Annual Reviews in Control 47 (2019) 423–440

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

A control-theoretic perspective on cyber-physical privacy: Where data privacy meets dynamic systems☆

Yang Lu[a,*], Minghui Zhu[a]

School of Electrical Engineering and Computer Science, Pennsylvania State University, 201 Old Main, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 21 January 2019
Revised 19 April 2019
Accepted 20 April 2019
Available online 24 April 2019

Keywords:
Cyber-physical systems
Privacy

ABSTRACT

A cyber-physical system (CPS) consists of a large number of geographically dispersed entities and distributed data sharing is necessary to achieve network-wide goals. However, distributed data sharing also raises the significant concern that private or confidential information of legitimate entities could be leaked to unauthorized entities. Privacy has become an issue of high priority to address before a certain CPS can be widely deployed. Existing privacy-preserving techniques solely focus on the cyber space but ignore the physical world, and hence they alone may not be adequate to ensure CPS privacy. This paper aims to summarize recent studies on how to develop control-theoretic approaches to complement existing privacy-preserving techniques and ensure CPS privacy.

Published by Elsevier Ltd.

Contents

☆ This work was partially supported by the grants ARO W911NF-13-1-0421 (MURI), NSF CNS-1505664, and NSF CAREER ECCS-1846706.

* Corresponding author.
E-mail address: yml5046@psu.edu (Y. Lu).

https://doi.org/10.1016/j.arcontrol.2019.04.010
1367-5788/Published by Elsevier Ltd.

1. Introduction

In the last decades, we have witnessed rapid advances in power, mobility and efficiency of computational devices. The advances have led to pervasive usage of information and communications technologies (ICT). Recently, ICT are increasingly integrated with control systems in the physical world. A new generation of engineering applications is emerging, including the smart grid, smart buildings, intelligent transportation systems, autonomous vehicles and medical device networks. The new-generation systems are referred to as cyber-physical systems (CPSs). In CPSs, synergy of ICT and physical processes introduces new functions into control systems, improves their operating performance and enables them to achieve unprecedented intelligence. The U.S. National Science Foundation (NSF) envisions that CPS technologies will transform the way people interact with engineered systems (Program (2019), CPS). A recent report estimates that CPS innovations could find direct applications in sectors currently accounting for more than $32.3 trillion in economic activities, and with the potential to grow to $82 trillion of output by 2025 – about one half of the global economy (Evans & Annunziata, 2012). CPS has become a priority area for research investment in both U.S. and Europe.

A CPS consists of a large number of geographically dispersed entities and distributed data sharing is necessary to achieve network-wide goals. However, distributed data sharing also raises the significant concern that private or confidential information of legitimate entities could be leaked to unauthorized entities. Here we present a few examples to show potential privacy issues of CPSs. In occupancy-based heating, ventilation, and air conditioning (HVAC) control systems, location traces of individual occupants can be inferred from occupancy data with auxiliary information such as office directories and user mobility patterns (Jia, Dong, Sastry, & Spanos, 2017). Such location privacy issue also arises in urban sensing networks, where using a cellphone for collecting information from the environment and tagging them with time and global positioning system (GPS) data will inevitably disclose sensitive personal information, including the user's location and identity (Krontiris, Freiling, & Dimitriou, 2010). The contextual information attached to location traces reveals much about individuals' habits, interests, activities and relationships (Lisovich, Mulligan, & Wicker, 2010). It can also lead to leakage of their personal or corporate secrets, expose them to unwanted advertisement and location-based spams, cause social reputation or economic damage, make them victims of blackmail or even physical violence (Shokri, Theodorakopoulos, Boudec, & Hubaux, 2011). In the smart grid, smart meters could lead to unintended consequences for customers' privacy (McDaniel & McLaughlin, 2009). Appliance usage patterns can be inferred from energy-use information stored at smart meters with auxiliary information of load signature libraries, and such usage patterns could expose customers' habits and behaviors.

Privacy has become an issue of high priority to address before certain CPSs can be widely deployed. For example, current absence of accepted solutions to tackle privacy concerns caused a deadlock in the mandatory deployment of smart meters in the Netherlands because of the common belief that smart metering is necessarily privacy-invasive (Cavoukian, 2012). In 2010, California's new law on smart meter privacy, for the first time, indicated strong demands to protect privacy of end-users' energy consumption data (California Public Utilities Commission, 2010). In addition, the Federal Trade Commission (FTC) provided recommendations on data collection and treatment to protect drivers' privacy in urban transportation (Cottrill & Thakuriah, 2011). In January 2015, FTC released a long-awaited report calling on companies that develop Internet-connected devices such as fitness monitors and connected cars to take proactive steps to protect consumers' privacy.

In the computer science community, a large number of technologies have been proposed to ensure data privacy of ICT systems. Typical examples include trusted computation, cryptographic computation and differential privacy. Comprehensive reviews can be found at recent surveys (Panackal & Pillai, 2013; Vaghashia & Ganatra, 2015). Trusted computation (Chase, 2007; Dent & Price, 2005; Kuntze, Mahler, & Schmidt, 2006) requires a trustworthy third party to perform all computations. Cryptographic computation relies on homomorphic encryption (Dijk, Gentry, Halevi, & Vaikuntanathan, 2010; Gentry, Halevi, & Smart, 2012; Goldwasser & Micali, 1982; Yi, Paulet, & Bertino, 2014) or secret sharing (Cramer, Damgård, & Nielsen, 2015; Lindell & Pinkas, 2009; Shamir, 1979). Homomorphic encryption allows certain algebraic operations to be carried out on ciphertexts, thus generating an encrypted result which, when decrypted, matches the result of operations performed on plaintexts. Secret sharing is a technique for distributing a secret among a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number of shares are combined together, while a smaller number of shares contains no information of the secret. Differentially private schemes (Dwork, 2006; Dwork & Lei, 2009; Dwork, Naor, Pitassi, & Rothblum, 2010; Dwork & Roth, 2014b; McSherry & Talwar, 2007) persistently add random noises into each individual's data in such a way that, with high probability, individual participation cannot be inferred by an

adversary, who can access arbitrary auxiliary information, via released data.

Existing techniques for data privacy of ICT systems are necessary but may not be sufficient to ensure CPS privacy. First, most existing techniques only apply to static data sources (Dwork, McSherry, Nissim, & Smith, 2006; Li, Li, & Venkatasubramanian, 2007; Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007; Samarati & Sweeney, 1998), with a few exceptions (Sankar, Rajagopalan, Mohajer, & Poor, 2013; Sankar, Rajagopalan, & Poor, 2013) that work for independent and identically distributed (IID) data sources. In contrast, the released data of a CPS is usually time series generated by dynamic systems and may not follow any probabilistic distribution. The structures of dynamic systems become a new attack vector. Second, current techniques do not take into account their impacts on the physical world. For many CPSs, e.g., power systems, control-theoretic specifications, e.g., stability and safety, may be more important than privacy. Third, current techniques need to be customized in order to address salient features of the physical world. For example, most existing cryptographic computation schemes only work for binary or non-negative integer data, while physical parameters and states of dynamic systems are usually signed real numbers. In addition, CPSs operate in real time and thus computational efficiency of privacy preserving schemes becomes a dominating factor. Further, entities of CPSs are distributed over an extended spatial area and it may lack of a centralized authority to carry over computations. The above emerging issues are mainly caused by the presence of dynamic systems. To address the issues, one should carefully examine unique characteristics of dynamic systems and study the interplay between data privacy and dynamic systems. In the control community, researchers have developed various approaches to address one or more of the aforementioned issues. In this paper, we will first comprehensively review these works, and then provide two representative example problems to illustrate in detail how some of the issues can be addressed.

