed in this dissertation holds regardless of the communication network topology. Moreover, the secure multiparty computation protocols are computationally quite expensive in comparison to the proposed privacy protocol.

Homomorphic encryption based privacy protocols [38, 39] for distributed optimization rely on public-key techniques and are computationally costlier than the proposed privacy protocol. Homomorphic encryption based privacy protocols rely on intractability (hardness) of certain decision problems for guaranteeing privacy assuming bounded computation power of the adversary. However, this dissertation investigates information-theoretic (refer. [7]) guarantee for internal privacy which holds regardless of the computation power of the adversary.

Techniques based on algebraic transformations (AT) have been proposed recently for internal privacy in multi-agent optimization [39–41]. Most of these works

¹¹Refer Section 1.4 for the definition of vertex cut.

are concerned about privacy in linear programming, however there do exist some work for privacy in a more general convex optimization problems [41]. For now, the results in [41] are only given for the centralized case wherein all the agents share their internal states with a central server.

Recently, there have been several proposals for differentially private distributed optimization algorithms [42, 43]. However, differential privacy is impossible to achieve without sacrificing the accuracy of distributed optimization [42]. That is, the agents only reach the approximate minima. The proposed privacy protocol in this dissertation does not affect the accuracy of the optimization problem.

To overcome the inaccuracies in the aforementioned differentially private solutions, Lou et al., [44] proposes a heterogeneous step-size consensus-based sub-gradient algorithm to preserve the privacy of the agents' costs while guaranteeing convergence to the minima. However, [44] has shown convergence of their algorithm only if the individual agents' costs are convex. The proposed privacy protocol in this dissertation does not require individual costs to be convex for its correctness as it relies on a privacy mechanism that masks individual agents' costs (which anyway preserves convexity) before the agents execute any existing distributed optimization protocol, including protocols that do not require agents' costs to be convex for their convergence (refer [45]).

In a closely related work, Gade and Vaidya [46] have proposed a similar privacy approach wherein each agent adds *correlated* random functions to their original costs and use the obtained effective costs for solving the original optimization problem. The addition of correlated random functions preserves the global aggregate cost

and the privacy of honest agents' costs is shown against passively adversarial agents that do not constitute a vertex cut of the communication network. The privacy in [46] is proven using the argument of compatibility of all possible costs that can be held by honest agents' in *view* of the adversarial agents. However, the privacy analysis in [46] does not recognize that distribution and structure of the random functions that are being added to the original costs are part of the adversary's information, as adversarial agents are also following the same prescribed protocol. In this dissertation, the structure and distribution of the random functions that are being added to the original costs are properly defined and assumed known to the adversary. Then, privacy is formally analyzed by measuring indistinguishability of possible honest agents' costs, in view of the adversarial agents, using relative entropy or KL-Divergence¹². The quantitative measure of privacy (or loss of privacy) given in this dissertation provides an insight on the dependence of internal privacy on the vertex expansion (ref. [65, 66]) of the communication network topology.

The comparison of the proposed privacy protocol for internal privacy in distributed optimization with the existing privacy protocols is summarized by the chart in Fig. 1.8. In the chart above, the communication cost is for each agent i using the standard 'big-theta' notation $\Theta(\cdot)$, where $|\cdot|$ denotes the cardinality of a set, \mathcal{N}_i denotes the set of neighbors of agent i , t is the assumed upper-bound on the total number of corrupted agents, m is the dimension of the arguments of the agents' costs and n is the total number of agents participating in distributed collaboration (distributed optimization in this case). Note that a distributed collaboration protocol

¹²Refer [47–49] for further details on the efficacy of KL-Divergence in privacy analysis.

	Topology	Accuracy	Comm. Cost	Synchronous	Remark
Proposed	Not Complete	Preserved	$\Theta(m \mathcal{N}_i)$	No	Privacy of only affine parts
MPC	Not Complete	Preserved	$\Theta(t \cdot mn)$	No	Privacy of entire cost function
DP	Not Complete	Not preserved	$\Theta(m \mathcal{N}_i)$	Yes	Can be made asynchronous
AT	Centralized	-	-	-	Only for Linear Programming
Gade and Vadiya, 2016	Not complete	Preserved	$\Theta(m \mathcal{N}_i)$	No	Privacy is not quantified
Lou, et al., 2017	Not complete	Preserved	$\Theta(m \mathcal{N}_i)$	No	Privacy is not quantified & requires convexity

Figure 1.8: Comparison of the proposed protocol with the existing protocols for internal privacy in distributed optimization.

is synchronous if it requires all the agents to operate in a synchronous manner.

1.3.3 Model-Based Scheme For External Privacy

Certain distributed collaboration algorithms can be modeled as linear time-invariant (LTI) networked control systems (NCS), such as the cooperative control of autonomous vehicles [11, 50], output synchronization of passive systems [51], supervisory control and data acquisition (SCADA) for smart homes or power grids [52, 53] and higher-order consensus [54, 55]. For such distributed collaboration algorithms this dissertation presents a model-based scheme for external privacy of agents' inputs.

Result: The proposed model-based scheme can guarantee external privacy of agents' inputs if the system parameters of the NCS equivalent to the distributed collaboration algorithm are related in a specific manner.

Comparison with Existing Solutions: An obvious solution for securing

the communication links is cryptographic encryption. The agents can simply encrypt their states before sharing them over network during the intermediate stage and guarantee some form of external privacy. However, standard cryptographic encryption schemes require generation and distribution of secret keys amongst the agents. Consequentially, cryptographic encryption schemes require sophisticated key management protocols for the purpose of generating and distributing keys *securely* amongst the agents [56] in order to guarantee privacy of the encrypted messages against eavesdroppers. These key management protocols add significant overhead communication and computation costs on the agents apart from the costs of encryption and decryption. The proposed model-based scheme proposes making of states in a way that preserves the accuracy of the distributed collaboration algorithm. Thus, it does not rely on secret keys for guaranteeing external privacy and saves the cost of key management protocols.

Further, it has been identified by the NIST¹³ that in case of real-time distributed collaboration of dynamic systems, such as formation control of vehicles or synchronization of dynamic systems, the communication delays introduced by standard cryptographic encryption schemes threaten the stability of the collaboration algorithm [57]. The proposed model-based scheme adds minimal communication delays and in fact, provides a gateway to avoid any additional communication delays altogether by appropriately re-designing the collaboration algorithm.

Differentially private mechanisms, wherein agents' obfuscate their states in a distributed collaboration algorithm with noise before sharing them with other

¹³National Institute of Standards and Technology

agents, have been recently investigated as an alternative to cryptographic encryptions for external privacy [2, 58]. Quite understandably, these obfuscations lead to inaccuracies in the final result of the distributed collaboration algorithm. As a result, differentially private mechanisms suffer from trade-off between privacy and accuracy [58, 59]. The model-based privacy scheme proposed in this dissertation preserves the accuracy of the distributed collaboration algorithm and is capable of achieving a desirable *degree of privacy* without any tradeoffs. However, for now the proposed model-based scheme can only guarantee external privacy for a specific class of distributed collaboration algorithms.

The proposed model-based scheme can guarantee external privacy only for those distributed collaboration algorithms that can be modeled as LTI networked control systems, and whose system parameters meet certain requirements. This is an obvious limitation of the proposed model-based privacy scheme when comparing with existing cryptographic encryptions and differentially private mechanisms. For now the effect of asynchronicity between agents has not been considered explicitly for the model-based scheme.

The comparison of the proposed model-based scheme with existing solutions for external privacy in distributed collaboration algorithms is summarized by the chart in Fig. 1.9

	Accuracy	Comm. Delays	Synchronous	Inputs	Secret Keys	Remark
Model-Based Scheme	Preserved	Low	Yes	Real	Not Required	Only for LTI dynamic systems
Cryptography Encryptions	Preserved	High	No	Integer	Required	General approach
Differential Privacy	Not preserved	Low	Yes	Real	Not Required	-

Figure 1.9: Comparison of the proposed model-based scheme with the existing solutions for external privacy in distributed collaboration algorithms.

1.4 Notation and Preliminaries

\mathbb{N} , $\mathbb{Z}_{\geq 0}$, $\mathbb{R}_{>0}$, \mathbb{R}^n and $\mathbb{R}^{n \times m}$ represent the set of natural numbers, non-negative integers, positive real numbers, n -dimensional real-valued vectors and $n \times m$ dimensional real-valued matrices, respectively. Let \mathbb{Z}_q denote the set of integers $\{0, \dots, q-1\}$. If $x \in \mathbb{R}^n$ (or \mathbb{Z}_q^n) then all the elements of x take value in \mathbb{R} (or \mathbb{Z}_q). For a positive integer q , $[q] = \{1, \dots, q\}$.

For any matrix $M \in \mathbb{R}^{m \times n}$, its nullspace is represented by $\mathcal{N}(M) \subset \mathbb{R}^n$. The inverse and generalized inverse of any square matrix M (real-valued elements) are denoted as M^{-1} and M^\dagger , respectively; pseudo-determinant¹⁴ of by $\det^*(M)$. $(\cdot)^T$ denotes the transpose. Notation $Diag(x)$ represents a diagonal matrix with elements of vector x as the diagonal entries. $\mathbf{0}_n$ represents an $n \times n$ matrix with all elements equal to 0.

Let $x^1 \in \mathbb{R}^n$ and $x^2 \in \mathbb{R}^n$, then $[x^1, x^2]$ represents the set of all vectors in \mathbb{R}^n that lie on the line segment joining x^1 and x^2 . A vector $x \in \mathbb{R}^n$ is said to have all

¹⁴Pseudo-determinant of a square real-valued matrix is equal to the product of its non-zero eigenvalues.

non-zero elements if $x[i] \neq 0, \forall i \in [n]$. Notation $frac(x)$ denotes the element-wise fractional part¹⁵ of a real-valued vector x . $1_n(0_n)$ represents a vector of dimension n with all elements equal to 1(0).

For a finite set S , $|S|$ denotes its cardinality; for an integer or real number x , $|x|$ denotes its absolute value.

For a continuous random vector X in $\mathcal{X} \subset \mathbb{R}^n$, its probability density at any point x is denoted by $f_X(x)$ and $Pr(X \in \Gamma)$ denotes the probability of X taking value in a compact set $\Gamma \subset \mathcal{X}$. $f_{X \in \Gamma}(x)$ is the conditional probability density of X , given that X belongs to the compact set Γ . Specifically,

$$f_{X \in \Gamma}(x) = \begin{cases} \frac{f_X(x)}{Pr(X \in \Gamma)} & \forall x \in \Gamma \\ 0 & \text{otherwise} \end{cases}$$

For two random vectors X and Y , the conditional probability density of X at x given Y is denoted by $f_{X|Y}(x|y)$. Let $\mathbb{E}(X)$ the mean of random vector X , the covariance matrix of X is given by $Cov(X) = \mathbb{E}(X^T X)$. (Note that the covariance matrix is always positive semi-definite.) The relative entropy or KL-divergence between two distributions f_X and f'_X over \mathcal{X} is denoted by $D_{KL}(f_X, f'_X)$, which is given as

$$D_{KL}(f_X, f'_X) = \int_{\mathcal{X}} f_X(x) \log \frac{f_X(x)}{f'_X(x)} dx$$

$U(\Gamma)$ represents the uniform probability density over a compact set $\Gamma \subset \mathbb{R}^n$ and $N(\mu, M)$ represents the multivariate normal distribution with mean $\mu \in \mathbb{R}^n$ and covariance $M \in \mathbb{R}^{n \times n}$.

For a discrete random variable X in \mathbb{Z}_q^n , $Pr(X = x)$ denotes its probability at any point $x \in \mathbb{Z}_q^n$.

¹⁵For example, $frac([1.2, 3.4]^T) = [0.2, 0.4]^T$.

1.4.1 Mathematical Model of the Communication Network

The communication network of the agents is represented by simple, undirected graphs. That is, the communication links in a MAS of n agents is modeled via a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where the nodes $\mathcal{V} \triangleq \{1, \dots, n\}$ denote the agents, and there is an edge $\{i, j\} \in \mathcal{E}$ iff there is a direct communication channel between agents i and j . We let \mathcal{N}_i denote the set of neighbors of an agent $i \in \mathcal{V}$, i.e., $j \in \mathcal{N}_i$ if and only if $\{i, j\} \in \mathcal{E}$. (Note that $i \notin \mathcal{N}_i$ since \mathcal{G} is a simple graph.)

We say two agents i, j are *connected* if there is a path from i to j ; since we consider undirected graphs, this notion is symmetric. We let $p_{i,j}$ denote an arbitrary path between i and j , when one exists. A graph \mathcal{G} is *connected* if every distinct pair of nodes is connected; note that a single-node graph is connected.

Definition 1.4.1. (Vertex cut) A set of nodes $\mathcal{V}_{cut} \subset \mathcal{V}$ is a *vertex cut* of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ if removing the nodes in \mathcal{V}_{cut} (and the edges incident to those nodes) renders the resulting graph unconnected. In this case, we say that \mathcal{V}_{cut} *cuts* $\mathcal{V} \setminus \mathcal{V}_{cut}$.

A graph is *k-connected* if the smallest vertex cut of the graph contains k nodes.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph. The *subgraph induced by* $\mathcal{V}' \subset \mathcal{V}$ is the graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ where $\mathcal{E}' \subset \mathcal{E}$ is the set of edges entirely within \mathcal{V}' (i.e., $\mathcal{E}' = \{\{i, j\} \in \mathcal{E} \mid i, j \in \mathcal{V}'\}$). We say a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ has c *connected components* if its vertex set \mathcal{V} can be partitioned into disjoint sets $\mathcal{V}_1, \dots, \mathcal{V}_c$ such that (1) \mathcal{G} has no edges between \mathcal{V}_i and \mathcal{V}_j for $i \neq j$ and (2) for all i , the subgraph induced by \mathcal{V}_i is connected. Clearly, if \mathcal{G} is connected then it has one connected component.

For a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, define its *incidence matrix* $\nabla \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ (see [60]) to be the matrix with $|\mathcal{V}|$ rows and $|\mathcal{E}|$ columns in which

$$\nabla_{i,e} = \begin{cases} 1 & \text{if } e = \{i, j\} \text{ and } i < j \\ -1 & \text{if } e = \{i, j\} \text{ and } i > j \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathbf{1}_n^T \cdot \nabla = 0$. Let $\nabla_{*,e}$ denote the column of ∇ corresponding to the edge $e \in \mathcal{E}$.

Then we have the following result [60, Theorem 8.3.1]:

Lemma 1.4.1. Let \mathcal{G} be an n -node graph with incidence matrix ∇ . Then $\text{rank}(\nabla) = n - c$, where c is the number of connected components of \mathcal{G} .

The graph-Laplacian \mathcal{L} is given as $\mathcal{L} = \nabla \nabla^T$, thus $\text{rank}(\mathcal{L}) = \text{rank}(\nabla)$. Let $\mu_1 \geq \dots \geq \mu_{n-c} > 0$ denote the non-zero eigenvalues of \mathcal{L} . (Eigenvalues of \mathcal{L} are real values because \mathcal{L} is a symmetric real-valued matrix.) From Spectral theorem we can decompose the graph-Laplacian \mathcal{L} as following:

$$\mathcal{L} = \mathcal{U} \text{Diag}([\mu_1, \dots, \mu_{n-c}, 0_c^T]^T) \mathcal{U}^T$$

where, $\mathcal{U} \in \mathbb{R}^{n \times n}$ is a unitary matrix constituting of the orthogonal eigenvectors of \mathcal{L} . For convenience, we denote the smallest non-zero eigenvalue of Laplacian \mathcal{L} as $\underline{\mu}(\mathcal{L})$.

The generalized inverse \mathcal{L}^\dagger of the graph-Laplacian of \mathcal{G} with c connected components is given as following(cf. [61]):

$$\mathcal{L}^\dagger = \mathcal{U} \text{Diag}([1/\mu_1, \dots, 1/\mu_{n-c}, 0_c^T]^T) \mathcal{U}^T \tag{1.1}$$

1.5 Dissertation Organization

Excluding this introductory chapter, the dissertation is divided into 4 chapters. Chapter 2 presents the proposed privacy protocol for internal privacy in distributed average consensus. Chapter 3 presents an extension of the privacy protocol proposed in Chapter 2 for internal privacy in distributed optimization. Chapter 4 is concerned with external privacy and presents the proposed model-based scheme for external privacy in distributed collaboration algorithms. Finally, Chapter 5 summarizes the contribution of this dissertation, outlines its limitations and discusses future research directions.

CHAPTER 2: INTERNAL PRIVACY IN DISTRIBUTED AVERAGE CONSENSUS

This chapter presents a distributed average consensus algorithm that guarantees information-theoretic (perfect) privacy of honest agents’ inputs against a passive adversary that corrupts certain agents (also known as semi-honest agents) in the network, as long as the set of corrupted agents is not a vertex cut of the underlying network topology \mathcal{G} . This implies that for a network with $(t + 1)$ -connectivity, the proposed privacy protocol guarantees information-theoretic privacy of honest agents’ inputs against any t passively adversarial agents. The proposed protocol is composition of a distributed privacy mechanism with any (non-private) distributed average consensus algorithm.

The protocol involves two phases:

1. In the first phase, each agent shares correlated random values with its neighbors and computes a new, “effective input” based on its original input and the random values it shared with its neighbors.
2. In the second phase, the agents run an arbitrary distributed average consensus protocol (e.g., flooding or any other protocol from the literature [4, 5, 62, 63]) using their effective inputs computed in the first phase (rather than their

original inputs) to retrieve the average of their (original) inputs.

The main contribution of this dissertation is the privacy mechanism run by the agents in the first phase, as in the second phase the agents rely on existing distributed average consensus algorithms.

It is shown that the above, two-step process correctly computes the average value of the agents' original inputs as long as the average consensus protocol used in the second phase is correct. This follows from the fact that the first phase is designed to ensure that the average of the agents' effective inputs is equal to the average of their original inputs. The privacy holds in this approach—in a formal sense and under certain conditions, as discussed in this chapter—regardless of the average consensus protocol used in the second phase. This is proved by showing that privacy holds even if all the effective inputs of the honest agents are revealed to the corrupted (or adversarial) agents.

The notion of privacy is information-theoretic, adopted from the literature on secure multi-party computation [7, 16], and holds regardless of the computation power of the adversary. Formally, the guarantee is that the entire view of the adversarial agents throughout the execution of the distributed average consensus protocol can be *simulated* by those agents given (1) their original inputs and (2) the average of the original inputs of the honest agents (or, equivalently, the average of the original inputs of all the agents in the network). This holds regardless of the true inputs of the honest agents. As a consequence, this means that the adversarial agents learn nothing (in statistical sense) about the collective inputs of the honest

agents from an execution of the protocol other than the average of the honest agents' inputs, and this holds regardless of any prior knowledge that the adversarial agents may have about the original inputs of (some of) the honest agents, or the distribution of those inputs. It is rigorously shown that the proposed protocol satisfies this notion of privacy as long as the set of adversarial agents does not constitute a vertex cut of the communication network topology \mathcal{G} .

2.1 Formal Problem Description

Consider a MAS of n agents where the communication network between agents is represented by an undirected, simple, connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$; that is, agents i and j have a direct communication link between them iff $\{i, j\} \in \mathcal{E}$. The communication channel between two nodes is assumed to be both private and authentic; equivalently, in the adversarial model an adversary can not eavesdrop on communications between honest agents, or tamper with their communication. (Alternately, private and authentic communication can be ensured using standard cryptographic techniques.)

Each agent i holds a (private) input, that is $\mathbf{ln}_i = s_i$, where s_i belongs to either a set of bounded integers or a set of bounded real values.

As mentioned before, distributed average consensus is a distributed collaboration protocol that allows the agents in the MAS to each compute the average of their inputs, i.e., after execution of the protocol each agent outputs the value

$$\bar{s} = \frac{1}{n} \cdot \sum_i s_i$$

2.1.1 Adversarial Model

Let $\mathcal{C} \subset \mathcal{V}$ denote the set of adversarial agents that are corrupted by a passive adversary, and let $\mathcal{H} = \mathcal{V} \setminus \mathcal{C}$ denote the remaining honest agents. As stated earlier, the adversary is passive and thus the corrupted agents run the prescribed protocol. Privacy requires that the entire *view* of the adversary—i.e., the inputs of the corrupted agents as well as their internal states and all the protocol messages they received throughout execution of the distributed average consensus protocol—does not leak (significant) information about the original inputs of the honest agents. Note that, by definition, the adversary learns \bar{s} (assuming it corrupts at least one agent) from which it can compute the sum of the inputs of the honest agents, and so the privacy definition requires that the adversary does not learn anything more than this.

2.1.2 Privacy Definition

Let $s_{\mathcal{C}}$ denote a set of inputs held by the corrupted agents, and $s_{\mathcal{H}}$ a set of inputs held by the honest agents. Let $s = [s_1, \dots, s_n]^T$.

Definition 2.1.1. For a distributed average consensus protocol, let $\text{View}_{\mathcal{C}}(s)$ be the random variable denoting the view of the corrupted adversarial agents in an execution of the distributed average consensus protocol where the agents begin holding inputs s .

Then:

Definition 2.1.2. A distributed average consensus protocol is (*perfectly*) \mathcal{C} -private if for all inputs s, s' (in the specified domain) such that $s_{\mathcal{C}} = s'_{\mathcal{C}}$ and $\sum_{i \in \mathcal{H}} s_i = \sum_{i \in \mathcal{H}} s'_i$, the distributions of $\text{View}_{\mathcal{C}}(s)$ and $\text{View}_{\mathcal{C}}(s')$ are identical.

Note that this definition makes sense even if $|\mathcal{C}| = n - 1$, though in that case the definition is vacuous since $s_{\mathcal{H}} = \sum_{i \in \mathcal{H}} s_i$ and so revealing the sum of the honest agents' inputs reveals the (single) honest agent's input.

An alternate, perhaps more natural, way to define privacy is to require that for any distribution S (known to the adversary) over the honest agents' inputs, the distribution of the honest agents' inputs conditioned on the adversary's view is identical to the distribution of the honest agents' inputs conditioned on their sum. It is not hard to see that this is equivalent to the above definition.

2.1.3 Assumptions

Assumptions made in this chapter are the following:

- (A1) The communication network \mathcal{G} is undirected in the first phase.
- (A2) Agents know the value of n .
- (A3) Inputs of all the agents belong to a common set of numbers; either bounded integers or real-values.
- (A4) The adversarial agents are passive, that is they obey the prescribed average consensus protocol without causing any disruptions.

2.2 Proposed Protocol

As described previously, the proposed protocol has a two-phase structure (as shown in Fig. 2.1). In the first phase, each agent i computes an “effective input” \tilde{s}_i based on its original input s_i and random values it shared with its neighbors; this is done while ensuring that $\sum_i s_i$ can be precisely retrieved from the value of $\sum_i \tilde{s}_i$. In the second phase, the agents use any (correct) distributed average consensus protocol Π to compute $\sum_i \tilde{s}_i$ (given by the average of $\{n\tilde{s}_i\}$), reduce that result to $\sum_i s_i$ by using a standard mathematical operation (specified later), and then divide by n . This clearly gives the correct average $\frac{1}{n} \cdot \sum_i s_i$, and thus all that remains is to analyze privacy.

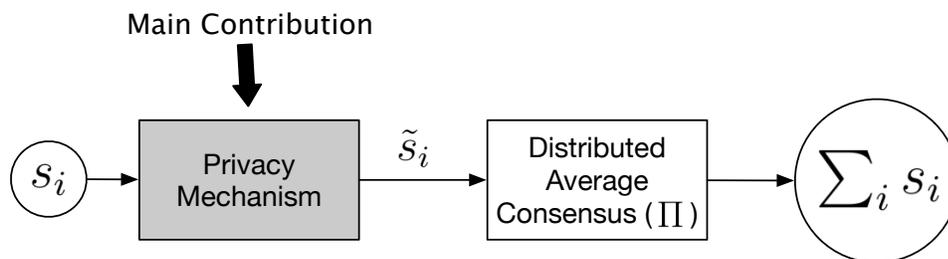


Figure 2.1: Proposed protocol for internal privacy in distributed average consensus.

It may at first seem strange that privacy of the proposed algorithm can be proved without knowing anything about the distributed average consensus protocol Π used in the second phase of the algorithm. This is done by making a “worst-case” assumption about Π , namely, that it simply reveals all the agents’ inputs to all the agents. Such an algorithm is, of course, not at all private; however, for our purposes, this does not immediately violate privacy because Π is run on the agents’ *effective*

inputs $\{\tilde{s}_i\}$ rather than their true inputs $\{s_i\}$.

From now on, then, let the view of the adversary consist of the initial inputs of the corrupted agents, their internal states and all the protocol messages they receive throughout execution of the first phase of the protocol, and the vector $\tilde{s} = [\tilde{s}_1, \dots, \tilde{s}_n]^T$ of all agents' effective inputs at the end of the first phase. The definition of privacy (cf. Definition 2.1.2) remains unchanged.

Before continuing with an analysis of privacy, first-phase algorithm is described as following for both class of inputs; bounded integers and bounded real-values.

2.2.1 Bounded Integral Inputs

The inputs $s_i \in \{0, \dots, q - 1\}$ for some publicly known, integer bound $q > 1$ and let p be an integer such that $p > n \cdot (q - 1) \geq \sum_i s_i$.

The **first phase** of the protocol proceeds as follows:

1. Each agent $i \in \mathcal{V}$ chooses independent, uniform values $r_{ij} \in \mathbb{Z}_p$ for all $j \in \mathcal{N}_i$, and sends r_{ij} to agent j .
2. Each agent $i \in \mathcal{V}$ computes a mask

$$a_i = \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) \bmod p, \quad (2.1)$$

where $a_i \in \mathbb{Z}_p$.

3. Each agent $i \in \mathcal{V}$ computes effective input

$$\tilde{s}_i = (s_i + a_i) \bmod p. \quad (2.2)$$

Note that

$$\sum_i \tilde{s}_i = \sum_i s_i + \sum_i a_i \bmod p.$$

Moreover,

$$\begin{aligned} \sum_i a_i &= \sum_i \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) \bmod p \\ &= 0 \bmod p, \end{aligned}$$

since \mathcal{G} is undirected. Thus, $\sum_i \tilde{s}_i = \sum_i s_i \bmod p$. Since $\sum_i s_i < p$ by choice of p , this implies that $\sum_i \tilde{s}_i \bmod p$ is equal to $\sum_i s_i$ over the integers, and hence correctness of overall algorithm (i.e., including the second phase) follows.

2.2.2 Bounded Real-Valued Inputs

By scaling appropriately¹, the inputs $s_i \in [0, 1/n)$, where n is the total number of agents in the network.

1. Each agent $i \in \mathcal{V}$ chooses independent, uniform values $r_{ij} \in [0, 1)$ for all $j \in \mathcal{N}_i$, and sends r_{ij} to agent j .
2. Each agent $i \in \mathcal{V}$ computes a mask

$$a_i = \text{frac} \left(\sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) \right), \quad (2.3)$$

where $a_i \in [0, 1)$.

3. Each agent $i \in \mathcal{V}$ computes effective input

$$\tilde{s}_i = \text{frac}(s_i + a_i). \quad (2.4)$$

¹Let each agent be associated with a finite (bounded) real-valued input $x_i \in [0, q)$, where q is a positive real value. Then, $s_i = x_i/nq \in [0, 1/n)$.

Note that²

$$\text{frac} \left(\sum_i \tilde{s}_i \right) = \sum_i s_i + \text{frac} \left(\sum_i a_i \right).$$

Moreover,

$$\text{frac} \left(\sum_i a_i \right) = \text{frac} \left(\sum_i \sum_{j \in N_i} (r_{ji} - r_{ij}) \right) = 0$$

since \mathcal{G} is undirected. Thus, $\text{frac}(\sum_i \tilde{s}_i) = \sum_i s_i$. Since $\sum_i s_i < 1$ (as $s_i \in [0, 1)$, this implies that $\text{frac}(\sum_i \tilde{s}_i)$ is equal to $\sum_i s_i$, and hence correctness of overall algorithm (i.e., including the second phase) follows.

The algorithm, for both the cases, is illustrated by an example in Section 2.3.

Note: In the above proposed protocols, the agents are not required to send $\{r_{ij}\}$ simultaneously and the delay between transmission and reception of these random values does not affect the protocol in any way as long as the agents are able to transmit these values successfully. The second phase can start once agents have successfully exchanged values of $\{r_{ij}\}$ with their neighbors in the network \mathcal{G} . Therefore, phase one does not require synchronicity between the agents, and an asynchronous Π (refer [5, 6]) renders the overall two-phase protocol asynchronous.

2.2.3 Privacy Analysis

It is shown here that \mathcal{C} -privacy holds as long as \mathcal{C} is not a vertex cut of \mathcal{G} .

The following arguments are given for the case of integral inputs. However, owing to the similarities between the mod operation and the fractional part³ the

² $\text{frac}(x + y) = \text{frac}(\text{frac}(x) + \text{frac}(y))$ and $\text{frac}(-x) = 1 - \text{frac}(x)$.

³Fractional part is essentially a mod operation for reals.

arguments given for proving the privacy result for the case of integral inputs extend readily for the case of real-valued inputs.

For an edge $e = \{i, j\}$ in the graph with $i < j$, define⁴

$$b_e = r_{ji} - r_{ij} \bmod p.$$

Let $b = [b_{e_1}, \dots]$ be the collection of such values for all the edges in \mathcal{G} . If we let $a = [a_1, \dots, a_n]^T$ denote the masks used by the agents, then we have⁵

$$a = \nabla \cdot b \bmod p.$$

Since the values $\{r_{ij}\}$ are uniform and independent in \mathbb{Z}_p , it is easy to see that the values $\{b_e\}_{e \in \mathcal{E}}$ are uniform and independent in \mathbb{Z}_p as well⁶. Thus, a is uniformly distributed over the vectors in the span (over \mathbb{Z}_p) of the columns of ∇ , which is denoted by $L(\nabla)$ and formally given as

$$L(\nabla) = \{\nabla \cdot b \bmod p \mid b \in \mathbb{Z}_p^{|\mathcal{E}|}\}$$

The following is easy to prove using the fact that $\text{rank}(\nabla) = n - 1$ when \mathcal{G} is connected (cf. Lemma 1.4.1):

Lemma 2.2.1. If \mathcal{G} is connected then a is uniformly distributed over all points in \mathbb{Z}_p^n subject to the constraint $\sum_i a_i = 0 \bmod p$.

⁴For case of reals, we have $b_e = \text{frac}(r_{ji} - r_{ij})$

⁵For case of reals, this reduces to $a = \text{frac}(\nabla \cdot b)$.

⁶If x and y are two independent random variables in \mathbb{Z}_p with at least one of them being uniformly distributed (in \mathbb{Z}_p), then $z = x + y \bmod p$ is uniformly distributed in \mathbb{Z}_p . Similarly, if x and y are two independent random variables in $[0, 1)$ with at least one of them being uniformly distributed (in $[0, 1)$), then $z = \text{frac}(x + y)$ is uniformly distributed in $[0, 1)$.

Proof. The proof is obvious for $n = 1$. Henceforth, $n > 1$.

Choose a subset \mathcal{E}' of \mathcal{E} with $n - 1$ edges such that $\mathcal{G}' = \{\mathcal{V}, \mathcal{E}'\}$ is connected (such a subset \mathcal{E}' is guaranteed to exist if \mathcal{G} is connected). Therefore, all the $n - 1$ columns of ∇' (incidence matrix of \mathcal{G}') are linearly independent. Combining this with the fact that all non-zero values in ∇' are either -1 or 1 implies,

$$a' = \nabla' \cdot b = \sum_{e \in \mathcal{E}'} \nabla'_{*,e} \cdot b_e \pmod{p}$$

is uniformly distributed over p^{n-1} points in

$$L(\nabla') = \{\nabla' \cdot b \pmod{p} \mid b \in \mathbb{Z}_p^{n-1}\},$$

as $\{b_e\}_{e \in \mathcal{E}'}$ are uniformly distributed in \mathbb{Z}_p^{n-1} .

Note that $1_n^T \cdot a' = 0 \pmod{p}$ (as $1_n^T \nabla' = 0_{|\mathcal{E}'|}^T$ for \mathcal{G}' being undirected). Therefore, the above result implies that a' is uniformly distributed over \mathbb{Z}_p^n subject to the constraint that $\sum_i a'_i = 0 \pmod{p}$.

The rest of the proof follows from induction of edges.

In case $\mathcal{E}' = \mathcal{E}$ (or \mathcal{E} has only $n - 1$ edges), the proof concludes here as $a = a'$. Otherwise, choose an edge e' from the set of remaining edges $\mathcal{E}' \setminus \mathcal{E}$ and let $\mathcal{E}'' = \mathcal{E}' \cup \{e'\}$. Note that as \mathcal{G}' is connected, therefore $\nabla_{*,e'}$ (column of ∇ corresponding to edge e') can be obtained by linearly combining the columns of ∇' as following⁷

$$\nabla_{*,e'} = \sum_{e \in \mathcal{E}'} \mu_e \nabla'_{*,e} \tag{2.5}$$

⁷It follows from the fact that there exists a path in \mathcal{G}' between the terminal nodes of the edge e' as \mathcal{G}' is connected.

where, $\mu_e \in \{-1, 0, 1\}$ for all $e \in \mathcal{E}'$.

From (2.5), each point a'' of $L(\nabla'')$ (span of the columns of the incidence matrix ∇'' of $\mathcal{G}'' = \{\mathcal{V}, \mathcal{E}''\}$ over \mathbb{Z}_p) takes the following form

$$\begin{aligned} a'' &= \sum_{e \in \mathcal{E}'} \nabla'_{*,e} \cdot b_e + \nabla'_{*,e'} \cdot b_{e'} \pmod{p} \\ &= \sum_{e \in \mathcal{E}'} \nabla'_{*,e} \cdot (b_e + \mu_e b_{e'}) \pmod{p} \end{aligned}$$

As $\{b_e\}_{e \in \mathcal{E}'}$ is uniformly distributed over all point in \mathbb{Z}_p^{n-1} and b_e is independent from all $\{b_e\}_{e \in \mathcal{E}'}$, therefore $\{b_e + \mu_e b_{e'} \pmod{p}\}_{e \in \mathcal{E}'}$ is uniformly distributed over all the points in \mathbb{Z}_p^{n-1} . This implies, a'' is uniformly distributed over p^{n-1} points in $L(\nabla'')$.

Induction of edges in this manner leads to the conclusion that a is uniformly distributed over p^{n-1} points in $L(\nabla)$

Note that $1_n^T \cdot a = 0 \pmod{p}$ as $1_n^T \nabla = 0_{|\mathcal{E}|}^T$ (due to the fact that \mathcal{G} is undirected).

Therefore, the above result implies that a is uniformly distributed over \mathbb{Z}_p^n subject to the constraint that $\sum_i a_i = 0 \pmod{p}$. ■

The above lemma when combined with the fact that $\tilde{s}_i = s_i + a_i \pmod{p}$ yields the following result.

Lemma 2.2.2. If \mathcal{G} is connected, then the effective inputs \tilde{s} are uniformly distributed in \mathbb{Z}_p^n subject to the constraint that $\sum_i \tilde{s}_i = \sum_i s_i \pmod{p}$, given the inputs $\{s_i\}$.

Proof.

Since $\tilde{s}_i = s_i + a_i \pmod p$ and s_i, a_i are independent, we get

$$Pr(\tilde{s}|s) = Pr(a = (\tilde{s} - s) \pmod p)$$

From Lemma 2.2.1 we know that⁸

$$Pr(a) = \begin{cases} 1/p^{n-1} & , \quad \sum_i a_i = 0 \pmod p \\ 0 & , \quad \text{o.w.} \end{cases}$$

when \mathcal{G} is connected.

Therefore,

$$Pr(\tilde{s}|s) = \begin{cases} 1/p^{n-1} & , \quad \sum_i \tilde{s}_i = \sum_i s_i \pmod p \\ 0 & , \quad \text{o.w.} \end{cases}$$

when \mathcal{G} is connected. ■

The above Lemma implies privacy for the case when $\mathcal{C} = \emptyset$, i.e., when there are no corrupted agents. In that case, the view of the adversary consists only of the effective inputs \tilde{s} , and Lemma 2.2.2 shows that the distribution of those values depends only on the sum of the agents' true inputs. Below, this line of argument is extended to the case of nonempty \mathcal{C} .

Fix some set \mathcal{C} of corrupted agents, and recall that $\mathcal{H} = \mathcal{V} \setminus \mathcal{C}$. Let $\mathcal{E}_{\mathcal{C}}$ denote the set of edges incident to \mathcal{C} , and let $\mathcal{E}_{\mathcal{H}} = \mathcal{E} \setminus \mathcal{E}_{\mathcal{C}}$ be the edges incident only to honest agents. Note that now the adversary's view contains (information that allows it to compute) $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$ in addition to the honest agents' effective inputs $\{\tilde{s}_i\}_{i \in \mathcal{H}}$.

The key observation enabling a proof of privacy is that the values $\{b_e\}_{e \in \mathcal{E}_{\mathcal{H}}}$ are uniform and independent in \mathbb{Z}_p *even conditioned on the values of* $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$. Thus,

⁸Total number of points in \mathbb{Z}_p^n whose elements add up to 0 under mod p are p^{n-1} .

as long as \mathcal{C} is not a vertex cut of \mathcal{G} , an argument as earlier implies that the masks $\{a_i\}_{i \in \mathcal{H}}$ are uniformly distributed in $\mathbb{Z}_p^{|\mathcal{H}|}$ subject to $\sum_{i \in \mathcal{H}} a_i = -\sum_{i \in \mathcal{C}} a_i \pmod p$ (conditioned on knowledge of the values $\{b_e\}_{e \in \mathcal{E}_\mathcal{C}}$), and hence the effective inputs $\{\tilde{s}_i\}_{i \in \mathcal{H}}$ are uniformly distributed in $\mathbb{Z}_p^{|\mathcal{H}|}$ subject to

$$\sum_{i \in \mathcal{H}} \tilde{s}_i = \sum_{i \in \mathcal{V}} s_i - \sum_{i \in \mathcal{C}} \tilde{s}_i \pmod p$$

given the values of $\{s_i\}$ and $\{\tilde{s}_i\}_{i \in \mathcal{C}}$ (again, even conditioned on knowledge of the $\{b_e\}_{e \in \mathcal{E}_\mathcal{C}}$). Since the right-hand side of the above equation can be computed from the effective inputs of the corrupted agents, the $\{b_e\}_{e \in \mathcal{E}_\mathcal{C}}$, and the sum of the honest agents' inputs, this implies:

Theorem 2.2.1. If \mathcal{C} is not a vertex cut of \mathcal{G} , then the proposed distributed average consensus protocol is perfectly \mathcal{C} -private.

Proof. Let $\mathcal{G}_\mathcal{H} = \{\mathcal{H}, \mathcal{E}_\mathcal{H}\}$ be the graph of honest agents (and edges incident to only honest agents) and $\nabla_\mathcal{H}$ be its *incidence matrix*.

Note that

$$\text{View}_\mathcal{C}(s) = \{\{s_i\}_{i \in \mathcal{C}}, \{\tilde{s}_i\}, \{b_e\}_{e \in \mathcal{E}_\mathcal{C}}\}.$$

Therefore, to prove \mathcal{C} -privacy it suffices to show that the joint distribution of $\tilde{s}_\mathcal{H}$ and $\{b_e\}_{e \in \mathcal{E}_\mathcal{C}}$ remains the same for any two sets of true inputs s, s' (in \mathbb{Z}_p), that satisfy $s_\mathcal{C} = s'_\mathcal{C}$ and $\sum_{i \in \mathcal{V}} s_i = \sum_{i \in \mathcal{V}} s'_i$, when $\mathcal{G}_\mathcal{H}$ is connected or \mathcal{C} is not a vertex cut of \mathcal{G} .

Each a_i can be decomposed as

$$a_i = \sum_{e \in \mathcal{E}_\mathcal{H}} \nabla_{i,e} \cdot b_e + \sum_{e \in \mathcal{E}_\mathcal{C}} \nabla_{i,e} \cdot b_e \pmod p$$

Note that the values $\{b_e\}_{e \in \mathcal{E}_\mathcal{H}}$ are uniformly and independently distributed in \mathbb{Z}_p (given the values $\{b_e\}_{e \in \mathcal{E}_\mathcal{C}}$).

From Lemma 2.2.1, the values $\{\sum_{e \in \mathcal{E}_\mathcal{H}} \nabla_{i,e} b_e \bmod p\}_{i \in \mathcal{H}}$ are uniformly distributed over $\mathbb{Z}_p^{|\mathcal{H}|}$ subject to the constraint $\sum_{i \in \mathcal{H}} (\sum_{e \in \mathcal{E}_\mathcal{H}} \nabla_{i,e} b_e) = 0 \bmod p$ when $\mathcal{G}_\mathcal{H}$ is connected.

Therefore, if $\mathcal{G}_\mathcal{H}$ is connected then

$$Pr(a_\mathcal{H} | \{b_e\}_{e \in \mathcal{E}_\mathcal{C}}) = \begin{cases} 1/p^{|\mathcal{H}|-1} & , \quad \sum_{i \in \mathcal{H}} a_i = -\sum_{i \in \mathcal{C}} a_i \bmod p \\ 0 & , \quad \text{o.w.} \end{cases}$$

where, $a_i = \sum_{e \in \mathcal{E}_\mathcal{C}} \nabla_{i,e} \cdot b_e \bmod p$, $\forall i \in \mathcal{C}$ and $a_\mathcal{H}$ denotes the vector of honest agents masks $\{a_i\}_{i \in \mathcal{H}}$.

Combining the above with Lemma 2.2.2 implies, ($\tilde{s}_\mathcal{H}$ is the vector of the effective inputs of the honest agents $\{\tilde{s}_i\}_{i \in \mathcal{H}}$)

$$Pr(\tilde{s}_\mathcal{H} | s_\mathcal{H}, \{b_e\}_{e \in \mathcal{E}_\mathcal{C}}) = 1/p^{|\mathcal{H}|-1} \quad (2.6)$$

for all the values $\tilde{s}_\mathcal{H}$ in $\mathbb{Z}_p^{|\mathcal{H}|}$ that satisfy

$$\sum_{i \in \mathcal{H}} \tilde{s}_i = \sum_{i \in \mathcal{H}} s_i - \sum_{i \in \mathcal{C}} a_i \bmod p$$

when $\mathcal{G}_\mathcal{H}$ is connected. ($a_i = \sum_{e \in \mathcal{E}_\mathcal{C}} \nabla_{i,e} b_e \bmod p$ for every $i \in \mathcal{C}$.)

Combining (2.6) with the fact that all the values of $\{b_e\}_{e \in \mathcal{E}}$ are independent to the inputs $\{s_i\}$ implies,

$$Pr(\tilde{s}_\mathcal{H}, \{b_e\}_{e \in \mathcal{E}_\mathcal{C}} | s_\mathcal{H}) \equiv Pr(\tilde{s}_\mathcal{H}, \{b_e\}_{e \in \mathcal{E}_\mathcal{C}} | s'_\mathcal{H})$$

for any two sets of original inputs $s_\mathcal{H}, s'_\mathcal{H}$ in $\mathbb{Z}_p^{|\mathcal{H}|}$ that satisfy $\sum_{i \in \mathcal{H}} s_i = \sum_{i \in \mathcal{H}} s'_i$.

Hence, if $\mathcal{G}_\mathcal{H}$ is connected then for an execution of the proposed distributed average

consensus protocol,

$$Pr(\text{View}_{\mathcal{C}}(s)) \equiv Pr(\text{View}_{\mathcal{C}}(s'))$$

for all s, s' in \mathbb{Z}_p^n that satisfy $s_{\mathcal{C}} = s'_{\mathcal{C}}$ and $\sum_{i \in \mathcal{V}} s_i = \sum_{i \in \mathcal{V}} s'_i$. ■

As a corollary, we have

Corollary 2.2.1. If \mathcal{G} is $(t+1)$ -connected, then for any \mathcal{C} with $|\mathcal{C}| \leq t$ the proposed distributed average consensus protocol is perfectly \mathcal{C} -private.

Note: The results states in Theorem 2.2.1 and Corollary 2.2.1 hold verbatim for the case of bounded real-valued inputs, and can be formally shown using similar arguments as above. The only difference is that for the case of bounded real-valued inputs the analysis involves fractional parts instead of modulus operations.

2.3 Illustration

For demonstration of the proposed distributed average consensus protocol consider a simple network of 3 agents with $\mathcal{V} = \{1, 2, 3\}$ and $\mathcal{E} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, as shown in Fig. 2.2.

2.3.1 Bounded Integral Inputs

Let the values of q and p be 10 and 30, respectively.