The remaining of this paper is organized as follows. As pointed out by Lindell and Pinkas (2009), data privacy has two important complementary issues: one is secure multiparty computation (SMC) (Cramer et al., 2015; Hazay & Lindell, 2010), whose goal is to enable collective computations with perfect correctness and meanwhile protect data privacy of the participants; and the other is private data release (Dwork et al., 2006; Li et al., 2007; Machanavajjhala et al., 2007; Samarati & Sweeney, 1998), whose goal is to perturb released data in such a way that one cannot infer private inputs from the perturbed outputs, and simultaneously, the perturbed outputs can still enable certain utilities. Sections 2 and 3 review widely adopted privacy notions for SMC and private data release, respectively. Section 4 then provides a cross-comparison between the privacy notions of the two sets. Section 5 provides a comprehensive review for recent works on CPS privacy in the control community. Different metrics are compared for the reviewed works, including privacy notions, privacy-enhancing techniques, attacker models, computation requirements, and impacts on control system performance. Furthermore, we also elaborate on how these works address one or more of the aforementioned new issues in CPS privacy. After that, Sections 6 and 7 present two recent representative works to exemplify how to tackle some of the new CPS issues. One is on privacy-preserving distributed optimization which belongs to SMC, and the other is on privacy-preserving dynamic data release which belongs to private data release. Finally, Section 8 concludes the paper.

1.1. Notions and notations

Let $\mathbb{N}$ denote the set of natural numbers and $\mathbb{Z}^n_{>0}$ denote the set of positive integer column vectors of size n. Given $w \in \mathbb{Z}_{>0}$, $\mathbb{Z}_w \triangleq \{0, 1, \cdots, w-1\}$ and $\mathbb{Z}^*_w$ denotes the set of positive integers which are smaller than w and do not have common factors other than 1 with w. Given a finite set C, denote by $|C|$ its cardinality. Given any $p, q \in \mathbb{Z}_{>0}$, $\gcd(p, q)$ and $\mathrm{lcm}(p, q)$ denote the greatest common divisor and the least common multiple of p and q, respectively. Given a polynomial function f, $\deg(f)$ denotes the degree of f. Given a non-empty, closed and convex set $Z \subseteq \mathbb{R}^m$, $\mathbb{P}_Z$ denotes the projection operation onto Z, i.e., for any $z \in \mathbb{R}^m$, $\mathbb{P}_Z[z] = \mathrm{argmin}_{y \in Z} \|y - z\|$. $\mathbf{0}_n$ denotes the column vector with n zeros. $\mathbf{0}_{m\times n}$ denotes the $m \times n$ matrix where all entries are zeros. I_n denotes the identity matrix of size n. $\mathbb{S}^n$ denotes the set of real symmetric matrices of size n. Given a finite index set Ω, let $[A_i]_{i\in\Omega}$ denote the column-wise stack of A_i for all $i \in \Omega$, where A_i's are matrices with the same number of columns. When there is no confusion in the context, we drop the subscript $i \in \Omega$ and use $[A_i]$. Given matrices $A_1, \cdots, A_N$ with the same column number, let A_{-i} denote $[A_j]_{j\neq i}$. Given column vectors $x_1, \cdots, x_N$, let x_{-i} denote $[x_j]_{j\neq i}$. The induced ℓ_1 and ℓ_2 norms of matrix A are denoted by $\|A\|_1$ and $\|A\|_2$, respectively. $\|A\|_0$ denotes the number of nonzero entries of matrix A and is referred to as its ℓ_0 norm. $\|A\|_*$ denotes the sum of the singular values of matrix A and is referred to as its nuclear norm. For a column vector $w \in \mathbb{R}^n$, define a quantity $\|\cdot\|_{\min}$ as $\|w\|_{\min} = \min_{\ell\in\{1,\cdots,n\}} |w_\ell|$. Given a matrix A, denote by $\mathrm{vec}(A)$ the column vector consisting of the entries of A. $\mathrm{rank}(A)$ and $\det(A)$ denote the rank and the determinant of matrix A, respectively. Denote by $\circ$ the operator of the Hadamard product of two matrices of the same size, i.e., given matrices $A = [A_{ij}] \in \mathbb{R}^{m\times n}$ and $B = [B_{ij}] \in \mathbb{R}^{m\times n}$, $T = [T_{ij}] = A \circ B$ means $T \in \mathbb{R}^{m\times n}$ and $T_{ij} = A_{ij}B_{ij}$. $1_{[\text{condition}]}$ denotes the indicator such that $1_{[\text{condition}]} = 1$ if condition is true and $1_{[\text{condition}]} = 0$ if condition is false. For any $k \in \mathbb{N}$, denote by $x_{[0,k]}$ the sequence of $\{x_0, \cdots, x_k\}$.

2. Review of secure multiparty computation

In this section, we first motivate the SMC problem by two real-world examples and present the general problem setting. After that, we review three most widely used privacy notions for SMC, namely, perfect secrecy, computational security, and plaintexts not efficiently solvable. Comprehensive discussions on SMC can be found at Yi et al. (2014), Cramer et al. (2015) and Goldreich (2004).

2.1. Motivating examples

SMC is ubiquitous in our daily life. For example, when we use credit cards or login in to our email accounts, there are some SMC protocols running to protect our passwords. We next use auction and benchmarking to motivate the problem of SMC. These examples are adopted from Chapter 1 of Cramer et al. (2015).

• *Auction.* In an auction, a set of bidders $\mathcal{V} = \{1, \cdots, N\}$ bid for an item and the one with the highest bid wins. Each bidder has a predetermined maximum amount x_i it is willing to pay for the item. Mathematically, the identity I of the winner is determined by $I = \mathrm{argmax}_{i\in\mathcal{V}} x_i$. Hence, the auction process can be mathematically described as the computation of $f(x) = \max_{i\in\mathcal{V}} x_i$. To guarantee fairness of the auction, for each $i \in \mathcal{V}$, x_i should be kept private for bidder i. Hence, there is a need for an algorithm that can correctly compute $f(x) = \max_{i\in\mathcal{V}} x_i$ and simultaneously does not disclose x_i for any $i \in \mathcal{V}$.

• *Benchmarking.* In a market, a set of companies $\mathcal{V} = \{1, \cdots, N\}$ in the same area have interests to evaluate their own performance through a comparison with others. This is known as a benchmark analysis. Such an analysis takes each company i's input x_i and computes a quantity $f_i(x)$ for each $i \in \mathcal{V}$ which reflects company i's performance. For example, x_i could be company i's annual net profit and $f_i(x) = \frac{x_i}{\sum_{j\in\mathcal{V}} x_j}$ represents the portion of x_i in the total annual

net profit $\sum_{j\in\mathcal{V}} x_j$ of the N companies. It is clear that each company i will insist that its input x_i must not be disclosed to its competitors during the benchmark analysis.

2.2. Problem setting

In this subsection, we present the general formulation of SMC. Consider a set of agents $\mathcal{V}=\{1,\cdots,N\}$. Each agent i holds a secret input x_i and aims to compute the value of $f_i(x_1,\cdots,x_N)$. Notice that agent i's function f_i in general depends on the other agents' inputs x_{-i} and thus needs to be collectively computed. However, some of the agents could be corrupted. Denote by $\mathcal{C}\subseteq\mathcal{V}$ the set of corrupted agents and $\mathcal{B}=\mathcal{V}\backslash\mathcal{C}$ the set of benign agents. We consider semi-honest attacker model (Cramer et al., 2015; Hazay & Lindell, 2010). That is, any corrupted agent $i\in\mathcal{C}$ correctly follows the prescribed algorithm but attempts to use its received messages throughout the execution of the algorithm to infer the private inputs of the benign agents. The goal of SMC is that the following two properties are simultaneously satisfied:

- Correctness: each agent $i\in\mathcal{V}$ obtains the correct value of $f_i(x)$;
- Privacy: the corrupted agents in set $\mathcal{C}$ cannot infer private inputs of the benign agents.

There have been various notions in defining what it means by "cannot infer private inputs". In the following three subsections, we present three widely used ones.

2.3. Perfect secrecy

The notion of perfect secrecy was proposed by Claude Shannon in his seminal work (Shannon, 1949). Informally speaking, a computing algorithm is perfectly secret if after the execution of the algorithm, the corrupted agents only know their own inputs and outputs, but do not know anything beyond them, even if they have unlimited computing power. To formally define perfect secrecy, we need to introduce several notions. First we provide the definition of perfect indistinguishability.