Consider an instance where $s_1 = 4$, $s_2 = 7$ and $s_3 = 3$.

In first phase, the agents execute the following steps

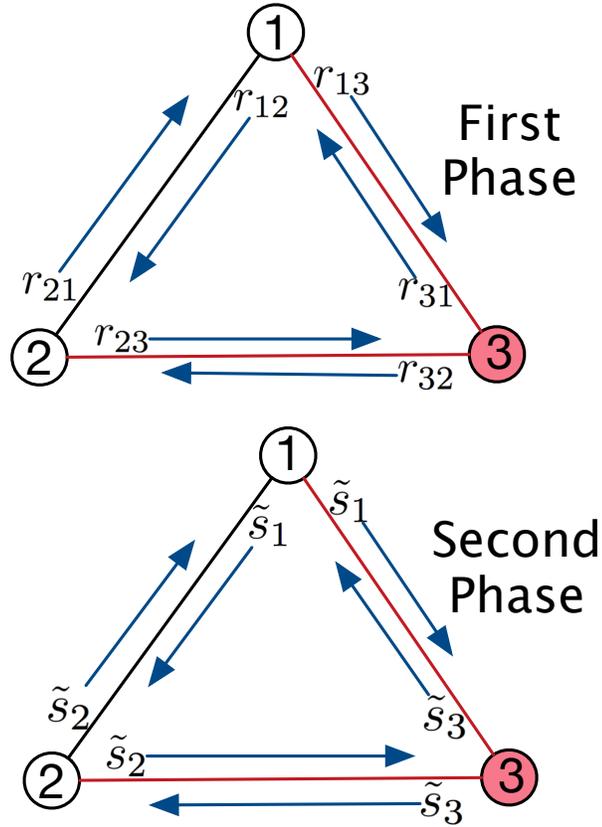


Figure 2.2: Illustration of the proposed distributed average consensus protocol. Arrows (in blue) indicate the flow of information over the communication links.

1. As shown in Fig. 2.2, all pair of adjacent agents i and j exchange the respective values of r_{ij} and r_{ji} (chosen independently and uniformly in \mathbb{Z}_p) with each other. Consider a particular instance where

$$r_{12} = 14, r_{21} = 11, r_{23} = 17, r_{32} = 5, r_{31} = 3, r_{13} = 8$$

2. The agents compute their respective masks,

$$\begin{aligned} a_1 &= ((r_{21} - r_{12}) + (r_{31} - r_{13})) \bmod p \\ &= ((11 - 14) + (3 - 8)) \bmod 30 = 22 \end{aligned}$$

$$\begin{aligned} a_2 &= ((r_{12} - r_{21}) + (r_{32} - r_{23})) \bmod p \\ &= ((14 - 11) + (5 - 17)) \bmod 30 = 21 \end{aligned}$$

$$\begin{aligned} a_3 &= ((r_{13} - r_{31}) + (r_{23} - r_{32})) \bmod p \\ &= ((8 - 3) + (17 - 5)) \bmod 30 = 17 \end{aligned}$$

(one can verify that $(a_1 + a_2 + a_3) \bmod 30 = 0$)

3. The agents compute their respective effective inputs,

$$\tilde{s}_1 = (s_1 + a_1) \bmod p = (4 + 22) \bmod 30 = 26$$

$$\tilde{s}_2 = (s_2 + a_2) \bmod p = (7 + 21) \bmod 30 = 28$$

$$\tilde{s}_3 = (s_3 + a_3) \bmod p = (3 + 17) \bmod 30 = 20$$

After the first phase, each agent uses a (non-private) distributed average consensus protocol Π in the second phase to compute $(1/3) \sum_i \tilde{s}_i$ (it can be easily verified that $\sum_i \tilde{s}_i \bmod 30 = \sum_i s_i = 14$).

Let $\mathcal{C} = \{3\}$ and so, $\mathcal{E}_{\mathcal{C}} = \{\{1, 3\}, \{2, 3\}\}$. It is easy to see that \mathcal{C} does not cut the graph \mathcal{G} and therefore, for any pair of inputs $s_1 \in \mathbb{Z}_{10}$ and $s_2 \in \mathbb{Z}_{10}$ that satisfy $s_1 + s_2 = 11$, \tilde{s}_1 and \tilde{s}_2 are uniformly distributed over \mathbb{Z}_{30}^2 subject to $\tilde{s}_1 + \tilde{s}_2 = 24 \bmod 30$.

2.3.2 Bounded Real-Valued Inputs

Let $s_1 = 0.1$, $s_2 = 0.2$ and $s_3 = 0.15$.

In first phase, the agents execute the following steps

1. As shown in Fig. 2.2, all pair of adjacent agents i and j exchange the respective values of r_{ij} and r_{ji} (chosen independently and uniformly in $[0, 1)$) with each other. Consider a particular instance where: $r_{12} = 0.1$, $r_{21} = 0.5$, $r_{23} = 0.7$, $r_{32} = 0.4$, $r_{31} = 0.3$ and $r_{13} = 0.8$.
2. The agents compute their respective masks,

$$a_1 = \text{frac}((r_{21} - r_{12}) + (r_{31} - r_{13})) = 0.9$$

Similarly, $a_2 = 0.3$ and $a_3 = 0.8$. (One can verify that $\text{frac}(a_1 + a_2 + a_3) = 0$.)

3. The agents compute their respective effective inputs,

$$\tilde{s}_1 = \text{frac}(s_1 + a_1) = \text{frac}(0.1 + 0.9) = 0.0$$

Similarly, $\tilde{s}_2 = 0.5$ and $\tilde{s}_3 = 0.95$.

After the first phase, each agent uses a (non-private) distributed average consensus protocol Π in the second phase to compute $(1/3) \sum_i \tilde{s}_i$ (it can be easily verified that $\text{frac}(\sum_i \tilde{s}_i) = \sum_i s_i = 0.45$).

Let $\mathcal{C} = \{3\}$ and so, $\mathcal{E}_{\mathcal{C}} = \{\{1, 3\}, \{2, 3\}\}$. It is easy to see that \mathcal{C} does not cut the graph \mathcal{G} and therefore, for any pair of inputs $s_1 \in [0, 1/3)$ and $s_2 \in [0, 1/3)$ that satisfy $s_1 + s_2 = .3$ the joint distribution of \tilde{s}_1 and \tilde{s}_2 is uniform over $[0, 1)^2$ such that $\text{frac}(\tilde{s}_1 + \tilde{s}_2) = 0.5$.

2.4 Extension

The privacy guarantee given by Theorem 2.2.1 can be easily extended for the case when \mathcal{C} is indeed a vertex cut by relaxing our definition of \mathcal{C} -privacy to $(\mathcal{C}, \mathcal{H})$ -privacy as following. Let set \mathcal{H} now represent a subset of $\mathcal{V} \setminus \mathcal{C}$, that is, now we are interested in privacy of only some of the honest agents instead of all the honest agents. Then, we can re-define privacy as following.

Definition 2.4.1. A distributed average consensus protocol is (*perfectly*) $(\mathcal{C}, \mathcal{H})$ -private if for all s, s' such that $s_{\mathcal{V} \setminus \mathcal{H}} = s'_{\mathcal{V} \setminus \mathcal{H}}$ and $\sum_{i \in \mathcal{H}} s_i = \sum_{i \in \mathcal{H}} s'_i$, the distributions of $\text{View}_{\mathcal{C}}(s)$ and $\text{View}_{\mathcal{C}}(s')$ are identical.

Similarly as above, this definition makes sense even if $|\mathcal{H}| = 1$, though in that case the definition is vacuous since $s_{\mathcal{H}} = \sum_{i \in \mathcal{H}} s_i$ and so revealing the sum of the inputs of the honest agents \mathcal{H} reveals the (single) honest agent's input!

Essentially, if the *view* of the adversary remains same for any two sets of inputs of the honest agents \mathcal{H} that sum up to the same value then no information is leaked to the adversary about the inputs of \mathcal{H} other than their common sum. Note that \mathcal{C} -privacy is equivalent $(\mathcal{C}, \mathcal{V} \setminus \mathcal{C})$ -privacy. Therefore, if a distributed average consensus protocol is \mathcal{C} -privacy then it is $(\mathcal{C}, \mathcal{H})$ -private for all $\mathcal{H} \subseteq \mathcal{V} \setminus \mathcal{C}$. (This is the reason why Definition 2.4.1 is a relaxation of Definition 2.1.2.)

From Theorem 2.2.1 we infer that -

Theorem 2.4.1. The proposed distributed average consensus protocol is $(\mathcal{C}, \mathcal{H})$ -private if \mathcal{C} does not *cut* \mathcal{H} .

Alternately, if the set of agents \mathcal{H} are connected in the sub-graph $\overline{G}_{\mathcal{C}} = \mathcal{G} \setminus \{\mathcal{C}, \mathcal{E}_{\mathcal{C}}\}$ then the proposed distributed average consensus protocol is $(\mathcal{C}, \mathcal{H})$ -private.

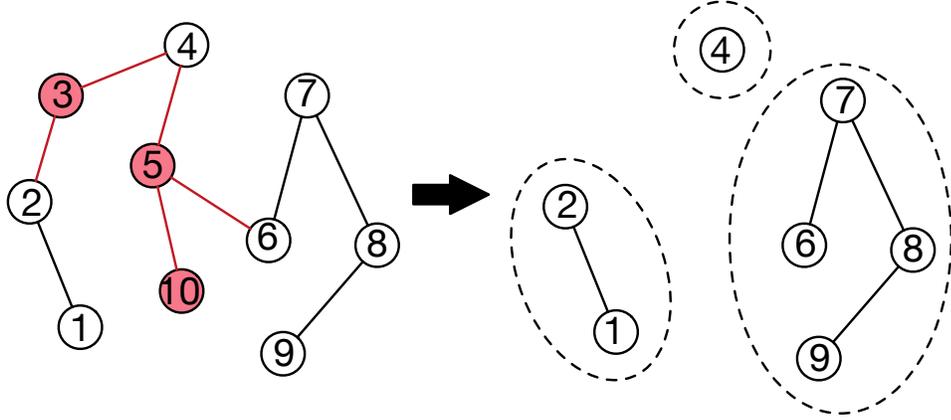


Figure 2.3: An example illustrating the internal privacy guarantee in the proposed distributed average consensus protocol.

For example, consider the scenario shown in Fig. 2.3, where the corrupted agents $\mathcal{C} = \{3, 5, 10\}$ cut the graph (communication network) into 3 connected components with set of agents $\mathcal{H}_1 = \{1, 2\}$, $\mathcal{H}_2 = \{4\}$ and $\mathcal{H}_3 = \{6, 7, 8, 9\}$ (and edges incident to the respective honest agents). As mentioned before, the proposed distributed protocol preserves the privacy of each group of honest agents $\mathcal{H}_i, i = 1, 2, 3$ in the sense of Definition 2.4.1. However, as $|\mathcal{H}_2| = 1$ the value of s_4 gets revealed to the adversary.

2.5 Concluding Remarks

The proposed two-phase approach for distributed average consensus guarantees internal privacy (perfectly) of honest agents if the passively adversarial agents

(or agents corrupted by a passive adversary) do not form a vertex cut in the underlying communication network \mathcal{G} . This implies that if the network has $(t + 1)$ -connectivity then the proposed distributed average consensus protocol preserves internal privacy of honest agents in presence of at most t number of arbitrary adversarial agents in the MAS. The only information that the adversary gets on the inputs of honest agents is their sum (or average). As an extension of this privacy result, it is shown that the proposed privacy protocol can preserve internal privacy of any subset of honest agents as long as that group of honest agents is not cut by the adversarial agents.

Now, consider a slightly different adversarial model where every agent in the network is dishonest, but are not colluding with each other. In that case, as a consequence of the obtained privacy result, a network of 2-connectivity is sufficient enough to protect the privacy of an agent's input against all the other agents in the network.

The main contribution of the proposed two-phase approach is the privacy mechanism, implemented by the agents in the first phase to mask their original inputs and generate effective inputs. The internal privacy then follows, as rigorously shown, from the inability (in statistical sense) of the adversarial agents to determine the masks used by the honest agents in the first phase to compute the effective inputs. As the privacy mechanism does not require synchronicity between agents, thus if the average consensus protocol used in the second phase is asynchronous, then the overall proposed distributed average consensus protocol is asynchronous. Furthermore, the agents can choose any distributed average consensus algorithm

in the second phase to compute the sum of their effective inputs. Consequentially, the proposed two-phase approach can be interpreted as a general approach for constructing distributed average consensus protocols to ensure internal privacy up to a certain extent.

2.5.1 Limitations

Limitations of the proposed privacy protocol are as following:

1. The agents are required to know the value of n , that is the size of the network.
2. The proposed privacy protocol does not prevent an adversary from knowing the sum of all the honest agents.
3. If the adversarial agents form a vertex cut then some of the honest agents might lose privacy of their inputs to the adversary. For instance, if all the neighbors of an honest agent are adversarial then internal privacy of that agent's input is not protected by the proposed privacy protocol.
4. The proposed privacy protocol can not preserve external privacy of the agents (i.e. privacy of agents' inputs against eavesdroppers). Alternately, it relies on existing cryptographic encryption schemes for external privacy.
5. The communication links between the agents during the privacy mechanism are assumed undirected. However, the agents are not required to communicate simultaneously and given the advancements in communication technology, full-duplex communication has become commonplace.

6. For now, the proposed protocol is only applicable for internal privacy in distributed collaborations wherein the collaboration objective is to compute the sum (or average) of agents' inputs. The protocol might not be effective for internal privacy in distributed collaborations wherein the collaboration objective is to compute any other general function on agents' inputs.
7. The adversarial agents are assumed passive. In practice, there can exist agents in the network that are actively adversarial, i.e. they can disrupt the prescribed distributed collaboration protocol including the privacy mechanism. In that case, the proposed privacy protocol might not be effective.

2.5.2 Future Research

In order to overcome the aforementioned limitations, future research can be pursued in the following directions.

1. Investigate if the sufficient condition for internal privacy of all honest agents is also necessary. This will reduce the required redundancy on connectivity of the communication network.
2. Explore conditions under which the proposed privacy protocol can also preserve external privacy, without relying on cryptographic encryptions.
3. Design a similar two-phase approach for internal privacy in distributed computations of more general functions on agents' inputs than just the average (or sum).

4. Explore the applicability of the proposed privacy protocol if some of the agents in the network are actively malicious.
5. Check if the proposed protocol can protect internal privacy of agents' internal states that are generated during the intermediate stage of other distributed collaboration algorithms. This might help in extending the applicability of the proposed privacy protocol for a larger class of distributed collaborations.

CHAPTER 3: INTERNAL PRIVACY IN DISTRIBUTED OPTIMIZATION

This chapter presents an extension of the privacy protocol proposed in Chapter 2, for internal privacy in distributed optimization. Specifically, it is shown that following a similar two-phase approach, as described in Chapter 2, honest agents can preserve privacy of the affine parts of their costs (or inputs) against passively adversarial agents as long as the set of adversarial agents do not constitute a vertex cut in the underlying communication network \mathcal{G} . The proposed protocol is a composition of a distributed privacy mechanism with any (non-private) distributed optimization algorithm.

Similar to the previous chapter, Chapter 2, the proposed approach constitutes of two phases:

1. In the first phase, each agent shares correlated random values with its neighbors and then computes a new, “effective cost” based on its original cost and the random values shared with its neighbors.
2. In the second phase, the agents run any arbitrary distributed optimization protocol to minimize the aggregate of their effective costs, instead of their original costs.

The privacy mechanism is designed to ensure that the aggregate of all agents' effective costs is equivalent to the aggregate of the agents' original costs. Therefore, the minima of the aggregate of effective costs is same as the minima of the aggregate of original costs. Furthermore, the privacy holds in this approach—in a formal sense and under certain conditions, as described in this chapter—regardless of the distributed optimization protocol used in the second phase. This is shown by proving privacy under the worst-case scenario where all the effective costs of the honest agents are revealed to all the agents (including the adversarial agents) in the second phase.

Informally speaking, internal privacy requires that the adversarial agents learn very little (in statistical sense) about the collective affine parts of the original costs of honest agents in an execution of the protocol (described above) other than the aggregate of the affine parts of the honest agents' costs. This holds regardless of any prior knowledge that the adversarial agents may have about the costs of (some of) the honest agents. It is proved that the privacy protocol satisfies this notion of privacy as long as the set of adversarial agents do not constitute a vertex cut of the communication network topology \mathcal{G} .

3.1 Formal Problem Description

Consider a MAS of n agents where the communication network between agents is represented by an undirected, simple, connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$; that is, agents i and j have a direct communication link between them iff $\{i, j\} \in \mathcal{E}$. The commu-

nication channel between two nodes is assumed to be both private and authentic; equivalently, in the adversarial model an adversary can not eavesdrop on communications between honest agents, or tamper with their communication. (Alternately, private and authentic communication can be ensured using standard cryptographic techniques.)

The input \ln_i of each agent is a function $h_i : \mathbb{R}^m \rightarrow \mathbb{R}$. A *distributed optimization algorithm* is an interactive protocol that enables agents to solve the following optimization problem in a distributed manner,

$$\underset{x \in \mathbb{R}^m}{\text{minimize}} \sum_i h_i(x) \tag{3.1}$$

3.1.1 Adversarial Model

Let $\mathcal{C} \subset \mathcal{V}$ denote the set of adversarial agents that are corrupted by a passive adversary, and let $\mathcal{H} = \mathcal{V} \setminus \mathcal{C}$ denote the remaining honest agents. As stated earlier, the adversary is passive and thus the corrupted agents run the prescribed protocol. Privacy requires that the entire *view* of the adversary—i.e., the costs of the corrupted agents as well as their internal states and all the protocol messages they received throughout execution of the protocol—does not leak (significant) information about the costs of the honest agents other than the aggregate of their costs in the solution set of (3.1). Note that, by definition, the adversary learns $\sum_i h_i(x^*)$ for every point x^* belonging to the solution set of (3.1) (assuming it corrupts at least one agent), from which it can compute the sum of the costs of the honest agents at x^* and in the worst-case this can allow the adversary to determine the aggregate of honest agents’

costs: $\sum_{i \in \mathcal{H}} h_i$. Therefore, in this dissertation, the privacy definition (as formulated in the subsequent subsection) requires that the adversary (or the adversarial agents) does not learn anything significant (quantified later) about the affine parts of the honest agents' costs apart from their aggregate.

3.1.2 Privacy Definition

Let $h_i^{(a)}$ denote the affine part¹ of cost h_i , i.e. $h_i^{(a)}(x) = \alpha_i^T x$, where $\alpha_i \in \mathbb{R}^m$.

Let α_i^k denote the k -th element of the affine coefficient of agent i and $\alpha^k = [\alpha_1^k, \dots, \alpha_n^k]^T$ be the vector of k -th elements of all the affine coefficients, for $k \in [m]$. Let $\alpha = \{\alpha_i\}$ and $\alpha' = \{\alpha'_i\}$ be two sets of affine coefficients, then the distance between α and α' , represented by $\text{dist}(\alpha, \alpha')$, is simply the accumulation of Euclidean distances between each of their elements:

$$\text{dist}(\alpha, \alpha') = \sqrt{\sum_{k=1}^m \|\alpha^k - \alpha'^k\|^2}$$

The privacy of the affine parts of honest agents' costs (against \mathcal{C}) is equivalent to the privacy of the coefficients of honest agents' costs $\{\alpha_i\}_{i \in \mathcal{H}}$. Thus, the privacy definition is formulated for the privacy of $\{\alpha_i\}_{i \in \mathcal{H}}$.

Let $\alpha_{\mathcal{C}} = \{\alpha_i\}_{i \in \mathcal{C}}$ denote the affine coefficients of corrupted agents' costs, and $\alpha_{\mathcal{H}} = \{\alpha_i\}_{i \in \mathcal{H}}$ denote the affine coefficients of honest agents' costs.

Definition 3.1.1. For a distributed optimization protocol, let $\text{View}_{\mathcal{C}}(\alpha)$ be the random variable denoting the information learned by the corrupted agents (or view

¹Every function $h(x)$ can be decomposed as $h(x) = h^{(na)}(x) + h^{(a)}(x)$, where $h^{(a)}(x)$ is affine in x (called the affine part of h) and $h^{(na)}(x)$ is not affine in x .

of the corrupted agents) in an execution of the distributed optimization protocol where the agents begin holding costs with affine coefficients $\alpha = \{\alpha_i\}$.

Then:

Definition 3.1.2. A distributed optimization protocol is (\mathcal{C}, ϵ) -*affine private* if for all pair of affine coefficients $\{\alpha_i\}, \{\alpha'_i\}$ that satisfy $\alpha_{\mathcal{C}} = \alpha'_{\mathcal{C}}$ and $\sum_{i \in \mathcal{H}} \alpha_i = \sum_{i \in \mathcal{H}} \alpha'_i$, the KL-divergence between the distributions of $\text{View}_{\mathcal{C}}(\alpha)$ and $\text{View}_{\mathcal{C}}(\alpha')$ is bounded above as following:

$$D_{KL}(\text{View}_{\mathcal{C}}(\alpha), \text{View}_{\mathcal{C}}(\alpha')) \leq \epsilon \text{dist}(\alpha, \alpha')^2$$

Note that this definition makes sense even if $|\mathcal{C}| = n - 1$, though in that case the definition is vacuous since $\alpha_{\mathcal{H}} = \sum_{i \in \mathcal{H}} \alpha_i$ and so revealing the sum of the honest agents' costs reveals the affine coefficients of honest agent's cost.

In other words, the privacy definition above says that it is very difficult for the adversary to distinguish between two possible sets of affine coefficients of honest agents' costs, if they are close enough (in terms of the Euclidean distance) to each other and has the same aggregate(or sum).

3.1.3 Assumptions

The assumptions made in this chapter are as following:

- (A1) The communication network \mathcal{G} is undirected in the first phase.
- (A3) All the agents' costs have a common domain \mathbb{R}^m .

(A4) The adversarial agents are passive, that is they obey the prescribed protocol without causing any disruptions.