Definition 2.1 (Cramer et al. (2015))**.** Let $\mathcal{X}=\{\mathcal{X}(\kappa)\}_{\kappa\in\mathbb{N}}$ and $\mathcal{Y}=\{\mathcal{Y}(\kappa)\}_{\kappa\in\mathbb{N}}$ be two distribution ensembles, where, for each $\kappa\in\mathbb{N}$, $\mathcal{X}(\kappa)$ and $\mathcal{Y}(\kappa)$ are two random variables with the same probability space and the same range $R(\kappa)$. We say that $\mathcal{X}$ and $\mathcal{Y}$ are perfectly indistinguishable, denoted by $\mathcal{X}\overset{p}{\equiv}\mathcal{Y}$, if the following holds:

$$\delta(\mathcal{X}(\kappa),\mathcal{Y}(\kappa))\triangleq\frac{1}{2}\sum_{r\in R(\kappa)}|\Pr[\mathcal{X}(\kappa)=r]-\Pr[\mathcal{Y}(\kappa)=r]|=0,\quad\forall\kappa\in\mathbb{N}.$$

□

In Definition 2.1, the term $\delta(\mathcal{X}(\kappa),\mathcal{Y}(\kappa))$ is called the statistical distance or total variation distance between $\mathcal{X}(\kappa)$ and $\mathcal{Y}(\kappa)$. The intuition behind Definition 2.1 is that two distribution ensembles that are perfectly indistinguishable cannot be distinguished because they follow the same distribution. Notice that Definition 2.1 does not have any restriction on the computing capability. Hence, if two random ensembles $\mathcal{X}$ and $\mathcal{Y}$ are perfectly indistinguishable, then one cannot distinguish them even if it has *unlimited* computing power. Next we introduce the notion of *view*. Roughly speaking, the view of an agent is the set of all the messages the agent can see after the execution of the algorithm.

Definition 2.2 (Cramer et al. (2015))**.** Let Π be a protocol for computing $f=\{f_i\}_{i\in\mathcal{V}}$. For an execution of Π on a joint input $\bar{x}$, the view of agent $i\in\mathcal{V}$, denoted by $\text{VIEW}_i^{\Pi}(\bar{x})$, is $\text{VIEW}_i^{\Pi}(\bar{x})\triangleq\{x_i,m_1^i,\cdots,m_{t_i}^i\}$, where t_i represents the total number of messages received by agent i, and for each $\ell\in\{1,\cdots,t_i\}$, m_ℓ^i represents the ℓth message agent i receives. □

Having introduced perfect indistinguishability and view, we are now ready to define perfect secrecy.

Definition 2.3 (Cramer et al. (2015))**.** Let Π be a protocol for computing $f=\{f_i\}_{i\in\mathcal{V}}$. Given a joint input $\bar{x}$, denote the joint view of the agents in a set $I\subseteq\mathcal{V}$ by $\text{VIEW}_I^{\Pi}(\bar{x})$. We say that Π t-privately computes f in the sense of perfect secrecy if there exists a probabilistic polynomial-time algorithm, denoted by S, such that for every $I\subseteq\mathcal{V}$ of cardinality at most t and for any admissible $\bar{x}$, it holds that

$$S(I,\{\bar{x}_i\}_{i\in I},\{f_i\}_{i\in I})\overset{p}{\equiv}\text{VIEW}_I^{\Pi}(\bar{x}).\tag{1}$$

□

In (1), $S(I,\{\bar{x}_i\}_{i\in I},\{f_i\}_{i\in I})$ and $\text{VIEW}_I^{\Pi}(\bar{x})$ denote the set of messages that the agents in set I can see after the execution of S and Π, respectively. Notice that the inputs to S are only I's own inputs $\{\bar{x}_i\}_{i\in I}$ and outputs $\{f_i\}_{i\in I}$, which must be known to I. Hence, the intuition behind (1) is that whatever can been seen by I during the execution of Π can be simulated by an algorithm S using only I's own inputs and outputs, and I cannot distinguish $S(I,\{\bar{x}_i\}_{i\in I},\{f_i\}_{i\in I})$ and $\text{VIEW}_I^{\Pi}(\bar{x})$ even if it has unlimited computing power. In other words, the execution of Π does not provide I any additional information other than what it must known (i.e., I's own inputs and outputs). Definition 2.3 requires that (1) holds for any $I\subseteq\mathcal{V}$ of cardinality at most t. Hence, if Π t-privately computes f in the sense of perfect secrecy, then as long as the number of the corrupted agents is no more than t, when pooling their information after the execution Π, the corrupted agents gain nothing about the private inputs of the benign agents.

One secure computing scheme that is perfectly secret is the Shamir's secret sharing scheme (S4) (Shamir, 1979). We next use an example to briefly introduce the S4. The following example is modified from the example in Section 3.3.4 of Cramer et al. (2015).

Example 2.1. Consider the case of three agents, P1, P2 and P3, and at most one of them is corrupted. Each agent i holds a private input x_i, and the three agents aim to collectively compute the function $f=x_1+x_2+x_3$. Assume that each x_i is an nonnegative integer and belongs to a finite field $\mathbb{Z}_p$, where p is a prime number and known to all the three agents. Also assume that $x_1+x_2+x_3$ belongs to $\mathbb{Z}_p$. Let $p=19$, $x_1=2$, $x_2=5$ and $x_3=9$.

Step 1: Each agent i picks $\alpha_i\in\mathbb{Z}_p$ uniformly at random and constructs a polynomial $g_i(X)=x_i+\alpha_i X$. Here, α_i is the key of agent i, and its value must be kept private to agent i. Let $\alpha_1=2$, $\alpha_2=6$ and $\alpha_3=1$. Then $g_1(X)=2+2X$, $g_2(X)=5+6X$, and $g_3(X)=9+X$.

Step 2: Each agent i computes $s_{ij}=g_i(j)\text{mod}p$ for all $j\in\{1,2,3\}$ and securely sends s_{ij} to agent j. We have $s_{11}=4$, $s_{12}=6$, $s_{13}=8$, $s_{21}=11$, $s_{22}=17$, $s_{23}=4$, $s_{31}=10$, $s_{32}=11$, and $s_{33}=12$.

Step 3: Each agent i computes $s_i=\sum_{j=1}^3 s_{ji}\text{mod}p$ and securely sends s_i to the other two agents. We have $s_1=6$, $s_2=15$, and $s_3=5$.

Step 4: Each agent i constructs polynomial $\delta_j(X)=\prod_{\ell\neq j}\frac{X-\ell}{j-\ell}\text{mod}p$, for all $j\in\{1,2,3\}$. We have $\delta_1(X)=10X^2+7X+3$, $\delta_2(X)=18X^2+4X+16$, and $\delta_3(X)=10X^2+8X+1$.

Step 5: Each agent i computes $s=\sum_{j=1}^3 s_j\delta_j(0)\text{mod}p$. We have $s=16=x_1+x_2+x_3$.

We next briefly explain why the above approach can guarantee correctness and privacy. At Step 2, each agent i distributes the shares s_{ij}'s of its private input x_i to the other two agents via the polynomial g_i constructed at Step 1. It is clear that if one can recover the polynomial g_i, then it can obtain x_i as $x_i=g_i(0)$. Notice that each g_i is a polynomial of degree one, i.e., a linear function. The crucial property of this secret sharing scheme that guarantees correctness and privacy is that at least two shares of x_i are needed to recover x_i, while any one share contains no information

of x_i. This property directly follows the fact that it takes at least $\tau+1$ points to define a polynomial of degree τ, and also any $\tau+1$ points are enough to do so.