3.2 Proposed Protocol

As described above, the protocol has a two-phase structure (as shown in Fig. 3.1). In the first phase, each agent i computes an “effective cost” \tilde{h}_i based on its original cost h_i and correlated random values it shares with its neighbors; this is done while ensuring that $\sum_i \tilde{h}_i(x)$ is equal to $\sum_i h_i(x)$ for all $x \in \mathbb{R}^m$ (shown below). In the second phase, the agents use any distributed optimization protocol Π to solve the following optimization problem

$$\underset{x \in \mathbb{R}^m}{\text{minimize}} \sum_i \tilde{h}_i(x) \tag{3.2}$$

This (as will be shown) gives the solution to the original optimization problem (3.1). Also, in the first phase the effective cost of each agent is obtained by adding an affine cost to the original cost, thus \tilde{h}_i is convex if h_i is convex. Hence, most of the existing distributed optimization (including the protocols that require individual costs to be convex) can be used in the second phase to solve (3.2) without any change in the order of their computational complexity.

It may at first seem strange that the privacy of the algorithm can be proved without knowing anything about the distributed optimization protocol Π used in the second phase. This is done by making a “worst-case” assumption about Π , namely, that it simply reveals all the agents’ effective costs to all the agents. Such an algorithm is, of course, not at all private; for our purposes, however, this does

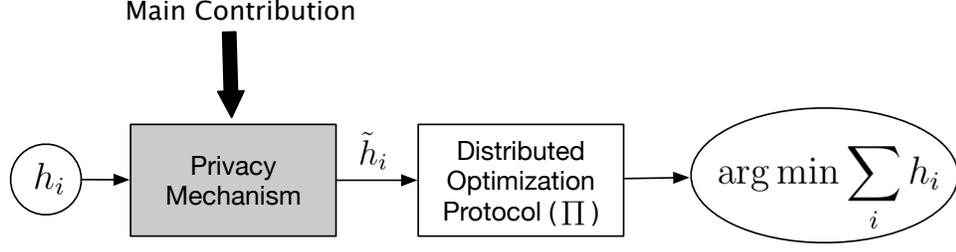


Figure 3.1: Proposed protocol for internal privacy in distributed optimization.

not immediately violate privacy because Π is run on the agents' *effective costs* $\{\tilde{h}_i\}$ instead of their original costs $\{h_i\}$.

From now on, then, view of the adversary consists of the original costs of the corrupted (adversarial) agents, their internal states and all the protocol messages they receive throughout execution of the first phase of our protocol, and all the agents' effective costs $\tilde{h} = [\tilde{h}_1, \dots, \tilde{h}_n]^T$ obtained at the end of the first phase. Our definition of privacy (cf. Definition 3.1.2) remains unchanged.

The first phase of the protocol proceeds as follows:

1. Each agent $i \in \mathcal{V}$ chooses independent vectors $r_{ij} \sim N(0_m, \sigma^2 \text{Diag}(1_m))$ from \mathbb{R}^m for all $j \in \mathcal{N}_i$, and sends r_{ij} to agent j . Here, $\sigma \in \mathbb{R}$.
2. Each agent $i \in \mathcal{V}$ computes a mask

$$a_i = \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}), \quad (3.3)$$

3. Each agent $i \in \mathcal{V}$ computes effective cost \tilde{h}_i , given as following,

$$\tilde{h}_i(x) = h_i(x) + a_i^T x, \forall x \in \mathbb{R}^m. \quad (3.4)$$

Note that

$$\sum_i \tilde{h}_i(x) = \sum_i h_i(x) + \sum_i a_i^T x, \forall x \in \mathbb{R}^m$$

Moreover,

$$\sum_i a_i = \sum_i \sum_{j \in \mathcal{N}_i} (r_{ji} - r_{ij}) = 0$$

since \mathcal{G} is undirected. Thus, $\sum_i \tilde{h}_i \equiv \sum_i h_i$ and hence correctness of the overall algorithm (i.e., including the second phase) follows. Now, all that remains is the privacy analysis.

Note that the privacy mechanism, described above, is asynchronous as the agents are not required to receive or send $\{r_{ij}\}_{\{i,j\} \in \mathcal{E}}$ simultaneously.

The proposed algorithm is illustrated by example in Section 3.3.

3.2.1 Privacy Analysis

It is shown here that (\mathcal{C}, ϵ) -affine privacy holds as long as \mathcal{C} is not a vertex cut of \mathcal{G} .

For an edge $e = \{i, j\}$ in the graph with $i < j$, define

$$b_e = r_{ji} - r_{ij}.$$

Let $b^k = [b_{e_1}^k, \dots]^T$ be the vector of the k -th elements of all such vectors b_e for all the edges in \mathcal{G} . Let $a^k = [a_1^k, \dots, a_n^k]^T$ be the column vector constituting of k -th elements of the masks computed by the agents, then

$$a^k = \nabla \cdot b^k.$$

Since the element of vector r_{ij} are identical and independent from each other with normal distribution $N(0, \sigma^2)$ for all $\{i, j\} \in \mathcal{E}$, it is easy to see that the values $\{b_e^k\}_{e \in \mathcal{E}}$ are independent and have identical normal distribution $N(0, 2\sigma^2)$ in \mathbb{R} , for all k in $[m]$. Consequentially, a^k is normally distributed over \mathbb{R}^n with mean and covariance equal to 0_n and $2\sigma^2\mathcal{L}$, respectively for all $k \in [m]$. Specifically, the probability density of a^k at any point $a \in \mathbb{R}^n$ is given by

$$f_{a^k}(a) = \frac{1}{\sqrt{\det^*(4\pi\sigma^2\mathcal{L})}} \exp\left(-\frac{1}{4\sigma^2}a^T\mathcal{L}^\dagger a\right) \quad (3.5)$$

where, $\det^*(4\pi\sigma^2\mathcal{L}) = (4\pi\sigma^2)^{n-c} \prod_{i=1}^{n-c} \mu_i$ (product of non-zero eigenvalues). As $\text{rank}(\mathcal{L}) = n - 1$ when \mathcal{G} is connected (cf. Lemma 1.4.1) and $1_n^T a^k = 0, \forall k \in [m]$ (as \mathcal{G} is undirected, thus $1_n^T \nabla = 0_{|\mathcal{E}|}^T$), we get:

Lemma 3.2.1. If \mathcal{G} is connected then a^k is normally distributed over all points in \mathbb{R}^n subject to the constraint that $\sum_i a_i^k = 0$ for all $k \in [m]$, with mean value $\mathbb{E}(a^k) = 0_n$ and covariance $\text{Cov}(a^k) = 2\sigma^2\mathcal{L}$.

Since $\tilde{\alpha}_i = \alpha_i + a_i$, the above lemma implies the following.

Lemma 3.2.2. If \mathcal{G} is connected then the k -th elements of the effective affine coefficients $\tilde{\alpha}^k = [\tilde{\alpha}_1^k, \dots, \tilde{\alpha}_n^k]^T$ are normally distributed in \mathbb{R}^n subject to the constraint that $\sum_i \tilde{\alpha}_i^k = \sum_i \alpha_i^k$ for all $k \in [m]$, with mean value $\mathbb{E}(\tilde{\alpha}^k) = \alpha^k = [\alpha_1^k, \dots, \alpha_n^k]^T$ and covariance $\text{Cov}(\tilde{\alpha}^k) = 2\sigma^2\mathcal{L}$, given the values of $\{\alpha_i\}$.

Proof. As $\tilde{\alpha}_i = \alpha_i + a_i, \forall i \in \mathcal{V}$, therefore $\tilde{\alpha}_i^k = \alpha_i^k + a_i^k$ for every $i \in \mathcal{V}$ and $k \in [m]$. ($\tilde{\alpha}_i^k$ and α_i^k denote the k -th elements of $\tilde{\alpha}_i$ and α_i , respectively.)

If we let $\tilde{\alpha}^k = [\tilde{\alpha}_1^k, \dots, \tilde{\alpha}_n^k]^T$ and $\alpha^k = [\alpha_1^k, \dots, \alpha_n^k]^T$ then $\tilde{\alpha}^k = \alpha^k + a^k, \forall k \in [m]$. As a_i is independent of α_i for all $i \in \mathcal{V}$, thus Lemma 3.2.1 implies that if \mathcal{G} is connected then $\tilde{\alpha}^k$ is normally distributed in \mathbb{R}^n subject to the constraint $\sum_i \tilde{\alpha}_i^k = \sum_i \alpha_i^k$ with $\mathbb{E}(\tilde{\alpha}^k) = \alpha^k$ and $\text{Cov}(\tilde{\alpha}^k) = 2\sigma^2\mathcal{L}$ for all $k \in [m]$, given the values of $\{\alpha_i\}$. ■

Let $f_{\tilde{\alpha}|\alpha}$ and $f_{\tilde{\alpha}|\alpha'}$ denote the conditional probability density function (or distribution) of the collective effective affine coefficients $\tilde{\alpha} = [\tilde{\alpha}_1, \dots, \tilde{\alpha}_n]$ given the true affine coefficients as $\alpha = \{\alpha_i\}$ and $\alpha' = \{\alpha'_i\}$, respectively. Then, using Lemma 3.2.2, we get:

Theorem 3.2.1. If \mathcal{G} is connected then

$$D_{KL}(f_{\tilde{\alpha}|\alpha} || f_{\tilde{\alpha}|\alpha'}) \leq \epsilon \text{dist}(\alpha, \alpha')^2$$

for all affine coefficients $\alpha = [\alpha_1, \dots, \alpha_n]$ and $\alpha' = [\alpha'_1, \dots, \alpha'_n]$ that satisfy $\sum_i \alpha_i = \sum_i \alpha'_i$ with $\epsilon = 1/(4\sigma^2\mu(\mathcal{L}))$, where $\mu(\mathcal{L})$ is the smallest non-zero eigenvalue of \mathcal{L} .

Proof. As $\tilde{\alpha}^k = \alpha^k + a^k$ and the value of a^k is independent of α^k for every $k \in [m]$, thus

$$f_{\tilde{\alpha}^k|\alpha}(\tilde{\alpha}^k) = f_{a^k}(\tilde{\alpha}^k - \alpha^k) \text{ and } f_{\tilde{\alpha}^k|\alpha'}(\tilde{\alpha}^k) = f_{a^k}(\tilde{\alpha}^k - \alpha'^k) \forall k \in [m]$$

where α^k and α'^k represent the k -th elements of coefficients $\{\alpha_i\}$ and $\{\alpha'_i\}$, respectively. This implies (cf. Lemma 3.2.1),

$$\log \frac{f_{\tilde{\alpha}^k|\alpha}(\tilde{\alpha}^k)}{f_{\tilde{\alpha}^k|\alpha'}(\tilde{\alpha}^k)} = \frac{1}{4\sigma^2} \times \{(\tilde{\alpha}^k - \alpha^k)^T \mathcal{L}^\dagger (\alpha^k - \alpha^k) - (\tilde{\alpha}^k - \alpha'^k)^T \mathcal{L}^\dagger (\tilde{\alpha}^k - \alpha'^k)\}, \forall k \in [m]$$

when \mathcal{G} is connected. By further simplifying the above equality, we get

$$\log \frac{f_{\tilde{\alpha}^k|\alpha}(\tilde{\alpha}^k)}{f_{\tilde{\alpha}^k|\alpha'}(\tilde{\alpha}^k)} = \frac{1}{4\sigma^2} (\alpha^k - \alpha'^k)^T \mathcal{L}^\dagger (2\tilde{\alpha}^k - \alpha^k - \alpha'^k), \forall k \in [m]$$

For simplicity, we let $a = \tilde{\alpha}^k - \alpha^k$. Then,

$$\begin{aligned}
D_{KL}(f_{\tilde{\alpha}^k|\alpha}||f_{\tilde{\alpha}^k|\alpha'}) &= \\
&= \frac{1}{4\sigma^2} \int_{a \in \mathbb{R}^n} (\alpha - \alpha')^T \mathcal{L}^\dagger(2a + \alpha - \alpha') f_{a^k}(a) da = \\
&= \frac{1}{2\sigma^2} (\alpha^k - \alpha'^k)^T \mathcal{L}^\dagger \mathbb{E}(a^k) + \frac{1}{4\sigma^2} (\alpha^k - \alpha'^k)^T \mathcal{L}^\dagger (\alpha^k - \alpha'^k) \\
&= \frac{1}{4\sigma^2} (\alpha^k - \alpha'^k)^T \mathcal{L}^\dagger (\alpha^k - \alpha'^k), \forall k \in [m]
\end{aligned}$$

As $1_n^T(\alpha^k - \alpha'^k) = 0_n$, i.e. $\alpha^k - \alpha'^k$ is orthogonal to 1_n and $\text{rank}(\mathcal{L}) = n - 1$ when \mathcal{G} is connected, therefore

$$D_{KL}(f_{\tilde{\alpha}^k|\alpha}||f_{\tilde{\alpha}^k|\alpha'}) \leq \frac{\|\alpha^k - \alpha'^k\|^2}{4\sigma^2 \mu_{n-1}} = \frac{\|\alpha^k - \alpha'^k\|^2}{4\sigma^2 \underline{\mu}(\mathcal{L})}, \forall k \in [m]$$

Note that different elements a^k and $a^{k'}$ of the masks are independent of each other, where $k \neq k' \in [m]$. Therefore,

$$f_{\tilde{\alpha}|\alpha} = \prod_{k=1}^m f_{\tilde{\alpha}^k|\alpha} \text{ and similarly, } f_{\tilde{\alpha}|\alpha'} = \prod_{k=1}^m f_{\tilde{\alpha}^k|\alpha'}$$

Hence,

$$D_{KL}(f_{\tilde{\alpha}|\alpha}||f_{\tilde{\alpha}|\alpha'}) = \sum_{k=1}^m D_{KL}(f_{\tilde{\alpha}^k|\alpha}||f_{\tilde{\alpha}^k|\alpha'}) \leq \frac{\text{dist}(\alpha, \alpha')^2}{4\sigma^2 \underline{\mu}(\mathcal{L})}$$

with $\epsilon = 1/(4\sigma^2 \underline{\mu}(\mathcal{L}))$. ■

The above implies (\mathcal{C}, ϵ) - affine privacy of the proposed distributed optimization algorithm for the case when $\mathcal{C} = \emptyset$, i.e., when there are no corrupted agents. In that case, the view of the adversary consists only of the effective affine coefficients $\tilde{\alpha}$, and Lemma 3.2.2 shows that the distribution of those values depends only on the sum of the agents' true affine coefficients. Below, this line of argument is extended to the case of nonempty \mathcal{C} .

Fix some set \mathcal{C} of corrupted agents, and recall that $\mathcal{H} = \mathcal{V} \setminus \mathcal{C}$. Let $\mathcal{E}_{\mathcal{C}}$ denote the set of edges incident to \mathcal{C} , and let $\mathcal{E}_{\mathcal{H}} = \mathcal{E} \setminus \mathcal{E}_{\mathcal{C}}$ be the edges incident only to honest agents. We refer to $\mathcal{G}_{\mathcal{H}} = \{\mathcal{H}, \mathcal{E}_{\mathcal{H}}\}$ as the *honest graph* and let $\mathcal{L}_{\mathcal{H}}$ denote the graph-Laplacian of $\mathcal{G}_{\mathcal{H}}$. Note that now the adversary's view contains (information that allows it to compute) $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$ in addition to the honest agents' affine coefficients $\{\tilde{\alpha}_i\}_{i \in \mathcal{H}}$.

The key observation enabling a proof of privacy is that the values $\{b_e^k\}_{e \in \mathcal{E}_{\mathcal{H}}}$ are independent in $\mathbb{R}^{|\mathcal{H}|}$ *even conditioned on the values of* $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$, for every $k \in [m]$. Thus, owing to Theorem 3.2.1, we get the following privacy guarantee:

Theorem 3.2.2. If \mathcal{C} is not a vertex cut of \mathcal{G} , then the proposed distributed optimization protocol is (\mathcal{C}, ϵ) -affine private, with $\epsilon = 1/(4\sigma^2\mu(\mathcal{L}_{\mathcal{H}}))$.

Proof. As \mathcal{C} is not a vertex cut of \mathcal{G} , this implies $\mathcal{G}_{\mathcal{H}}$ is connected.

The view of the adversarial agents is given by

$$\text{View}_{\mathcal{C}}(\alpha) = \{\{\alpha_i\}_{i \in \mathcal{C}}, \{\tilde{\alpha}_i\}, \{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}\}$$

As $\tilde{\alpha}_i = \alpha_i + a_i, \forall i$ and $\sum_i a_i = 0_m$ (as \mathcal{G} is undirected), this implies

$$\sum_{i \in \mathcal{H}} \tilde{\alpha}_i = \sum_{i \in \mathcal{H}} \alpha_i - \sum_{i \in \mathcal{C}} a_i$$

where, the k -th element of a_i is given as

$$a_i^k = \sum_{e \in \mathcal{E}_{\mathcal{C}}} \nabla_{i,e} \cdot b_e^k, \forall i \in \mathcal{C}$$

for every $k \in [m]$.

Let $f_{\tilde{\alpha}_{\mathcal{H}}|\alpha_{\mathcal{H}},\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}}$ and $f_{\tilde{\alpha}_{\mathcal{H}}|\alpha'_{\mathcal{H}},\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}}$ represent the conditional probability density function of $\alpha_{\mathcal{H}}$ given $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$ and the values of the original affine coefficients

of honest agents as $\alpha_{\mathcal{H}} = \{\alpha_i\}_{i \in \mathcal{H}}$ and $\alpha'_{\mathcal{H}} = \{\alpha'_i\}_{i \in \mathcal{H}}$, respectively. Then, Theorem 3.2.1 implies

$$D_{KL}(f_{\tilde{\alpha}_{\mathcal{H}}|\alpha_{\mathcal{H}},\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}} \| f_{\tilde{\alpha}_{\mathcal{H}}|\alpha'_{\mathcal{H}},\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}}) \leq \epsilon \text{dist}(\alpha_{\mathcal{H}}, \alpha'_{\mathcal{H}})^2$$

when $\mathcal{G}_{\mathcal{H}}$ is connected, for all affine coefficients $\alpha_{\mathcal{H}}$ and $\alpha'_{\mathcal{H}}$ that satisfy $\sum_{i \in \mathcal{H}} \alpha_i = \sum_{i \in \mathcal{H}'} \alpha'_i$ with $\epsilon = 1/(4\sigma^2 \underline{\mu}(\mathcal{L}_{\mathcal{H}}))$, where $\underline{\mu}(\mathcal{L}_{\mathcal{H}})$ is the smallest non-zero eigenvalue of $\mathcal{L}_{\mathcal{H}}$.

As $\{b_e\}_{e \in \mathcal{E}_{\mathcal{C}}}$ are chosen independently of the affine coefficients $\{\alpha_i\}$, the above implies

$$D_{KL}(\text{View}_{\mathcal{C}}(\alpha) \| \text{View}_{\mathcal{C}}(\alpha')) \leq \epsilon \text{dist}(\alpha, \alpha')^2$$

when $\mathcal{G}_{\mathcal{H}}$ is connected, for any two sets of affine coefficients $\alpha = \{\alpha_i\}$ and $\alpha' = \{\alpha'_i\}$ that satisfy $\alpha_i = \alpha'_i, \forall i \in \mathcal{C}$ and $\sum_i \alpha_i = \sum_i \alpha'_i$. ■

As a corollary, we have

Corollary 3.2.1. If \mathcal{G} is $(t+1)$ -connected, then for any \mathcal{C} with $|\mathcal{C}| \leq t$ the proposed distributed optimization protocol is (\mathcal{C}, ϵ) -affine private.

The subsequent subsection discusses the relationship between the vertex expansion of $\mathcal{G}_{\mathcal{H}}$ and privacy strength of the proposed privacy protocol.

3.2.2 Vertex Expansion and Privacy Strength

The value of ϵ is a quantitative measure of privacy, and smaller is the value of ϵ higher is the privacy. As is shown in Theorem 3.2.2, ϵ is inversely proportional to the variance of random values added to the affine coefficients, which is quite

intuitive. As the proposed privacy mechanism does not affect the accuracy of the distributed optimization protocol, thus we can choose value of σ appropriately for desirable privacy strength.

It is interesting to note that the value of ϵ is inversely proportional to the smallest non-zero eigenvalue of the Laplacian of the graph $\mathcal{L}_{\mathcal{H}}$. From Cheeger's Theorem [64], we know that

$$\frac{\phi_{\mathcal{G}_{\mathcal{H}}}^2}{2} \leq \underline{\mu}(\mathcal{L}_{\mathcal{H}}) \leq 2\phi_{\mathcal{G}_{\mathcal{H}}} \quad (3.6)$$

where, $\phi_{\mathcal{G}_{\mathcal{H}}}$ is a non-negative real number that is known as the *vertex expansion* of graph $\mathcal{G}_{\mathcal{H}}$. Value of $\phi_{\mathcal{G}_{\mathcal{H}}}$ roughly indicates how close $\mathcal{G}_{\mathcal{H}}$ is to being not connected (specific form is omitted here, interested readers can refer to [65, 66]). The value of $\phi_{\mathcal{G}_{\mathcal{H}}}$ is zero if and only if $\mathcal{G}_{\mathcal{H}}$ is not connected. Thus, owing to the Cheeger's inequality (3.6) the value of ϵ in Theorem 3.2.2 is bounded above as

$$\epsilon \leq \frac{1}{2\sigma^2\phi_{\mathcal{G}_{\mathcal{H}}}^2}$$

This means that higher vertex expansion of the honest graph $\mathcal{G}_{\mathcal{H}}$ implies better privacy.

3.3 Illustration

Consider a simple network of 3 agents with $\mathcal{V} = \{1, 2, 3\}$ and

$\mathcal{E} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, as shown in Fig. 3.2.

Let $h_1(x) = (x - x_1)^2$, $h_2(x) = (x - x_2)^2$ and $h_3(x) = (x - x_3)^2$, where $x \in \mathbb{R}$ is the variable and $x_i \in \mathbb{R}$, $\forall i$ are the constants. Let $\mathcal{C} = \{3\}$ and so, $\mathcal{H} = \{1, 2\}$. If

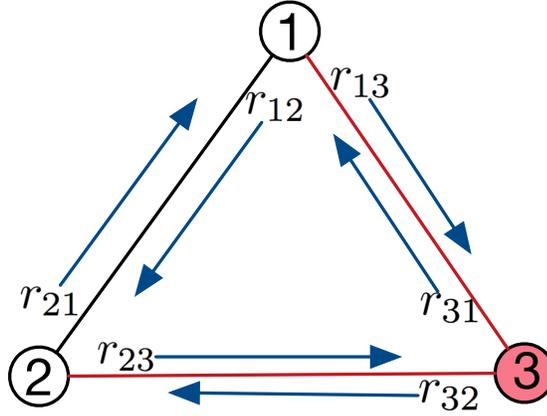


Figure 3.2: Illustration of the proposed distributed optimization protocol. Arrows (in blue) indicate the flow of information over communication links.

the agents use a consensus-based gradient method [29] for distributed optimization of the aggregate $\sum_i h_i(x)$ then the adversarial agent 3 acquires knowledge of x_1 and x_2 (using the received optimal estimates of 1 and 2 at each time-step of the optimization algorithm). Now, to prevent this loss of privacy, the agents implement the proposed privacy protocol in the following manner.

In phase one, the agents execute the following steps

1. As shown in Fig. 3.2, all pairs of adjacent agents i and j exchange the respective values r_{ij} and r_{ji} (chosen independently and following a normal distribution, with mean 0 and variance $\sigma = 1$, in \mathbb{R}) with each other. Consider a particular instance where

$$r_{12} = 0.1, r_{21} = 0.5, r_{23} = 0.7, r_{32} = 0.4, r_{31} = 0.3, r_{13} = 0.8$$

2. The agents compute their respective masks,

$$a_1 = (r_{21} - r_{12}) + (r_{31} - r_{13}) = -0.1$$

Similarly, $a_2 = -0.7$ and $a_3 = 0.8$. (One can verify that $(a_1 + a_2 + a_3) = 0$.)

3. The agents compute their respective effective costs,

$$\tilde{h}_1(x) = (x - x_1)^2 - 0.1x = x^2 - (2x_1 + 0.1)x + x_1^2$$

Similarly, $\tilde{h}_2(x) = x^2 - (2x_2 + 0.7)x + x_2^2$ and $\tilde{h}_3(x) = x^3 - (2x_3 - 0.8)x + x_3^2$.

After phase one, each agent uses a (non-private) distributed optimization algorithm Π in the second phase to compute the value of x that minimizes $\sum_i \tilde{h}_i$ (it can be easily verified that $\sum_i \tilde{h}_i \equiv \sum_i h_i$).

Here, as agent 3 does not cut the honest agents 1 and 2, therefore an adversary that corrupts agent 3 can only determine $2x_1 + 2x_2$ with certainty and not the individual values of x_1 and x_2 (cf. Theorem 3.2.2). Specifically, as $\sigma = 1$ and $\underline{\mu}(\mathcal{L}_{\mathcal{H}}) = 2$

$$\begin{aligned} D_{KL}(\mathbf{View}_{\{3\}}(2x_1, 2x_2) || \mathbf{View}_{\{3\}}(2x'_1, 2x'_2)) \\ \leq \frac{1}{c} \{(x_1 - x'_1)^2 + (x_2 - x'_2)^2\} \end{aligned}$$

for any two (x_1, x_2) and (x'_1, x'_2) that satisfy

$$x_1 + x_2 = x'_1 + x'_2$$

3.4 Concluding Remarks

The proposed two-phase approach for distributed optimization guarantees internal privacy of the affine parts of honest agents' costs if the passively adversarial

agents (or agents corrupted by a passive adversary) do not form a vertex cut in the underlying communication network \mathcal{G} . This implies that the proposed distributed optimization protocol preserves internal privacy of the affine parts of agents' costs in presence of at most t arbitrary adversarial agents if the communication network has $(t + 1)$ -connectivity. The only information that the adversary gets on the affine parts of honest agents' costs is their sum (or average). Using similar arguments as in Section 2.4, it can be shown that the proposed privacy protocol preserves privacy of the affine parts of the costs of a group of honest agents – in a formal sense as defined for all the honest agents – that are not cut by the adversarial agents.

Consider a slightly different adversarial model wherein every agent in the network is dishonest, but are not colluding with each other. In that case, as a consequence of the obtained privacy result, a network of 2-connectivity is sufficient enough to protect the privacy of the affine parts of an agent's costs against all the other agents in the network.

The main contribution of the proposed two-phase approach is the privacy mechanism, implemented by the agents in the first phase to mask their original costs and generate effective costs. The internal privacy then follows, as rigorously shown, from the inability (in statistical sense) of the adversarial agents to determine the masks used by the honest agents. As the privacy mechanism does not require synchronicity between agents, thus if the distributed optimization protocol used in the second phase is asynchronous, then the overall proposed protocol is asynchronous as well. Moreover, the agents can choose any distributed optimization algorithm in the second phase on their effective costs. Therefore, the proposed

two-phase approach can be interpreted as a general way of constructing distributed optimization protocols to protect internal privacy of the participating agents.

3.4.1 Limitations

Limitations of the proposed privacy protocol are as following:

1. The proposed privacy protocol only guarantees internal privacy of the *affine parts* of honest agents' costs and not the entire costs. However, similar approach can be readily used to protect internal privacy of other higher-order coefficients in the agents' costs.
2. The proposed privacy protocol does not prevent the adversarial agents from knowing the aggregate of the affine parts of honest agents' costs.
3. If the adversarial agents form a vertex cut then some of the honest agents might lose privacy of their entire costs to the adversary. For instance, if all the neighbors of an honest agent are adversarial then privacy of that honest agent's cost can not be protected by the proposed privacy protocol.
4. The proposed privacy protocol can not preserve external privacy of the agents (privacy of agents' costs against eavesdroppers). Alternately, it relies on existing cryptographic encryptions for external privacy.
5. The communication links between the agents in the first phase are assumed undirected. However, the agents are not required to communicate simultaneously and given the advancements in communication technology, full-duplex

communication has become commonplace.

6. The adversarial agents are assumed passive. In practice, there can exist agents in the network that are actively adversarial, i.e. they can disrupt the prescribed distributed collaboration protocol including the privacy mechanism. In that case, the proposed privacy protocol might not be effective.

3.4.2 Future Research

In order to overcome the aforementioned limitations, future research can be pursued in the following directions.

1. Investigate if the sufficient condition for internal privacy of all the honest agents is also necessary.
2. Explore the conditions under which the proposed two-phase approach can also preserve external privacy, without relying on cryptographic encryptions.
3. Extend the applicability of the proposed privacy protocol for internal privacy of the entire agents' costs instead of just their affine parts.
4. Investigate the efficacy of the proposed privacy protocol in case where some of the agents in the network are actively adversarial.

CHAPTER 4: MODEL-BASED SCHEME FOR EXTERNAL PRIVACY IN DISTRIBUTED COLLABORATION ALGORITHMS

This chapter investigates a model-based scheme for external privacy of agents in distributed collaboration algorithms that can be modeled as 'one-channel feedback' linear time-invariant (LTI) networked control systems (NCS), such as the cooperative control of autonomous vehicles [11,50], output synchronization of passive systems [51], supervisory control and data acquisition (SCADA) for smart homes or power grids [52,53] and higher-order consensus [54,55]. A one-channel feedback¹ LTI NCS constitutes of a dynamic plant whose states evolve in a linear time-invariant manner, and are measured by sensors that are remotely located. Whereas, the controller is co-located with the actuators the drive the states of the plant [67].

For an example of such a distributed collaboration algorithm, consider a MAS of n LTI control systems (or agents) wherein the collaboration objective is to achieve consensus on states of all these control systems. Note that the agents in this case are LTI control systems. Each control system i is associated with state $x_{i,t} \in \mathbb{R}^n$

¹If both sensors and controller are remotely located from the actuators, then the NCS is referred as 'two-channel feedback' NCS [67]

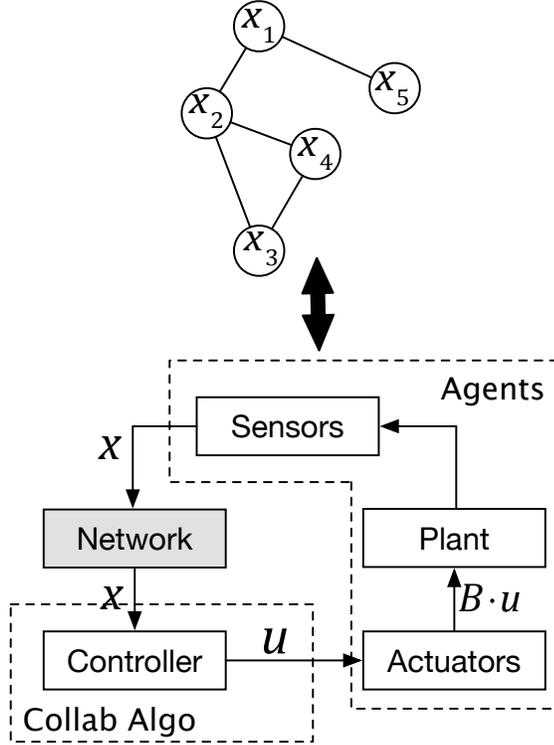


Figure 4.1: An illustration of the modeling of a distributed state-consensus algorithm by a one-channel feedback networked control system.

and control input $u_{i,t} \in \mathbb{R}^m$ for every time $t \in \mathbb{Z}_{\geq 0}$, where \underline{n} , \underline{m} are positive integers. In this particular case, the input In_i , internal states State_i and output Out_i of each agent i is given by the agent's initial state $x_{i,0}$, collection of states $\{x_{i,t}\}_{t \in \mathbb{N}}$ and asymptotic state $\lim_{t \rightarrow \infty} x_{i,t}$, respectively.

The states evolve in time as following,

$$x_{i,t+1} = A_i x_{i,t} + B_i u_{i,t}, \forall t \in \mathbb{Z}_{\geq 0}, \forall i \in [n]$$

where, $A_i \in \mathbb{R}^{n \times n}$ and $B_i \in \mathbb{R}^{n \times m}$ for all $i \in [n]$. To achieve the objective of state consensus, that is to ensure $\lim_{t \rightarrow \infty} (x_{i,t} - x_{j,t}) = 0, \forall i, j \in [n]$, each agent i

applies the following linear time-invariant static feedback control:

$$u_{i,t} = K_i \sum_{j \in \mathcal{N}_i} (x_{i,t} - x_{j,t}), \forall t \in \mathbb{Z}_{\geq 0}$$

where, $K_i \in K \in \mathbb{R}^{\underline{m} \times \underline{n}}$ for all $i \in [n]$. Hence, the resultant mathematical model of this distributed collaboration algorithm is equivalent to the following LTI control system,

$$x_{t+1} = A x_t + B u_t, u_t = K x_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.1)$$

where $x_t = [x_{1,t}^T, \dots, x_{n,t}^T]^T \in \mathbb{R}^{nn}$ is the collection of all the states of the agents and $u_t = [u_{1,t}^T, \dots, u_{n,t}^T]^T \in \mathbb{R}^{nm}$ is the collection of all the control inputs of the agents. The system parameters $A \in \mathbb{R}^{nn \times nn}$, $B \in \mathbb{R}^{nn \times nm}$ and $K \in \mathbb{R}^{nm \times nn}$ are as following:

$$A = \begin{bmatrix} A_1 & \cdots & \mathbf{0}_{\underline{n}} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{\underline{n}} & \cdots & A_n \end{bmatrix}, B = \begin{bmatrix} B_1 & \cdots & \mathbf{0}_{\underline{n} \times \underline{m}} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{\underline{n} \times \underline{m}} & \cdots & B_n \end{bmatrix}$$

and $K = [K_{ij}]_{i,j \in [n]}$, where

$$K_{ij} = \begin{cases} |\mathcal{N}_i| K_i & , \quad i = j \\ -K_i & , \quad \{i, j\} \in \mathcal{E} \\ \mathbf{0}_{\underline{n} \times \underline{m}} & , \quad \text{o.w.} \end{cases}$$

Therefore, this particular distributed collaboration algorithm can be modeled as a networked control system that constitutes of a linear time-invariant plant, and whose states are measured by remote sensors (as shown in Fig. 4.1). Here, the sensor measurements represent the states $\{\text{State}_i\}$ that are exchanged between agents in the intermediate stage of the distributed collaboration algorithm.

Note: The external privacy of agents in such distributed collaboration algorithms is equivalent to the privacy of sensor measurements against an eavesdropper in the resultant NCS.

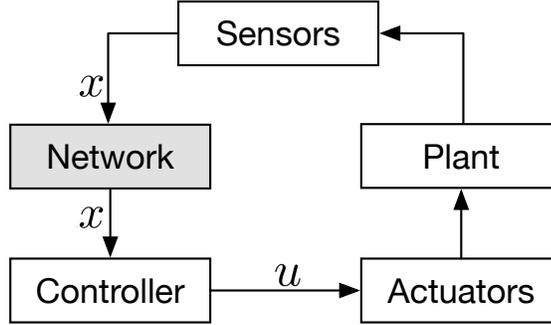


Figure 4.2: Schematic of a networked control system with remote sensors for measuring states.

Henceforth, the discussion in this chapter pertains to the privacy of the sensor measurements (against eavesdroppers) of the 'one-channel feedback' NCS (shown in Fig. 4.2) whose states follow the dynamics given by (4.1). For notational convenience, the dimension of x_t and u_t are simply represented by n and m , respectively (i.e. n_n and n_m are replaced by n and m , respectively).

4.1 Formal Problem Description

Consider a distributed collaboration algorithm that can be modeled by an LTI NCS with remote sensors (as shown in Fig. 4.2), whose states evolve according to the dynamics (4.1). The sensor measurements denote the states that the agents generate and exchange in the intermediate stage of the collaboration algorithm. Therefore, privacy of the agents' states (and inputs) against a passive adversary listening to the messages exchanged between the agents in the distributed collaboration algorithm

is equivalent to the privacy of sensor measurements against an eavesdropper that is listening to the messages transmitted by the sensors (across the network) in the resultant NCS shown in Fig.4.2.

4.1.1 Adversarial Model

An eavesdropper is defined as following.

Definition 4.1.1. (*Eavesdropper*) An eavesdropper is a passive adversary that can read all the messages being transmitted by the sensors over the network in the NCS shown in Fig.4.2. An eavesdropper has precise knowledge of the system parameters (A , B and K) and the encryption scheme, if used, to encrypt the messages by the sensors.

As an eavesdropper is assume passive, implies it can not tamper with the messages in any form.

The view of an eavesdropper is the information stored in the messages transmitted by the sensors over the network to the control. As defined formally in the subsequent subsection, privacy of sensor measurements against an eavesdropper requires that the view of the eavesdropper reveals little information (in statistical sense) about the elements of x_t at any time $t \in \mathbb{Z}_{\geq 0}$.

4.1.2 Assumptions

The assumptions made in this chapter are as following.

(A1) The distributed collaboration algorithm can be modeled as a 'one-channel feedback' NCS with LTI closed-loop dynamics.

(A2) There is no delay in the network.

(A3) The eavesdropper is a passive adversary, i.e. it does not tamper with the messages.

(A4) In the state-dynamics (4.1), $A + BK$ is non-singular.

4.2 Model-Based Privacy Scheme

This section is divided into two subsections. In the first subsection, the model-based privacy scheme, which is referred as *Masking With System Kernel* (MSK), is presented. In the second subsection, the privacy of states $\{x_t\}$ against an eavesdropper is formally defined along with its implications.

4.2.1 Masking With System Kernel (MSK)

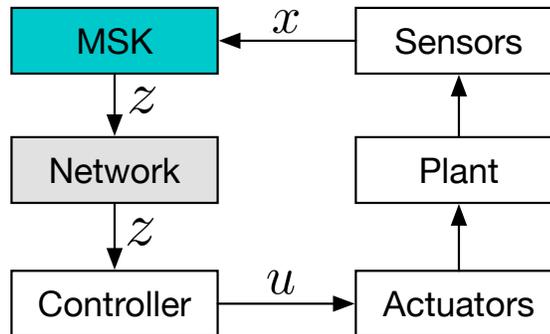


Figure 4.3: Schematic of the proposed model-based scheme, referred as masking with system kernel (MSK).

The proposed model-based scheme is referred as masking with system kernel (MSK) to specify the mechanism that is used for preserving the privacy of the sensor measurements, against an eavesdropper, in the aforementioned NCS. In masking with system kernel (MSK), the sensor measurements $\{x_t\}_{t \geq 0}$ are masked before transmission by adding a vector w_t , that is randomly chosen from the kernel² of the control gain K , to obtain the following masked sensor measurements (or masked states) z_t ,

$$z_t = x_t + w_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.2)$$

Here, w_t is distributed over a compact set $N_t \subset \mathcal{N}(K)$ at each $t \in \mathbb{Z}_{\geq 0}$. The specifications on the probability density of w_t and the compact set N_t are given in the later part of the chapter.

Instead of the true state measurements x_t the sensors transmit z_t over the network. The controller now computes the control input u_t using the received masked state measurements as following.

$$u_t = Kz_t = Kx_t, \forall t \in \mathbb{Z}_{\geq 0}$$

Note As $\{u_t\}_{t \in \mathbb{Z}_{\geq 0}}$ remains unchanged, therefore masking of state measurements as above does not affect the state dynamics of the NCS and so the internal states $\{x_t\}_{t \in \mathbb{Z}_{\geq 0}}$ generated by the agents in the distributed collaboration algorithm is preserved.

Now, the view of an eavesdropper constitutes of the masked states $\{z_t\}_{t \in \mathbb{Z}_{\geq 0}}$. Therefore, privacy of sensor measurements against the eavesdropper requires that

²A vector $v \in \mathbb{R}^n$ belongs to the kernel of the matrix $M \in \mathbb{R}^{n \times n}$ if and only if $Mv = 0$.

the masked states $\{z_t\}_{t \in \mathbb{Z}_{\geq 0}}$ leak very little information about the elements of x_t at any time $t \in \mathbb{Z}_{\geq 0}$. Note that, by definition the probability distribution of w_t is also part of eavesdropper's prior knowledge in addition to A , B and K .

4.2.2 Privacy Definition

The privacy of states is defined based on the (information-theoretic) notion of *perfect secrecy* that was introduced by Shannon in his seminal work [9] and has become an integral part of cryptography³ for analyzing privacy by encryption schemes. The definition is the basis for deriving necessary and sufficient conditions under which MSK guarantees privacy of states against an eavesdropper. The following notation is introduced to formally define privacy of states.

Notation: Let X_t , Z_t , W_t and U_t represent the random vectors corresponding to x_t , z_t , w_t and u_t . The sequences of random vectors Z_t and U_t are represented as $\mathcal{Z}_t = \{z_0, \dots, z_t\}$ and $\mathcal{U}_t = \{U_0, \dots, U_t\}$.

Owing to (4.1), the random vectors X_t , Z_t , W_t and U_t are related as following.

$$X_t = (A + BK)^t X_0, Z_t = X_t + W_t \text{ and } U_t = KZ_t \quad (4.3)$$

for every $t \in \mathbb{Z}_{\geq 0}$.

As $U_t = KZ_t, \forall t \in \mathbb{Z}_{\geq 0}$ and K is assumed public, thus

$$f_{X_r | Z_t, \mathcal{U}_t}(x_r) = f_{X_r | Z_t}(x_r), \forall r, t \in \mathbb{Z}_{\geq 0} \quad (4.4)$$

³For further details on *perfect secrecy*, refer to Chapter 2 in [68]

In other words, the control inputs $u_t, \forall t \in \mathbb{Z}_{\geq 0}$ provide no additional information on x_r (for any $r \in \mathbb{Z}_{\geq 0}$) than what is already available from the masked states $z_t, \forall t \in \mathbb{Z}_{\geq 0}$. In fact, any transformation of z_t that depends on the system parameters A, B or K provides no additional information on x_r (for any $r \in \mathbb{Z}_{\geq 0}$) than what is already available from $z_t, \forall t \in \mathbb{Z}_{\geq 0}$ as A, B and K are assumed public (known to the eavesdropper).

For a given time $t \in \mathbb{Z}_{\geq 0}$ and $Z_s = z_s$ for all $s \in \{0, \dots, t\}$, X_s is contained in the set $M_s(z_s)$ (as $w_s \in N_s, \forall s \in \{0, \dots, t\}$), which is defined as

$$M_s(z_s) = (x_s \in \mathbb{R}^n; z_s - x_s \in N_s)$$

for all $s \in \{0, \dots, t\}$. As $x_s = (A + BK)^s x_0$, this implies that X_0 is contained in

$$\mathcal{K}_{t,0}(\mathbf{z}_t) = \bigcap_{s=0}^t K_s(z_s) \quad (4.5)$$

where,

$$K_s(z_s) = (x_0 \in \mathbb{R}^n; z_s - (A + BK)^s x_0 \in N_s), \forall s \in \{0, \dots, t\} \quad (4.6)$$

for a given sequence of masked states $\mathbf{z}_t = \{z_0, \dots, z_t\}$.

This implies that

$$f_{X_0|z_t}(x_0|\mathbf{z}_t) = 0, \forall x_0 \notin \mathcal{K}_{t,0}(\mathbf{z}_t) \quad (4.7)$$

This equality holds regardless of the probability distribution of w_t . Evidently, from (4.7) it is quite obvious that MSK can not prevent disparity between the *priori* (before observing the masked states) and *posteriori* (after observing the masked states) probability distributions of X_s for any $s \in \{0, \dots, t\}$.

It is clear from the above discussion that an eavesdropper can determine the set of possible values of the initial state x_0 , i.e. $\mathcal{K}_{t,0}(\mathbf{z}_t)$ by observing the masked states from time 0 to t . Consequentially, it can also determine the sets of possible values for states at subsequent times. However, if the design of MSK can ensure that it is impossible to distinguish between any two state values of x_0 in $\mathcal{K}_{t,0}(\mathbf{z}_t)$, privacy of states can be guaranteed. Alternately, privacy of x_0 can be guaranteed if the knowledge of masked states does not affect the *a posteriori* probability distribution of X_0 in the set $\mathcal{K}_{t,0}(\mathbf{z}_t)$. This intuition is used to formulate the following definition of privacy of states.

Definition 4.2.1. MSK guarantees the *privacy of states* if both the following conditions, **C1** and **C2**, hold for every $t \in \mathbb{Z}_{\geq 0}$ and for any given sequence of masked states $\mathbf{z}_t = \{z_0, \dots, z_t\}$.

C1 $f_{X_r|\mathcal{Z}_t}(x_r|\mathbf{z}_t) = f_{X_r \in \mathcal{K}_{t,r}(\mathbf{z}_t)}(x_r)$ for every $r \in \{0, \dots, t\}$.

C2 For every $r \in \{0, \dots, t\}$, there exists a vector $v_r \in \mathbb{R}^n$ with all non-zero elements ($0 < |v_r[i]|, \forall i$) such that

$$[z_r - v_r, z_r + v_r] \subseteq \mathcal{K}_{t,r}(\mathbf{z}_t)$$

Here, $\mathcal{K}_{t,r}(\mathbf{z}_t) = (x \in \mathbb{R}^n; (A + BK)^{-r}x \in \mathcal{K}_{t,0}(\mathbf{z}_t))$.