In our example, we assume that there is only one corrupted agent, saying P1. Since g_2 and g_3 are polynomials of degree one and P1 only knows one share of x_2 and x_3, i.e., s_{21} and s_{31}, respectively, it has no information of x_2 and x_3. In other words, any pair of $x_2, x_3 \in \mathbb{Z}_p$ such that $x_2+x_3 = f-x_1 = 14$ is equally possible to P1, and hence the scheme is perfectly secret. In general, to resist τ corrupted agents, each agent needs to share its private input by a polynomial of degree at least τ.

The correctness of the scheme is a result of the homomorphic property of Step 3 and the Lagrange interpolation of Steps 4 and 5. In particular, the homomorphic property of Step 3 is that the sum of the shares received by one agent is a share of the sum of the private inputs, i.e., for each $i \in \{1, 2, 3\}$, $s_i = \sum_{j=1}^{3} s_{ji} \bmod p$ is a share of $s = \sum_{j=1}^{3} x_j$ generated by the polynomial $g = \sum_{j=1}^{3} g_j$. Steps 4 and 5 use Lagrange interpolation to reconstruct the polynomial $g = \sum_{j=1}^{3} g_j$ and $s = g(0)$. By Section 3.2 of Cramer et al. (2015), given a polynomial g of degree τ and an arbitrary set $C \subseteq \mathbb{R}$ with $|C| \geq \tau+1$, the polynomial g can be uniquely reconstructed as $g(X) = \sum_{j \in C} g(j)\delta_j(X)$, where δ_j is defined in Step 4.

2.4. Computational security

Perfect secrecy has two requirements: (i) the left-hand-side and the right-hand-side of (1) have exactly the same distribution and (ii) it must hold even if the corrupted agents have unlimited computing power. To satisfy these two requirements, the key must be at least as long as the private inputs and the communication links must be secure (Shannon, 1949). In the S4 of the above subsection, to share two private inputs x_{i1} and x_{i2}, agent i needs to adopt two different keys α_{i1} and α_{i2}. Otherwise, if the same α is used to share x_{i1} and x_{i2}, then any other agent can infer $x_{i1} - x_{i2} \bmod p$ using its shares of x_{i1} and x_{i2}. We then seek for an alternative privacy notion which does not have the limitations on key length and communication links. On the other hand, such an alternative privacy notion could be weaker than perfect secrecy in terms of privacy level. Computational security is an example of such privacy notion. Roughly speaking, computational security requires that any non-negligible amount of information cannot be feasibly extracted. To provide the formal definition of computational security, we first need to introduce computational indistinguishability.

Definition 2.4 (Cramer et al. (2015))**.** Let $\mathcal{X} = \{\mathcal{X}(\kappa)\}_{\kappa \in \mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}(\kappa)\}_{\kappa \in \mathbb{N}}$ be two distribution ensembles, where, for each $\kappa \in \mathbb{N}$, $\mathcal{X}(\kappa)$ and $\mathcal{Y}(\kappa)$ are two random variables with the same probability space and the same range $R(\kappa)$. We say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, denoted by $\mathcal{X} \overset{c}{\equiv} \mathcal{Y}$, if for every non-uniform probabilistic polynomial-time distinguisher D, every positive polynomial $p : \mathbb{N} \to \mathbb{R}_{>0}$, and every sufficiently large $\kappa \in \mathbb{N}$, the following holds:

$$|\Pr[D(\mathcal{X}(\kappa)) = 1] - \Pr[D(\mathcal{Y}(\kappa)) = 1]| < \frac{1}{p(\kappa)}.$$

□

Definition 2.4 relaxes Definition 2.1 in two aspects. First, Definition 2.4 only considers polynomial-time distinguishers, while Definition 2.1 does not have restrictions on the computing power of the distinguishers. Second, Definition 2.4 requires that the difference between the two concerned distributions is negligible, while Definition 2.1 requires that the two distributions must be exactly the same. Hence, Definition 2.4 states that two distributions are computationally indistinguishable if the difference identified by any efficient distinguisher is negligible. The definition of computational security is provided next.

Definition 2.5 (Cramer et al. (2015))**.** Let Π be a protocol for computing $f = \{f_i\}_{i \in \mathcal{V}}$. Given a joint input $\bar{x}$, denote the joint view of the agents in a set $I \subseteq \mathcal{V}$ by $\text{VIEW}_I^{\Pi}(\bar{x})$. We say that Π privately computes f in the sense of computational security if there exists a probabilistic polynomial-time algorithm, denoted by S, such that for every $I \subseteq \mathcal{V}$ and for any admissible $\bar{x}$, it holds that

$$S(I, \{\bar{x}_i\}_{i \in I}, \{f_i\}_{i \in I}) \overset{c}{\equiv} \text{VIEW}_I^{\Pi}(\bar{x}). \tag{2}$$

□

Definition 2.5 states that whatever can been seen by I during the execution of Π can be simulated by an algorithm S using only I's own inputs and outputs, and I cannot distinguish any non-negligible difference between $S(I, \{\bar{x}_i\}_{i \in I}, \{f_i\}_{i \in I})$ and $\text{VIEW}_I^{\Pi}(\bar{x})$ by any efficient distinguisher. In other words, the execution of Π does not provide I any additional non-negligible information that can be efficiently extracted.

Well-known computationally secure encryption schemes include public key encryption schemes Goldwasser–Micali, ElGamal, Paillier and Boneh–Goh–Nissim. Here we use the Paillier encryption scheme as an example. The example is modified from the example in Section 2.4 of Yi et al. (2014).

Example 2.2. Consider the case of three agents, P1, P2 and P3, where P1 is an entity that is equipped with superior computing resources and could be corrupted. Each agent i holds a private input x_i. Let $x_1 = 2$, $x_2 = 5$ and $x_3 = 9$. P2 requests P1 to compute $f = x_1 + x_2 + x_3$ for it. Here, we consider sum computation because the Paillier encryption scheme is additively homomorphic. Since P1 could be corrupted, P2 and P3 cannot directly send x_2 and x_3 to P1. Instead, P2 could adopt the Paillier encryption scheme to achieve secure computation of f. The steps are as follows.

Key generation:P2 chooses two large prime numbers p and q randomly and independently, such that $\gcd(pq, (p-1)(q-1)) = 1$; computes $\alpha = pq$ and $\nu = \text{lcm}(p-1, q-1)$; selects random integer $\beta \in \mathbb{Z}_{\alpha^2}^*$ such that the modular multiplicative inverse $\pi = (\frac{(\beta^{\nu} \bmod \alpha^2)-1}{\alpha})^{-1} \bmod \alpha$ exists, i.e., $\pi \frac{(\beta^{\nu} \bmod \alpha^2)-1}{\alpha} \equiv 1 \bmod \alpha$. The public keys are (α, β) and the private keys are (ν, π). P2 publicizes (α, β) to P1 and P3 while keeps (ν, π, p, q) private to itself.

Let $p = 17$ and $q = 19$. It can be checked that $\gcd(pq, (p-1)(q-1)) = 1$. We have $\alpha = pq = 323$ and $\nu = \text{lcm}(p-1, q-1) = 144$. Let $\beta = 324$. We then have $\pi = 83$.

Encryption: Each agent i selects a random number $r_i \in \mathbb{Z}_{\alpha}^*$ and computes the ciphertext by the encryption operation $E(\cdot)$ as $y_i = E(x_i, \alpha, \beta, r_i) = \beta^{x_i} \cdot r_i^{\alpha} \bmod \alpha^2$. P2 and P3 send y_2 and y_3 to P1.

Let $r_1 = 5$, $r_2 = 7$ and $r_3 = 8$. We then have $y_1 = 324^2 \cdot 5^{323} \bmod 323^2 = 100153$, $y_2 = 324^5 \cdot 7^{323} \bmod 323^2 = 87145$, and $y_3 = 324^9 \cdot 8^{323} \bmod 323^2 = 51920$.

Evaluation: P1 computes $\bar{f} = y_1 y_2 y_3 \bmod \alpha^2$ and sends $\bar{f}$ to P2.

We have $\bar{f} = 100153 \cdot 87145 \cdot 51920 \bmod 323^2 = 11358$.