The value $\delta_r = \min_{i=1}^n |v_r[i]|$ is the *degree of privacy* for state at time $r \in \{0, \dots, t\}$.

Note: v_t for any time $t \in \mathbb{Z}_{\geq 0}$ in the definition above should not independent on the values of the masked states $\{z_0, z_1, \dots\}$.

Condition **C1** ensures that for any given sequence of masked states

$$\mathcal{Z}_t = \mathbf{z}_t = \{z_0, \dots, z_t\}, t \in \mathbb{Z}_{\geq 0},$$

the *posteriori* probability distribution of X_r , $r \in \{0, \dots, t\}$ should be the same as its *priori* probability distribution in $\mathcal{K}_{t,r}(\mathbf{z}_t)$. Clearly, **C1** is not as strong a condition as the *perfect secrecy*⁴, which is impossible to achieve using MSK⁵. However, condition **C1** implies the following equality for every $t \in \mathbb{Z}_{\geq 0}$ and $r \in \{0, \dots, t\}$,

$$f_{\mathcal{Z}_t|X_r}(\mathbf{z}_t|x_r) = f_{\mathcal{Z}_t|X_r}(\mathbf{z}_t|x'_r) \quad (4.8)$$

where, x_r and x'_r are any two points in $\mathcal{K}_{t,r}(\mathbf{z}_t)$. The above implies that it is impossible to distinguish between any two values of X_r , $r \in \{0, \dots, t\}$ in $\mathcal{K}_{t,r}(\mathbf{z}_t)$ given the masked states \mathbf{z}_t .

Now, for a given sequence of masked states \mathbf{z}_t the set of possible values of X_0 is given by $\mathcal{K}_{t,0}(\mathbf{z}_t)$ (refer (4.5)). It is not difficult to see that the set $\mathcal{K}_{t,0}(\mathbf{z}_t)$ is non-increasing with respect to t , that is

$$\mathcal{K}_{t+1,0}(\mathbf{z}_{t+1}) \subseteq \mathcal{K}_{t,0}(\mathbf{z}_t), \forall t \in \mathbb{Z}_{\geq 0}$$

Thus, it is very much possible that there exists a $T \in \mathbb{N}$ such that $\mathcal{K}_{t,0}(\mathbf{z}_t)$ is singleton⁶ for all $t \geq T$. This is demonstrated by the following numerical example

⁴This is an obvious limitation as *perfect secrecy* is usually studied for the case of finite fields, whereas here we are dealing with state vectors in n -dimensional space of real numbers

⁵In short, perfect secrecy would have required equivalence between the priori and posteriori probability distributions, that is $f_{X_0|\mathcal{Z}_t}(x_0|\mathbf{z}_t) = f_{X_0}(x_0)$, $\forall x_0 \in \mathbb{R}^n$. Hence, perfect secrecy is impossible to achieve here as $x_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)$ given the masked states \mathbf{z}_t .

⁶ $\mathcal{K}_{t,r}(\mathbf{z}_t)$ is guaranteed to be non-empty from the very design of the MSK.

in which the set $\mathcal{K}_{t,0}(\mathbf{z}_t)$ indeed reduces to singleton set for t greater than some particular value. (Note that $\mathcal{K}_{t,0}(\mathbf{z}_t)$ is singleton if and only if $\mathcal{K}_{t,r}(\mathbf{z}_t)$ is singleton for all $r \in \{1, \dots, t\}$.)

Example to justify the need for condition C2:

Consider the following LTI state-dynamics of the plant

$$x_{t+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t, u_t = \begin{bmatrix} 1 & 1 \end{bmatrix} x_t$$

where $x_t \in \mathbb{R}^2$ for every $t \in \mathbb{Z}_{\geq 0}$. The system parameters are as follows

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ and } K = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Here, $\mathcal{N}(K) = \text{span}\{v_o\}$ with $v_o = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$.

Let the states be masked using MSK as given by (4.2) with $W_t \in U(N_t)$, where N_t is a line segment of finite length along the vector v_o for $\forall t \in \mathbb{Z}_{\geq 0}$.

Now, in this case w_t is of the form $\lambda_t v_o$, $\lambda_t \in \mathbb{R}$ for all $t \in \mathbb{Z}_{\geq 0}$. Consider the mask vectors z_0 and z_1 , given as following

$$z_0 = x_0 + w_0, z_1 = x_1 + w_1$$

As $x_1 = Ax_0$, $w_0 = \lambda_0 v_o$, $w_1 = \lambda_1 v_o$ and $v_o = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$, we get the following set of equations

$$z_0[1] = x_0[1] - \lambda_0, z_0[2] = x_0[2] + \lambda_0$$

$$z_1[1] = x_0[1] + x_0[2] - \lambda_1, z_1[2] = x_0[2] + \lambda_1$$

or simply,

$$\begin{bmatrix} z_0[1] \\ z_0[2] \\ z_1[1] \\ z_1[2] \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0[1] \\ x_0[2] \\ \lambda_0 \\ \lambda_1 \end{bmatrix} \quad (4.9)$$

As the matrix in the RHS of (4.9) is non-singular, this implies that state vector x_0 can be uniquely determined for the given masked states z_0 and z_1 . In other words, the set $\mathcal{K}_{t,0}(\mathbf{z}_t)$ is singular for all $t \geq 1$.

From Lemma 4.2.1, the masking of states in the form of MSK (4.2) with the aforementioned specifications on the masks w_t satisfies condition **C1** of Definition 4.2.1. However, it has been shown that the states can be uniquely determined from just observing the first 2 masked states. Therefore, not only this example explains the reason for condition **C2** in Definition 4.2.1, it also suggests that MSK can not preserve privacy of states for *any* NCS.

Clearly, the privacy of state x_0 is immediately lost if $\mathcal{K}_{t,0}(\mathbf{z}_t)$ reduces to a singleton set after some time. Therefore, we require an additional condition **C2** to ensure a lower bound on the size of $\mathcal{K}_{t,r}(\mathbf{z}_t)$, $\forall r \in \{0, \dots, t\}$ for any sequence of masked states $\mathcal{Z}_t = \mathbf{z}_t$, $t \in \mathbb{Z}_{\geq 0}$. Condition **C2** implies that for every $r \in \{0, \dots, t\}$ there exists a vector v_r with all non-zero elements such that the set $[z_r - v_r, z_r + v_r]$ is contained in $\mathcal{K}_{t,r}(\mathbf{z}_t)$.

The reason v_r should have all non-zero elements is to ensure privacy of each and every element of the state at time $r \in \{0, \dots, t\}$.

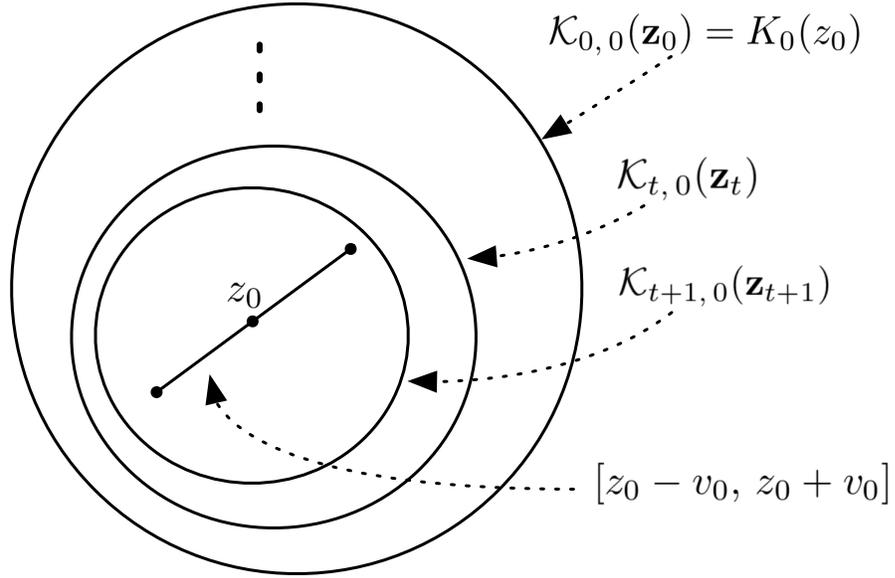


Figure 4.4: Illustration of the evolution of sets $\mathcal{K}_{t,0}(\mathbf{z}_t)$ with time t and the implication of condition **C2** given in the definition of privacy of states.

The implication of **Definition 4.2.1** (and perhaps a more natural definition of privacy of states in this case) can be summarized in the following remark⁷.

Remark 4.2.1. If MSK guarantees privacy of states, then for any $t \in \mathbb{Z}_{\geq 0}$ and any given sequence of t masked states \mathbf{z}_t there exists a vector $v_r \in \mathbb{R}^n$ (with all non-zero elements), such that

$$f_{\mathbf{z}_t|X_r}(\mathbf{z}_t|x_r) = f_{\mathbf{z}_t|X_r}(\mathbf{z}_t|x'_r), \forall r \in \{0, \dots, t\} \quad (4.10)$$

for all x_r, x'_r in $[z_r - v_r, z_r + v_r]$.

Thus, *privacy of states* as defined above guarantees that it is impossible to

⁷To verify this, refer to the supporting argument of (4.8)

distinguish between two state values from the set $[z_t - v_t, z_t + v_t]$ at any time t , even after observing the entire sequence of masked states.

Note the following remark expositing the importance of $v_r, \forall r \in \{0, \dots, t\}$ having all non-zero elements and *degree of privacy*.

Remark 4.2.2. From (4.10), each element $X_r[i], i \in \{1, \dots, n\}$ can take any value in $[z_r[i] - v_r[i], z_r[i] + v_r[i]]$ for a given sequence of masked states \mathbf{z}_t . Therefore, the size of the range of possible values for i -th element of state at time $r \in \{0, \dots, t\}$ is equal to $2|v_r[i]|$. Hence, δ_r is the factor that determines lower bound⁸ on the size of the set of possible values for each element of the state at r (as v_r at any time $r \in \mathbb{Z}_{\geq 0}$ is independent of the sequence of masked states). Thus, *degree of privacy* δ_t is a critical measure of the extent of privacy of state at any time $t \in \mathbb{Z}_{\geq 0}$.

The subsequent subsection presents the necessary and sufficient conditions under which MSK guarantees privacy of states, in the formal sense as defined above.

4.2.3 Privacy Analysis

The following result follows from an existing result on state estimation of an LTI system in the presence of independent and uniformly distributed measurement noise (cf. Servi and Ho, 1981 [69]).

Lemma 4.2.1. (Modified version of the Theorem in [69]) If $W_t \sim U(N_t), \forall t \in \mathbb{Z}_{\geq 0}$ where N_t is any compact subset of $\mathcal{N}(K)$ then MSK satisfies **C1**.

⁸It is a lower bound as it is concerned with only the possible values along a single vector v_r at any time r .

Proof. Let the probability density of W_t be

$$f_{W_t}(w_t) = \begin{cases} \gamma_t & \forall w_t \in N_t \\ 0 & \text{otherwise} \end{cases}, \forall t \in \mathbb{Z}_{\geq 0}$$

where γ_t is some positive real-valued number for every $t \in \mathbb{Z}_{\geq 0}$. (The actual value of γ_t is not consequential in the proof.)

As the random vectors w_t are independent of state vector x_t at every $t \in \mathbb{Z}_{\geq 0}$, this implies

$$f_{W_t|X_0}(w_t) = \begin{cases} \gamma_t & \forall w_t \in N_t \\ 0 & \text{otherwise} \end{cases}, \forall t \in \mathbb{Z}_{\geq 0}$$

It should be noted that for deriving the conditional probability above, we have implicitly used the fact that $x_0 = (A + BK)^{-t}x_t, \forall t \in \mathbb{Z}_{\geq 0}$.

Consequentially, the conditional probability distribution of Z_t at any time $t \in \mathbb{Z}_{\geq 0}$ for a given $X_0 = x_0$ is as following (refer to (4.2))

$$f_{Z_t|X_0}(z_t|x_0) = \begin{cases} \gamma_t & \forall x_0 \in K_t(z_t) \\ 0 & \text{otherwise} \end{cases}$$

As the random vectors $w_t, t = 0, 1, \dots$ are independent of each other, this implies that distributions of $z_t, t = 0, 1, \dots$ are independent of each other, given x_0 . Therefore,

$$f_{Z_t|X_0}(\mathbf{z}_t|x_0) = \begin{cases} \prod_{s=0}^t \gamma_s & \forall x_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t) \\ 0 & \text{otherwise} \end{cases} \quad (4.11)$$

where, $\mathcal{K}_{t,0}(\mathbf{z}_t) = \bigcap_{s=0}^t K_s(z_s)$.

Now, from Baye's rule we get

$$f_{X_0|Z_t}(x_0|\mathbf{z}_t) = \frac{f_{Z_t|X_0}(\mathbf{z}_t|x_0)f_{X_0}(x_0)}{\int_{x \in \mathbb{R}^n} f_{Z_t|X_0}(\mathbf{z}_t|x)f_{X_0}(x)dx}$$

From (4.11),

$$\int_{x \in \mathbb{R}^n} f_{Z_t|X_0}(\mathbf{z}_t|x)f_{X_0}(x)dx = Pr(X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)) \prod_{s=0}^t \gamma_s$$

This implies,

$$f_{X_0|Z_t}(x_0|\mathbf{z}_t) = \frac{f_{Z_t|X_0}(\mathbf{z}_t|x_0)f_{X_0}(x_0)}{Pr(X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)) \prod_{s=0}^t \gamma_s}$$

or simply

$$f_{X_0|Z_t}(x_0|\mathbf{z}_t) = f_{X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)}(X_0 = x_0),$$

for every $t \in \mathbb{Z}_{\geq 0}$. As $A + BK$ is assumed non-singular and $X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)$ is equivalent to $X_r \in \mathcal{K}_{t,r}(\mathbf{z}_t)$ for any $r \in \mathbb{N}$, we get

$$f_{X_r|Z_t}(x_r|\mathbf{z}_t) = f_{X_r \in \mathcal{K}_{t,r}(\mathbf{z}_t)}(X_r = x_r), \forall r, t \in \mathbb{Z}_{\geq 0}$$

This concludes the proof. ■

According to Lemma 4.2.1, if the mask vectors w_t are independently and uniformly distributed in some compact subset N_t of $\mathcal{N}(K)$ for every $t \in \mathbb{Z}_{\geq 0}$ then **C1** holds. However, mere random selections of w_t , $t \in \mathbb{Z}_{\geq 0}$ does not guarantee **C2**, as has been shown by the example in Section 4.2.2.

Using Lemma 4.2.1 and the fact that at any time t the states x_t are masked/masked by adding a random vector from $\mathcal{N}(K)$, we get the following necessary condition for MSK to guarantee privacy of states.

Theorem 4.2.1. For the considered NCS with state dynamics (4.1), MSK guarantees privacy of states *only if* $\exists v \in \mathcal{N}(K)$ with all non-zero elements such that $A^t v \in \mathcal{N}(K), \forall t \in \mathbb{Z}_{\geq 0}$.

Proof. If MSK guarantees privacy of states, then both **C1** and **C2** should hold. From Lemma 4.2.1 we know that **C1** holds if $w_t \sim U(N_t), \forall t \in \mathbb{Z}_{\geq 0}$, regardless of A, B and K (as long as $A + BK$ is non-singular).

If **C2** holds then for every $t \in \mathbb{Z}_{\geq 0}$ there exists a vector $v_t \in \mathcal{N}(K)$ (with all non-zero elements) such that for a given value of z_t , two *distinct* possible values of state at t are

$$x_t = z_t - (1 - 2\lambda_t)v_t,$$

$$x'_t = z_t - (1 - 2\lambda'_t)v_t$$

As the state evolution (given by (4.1)) is preserved under MSK, corresponding to the above possible values of state at t , we have the following possibilities for state at $t + 1$

$$x_{t+1} = (A + BK)z_t - (1 - 2\lambda_t)Av_t,$$

$$x'_{t+1} = (A + BK)z_t - (1 - 2\lambda'_t)Av_t$$

For a given value of z_{t+1} , the values x_{t+1} and x'_{t+1} are possible for state at $t + 1$ only if $x_{t+1} - x'_{t+1} \in \mathcal{N}(K)$, i.e $Av_t \in \mathcal{N}(K)$.

This implies, $(A + BK)^\tau v_t = A^\tau v_t, \forall \tau \in \mathbb{Z}_{\geq 0}$. Therefore, after τ steps, the

following two distinct states

$$x_{t+\tau} = (A + BK)^\tau z_t - (1 - 2\lambda_t)A^\tau v_t,$$

$$x'_{t+\tau} = (A + BK)^\tau z_t - (1 - 2\lambda'_t)A^\tau v_t$$

are possible only if $A^\tau v_t \in \mathcal{N}(K), \forall \tau \in \mathbb{Z}_{\geq 0}$. ■

It is worth noting that the aforementioned condition is necessary for having multiple possibilities of x_t with distinct elements (at all times), given the masked states $\{z_0, \dots, z_t, \dots\}$. Therefore, Theorem 4.2.1 holds regardless of how the privacy of states is defined.

It is interesting to note here that it is impossible to *observe* the states x_t from the control input u_t under the necessary condition given above, as is formally shown by the following Lemma:

Lemma 4.2.2. Unobservability of the pair $(A + BK, K)$ is equivalent to existence of a *non-zero* vector $v \in \mathcal{N}(K)$ s.t. $A^t v \in \mathcal{N}(K), \forall t \in \mathbb{Z}_{\geq 0}$.

Proof. We know that $(A + BK, K)$ is unobservable if and only if the observability matrix

$$\mathcal{O} = \begin{bmatrix} K \\ K(A + BK) \\ \vdots \\ K(A + BK)^{n-1} \end{bmatrix}$$

has rank less than n . To verify the ‘if’ part, just assume a non-zero v in $\mathcal{N}(K)$ (dimension of $\mathcal{N}(K)$ is at least one) such that $Av \in \mathcal{N}(K)$ then it is quite obvious

that $\mathcal{O}v = 0_{nm}$ and so $\text{rank}(\mathcal{O}) < n$. To verify the ‘only if’ part, let \mathcal{O} be rank-deficient then $\exists v \in \mathbb{R}^n$ such that

$$\mathcal{O}v = \begin{bmatrix} Kv \\ K(A+BK)v \\ \vdots \\ K(A+BK)^{n-1}v \end{bmatrix} = 0_{nm}$$

For this to happen, v should belong to $\mathcal{N}(K)$. Further, this v should also satisfy $K(A+BK)^t v = KA^t v = 0_m, \forall t \in \{1, \dots, n-1\}$ (as $v \in \mathcal{N}(K)$ is already inferred). Therefore, if $A^t v \notin \mathcal{N}(K)$ for any t then \mathcal{O} does not have a non-trivial kernel (using the Cayley-Hamilton Theorem). ■

It is well-known in control theory that one can observe x_t of system (4.1) from the control input u_t if and only if $(A+BK, K)$ is observable. Thus, from the above result it is obvious that we can not observe x_t from u_t if the condition given in Theorem 4.2.1 is satisfied.

It turns out, a slight variation of the above necessary condition is also sufficient for MSK to guarantee privacy of states, as is formally demonstrated in the following theorem.

Theorem 4.2.2. For the considered NCS with state dynamics (4.1), if there exists a vector $v \in \mathcal{N}(K)$ with all non-zero elements such that $Av = \mu v, \mu \neq 0$ then MSK guarantees privacy of the states with degree of privacy

$$\delta_t = 2\mu^t d_o \min_{i=1}^n |v[i]|, \text{ where}$$

1. $W_0 \sim U(N_0), N_0 = [-d_o v, d_o v], d_o \in \mathbb{R}_{>0},$

$$2. w_t = \mu^t w_0, \forall t \in \mathbb{N}$$

and d_o is any positive real-valued number.

Proof. Consider any $t \in \mathbb{Z}_{\geq 0}$ and sequence of masked states till time t , $\mathbf{z}_t = \{z_0, \dots, z_t\}$.

As $w_t = \mu^t w_0$, we get

$$z_t = x_t + w_t = (A + BK)^t z_0, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.12)$$

Consequentially,

$$Z_t = (A + BK)^t Z_0, \forall t \in \mathbb{Z}_{\geq 0}$$

The aforementioned relationship between the masked states in time implies that

$$f_{X_0|Z_t}(x_0|\mathbf{z}_t) = f_{X_0|Z_0}(x_0|z_0), \forall t \in \mathbb{Z}_{\geq 0} \quad (4.13)$$

From Baye's rule, we know that

$$f_{X_0|Z_0}(x_0|z_0) = \frac{f_{Z_0|X_0}(z_0|x_0)f_{X_0}(x_0)}{\int_{x \in \mathbb{R}^n} f_{Z_0|X_0}(z_0|x)f_{X_0}(x)dx} \quad (4.14)$$

As $w_0 \sim U([-d_o v, d_o v])$ and is independent of x_0 , this implies

$$f_{Z_0|X_0}(z_0|x_0) = \begin{cases} \gamma_0 & \forall x_0 \in K_0(z_0) \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

where γ_0 is some positive real number (the actual value is not required for the proof).

(From (4.6), the set $K_0(z_0) = (x \in \mathbb{R}^n; z_0 - x \in N_0)$.)

Thus,

$$\int_{x \in \mathbb{R}^n} f_{Z_0|X_0}(z_0|x) f_{X_0}(x) dx = \gamma_0 \int_{x \in K_0(z_0)} f_{X_0}(x) dx = \gamma_0 Pr(X_0 \in K_0(z_0)) \quad (4.16)$$

Substituting (4.15) and (4.16) in (4.14) yields

$$f_{X_0|Z_0}(x_0|z_0) = \begin{cases} \frac{f_{X_0}(x_0)}{Pr(X_0 \in K_0(z_0))} & \forall x_0 \in K_0(z_0) \\ 0 & \text{otherwise} \end{cases}$$

or simply,

$$f_{X_0|Z_0}(x_0|z_0) = f_{X_0 \in K_0(z_0)}(x_0) \quad (4.17)$$

Now, we first show that $K_0(z_0)$ and $\mathcal{K}_{t,0}(\mathbf{z}_t)$ are indeed equivalent in this case.

As $w_0 \in N_0$ and $N_0 = [-d_o v, d_o v]$, this implies that for any $x_0 \in K_0(z_0)$ we can write

$$z_0 - x_0 = -\lambda d_o v + (1 - \lambda) d_o v \quad (4.18)$$

where, λ is some value in $[0, 1]$. As $v \in \mathcal{N}(K)$ is the right eigenvector of A with $Av = \mu v$, we get

$$(A + BK)^t v = \mu^t v, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.19)$$

From (4.18) and (4.19), we get

$$(A + BK)^t z_0 - (A + BK)^t x_0 = -\lambda \mu^t v + (1 - \lambda) \mu^t v \quad (4.20)$$

for every $t \in \mathbb{Z}_{\geq 0}$. From (4.12) and the fact that $w_t = \mu^t w_0$, (4.20) is equivalent to the following

$$z_t - (A + BK)^t x_0 = w_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.21)$$

Therefore, from (4.21) it is quite evident that if $x_0 \in K_0(z_0)$ then $x_0 \in K_t(z_t)$, $\forall t \in \mathbb{Z}_{\geq 0}$. Inferentially, $K_0(z_0) \subseteq \mathcal{K}_{t,0}(\mathbf{z}_t)$. However, $\mathcal{K}_{t,0}(\mathbf{z}_t) \subseteq K_0(z_0)$ as $\mathcal{K}_{t,0}(\mathbf{z}_t) = \bigcap_{s=0}^t K_s(z_s)$. This implies that

$$\mathcal{K}_{t,0}(\mathbf{z}_t) = K_0(z_0), \forall t \in \mathbb{Z}_{\geq 0} \quad (4.22)$$

Equation (4.22) has the following twofold implications.

(i) Combining (4.22) with (4.13) and (4.17) gives

$$f_{X_0|\mathbf{z}_t}(x_0|\mathbf{z}_t) = f_{X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)}(x_0)$$

As $A + BK$ is non-singular, the equality above implies that⁹

$$f_{X_r|\mathbf{z}_t}(x_r|\mathbf{z}_t) = f_{X_r \in \mathcal{K}_{t,r}(\mathbf{z}_t)}(x_r)$$

for every $r \in \{0, \dots, t\}$. Here, $\mathcal{K}_{t,r}(\mathbf{z}_t) = \{x \in \mathbb{R}^n; (A + BK)^{-r}x \in \mathcal{K}_{t,0}(\mathbf{z}_t)\}$.

Hence, condition **C1** of Definition 4.2.1 is satisfied.

(ii) From (4.22), we get $\mathcal{K}_{t,0}(\mathbf{z}_t) = [z_0 - d_0v, z_0 + d_0v]$. Now, using (4.12) and (4.19) here implies that

$$\mathcal{K}_{t,r}(\mathbf{z}_t) = [z_r - \mu^r d_0v, z_r + \mu^r d_0v], \forall r \in \{0, \dots, t\} \quad (4.23)$$

Hence, condition **C2** of Definition 4.2.1 is satisfied with $v_r = \mu^r v$, $\forall r \in \mathbb{Z}_{\geq 0}$.

⁹If two random vectors S and R satisfy $S = TR$, where T is a (known) non-singular matrix (ref. [70]), then

$$f_S(s) = \frac{1}{|\det(T)|} f_R(T^{-1}s)$$

Note that the conclusions above holds for any $t \in \mathbb{Z}_{\geq 0}$ and any sequence of masked states \mathbf{z}_t generated by the MSK (in (4.2)) with the given specifications.

Clearly, from (4.23) the degree of privacy of the state at time $t \in \mathbb{Z}_{\geq 0}$ is given by $\delta_t = 2\mu^t d_o \min_{i=1}^n |v[i]|$. ■

As a corollary of Theorems 4.2.1 and 4.2.2, we have

Corollary 4.2.1. If the dimension of $\mathcal{N}(K)$ is one, then MSK guarantees privacy of states (for the considered NCS) *if and only if* there exists a vector v in $\mathcal{N}(K)$ with all non-zero elements and $Av = \mu v$, $\mu \neq 0$.

In case the dimension of $\mathcal{N}(K)$ is equal to $m > 1$, then it is not necessary for a single vector v *with all non-zero elements* to be the right-eigenvector of A . Instead, we can have $m \in \{1, \dots, n-1\}$ independent vectors $v_j \in \mathcal{N}(K)$ that satisfy $Av_j = \mu_j v_j$, $\mu_j \neq 0$, $\forall j \in \{1, \dots, m\}$, and each vector v_j , $j \in \{1, \dots, m\}$ *can have some elements equal to zero*. In this scenario, MSK can guarantee privacy of states if the sum (or weighted sum) of these vectors, $\sum_{j=1}^m v_j$, has all non-zero elements. This result is formally stated in the following theorem.

Theorem 4.2.3. For the considered NCS with state dynamics (4.1), *if* there exists $m > 0$ independent vectors $v_j \in \mathcal{N}(K)$ such that $Av_j = \mu_j v_j$, $\mu_j \neq 0$, $\forall j \in \{1, \dots, m\}$ and

$$v = \sum_{i=1}^m v_i \tag{4.24}$$

has all non-zero elements then MSK guarantees privacy of states with degree of privacy $\delta_t = 2d_o \min_{i=1}^n |\sum_{j=1}^m v_j[i]|$, where

1. $w_0 = \sum_{j=1}^m \omega_j$, where each ω_j is uniformly and independently chosen from $[-d_o v_i, d_o v_i]$, $\forall j \in \{1, \dots, m\}$,
2. $w_t = \sum_{j=1}^m \mu_j^t \omega_j$, $\forall t \in \mathbb{N}$

and d_o is any positive real-valued number and $m \in \{1, \dots, n-1\}$.

Proof. (The proof uses the results deduced in the proof of Theorem 4.2.2.)

Consider any $t \in \mathbb{Z}_{\geq 0}$ and let $\mathbf{z}_t = \{z_0, \dots, z_t\}$ be a given sequence of $t+1$ masked states generated by the MSK (refer to (4.2)) with the given specifications for w_t .

As $Av_j = \mu_j v_j$ and $v_j \in \mathcal{N}(K)$ for every $j \in \{1, \dots, m\}$, this implies

$$(A + BK)^t v_j = \mu_j^t v_j, \quad \forall j \in \{1, \dots, m\}$$

for every $t \in \mathbb{Z}_{\geq 0}$. Therefore $w_t = (A + BK)^t w_0$, $\forall t \in \mathbb{Z}_{\geq 0}$ and so,

$$Z_t = X_t + W_t = (A + BK)^t Z_0, \quad \forall t \in \mathbb{Z}_{\geq 0}$$

Consequentially,

$$f_{X_0|\mathbf{z}_t}(x_0|\mathbf{z}_t) = f_{X_0|Z_0}(x_0|z_0), \quad \forall t \in \mathbb{Z}_{\geq 0}$$

Using the arguments given in the proof of Theorem 4.2.2 it is easy to infer that

$$f_{X_0|Z_0}(x_0|z_0) = f_{X_0 \in K_0(z_0)}(x_0) \tag{4.25}$$

(only here, unlike Theorem 4.2.2, N_t , $t \in \mathbb{Z}_{\geq 0}$ is not just a line segment but is given as following)

$$N_t = \left(\sum_{j=1}^m \left\{ -\lambda_j \mu_j^t d_j v_j + (1 - \lambda_j) \mu_j^t d_j v_j \right\} \middle| \lambda_j \in [0, 1] \right) \tag{4.26}$$

For any $x_0 \in K_0(z_0)$ (from (4.26)),

$$z_0 - x_0 = w_0 = \sum_{j=1}^m \{\lambda_j(-d_j v_j) + (1 - \lambda_j)(d_j v_j)\}$$

where, $\lambda_j \in [0, 1], \forall j \in \{1, \dots, m\}$. This implies

$$\begin{aligned} & (A + BK)^t z_0 - (A + BK)^t x_0 \\ &= \sum_{j=1}^m \{\lambda_j(-\mu_j^t d_j v_j) + (1 - \lambda_j)(\mu_j^t d_j v_j)\}, \\ &= \sum_{j=1}^m \mu_j^t \{\lambda_j(-d_j v_j) + (1 - \lambda_j)(d_j v_j)\} = w_t, \forall t \in \mathbb{Z}_{\geq 0} \end{aligned}$$

This is equivalent to

$$z_t - (A + BK)^t x_0 = w_t \in N_t, \forall t \in \mathbb{Z}_{\geq 0}$$

Therefore, if $x_0 \in K_0(z_0)$ then $x_0 \in K_t(z_t), \forall t \in \mathbb{Z}_{\geq 0}$.

Inferentially, $K_0(z_0) \subseteq \mathcal{K}_{t,0}(\mathbf{z}_t)$. However, we know that $\mathcal{K}_{t,0}(\mathbf{z}_t) \subseteq K_0(z_0)$ as $\mathcal{K}_{t,0}(\mathbf{z}_t) = \bigcap_{s=0}^t K_s(z_s)$. This implies

$$\mathcal{K}_{t,0}(\mathbf{z}_t) = K_0(z_0), \forall t \in \mathbb{Z}_{\geq 0} \quad (4.27)$$

Equations (4.25) and (4.27) implies

$$f_{X_0|Z_0}(x_0|z_0) = f_{X_0 \in \mathcal{K}_{t,0}(\mathbf{z}_t)}(x_0)$$

This equality combined with the fact that $A + BK$ is non-singular implies the following for every $r \in \{0, \dots, t\}$

$$f_{X_r|Z_r}(x_r|z_r) = f_{X_r \in \mathcal{K}_{t,r}(\mathbf{z}_t)}(x_r)$$

Hence, condition **C1** of Definition 4.2.1 is satisfied.

Now, note that the line segment $[-d_o v_0, d_o v_0] \subset N_0$, where v_0 is the same as defined in (4.24).

Therefore, $[z_0 - d_o v_0, z_0 + d_o v_0] \subset K_0(z_0) = \mathcal{K}_{t,0}(\mathbf{z}_t)$. Consequentially, as $z_t = (A + BK)^t z_0$, this implies that

$$[z_r - d_o v_r, z_r + d_o v_r] \subset \mathcal{K}_{t,r}(\mathbf{z}_t), \forall r \in \{1, \dots, t\} \quad (4.28)$$

where $v_r = (A + BK)^r v_0 = \sum_{j=1}^m \mu_j^r v_j$. Hence, condition **C2** of Definition 4.2.1 is also satisfied.

Note that the conclusions above holds for any $t \in \mathbb{Z}_{\geq 0}$ and corresponding sequence of masked states till t generated by the MSK (as given by (4.2)) with the given specifications.

Clearly, from (4.28) the degree of privacy for the state at any time t is given by $\delta_t = 2d_o \min_{i=1}^n |\sum_{j=1}^m \mu_j^t v_j[i]|, \forall t \in \mathbb{Z}_{\geq 0}$. ■

The above necessary and sufficient conditions clearly indicate that MSK in the current form can guarantee privacy of states only for a subset of one-channel feedback NCS. Moreover, it is not difficult to see (from the necessary condition in 4.2.1) that if MSK were to guarantee the privacy of states then we can not have desirable pole-placement for the close-loop transfer function.

It is however possible to ensure asymptotic stability of the NCS by appropriately choosing K that simultaneously satisfies the above necessary and sufficient conditions, and Schur stability criterion of the transition matrix $A + BK$. A simple numerical example in the next section demonstrates this. However, in general this not a trivial task, as choosing such a K requires solving a system of bilinear

inequalities.

4.3 Numerical Examples

This section presents an example to illustrate the proposed model-based scheme. First, a simple numerical example (Example I) demonstrates the result in Theorem 4.2.2. Then, a system of two quadrotors (or agents) is considered in Example II, wherein the collaboration objective of the quadrotors is to reach consensus on their elevations while keeping their individual initial elevations private from an eavesdropper listening to the messages being exchanged between the rotors.

4.3.1 Example I

Consider the following state-dynamics of the plant in the considered NCS

$$x_{t+1} = Ax_t + Bu_t, u_t = Kx_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.29)$$

where $x_t \in \mathbb{R}^2, \forall t \in \mathbb{Z}_{\geq 0}$,

$$A = \begin{bmatrix} 1 & \epsilon \\ 0 & -\epsilon k_d + 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \epsilon \end{bmatrix} \text{ and } K = -[k_1, k_2].$$

Here k_d, k_1, k_2 and ϵ are positive real values such that

$$k_1 = k_d k_2 \text{ and } \epsilon \notin 1/k_d.$$

Here,

$$\mathcal{N}(K) = \text{span}\{v\}, v = \begin{bmatrix} -k_2 \\ k_1 \end{bmatrix} \quad (4.30)$$

Clearly, both the elements of v are non-zero (this is one of the required conditions in Theorem 4.2.2). From simple calculations, we get

$$Av = \begin{bmatrix} -k_2 + \epsilon k_1 \\ -\epsilon k_1 k_d + k_1 \end{bmatrix}$$

As $k_1 = k_d k_2$, we get

$$Av = \begin{bmatrix} -k_2 + \epsilon k_1 \\ -\epsilon k_1 k_d + k_1 \end{bmatrix} = (1 - \epsilon k_d)v \quad (4.31)$$

where, $(1 - \epsilon k_d) \neq 0$. Also, note that

$$A + BK = \begin{bmatrix} 1 & \epsilon \\ -\epsilon k_d k_2 & 1 - \epsilon(k_2 + k_d) \end{bmatrix}$$

is non-singular with eigenvalues $1 - \epsilon k_d$ and $1 - \epsilon k_2$.

Therefore, the system parameters in this case satisfy the conditions prescribed in Theorem 4.2.2 with v as given in (4.30) and $\mu = (1 - \epsilon k_d)$. Hence, MSK as given by (4.2) or specifically the masking of x_t as following

$$z_t = x_t + w_t, \forall t \in \mathbb{Z}_{\geq 0}$$

where w_0 is chosen uniformly from the set $[-d_o v, d_o v]$ and $w_t = (1 - \epsilon k_d)^t w_0, \forall t \in \mathbb{N}$ guarantees privacy of state x_t at any time $t \in \mathbb{Z}_{\geq 0}$ against an eavesdropper with unbounded computation power with degree of privacy $d_o(1 - \epsilon k_d)^t \min\{k_2, k_d k_2\}$ for any $t \in \mathbb{Z}_{\geq 0}$. Here, d_o can any positive real number. Consequentially, the control input u_t in (4.29) is replaced by $u_t = K z_t, \forall t \in \mathbb{Z}_{\geq 0}$.

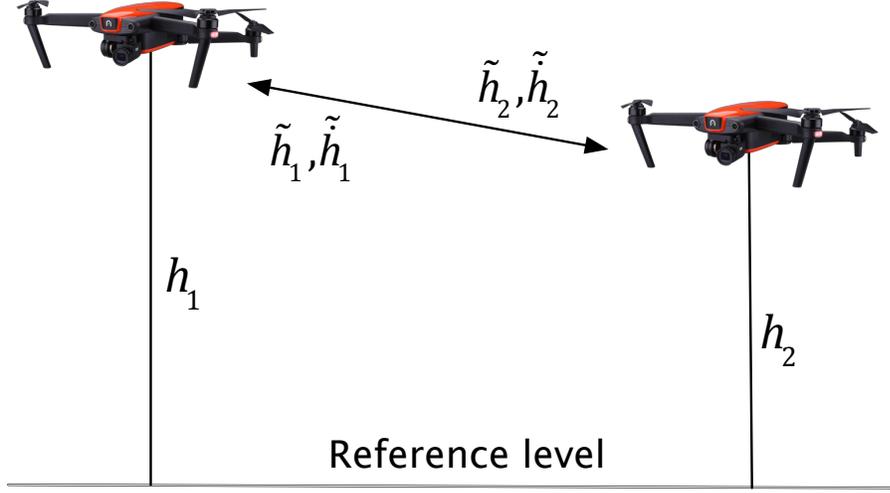


Figure 4.5: An example for illustrating the proposed model-based scheme.

4.3.2 Example II

Consider two quadrotors (hereafter referred to as agents) as shown in Fig. 4.5, collaborating to reach consensus on their elevation. The discrete-time dynamics of the elevation of each agent $i \in \{1, 2\}$ is given as following for all $t \in \mathbb{Z}_{\geq 0}$ (ref. [11]),

$$\begin{aligned} h_{i,t+1} &= h_{i,t} + \epsilon k_d \dot{h}_{i,t} \\ \dot{h}_{i,t+1} &= (1 - \epsilon k_d) \dot{h}_{i,t} + u_{i,t} \end{aligned}$$

where, $h_{i,t}$ and $\dot{h}_{i,t}$ are the elevation and rate of change in elevation, respectively of agent i . Here, k_d and ϵ are positive real-valued constants.

In this example, it is demonstrated that the agents can mask their states $(\{(h_{i,t}, \dot{h}_{i,t})\}_{t \in \mathbb{Z}_{\geq 0}})$ using MSK to protect privacy of their initial elevations $(h_{1,0}, h_{2,0})$ from an eavesdropper and at the same time fulfill their objective of reaching a

common elevation.

Let $x_{i,t} = [h_{i,t}, \dot{h}_{i,t}]^T$. Then, we get

$$x_{i,t+1} = Ax_{i,t} + Bu_{i,t}, \forall t \in \mathbb{Z}_{\geq 0}, i \in \{1, 2\} \quad (4.32)$$

where, $x_{i,t} \in \mathbb{R}^2$ and $u_{i,t} \in \mathbb{R}$ represent the states and the control inputs (commands of the collaboration algorithms) of the agents $i = \{1, 2\}$ at time t . Note that the system parameters A and B are same as in (4.29). However, range of ϵ is different, which is specified later. The combined state dynamics of the two agents is given as following:

$$\begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \end{bmatrix} = \begin{bmatrix} A & \mathbf{0}_2 \\ \mathbf{0}_2 & A \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} + \begin{bmatrix} B & 0_2 \\ 0_2 & B \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} \quad (4.33)$$

To reach consensus on their elevations, the agents use the following control inputs,

$$u_{i,t} = K(x_{1,t} - x_{2,t}), u_{2,t} = K(x_{2,t} - x_{1,t}), \forall t \in \mathbb{Z}_{\geq 0} \quad (4.34)$$

where $K = -[k_1, k_2]$ is the same as in (4.29) with $k_1 \in \mathbb{R}_{>0}$ and $k_2 \in \mathbb{R}_{>0}$ and $k_1 = k_d k_2$.

Substituting $u_{1,t}$ and $u_{2,t}$ in (4.33) yields

$$x_{t+1} = \begin{bmatrix} A + BK & \mathbf{0}_2 \\ \mathbf{0}_2 & A + BK \end{bmatrix} x_t - \begin{bmatrix} \mathbf{0}_2 & BK \\ BK & \mathbf{0}_2 \end{bmatrix} x_t \quad (4.35)$$

where, $x_t = [(x_{1,t})^T, (x_{2,t})^T]^T$ for every $t \in \mathbb{Z}_{\geq 0}$. We get the following result:

Lemma 4.3.1. If the agents apply the control inputs given in (4.34) with $\epsilon \in (0, \min\{1/k_d, 1/2k_2\})$, then the states $x_{1,t}$ and $x_{2,t}$ exponentially converge to each other.

Proof. Let $e_t = x_{1,t} - x_{2,t}$ be the error between the states of the plants at each time $t \in \mathbb{Z}_{\geq 0}$.

From the state dynamics given in (4.33), we get

$$e_{t+1} = Ae_t + B(u_{1,t} - u_{2,t}), \forall t \in \mathbb{Z}_{\geq 0}$$

Substituting the control inputs given in (4.34) above yields the following error dynamics

$$e_{t+1} = (A + 2BK)e_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.36)$$

Axiomatically, the state-synchronization is equivalent to the origin being asymptotically stable for (4.36). From the well-known result in discrete-time linear systems we know that the origin of (4.36) is asymptotically stable if and only if $A + 2BK$ is a Schur matrix [71]. From simple calculations, we have

$$A + 2BK = \begin{bmatrix} 1 & \epsilon \\ -2\epsilon k_2 k_d & 1 - \epsilon(k_d + 2k_2) \end{bmatrix}$$

with eigenvalues equal to $1 - \epsilon k_d$ and $1 - 2\epsilon k_2$.

Therefore, the necessary and sufficient condition for state-synchronization of (4.33) with the control inputs (4.34) requires:

$$0 < 1 - \epsilon k_d < 1 \text{ and } 0 < 1 - 2\epsilon k_2 < 1 \quad (4.37)$$

For the sake of convenience let $\alpha_t = h_{1,t} + h_{2,t}$ and $\beta_t = \dot{h}_{1,t} + \dot{h}_{2,t}$. From (4.33), we get

$$\alpha_{t+1} = \alpha_t + \epsilon\beta_t \quad (4.38)$$

$$\beta_{t+1} = (1 - \epsilon k_d)\beta_t$$

for every $t \in \mathbb{Z}_{\geq 0}$. From (4.38), for every $t \in \mathbb{N}$ we get

$$\alpha_t = \alpha_0 + \epsilon\beta_0 \sum_{s=0}^{t-1} (1 - \epsilon k_d)^s \text{ and } \beta_t = (1 - \epsilon k_d)^t \beta_0 \quad (4.39)$$

Therefore, if (4.37) is satisfied then from (4.36) and (4.39), we get the following.

$$\lim_{t \rightarrow \infty} x_{1,t} = \lim_{t \rightarrow \infty} x_{2,t} = \frac{1}{2} \begin{bmatrix} \alpha_0 + \frac{1}{k_d} \beta_0 \\ 0 \end{bmatrix} \quad (4.40)$$

Hence, the elevation of both agents converge to the same value, which is given by (4.40). ■

In order to compute the control inputs (4.34), the agents need to exchange values of $\{(h_{i,t}, \dot{h}_{i,t})\}_{t \in \mathbb{Z}_{\geq 0}}$ on the communication link, revealing their initial elevations to any eavesdropper listening to the communication link. Thus, in the following we present how each agent i can mask $\{(h_{i,t}, \dot{h}_{i,t})\}_{t \in \mathbb{Z}_{\geq 0}}$ as $\{(\tilde{h}_{i,t}, \tilde{\dot{h}}_{i,t})\}_{t \in \mathbb{Z}_{\geq 0}}$ using MSK to protect privacy of their initial elevations against such an eavesdropper.

Now, let

$$\mathbf{A} = \begin{bmatrix} A + BK & \mathbf{0}_2 \\ \mathbf{0}_2 & A + BK \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -B & 0_2 \\ 0_2 & -B \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 0_2^T & K \\ K & 0_2^T \end{bmatrix}$$

Using the above notation, (4.35) can be written as

$$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t, u_t = \mathbf{K}x_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.41)$$

Note that $\mathbf{A} + \mathbf{BK}$ is non-singular.