Decryption: P2 decrypts $\bar{f}$ by the decryption operation $D(\cdot)$ as $\hat{f} = D(\bar{f}, \alpha, \nu, \pi) = \frac{(\bar{f}^{\nu} \bmod \alpha^2)-1}{\alpha} \cdot \pi \bmod \alpha$.

We have $\hat{f} = \frac{(11358^{144} \bmod 323^2)-1}{323} \cdot 83 \bmod 323 = 16 = x_1 + x_2 + x_3$.

The correctness of the Paillier encryption scheme follows its additively homomorphic property: the product of encryptions is an encryption of the sum of the private inputs. The computational security of the Paillier encryption scheme is established under the decisional composite residuosity assumption (DCRA): Given a composite C and an integer z, it is computationally intractable to decide whether z is a C-residue modulo C^2 or not, i.e., whether there exists y such that $z = y^C \bmod C^2$. Notice that the Paillier encryption scheme is probabilistic as the encryption step adopts randomness r_i. As a result, the encryption of a plaintext under the same encryptions keys is not unique. Due to this probabilistic nature, under the DCRA, it is computationally intractable to determine whether or not a guessed plaintext can produce a given

encryption. Hence, the Paillier encryption scheme is computationally secure.

2.5. Plaintexts not efficiently solvable

Another privacy notion is *plaintexts not efficiently solvable*. An encryption scheme is secure in this sense if the adversaries cannot solve plaintexts via observing ciphertexts by any polynomial-time algorithm. This privacy notion is weaker than computational security and perfect secrecy since indistinguishability implies unsolvability, but the reverse direction may not be true. For this privacy notion, the security of an encryption scheme is usually not established by a proof, but claimed in a heuristic manner by checking all existing solving algorithms and claiming that none of them is efficient. Well-known encryption schemes that adopt this privacy notion include RSA, DES (Data Encryption Standard) and AES (Advanced Encryption Standard). We next use RSA as an example to illustrate this privacy notion. The example is modified from the example in Section 1.3.2 of Yi et al. (2014).

Example 2.3. Again, consider the case of three agents, P1, P2 and P3, where P1 is an entity that is equipped with superior computing resources and could be corrupted, while P2 and P3 are benign. Each agent i holds a private input x_i. Let $x_1 = 2,\ x_2 = 5$ and $x_3 = 9$. P2 requests P1 to compute $f = x_1x_2x_3$ for it. Here, we consider product computation because the RSA encryption scheme is multiplicatively homomorphic. Since P1 could be corrupted, P2 and P3 cannot directly send x_2 and x_3 to P1. Instead, P2 could adopt the RSA encryption scheme to achieve secure computation of f. The steps are as follows.

Key generation: P2 randomly chooses two large prime numbers p and q; computes $n = pq$ and $\phi = (p-1)(q-1)$; chooses an integer $1 < e < \phi$ such that $\gcd(e, \phi) = 1$; computes $d = e^{-1} \bmod \phi$. The public keys are (n, e) and the private key is d. P2 publicizes (n, e) to P1 and P3 while keeps (d, p, q, ϕ) private to itself.

Let $p = 17$ and $q = 19$. We have $n = pq = 323$ and $\phi = 288$. Let $e = 5$. It can be checked that $\gcd(e, \phi) = 1$. We then have $d = 173$.

Encryption: Each agent i encrypts x_i by the encryption operation $E(\cdot)$ as $y_i = E(x_i, n, e) = x_i^e \bmod n$. P2 and P3 send y_2 and y_3 to P1.

We have $y_1 = 2^5 \bmod 323 = 32,\ y_2 = 5^5 \bmod 323 = 218$, and $y_3 = 9^5 \bmod 323 = 263$.

Evaluation: P1 computes $\bar{f} = y_1y_2y_3 \bmod n$ and sends $\bar{f}$ to P2.

We have $\bar{f} = 32 \cdot 218 \cdot 263 \bmod 323 = 48$.

Decryption: P2 decrypts $\bar{f}$ by the decryption operation $D(\cdot)$ as $\hat{f} = D(\bar{f}, n, d) = \bar{f}^d \bmod n$.

We have $\hat{f} = 48^{173} \bmod 323 = 90 = x_1x_2x_3$.

The correctness of RSA follows its multiplicatively homomorphic property: the product of encryptions is an encryption of the product of the private inputs. The privacy of RSA is established under the assumption that large integer factorization is computationally intractable. Notice that if one can factorize n to obtain the values of p and q, then it can easily compute ϕ. By knowing the public key e, it can further compute the value of d and perform decryption operations to obtain the values of the private inputs. However, when p and q are very large, e.g., in the magnitude of 2000, there is no known way that can factorize n in a reasonable amount of time. Hence, RSA is secure in the sense of plaintexts not efficiently solvable. On the other hand, since RSA is a deterministic encryption scheme, i.e., the encryption of a plaintext is unique under the same encryption keys, one can easily determine which values are not the private inputs. Hence, RSA is not computationally secure.

Table 1
Original inpatient microdata.

	Non-sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13,053	28	Russian	Heart Disease
2	13,068	29	American	Heart Disease
3	13,068	21	Japanese	Viral Infection
4	13,053	23	American	Viral Infection
5	14,853	50	Indian	Cancer
6	14,853	55	Russian	Heart Disease
7	14,850	47	American	Viral Infection
8	14,850	49	American	Viral Infection
9	13,053	31	American	Cancer
10	13,053	37	Indian	Cancer
11	13,068	36	Japanese	Cancer
12	13,068	35	American	Cancer

3. Review of private data release

In this section, we first motivate private data release by a real-world example. After that, we review four most widely used privacy notions for private data release, namely, k-anonymity, ℓ-diversity, t-closeness, and differential privacy.

3.1. A motivating example

Data release is commonplace in our daily life. On the one hand, analyzers can use the released data to perform statistical studies. On the other hand, the participants may insist that their privacy must not be breached through the released data. We next use the example of medical data publish to motivate the problem of private data release. This example is adopted from Machanavajjhala et al. (2007).

• *Publishing medical data.* Consider the scenario where a hospital needs to publish a set of medical data for some research purposes. Table 1 is the original table of the medical data. In Table 1, Zip code, Age and Nationality are non-sensitive attributes, called quasi-identifiers, which can be used to potentially identify an individual. While, Condition (disease) is a sensitive attribute and it is necessary that one cannot tell the Condition type of any individual patient through published data. If Table 1 is directly published, then if one knows an individual patient's Zip code, Age and Nationality, say, 13,068 and 21 and Japanese, respectively, then it can tell the identity of this patient in the table, that is, the third row, and further tell that this patient has viral infection.

3.2. k-anonymity, ℓ-diversity and t-closeness

The above example shows that a dataset has to be processed before release for privacy preservation. In this subsection, we review three notions for privacy preserving data release, namely, k-anonymity, ℓ-diversity, and t-closeness.

3.2.1. k-anonymity

The notion of k-anonymity was introduced by Pierangela Samarati and Latanya Sweeney in their work (Samarati & Sweeney, 1998). Roughly speaking, k-anonymity protects identity privacy by requiring that each group of records with the same values of quasi-identifiers (e.g., gender, zip code, age) must include at least k records. Its formal definition is given next.

Definition 3.1 (Li et al. (2007))**.** An equivalence class of a table is a set of records that have the same values for the quasi-identifiers. □

Table 2
4-anonymous inpatient microdata.

	Non-sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Definition 3.2 (Samarati & Sweeney (1998))**.** A table satisfies k-anonymity if each equivalence class has at least k individuals. □

There are two common methods to achieve k-anonymity for a given value of k: suppression and generalization. In particular, the method of suppression replaces certain values of non-sensitive attributes by an asterisk '*'; and the method of generalization replaces individual values of non-sensitive attributes by a broader category. The following Table 2 is an anonymized version of Table 1, in which the entries of Zip code, the last four entries of Age, and the entries of Nationality are anonymized by the method of suppression, and the first eight entries of Age are anonymized by the method of generalization. By Definition 3.1, Table 2 has three equivalence classes: rows 1–4, rows 5–8, and rows 9–12. Since each equivalence class has four records, Table 2 has 4-anonymity.

3.2.2. ℓ-diversity

A major weakness of the k-anonymity model is that it may not protect sensitive attributes. For example, in Table 2, if one knows that an individual participant has Zip code beginning with 130 and is in its thirties, then he can uniquely infer that this participant has cancer. This is because the values of the sensitive attributes within the equivalence class lack diversity. To overcome this weakness, Ashwin Machanavajjhala et al. extends k-anonymity and proposes the notion of ℓ-diversity in their work (Machanavajjhala et al., 2007). In principle, ℓ-diversity requires that each equivalence class has at least ℓ "well-represented" values for each sensitive attribute. The simplest understanding of "well-represented" would be to require that there are at least ℓ distinct values for each sensitive attribute in each equivalence class. The corresponding definition is given next.

Definition 3.3 (Machanavajjhala et al. (2007))**.** A table is (distinct) ℓ-diverse if each equivalence class has at least ℓ different values for all the sensitive attributes. □

In the following Table 3, each equivalence class has at least three different types of disease. Hence, Table 3 has (distinct) 3-diversity. The paper (Machanavajjhala et al., 2007) proposed two other interpretations of "well-represented": entropy diversity and recursive diversity. In particular, entropy diversity quantifies diversity by the entropy of an equivalence class; and recursive diversity sets a relation between the most frequent values and the most infrequent values, so as to ensure that the most frequent values do not appear too frequently, and the most infrequent ones do no appear too rarely. Please refer to Machanavajjhala et al. (2007) for the details of these two interpretations.

Table 3
3-diverse inpatient microdata.

	Non-sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Table 4
Original salary/disease table.

	Non-sensitive		Sensitive	
	Zip Code	Age	Salary	Disease
1	47,677	29	3K	Gastric Ulcer
2	47,602	22	4K	Gastritis
3	47,678	27	5K	Stomach Cancer
4	47,905	43	6K	Gastritis
5	47,909	52	11K	Flu
6	47,906	47	8K	Bronchitis
7	47,605	30	7K	Bronchitis
8	47,673	36	9K	Pneumonia
9	47,607	32	10K	Stomach Cancer

Table 5
A 3-diverse version of Table 4.

	Non-sensitive		Sensitive	
	Zip Code	Age	Salary	Disease
1	476**	2*	3K	Gastric Ulcer
2	476**	2*	4K	Gastritis
3	476**	2*	5K	Stomach Cancer
4	4790*	≥ 40	6K	Gastritis
5	4790*	≥ 40	11K	Flu
6	4790*	≥ 40	8K	Bronchitis
7	476**	3*	7K	Bronchitis
8	476**	3*	9K	Pneumonia
9	476**	3*	10K	Stomach Cancer

3.2.3. t-closeness

One weakness of ℓ-diversity is that it may not be enough to protect privacy in the case where one has auxiliary information of some *a priori* skewed distribution of sensitive attributes. For example, assume that 1% of the overall population have cancer. If one knows that an individual is in the first equivalence class of Table 3, then this individual would be considered to have 50% probability of having cancer, which is much higher than the prior probability 1%. Another weakness of ℓ-diversity is that it does not take into account the semantical closeness of sensitive attribute values in an equivalence class. Distinct but semantically similar sensitive attribute values could disclose much information. Consider the following example adopted from Li et al. (2007). Table 4 is an original salary/disease table, in which Zip code and Age are non-sensitive attributes, and Salary and Disease are sensitive attributes. By using the methods of suppression and generalization, we can obtain Table 5, which has 3-diversity. Assume that ones knows that an individual belongs to the first equivalence class, then it knows that this individual's salary is in the range 3K–5K and can further infer that this individual belongs to the low salary population. Moreover, it can also conclude that this individual has certain stomach-related problem, because all the three diseases in the first

Table 6
Table that has 0.167-closeness w.r.t. Salary and 0.278-closeness w.r.t. Disease.

	Non-sensitive		Sensitive	
	Zip Code	Age	Salary	Disease
1	4767*	≤40	3K	Gastric Ulcer
3	4767*	≤40	5K	Stomach Cancer
8	4767*	≤40	9K	Pneumonia
4	4790*	≥40	6K	Gastritis
5	4790*	≥40	11K	Flu
6	4790*	≥40	8K	Bronchitis
2	4760*	≤40	4K	Gastritis
7	4760*	≤40	7K	Bronchitis
9	4760*	≤40	10K	Stomach Cancer

equivalence are stomach-related. Such information leakage is due to that the sensitive attribute values in an equivalence class are semantically close to each other (although distinct).

To overcome the above issues, Ninghui Li et al. proposed the notion of t-closeness in their work (Li et al., 2007). This notion is a further extension of ℓ-diversity. In principle, t-closeness requires that the distribution of each sensitive attribute in any equivalence class is close to the distribution of this attribute in the overall table. The work of Li et al. (2007) proposed to use the Earth Mover's distance (EMD) (Rubner, Tomasi, & Guibas, 2000) to quantify the closeness between two distributions. The EMD is formulated as the minimal amount of work required to transform one distribution to another via moving probability mass between each other. We refer to Sections 4 and 5 of Li et al. (2007) for the formal definition of the EMD and methods to calculate it, respectively. The formal definition of t-closeness is given next.

Definition 3.4 (Li et al. (2007))**.** An equivalence class is said to have t-closeness if the EMD between the distribution of any sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness. □

The following Table 6 is another anonymized version of Table 4. It is easy to see that Table 6 also has 3-diversity. By computing the EMD, we can obtain that Table 6 has 0.167-closeness with respect to Salary and 0.278-closeness with respect to Disease. Compared with Table 5, the similarity attack is avoided in Table 6. In particular, given an individual in any of the three equivalence classes of Table 6, one cannot tell whether this individual belongs to the low salary or high salary population, or whether this individual has a stomach-disease or not.

3.3. Differential privacy

Differential privacy was proposed by Dwork et al. (2006). Different from the three notions introduced in Section 3.2 which anonymize released data via suppression and generalization, differentially private schemes protect data privacy by adding random perturbations to released data. The amount of perturbations embodies the tradeoff between data utility and data privacy: larger perturbations lead to less useful results but higher levels of privacy, while smaller perturbations achieve the opposite. An advantage of differential privacy is that differentially private schemes can resist arbitrary auxiliary information. Informally speaking, a random mechanism is differentially private if its behaviors (outputs) over similar databases (inputs) are also similar, so that an adversary cannot tell whether an individual's record is in the released data or not from the outputs of the mechanism. The similarity between databases is defined by the following notion of adjacency.

Definition 3.5 (Dwork & Roth (2014b))**.** Two databases $D = \{d_1, \cdots, d_n\}$ and $D' = \{d'_1, \cdots, d'_n\}$ are said to be adjacent if there exists $i \in \{1, \cdots, n\}$ such that $d_j = d'_j$ for all $j \neq i$. □

By Definition 3.5, we see that two databases are adjacent if they differ on at most one element. The formal definition of differential privacy is given next.

Definition 3.6 (Dwork & Roth (2014b))**.** Given ε, $\delta \geq 0$, a randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and all adjacent databases $D, D' \in \mathcal{D}$, it holds that:

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta. \tag{3}$$

□

In the above definition, if $\delta = 0$, then we say that $\mathcal{M}$ is ε-differentially private. The inequality (3) defines the similarity between the outputs generated over two adjacent databases required by differential privacy. Notice that the smaller the values ε and δ, the more similar the outputs $\mathcal{M}(D)$ and $\mathcal{M}(D')$. If ε and δ are both 0, then $\mathcal{M}(D)$ and $\mathcal{M}(D')$ are exactly the same. The privacy level guaranteed by Definition 3 can be interpreted as follows. Consider an individual i whose data d_i is to be protected. Since (3) must hold for all adjacent databases, if a mechanism $\mathcal{M}$ is differentially private, then even if an adversary knows the exact values of all the other elements of the input database, it cannot tell an individual's participation status (i.e., whether an individual's record is in the released data or not) from the output of $\mathcal{M}$.