The agents mask their states as following before transmitting them over the communication link.

$$z_{i,t} = x_{i,t} + w_{i,t}, \forall t \in \mathbb{Z}_{\geq 0}, i = \{1, 2\} \quad (4.42)$$

with

$$w_{i,0} \sim U([-d_o v, d_o v]) \text{ and } w_{i,t} = (1 - \epsilon k_d)^t w_{i,0}, \forall t \in \mathbb{N}.$$

and v is same as given in (4.30). Equivalently, $\tilde{h}_{i,t}$ and $\tilde{\tilde{h}}_{i,t}$ are given by the first and the second element $z_{i,t}$, respectively for each $i \in \{1, 2\}$.

Thus, the new control inputs for each each agents are now given by $u_t = \mathbf{K}z_t$.

Equivalently, we get

$$z_t = x_t + w_t, \forall t \in \mathbb{Z}_{\geq 0} \quad (4.43)$$

where, $z_t = [(z_{1,t})^T, (z_{2,t})^T]^T$ and $w_t = [(w_{1,t})^T, (w_{2,t})^T]^T$. Note that the masking of states in (4.43) is equivalent to MSK (refer to (4.2)).

Let

$$v_1 = \begin{bmatrix} v \\ 0_2 \end{bmatrix}, v_2 = \begin{bmatrix} 0_2 \\ v \end{bmatrix} \text{ and } v_0 = v_1 + v_2 = \begin{bmatrix} v \\ v \end{bmatrix} \quad (4.44)$$

Clearly, as v has all non-zero elements, this implies that v_0 also has all non-zero elements. It is easy to verify that both v_1 and v_2 belong to $\mathcal{N}(\mathbf{K})$ with $\mathbf{A}v_1 = (1 - \epsilon k_d)v_1$ and $\mathbf{A}v_2 = (1 - \epsilon k_d)v_2$ where $0 < (1 - \epsilon k_d) < 1$. Therefore, the system parameters satisfy the relationship given in Theorem 4.2.3.

Now, it is easy to verify that masking of x_t in (4.43) satisfies the specifications of MSK given in Theorem 4.2.3.

Therefore, from Theorem 4.2.3, for any given $t \in \mathbb{Z}_{\geq 0}$ and a corresponding sequence of masked states (as generated by (4.43)) \mathbf{z}_t ,

$$f_{\mathbf{z}_t|X_r}(\mathbf{z}_t|x_0) = f_{\mathbf{z}_t|X_r}(\mathbf{z}_t|x_0) \quad (4.45)$$

for all x_0, x'_0 in the set $[z_0 - d_o v_0, z_0 + d_o v_0]$. Here, $X_0 = [(X_{1,0})^T, (X_{2,0})^T]$ represents the random vector for x_0 .

4.4 Concluding Remarks

The proposed model-based scheme can preserve external privacy of agents' inputs in a distributed collaboration algorithms that can be modeled as a one-channel feedback linear time-invariant networked control system whose system parameters satisfy certain conditions. The model-based scheme, referred to as masking with system kernel (MSK), exploits the inherent property of unobservability in certain distributed collaboration algorithms, that involve dynamical systems, to preserve the external privacy of agents' inputs to overcome the shortcomings of cryptographic encryptions.

4.4.1 Limitations

The limitations of the model-based privacy scheme or MSK are as following.

1. The model-based based scheme is only effective for those distributed collaboration algorithms that can be modeled as linear time-invariant networked control systems whose system parameters satisfy a specific condition (given in Theorem 4.2.1). This is a strong limitation of the scheme as in general distributed collaboration algorithms need not be linear or even time-invariant.
2. Often in practical scenarios, the communication links in the network suffer from delays, leading to asynchronicity between agents. In this chapter, delays

in communication links are not considered and therefore agents are assumed to operate in a synchronous manner.

4.4.2 Future Research

In order to overcome the aforementioned limitations, future research can be pursued in the following directions.

1. Investigate a methodology for altering the collaboration algorithm (without affecting the collaboration objective) to satisfy the conditions that are required for guaranteeing external privacy by the proposed model-based scheme.
2. Investigate the extension of external privacy guarantee to the case where systems parameters of the resultant NCS do not satisfy the prescribed sufficient conditions.
3. Study the effect of delays in the communication links for the proposed model-based scheme.
4. Investigate the efficacy of the model-based scheme for distributed collaboration algorithms that have non-linear and time-varying dynamics.
5. Investigate the possibility of using this model-based privacy scheme for exchanging information across network in a secure manner, i.e. explore the possibility of using the model-based scheme for encryption purposes.
6. Formulate a more general privacy definition, where the obscurity in the agents' states in view of the eavesdropper is multi-dimensional.

CHAPTER 5: SUMMARY

This dissertation addressed the issue of privacy in distributed collaboration algorithms by classifying privacy into two types, internal and external privacy, depending on the means through which an adversary (passive) learns information about agents' inputs.

1. *Internal Privacy*: refers to the privacy of agents' inputs against a passive adversary that corrupts some of the agents that are participating in the distributed collaboration.
2. *External Privacy*: refers to the privacy of agents' inputs against a passive adversary reading messages that are exchanged between agents in the intermediate stage of the distributed collaboration.

This dissertation proposed a privacy protocol that can be used for preserving internal privacy of agents in two particular distributed collaboration algorithms; *distributed average consensus* and *distributed optimization*. Further, this dissertation investigated a model-based scheme for external privacy in distributed collaboration algorithms that can be modeled as linear time-invariant networked control systems, as an alternative to cryptographic encryptions.

The contribution, limitations and future directions of the proposed solutions are summarized as following.

5.1 Contribution: Internal Privacy in Distributed Collaboration

The proposed privacy protocol can be interpreted as a general approach for constructing distributed average consensus and distributed optimization algorithms that ensures internal privacy—in a formal sense as described in Chapters 2-3—if the passively adversarial agents (i.e. agents corrupted by a passive adversary) do not constitute a vertex cut of the underlying communication network of the MAS.

The approach constitutes of two phases (as illustrated in Fig. 5.1):

1. In the first phase, the agents run a privacy mechanism to compute new “effective inputs”. The effective inputs are computed based on the original inputs and the correlated random values that are shared between adjacent agents in the privacy mechanism.
2. In the second phase, the agents run an arbitrary distributed collaboration algorithm (Π) on their effective inputs, instead of their original inputs.

The above two-step process has been discussed in detail for the case of distributed average consensus and distributed optimization in Chapters 2 and 3, respectively. The effective inputs, denoted by $\{\tilde{In}_i\}$, that are computed in the first phase satisfy: $\text{Collab}_i(\{In_k\}) = \text{Collab}_i(\{\tilde{In}_k\}), \forall i \in \mathcal{V}$, i.e. the collaboration out-

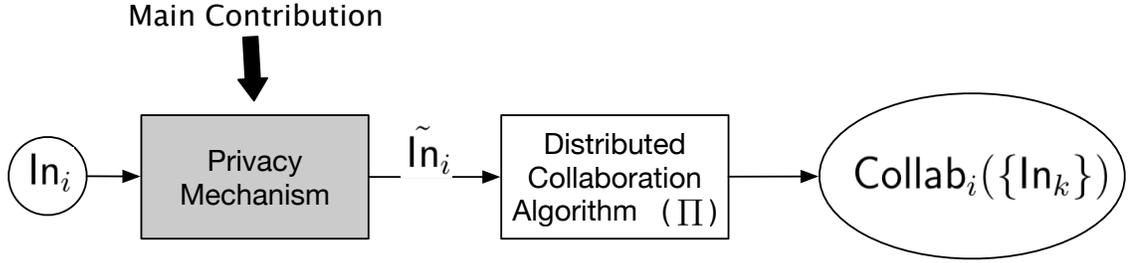


Figure 5.1: In the proposed protocol for internal privacy in distributed collaborations, the agents run a privacy mechanism on their inputs $\{\ln_i\}$ before execute the distributed collaboration algorithm, to compute effective inputs $\{\tilde{\ln}_i\}$ that satisfy $\text{Collab}_i(\{\ln_k\}) = \text{Collab}_i(\{\tilde{\ln}_k\})$.

puts are preserved. The internal privacy in this approach relies on the randomness introduced by the privacy mechanism and the privacy guarantee holds regardless of the distributed collaboration protocol used in the second phase. This is proved by showing that internal privacy holds even if all the effective inputs of the honest agents are revealed to the adversarial agents (worst-case scenario).

5.1.1 Limitations

Limitations of the proposed two phase approach for internal privacy in distributed collaboration (average consensus and optimization) are as following.

1. For now, the privacy protocol has been investigated for internal privacy in only two particular distributed collaborations; distributed average consensus and distributed optimization.
2. In case of distributed average consensus, the agents are required to know the value of n , that is the size of the network.

3. In case of distributed optimization, for now, internal privacy can only be guaranteed for the affine parts of the agents' costs. However, the proposed privacy protocol can be extended easily for internal privacy of higher-order parts (or coefficients) of the agents' costs.
4. This protocol does not prevent loss of information on honest agents' inputs due to their contribution in the adversarial agents' collaboration outputs. Consequentially, the adversary readily knows the aggregate of the honest agents' inputs in both distributed average consensus and distributed optimization.
5. If the adversarial agents form a vertex cut in the communication network, then some of the honest agents might lose privacy of their inputs to the adversary. For instance, if all the neighbors of an honest agent are adversarial then internal privacy of that agent's input is not be protected by the proposed privacy protocol.
6. The proposed privacy protocol can not preserve external privacy of the agents (i.e. privacy of agents' inputs against eavesdroppers). Alternately, it relies on the existing cryptographic encryptions for external privacy.
7. The communication links between the agents in the privacy mechanism (first phase) are assumed undirected. However, the agents are not required to communicate simultaneously and given the advancements in communication technology, full-duplex communication links are commonplace.
8. The adversarial agents (or the adversary) are assumed passive. In practice,

there can exist agents in the network that are actively adversarial, i.e. they can disrupt the prescribed distributed collaboration protocol including the privacy mechanism. Under this scenario, the proposed privacy protocol might not be effective.

5.1.2 Future Research

In order to overcome the aforementioned limitations, future research can be pursued in the following directions.

1. Explore the conditions under which the proposed two-phase approach can also preserve external privacy, without relying on cryptographic encryptions.
2. Explore the applicability of the proposed privacy protocol if some of the agents in the network are actively malicious.
3. Investigate if the proposed protocol can preserve internal privacy of the internal states generated in the intermediate stage of a distributed collaboration algorithm. This can enhance the applicability of the proposed protocol to more general distributed collaborations.
4. Investigate if the sufficient condition for guaranteeing internal privacy of all the honest agents is also necessary. Otherwise, investigate the least conservative sufficient condition.
5. Investigate conditions under which the proposed protocol can also preserve external privacy, without relying on cryptographic encryptions.

6. Investigate the efficacy of the proposed protocol if some of the agents in the network are actively malicious.
7. In case of distributed average consensus, design a similar two-phase approach for internal privacy in distributed computation of more general functions than just the average. In case of distributed optimization, extend the proposed protocol for internal privacy of the entire agents' costs instead of just their affine parts.

5.2 Contribution: External Privacy in Distributed Collaboration

This dissertation investigates model-based scheme, also referred as masking with system kernel or MSK, for external privacy in distributed collaboration algorithms that can be modeled as linear time-invariant networked control systems. In contrast to the cryptographic encryptions, privacy of the proposed model-based scheme does not rely on secure generation and distribution of keys amongst the agents. Instead, the model-based scheme exploits the inherent redundancies in certain distributed collaboration algorithms. The model-based scheme introduces minimal overhead delays and thus, it does not threaten the stability of the collaboration algorithm, which is especially critical if the agents participating in the collaboration are physical dynamic systems.

The external privacy is guaranteed by the model-based scheme, if certain conditions are met by the distributed collaboration algorithm, in an information-

theoretic manner using a relaxed notion of Shannon’s perfect secrecy. It is interesting to note that the required conditions are closely related to the well-known concept of unobservability in control systems theory. The details of the results are given in Chapter 4.

5.2.1 Limitations

The limitations of the model-based privacy scheme or MSK are as following.

1. The model-based based scheme is only effective for those distributed collaboration algorithms that can be modeled as linear time-invariant networked control systems, whose system parameters satisfy certain conditions (as specified in Chapter 4). This is a strong limitation of the scheme as in general distributed collaboration algorithms need not be linear or even time-invariant. However, the restrictions on the system parameters can be met by appropriately designing the collaboration algorithm as demonstrated by the numerical examples, Example I and Example II, in Chapter 4
2. Often in practical scenarios, the communication links in the network suffer from package drops and delays, leading to asynchronicity between the agents. In this dissertation, delays in communication links have not been considered and therefore agents are assumed to operate in a synchronous manner.

5.2.2 Future Research

In order to overcome the aforementioned limitations, future research can be pursued in the following directions.

1. For now, the formulated privacy definition gives a minimal requirement for guaranteeing privacy of agents' inputs. A more general privacy definition can be formulated, wherein the possible values of agents inputs might belong to multi-dimensional set (instead of a one-dimensional set as in the current form) in view of the eavesdropper.
2. Investigate a methodology for altering a collaboration algorithm (without affecting the collaboration objective) to satisfy the conditions that are required for guaranteeing external privacy by the proposed model-based scheme.
3. Extend (or investigate) the external privacy guarantee for the case when systems parameters of the resultant NCS do not satisfy the conditions prescribed in Chapter 4.
4. Study the effect of delays in the communication links on the proposed model-based scheme.
5. Investigate the efficacy of the model-based scheme for distributed collaboration algorithms that have non-linear and time-varying dynamics.
6. Investigate the possibility of using this model-based privacy scheme for exchanging information across network in a secure manner.

Bibliography

- [1] Shanhe Yi, Zhengrui Qin, and Qun Li. Security and privacy issues of fog computing: A survey. In *International conference on wireless algorithms, systems, and applications*, pages 685–695. Springer, 2015.
- [2] Zhenqi Huang, Sayan Mitra, and Geir Dullerud. Differentially private iterative synchronous consensus. In *Proc. ACM Workshop on Privacy in the Electronic Society*, pages 81–90. ACM, 2012.
- [3] Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2483–2493, 2013.
- [4] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.
- [5] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [6] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [7] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM, 1988.
- [8] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [9] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

- [10] Luca Scardovi, Murat Arcak, and Eduardo D Sontag. Synchronization of interconnected systems with an input-output approach. part i: Main results. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 609–614. IEEE, 2009.
- [11] Wei Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory & Applications*, 1(2):505–512, 2007.
- [12] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.
- [13] Shiping Yang, Sicong Tan, and Jian-Xin Xu. Consensus based approach for economic dispatch problem in a smart grid. *IEEE Transactions on Power Systems*, 28(4):4416–4426, 2013.
- [14] Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11(5):1663–1707, 2010.
- [15] David Chaum, Claude Crépeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 11–19. ACM, 1988.
- [16] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, 2004.
- [17] Juan Garay and Rafail Ostrovsky. Almost-everywhere secure computation. In *Advances in Cryptology—Eurocrypt 2008*, Lecture Notes in Computer Science, pages 307–323. Springer, 2008.
- [18] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.
- [19] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81:221–231, 2017.
- [20] Paolo Braca, Riccardo Lazzeretti, Stefano Marano, and Vincenzo Matta. Learning with privacy in consensus + obfuscation. *IEEE Signal Processing Letters*, 23(9):1174–1178, 2016.
- [21] Nicolaos E Manitara and Christoforos N Hadjicostis. Privacy-preserving asymptotic average consensus. In *European Control Conference*, pages 760–765. IEEE, 2013.

- [22] Sérgio Pequito, Soumya Kar, Shreyas Sundaram, and A Pedro Aguiar. Design of communication networks for distributed computation with privacy guarantees. In *53rd IEEE Conference on Decision and Control*, pages 1370–1376. IEEE, 2014.
- [23] Nirupam Gupta and Nikhil Chopra. Confidentiality in distributed average information consensus. In *55th IEEE Conf. on Decision and Control*, pages 6709–6714. IEEE, 2016.
- [24] Minghao Ruan, Muaz Ahmad, and Yongqiang Wang. Secure and privacy-preserving average consensus. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, pages 123–129. ACM, 2017.
- [25] Yilin Mo and Richard M Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.
- [26] Riccardo Lazzeretti, Steven Horn, Paolo Braca, and Peter Willett. Secure multi-party consensus gossip algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 7406–7410. IEEE, 2014.
- [27] Emmanuel A Abbe, Amir E Khandani, and Andrew W Lo. Privacy-preserving methods for sharing financial risk exposures. *American Economic Review*, 102(3):65–70, 2012.
- [28] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
- [29] A Nedić, Dimitri P Bertsekas, and Vivek S Borkar. Distributed asynchronous incremental subgradient methods. *Studies in Computational Mathematics*, 8(C):381–407, 2001.
- [30] Håkan Terelius, Ufuk Topcu, and Richard M Murray. Decentralized multi-agent optimization via dual decomposition. *IFAC Proceedings Volumes*, 44(1):11245–11251, 2011.
- [31] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [32] S Sundhar Ram, A Nedić, and Venugopal V Veeravalli. A new class of distributed optimization algorithms: Application to regression of distributed data. *Optimization Methods and Software*, 27(1):71–88, 2012.
- [33] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27. ACM, 2004.

- [34] Robin L Raffard, Claire J Tomlin, and Stephen P Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2453–2459. IEEE, 2004.
- [35] Octavian Catrina and Sebastiaan De Hoogh. Secure multiparty linear programming using fixed-point arithmetic. In *European Symposium on Research in Computer Security*, pages 134–150. Springer, 2010.
- [36] Jiangtao Li and Mikhail J Atallah. Secure and private collaborative linear programming. In *Collaborative Computing: Networking, Applications and Work-sharing, 2006. CollaborateCom 2006. International Conference on*, pages 1–8. IEEE, 2006.
- [37] Tomas Toft. Solving linear programs using multiparty computation. In *International Conference on Financial Cryptography and Data Security*, pages 90–107. Springer, 2009.
- [38] Marius C Silaghi and Debasis Mitra. Distributed constraint satisfaction and optimization with privacy enforcement. In *Intelligent Agent Technology, 2004.(IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on*, pages 531–535. IEEE, 2004.
- [39] Yuan Hong, Jaideep Vaidya, Nicholas Rizzo, and Qi Liu. Privacy preserving linear programming. *arXiv preprint arXiv:1610.02339*, 2016.
- [40] Olvi L Mangasarian. Privacy-preserving linear programming. *Optimization Letters*, 5(1):165–172, 2011.
- [41] Pradeep Chathuranga Weeraddana, Georgios Athanasiou, Carlo Fischione, and John S Baras. Per-se privacy preserving solution methods based on optimization. In *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, pages 206–211, 2013.
- [42] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private distributed convex optimization via functional perturbation. *IEEE Transactions on Control of Network Systems*, 5(1):395–408, 2018.
- [43] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, page 4. ACM, 2015.
- [44] Youcheng Lou, Lean Yu, Shouyang Wang, and Peng Yi. Privacy preservation in distributed subgradient optimization algorithms. *IEEE transactions on cybernetics*, 2017.
- [45] Shripad Gade and Nitin H Vaidya. Distributed optimization of convex sum of non-convex functions. *arXiv preprint arXiv:1608.05401*, 2016.

- [46] Shripad Gade and Nitin H Vaidya. Private learning on networks. *arXiv preprint arXiv:1612.05236*, 2016.
- [47] Nirupam Gupta, Jonathan Katz, and Nikhil Chopra. Privacy in distributed average consensus. *IFAC-PapersOnLine*, 50(1):9515–9520, 2017.
- [48] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.
- [49] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.
- [50] J Shao, G Xie, and L Wang. Leader-following formation control of multiple mobile vehicles. *IET Control Theory & Applications*, 1(2):545–552, 2007.
- [51] Nikhil Chopra and Mark W Spong. On synchronization of networked passive systems with time delays and application to bilateral teleoperation. In *proceedings of the SICE annual conference*, pages 3424–3429, 2005.
- [52] Yu-Ping Tsou, Jun-Wei Hsieh, Cheng-Ting Lin, and Chun-Yu Chen. Building a remote supervisory control network system for smart home applications. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 3, pages 1826–1830. IEEE, 2006.
- [53] Farid Katiraei, Reza Iravani, Nikos Hatziargyriou, and Aris Dimeas. Microgrids management. *IEEE power and energy magazine*, 6(3), 2008.
- [54] Wei Ren, Kevin L Moore, and YangQuan Chen. High-order and model reference consensus algorithms in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):678–688, 2007.
- [55] Nirupam Gupta, Yimeng Dong, and Nikhil Chopra. Robustness of distributive double-integrator consensus to loss of graph connectivity. In *American Control Conference (ACC), 2017*, pages 4516–4521. IEEE, 2017.
- [56] Yi-Sheng Shiu, Shih Yu Chang, Hsiao-Chun Wu, Scott C-H Huang, and Hsiao-Hwa Chen. Physical layer security in wireless networks: A tutorial. *IEEE wireless Communications*, 18(2), 2011.
- [57] Kerry A McKay, Kerry A McKay, Larry Bassham, Meltem Sonmez Turan, and Nicky Mouha. *Report on lightweight cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2017.
- [58] Heng Zhang, Yuanchao Shu, Peng Cheng, and Jiming Chen. Privacy and performance trade-off in cyber-physical systems. *IEEE Network*, 30(2):62–66, 2016.
- [59] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private average consensus with optimal noise selection. *IFAC-PapersOnLine*, 48(22):203–208, 2015.

- [60] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer, 2001.
- [61] Ivan Gutman and W Xiao. Generalized inverse of the laplacian matrix and some applications. *Bulletin de l'Academie Serbe des Sciences at des Arts (Cl. Math. Natur.)*, 129:15–23, 2004.
- [62] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [63] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.
- [64] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Proceedings of the Princeton conference in honor of Professor S. Bochner*, 1969.
- [65] Fan RK Chung. Laplacians of graphs and cheeger's inequalities. *Combinatorics, Paul Erdos is Eighty*, 2(157-172):13–2, 1996.
- [66] Anne Marsden. Eigenvalues of the laplacian and their relationship to the connectedness of a graph. *University of Chicago, REU*, 2013.
- [67] Joo P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.
- [68] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.
- [69] L Servi and Y Ho. Recursive estimation in the presence of uniformly distributed measurement noise. *IEEE Transactions on Automatic Control*, 26(2):563–565, 1981.
- [70] Bernard Lindgren. *Statistical theory*. Routledge, 2017.
- [71] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.