There have been various differentially private mechanisms, among which the most common one is the Laplace mechanism, introduced next.

• Laplace mechanism. Given a random variable x, we write $x \sim \text{Lap}(b)$ to denote that x is drawn from the Laplace distribution with parameter b, that is, the density function $p(x)$ of x is given by $p(x) = \frac{1}{2b}\exp(-|x|/b)$. Roughly speaking, given a function f, the Laplace mechanism perturbs the output of f by adding a noise drawn from the Laplace distribution. The magnitude of the noise is determined by the sensitivity of f and the desired privacy level. In particular, the sensitivity of a function f is defined as

$$\Delta f = \max_{D, D' \in \mathcal{D}: D, D' \text{are adjacent}} |f(D) - f(D')|.$$

That is, the sensitivity of f captures the largest deviation in the output of f when a single individual's data changes. Intuitively, the added noise should flatten this largest possible deviation in order to hide the change in a single individual's data. The following theorem formalizes this intuition.

Theorem 3.1 (Dwork & Roth (2014b))**.** *Given a function $f : \mathcal{D} \to \mathbb{R}$ with sensitivity Δf. The Laplace mechanism $\mathcal{M}(D) = f(D) + x$ with $x \sim \text{Lap}(\Delta(f)/\varepsilon)$ is ε-differentially private.* □

4. Cross-comparison between privacy notions

In the previous two sections, we review two sets of privacy notions, one for SMC and the other for private data release, and provide inter-comparisons for each set of notions. In this section, we further provide a cross-comparison between the two sets.

The most essential distinction is that the problem settings are different. In particular, SMC studies how to securely and correctly compute given functions, while private data release studies how to privately output given set of data while maintaining certain utilities. There is a fundamental utility-privacy tradeoff in private data release: disclosing fully accurate information maximizes data utility but minimizes data privacy, while disclosing random noises achieves the opposite (Li & Li, 2009). Such utility-privacy tradoff

does not exist in SMC. On the other hand, SMC requires perfect correctness, which is absent in private data release.

Due to the above distinction, the two sets of privacy notions seek for different privacy objectives. Informally speaking, the objective of the privacy notions for SMC is that nothing about the concerned private data can be learned from the observations (e.g., shares or encryptions of the private data). However, the "nothing is learned" definition cannot be adopted for private data release because such a strong privacy requirement intrinsically inhibits any meaningful data utility (Dwork & Roth, 2014a). In contrast, the notions for private data release aim to ensure that there is enough uncertainty in the concerned private data by observing the released outputs.

The above distinctions in the privacy objectives necessitate different techniques to solve the problems. The techniques for SMC can achieve perfect correctness, but are usually time-consuming, due to, e.g., operations over large integers involved in cryptographic techniques. In contrast, the techniques for private data release are more computationally efficient, but cannot achieve perfect correctness, due to, e.g., usage of perturbations.

5. Recent works on CPS privacy

CPS privacy has been attracting increasing research interests in the control community. In this section, we summarize recent works in this area. This section compare the works from aspects important and unique to CPS privacy, including privacy notions, privacy-enhancing techniques, attacker models, computation requirements, and impacts on control system performance.

5.1. SMC-related works

There are mainly two branches for SMC-related works, classified by the privacy notions.

Standard notions for SMC. One branch of works used the standard privacy notions in Section 2. The work of Lu and Zhu (2015a) used the Shamir's secret sharing scheme to achieve perfect secrecy for distributed optimization problems on tree topologies in the presence of semi-honest attackers. In particular, the joint computations in a gradient method are abstracted as collective sum computations over the agents' private inputs. Each agent shares its private input to its neighbors and then follows an iterative protocol to sum up the shares from its further neighbors in the network. The approach is fully decentralized and extends the traditional Shamir's secret sharing scheme from complete communication topologies to tree topologies. Recently, homomorphic encryption has attracted growing attentions in the control community. The works of Lu and Zhu (2015b), Shoukry et al. (2016) and Lu and Zhu (2018b) applied homomorphic encryption to securely compute gradient-based algorithms for solving potential games, quadratic programs and distributed optimization, respectively. In particular, the works of Lu and Zhu (2015b) and Lu and Zhu (2018b) considered the scenario where both the agents and the computing entity had private inputs and the joint functions could be arbitrary polynomials. The work of Lu and Zhu (2015b) proposed a privacy preserving algorithm based on a private key fully homomorphic encryption scheme, where multiplications and additions had to be carried out by two different computing entities. The work of Lu and Zhu (2018b) first extended the result of Lu and Zhu (2015b) such that only one computing entity was needed to carry out both multiplications and additions, under a temporarily independent attack assumption. It then proposed a Paillier encryption based privacy preserving algorithm for weighted sum computations, which did not require this assumption. The work of Shoukry et al. (2016) considered the scenario where the joint functions were weighted sum computations and only the agents had private inputs. It proposed a privacy preserving algorithm based on the Paillier encryption scheme and discussed how to securely carry out projection operations for non-negative dual variables. The works of Kogiso and Fujita (2015) and Farokhi, Shames, and Batterham (2017) studied linear control systems and used homomorphic encryption to securely carry out linear state or output feedback controllers. In particular, the work of Kogiso and Fujita (2015) adopted multiplicatively homomorphic encryption schemes (RSA and ElGamal) to securely compute the individual products in the multiplication of the gain matrix and the state/output vector in the case where the computing entity did not know either the gain or the state/output. The work of Farokhi et al. (2017) adopted additively homomorphic encryption scheme (Paillier) to securely compute the overall inner product of the gain matrix and the state/output in the case where the computing entity knew the gain but did not know the state/output. The work of Freris and Patrinos (2016) adopted the Paillier encryption to securely compute the weighted sums in consensus protocols where the weights were known to the computing entity. The aforementioned homomorphic encryption-based works all considered semi-honest attackers. Most of the works only briefly mentioned that fixed-point arithmetic were used to handle real-valued data, but did not give implementation details. One exception is the work of Lu and Zhu (2018b), which proposed a transformation scheme between non-negative integers and signed real numbers and elaborated on how to integrate the transformation scheme into standard homomorphic encryption schemes to guarantee computing correctness. These works required a third entity, e.g., an aggregator or an operator, to carry out computations over encrypted data. Under such a framework, these works achieved computational security.

Non-unique determination. Another branch of works defined privacy as that the private data and/or a finite range of the private data cannot be uniquely determined. Such privacy notions are weaker than the standard privacy notions for SMC in Section 2. This is because the notions allow disclose of partial information, e.g., linear combinations of the private inputs, as long as the private inputs and/or their ranges cannot be uniquely determined from the disclosed partial information, while the standard notions for SMC require that nothing is disclosed about the private inputs. On the other hand, the weaker privacy notions allow for other properties desirable for CPSs. Along this line, obfuscation techniques have been proposed to protect coefficient confidentiality for optimization problems in cloud computing in the presence of semi-honest attackers. Roughly speaking, the legitimate problem holder applies an obfuscation transformation to the original optimization problem and sends the obfuscated problem to the cloud. Upon receiving an optimal solution of the obfuscated problem, the legitimate holder can retrieve an optimal solution of the original problem by inverting the obfuscation transformation. For example, the works of Borden, Molzahn, Ramanathan, and Lesieutre (2012), Borden, Molzahn, Lesieutre, and Ramanathan (2013) and Wang, Ren, and Wang (2016) applied obfuscation techniques to optimal power flow (OPF) problems and linear programming problems, respectively. In particular, the work of Borden et al. (2012) adopted obfuscation techniques to protect confidential power system parameters, which appeared as input matrices and vectors in the objective function and constraints of OPF problems. The work of Borden et al. (2013) extended (Borden et al., 2012) by removing a restrictive positive monomial condition on the transformation matrix so that the transformation could be applied to a broader class of OPF problems. The work of Wang et al. (2016) studied obfuscation techniques for general linear programming problems to protect the optimal solution as well as the input matrices and vectors in the objective function and equality and inequality constraints. Without using operations over

large integers, obfuscation-based approaches are in general more computationally efficient than cryptography-based approaches. On the other hand, existing obfuscation techniques are only applicable to single-agent centralized optimization problems with linear or quadratic cost functions in order to have the property that inverting the linear obfuscation transformation returns the optimal solution of the original problem. Using the same type of privacy notions, another set of works combined homomorphic encryption with unique properties of traditional decentralized algorithms for certain control problems and the resulted algorithms are fully decentralized. Roughly speaking, these works exploited the presence of some penalty or weight parameters that can be randomly chosen to eliminate unique determination of private data, and further adopted homomorphic encryption to resist eavesdroppers. Representative works include Zhang, Ahmand, and Wang (2019) and Zhang and Wang (to appear) which studied decentralized optimization, and Ruan, Ahmand, and Wang (2017), Gao, Zhang, Ahmand, and Wang (2018) and Ruan, Gao, and Wang (to appear) which studied average consensus. In particular, the work of Zhang et al. (2019) introduced a new alternating direction method of multipliers (ADMM) with time-varying penalty matrices which allowed random weight choices. The work of Zhang and Wang (to appear) leveraged the property that the weights in the estimate update rule of projected subgradient algorithm for decentralized optimization could be randomly chosen. The work of Ruan et al. (2017) and its journal version (Ruan et al., to appear) decomposed the weights in the adjacency matrix of the underlying average consensus problem so that the two decomposed weights could be randomly chosen. The work of Gao et al. (2018) proposed to use randomly chosen weights in the push-sum algorithm for average consensus problems. On top of the random weight-chosen mechanisms, these works adopted the Paillier encryption scheme to securely carry out the involved weighted sum computations. All these works considered semi-honest attackers. The works of Ruan et al. (2017, to appear) also considered active attackers and used the technique of digital signature to enable detection of message forging or tampering attacks. Besides the above two sets of works, more recently, there emerges a singular work (Altafini, 2019) using a different technique to achieve the same type of privacy notions for initial states in average consensus against semi-honest attackers. This work adopted time-varying output maps to mask intermediate states such that the masked system asymptotically converges to the original system. This approach is fully decentralized as the output maps are implemented locally.

5.2. Private data release-related works

There are mainly three branches for private data release-related works, classified by the privacy notions.

Differential privacy. The first branch of works used differential privacy as the privacy notion. For example, differentially private algorithms have been developed for distributed consensus (Huang, Mitra, & Dullerud, 2012), optimization (Hale & Egerstedt, 2015; 2018; Han, Topcu, & Pappas, 2017; Nozari, Tallapragada, & Cortes, 2016), filtering (Ny & Pappas, 2014), linear distributed control systems (Wang, Huang, Mitra, & Dullerud, 2017), routing games (Dong, Krichene, Bayen, & Sastry, 2015), and linear quadratic Gaussian (LQG) controller design (Zhang, Shu, Cheng, & Chen, 2016). These works considered semi-honest attackers and extended traditional differential privacy mechanisms from static settings to dynamic settings by properly redefining adjacency relations and sensitivity analysis. The papers of Cortes et al. (2016) and Han and Pappas (2018) provided excellent reviews on differential privacy in control applications. Please refer to them for more detailed discussions on this set of works. A major advantage of differential privacy is that differentially private schemes can resist arbitrary auxiliary information of the attackers. Moreover, by following systematic ways of adding random noises, differentially private schemes can achieve some predetermined privacy level in a mathematically rigorous and convenient manner. On the other hand, to achieve differential privacy, it is necessary to add persistent randomized perturbations into released data (Geng & Viswanath, 2014). For control systems, such persistent random perturbations could potentially deteriorate system utilities.

Mutual information. The second branch of works quantified privacy levels using mutual information. Roughly speaking, the mutual information of two random variables is a measure of their mutual dependence, and it quantifies the amount of information, called information entropy, obtained about one random variable through observing the other random variable (Cover & Thomas, 1991). In the context of CPS privacy, the mutual information of the private data and the released data is used to measure how much information of the private data is disclosed through the released data. For example, the work of Venkitasubramaniam, Yao, and Pradhan (2015) studied both information extraction and information injection attacks in stochastic control systems and used Shannon entropy to quantify the tradeoffs between information security and physical system performance. The work of Han, Topcu, and Pappas (2016) proposed an event-based mutual information metric against information extraction attacks and analyzed this metric for the best-effort policy in smart grid. The work of Tanaka, Skoglund, Sandberg, and Johansson (2017) used causally conditioned directed information to measure privacy loss against information extraction attacks in cloud-based control and proposed a private filter that attained the optimal tradeoff between privacy loss and control quality. An advantage of the mutual information-based approach is that privacy loss is quantified by information entropy, which is standard in information theory, and various well-developed mathematical models and tools in probability theory and information theory can be used to analyze privacy loss. On the other hand, a disadvantage is that it requires explicit statistical models of source data and auxiliary information (Venkitasubramaniam et al., 2015). In contrast, in many CPSs, system states and inputs (source data) and/or prior knowledge of the system (auxiliary information) do not follow any probabilistic distribution.

Unobservability. The third branch of works defined privacy in the spirit of unobservability in control theory. This type of notions is consistent with k-anonymity and ℓ-diversity, which requires that there is enough uncertainty/diversity in the private data. For example, the works of Xue, Wang, and Roy (2014) and Roy, Xue, and Das (2012) used unobservability as the privacy notion and provided algebraic and graph-theoretic characterizations for the security and discoverability of network spread dynamics. The work of Mo and Murray (2017) also used unobservability to define privacy of initial states in average consensus and proposed a perturbation-based mechanism to minimize the observable subspace. The works of Lu and Zhu (2018a); Zhu and Lu (2015) considered privacy of both initial states and inputs in linear dynamic networks. They adopted non-strong observability-inspired privacy notions and proposed a closed-loop perturbation-based method to simultaneously achieve privacy, maintain controllability, and minimize perturbation costs. All these works considered semi-honest attackers. For event-driven dynamic systems, there is a set of works that defined privacy in terms of opacity, which is similar to system unobservability in spirit. Roughly speaking, a system is opaque if for every secret-induced behavior, there exists a different behavior not induced by the secret that generates identical observations. If a system is not originally opaque, one popular opacity-enforcing approach is to design a minimally restrictive supervisor to disable behaviors that violate opacity (Dubreil, Darondeau, & Marchand, 2010; Saboori &

Hadjicostis, 2012; Yin & Lafortune, 2016). Another representative opcity-enforcing approach is to insert fictitious events at system outputs (Ji, Wu, & Lafortune, 2018; Wu & Lafortune, 2014). The paper of Jacob, Lesage, and Faure (2016) provided an excellent review on opacity-based works for discrete event systems. Please refer to it for more detailed discussions on this set of works. All the above opacity-based works also considered semi-honest attackers. An advantage of the observability-inspired and opacity-based notions is that privacy is defined as a system property so that control-theoretic tools can be used to design protection schemes to simultaneously achieve privacy and other desirable system utilities, e.g., controllability. On the other hand, existing works in this branch cannot resist arbitrary auxiliary information of the attackers.

6. Privacy preserving distributed optimization

In this section, we use our recent work (Lu & Zhu, 2018b) on privacy preserving distributed optimization to exemplify how to achieve SMC in CPS in which there exists a third party. This work is representative in the following aspects. First, our work achieved the standard privacy notions for SMC reviewed in Section 2. Second, other cryptography-based works achieving the standard SMC privacy notions only studied partial homomorphic encryption schemes, while our work studied both fully and partial homomorphic encryption schemes. Third, our work provided an elaborate transformation mechanism to handle real-valued data, while the other works only briefly mentioned that fixed-point arithmetic was used to do so. Fourth, our work studied privacy preserving computation for general gradient-based algorithms for distributed optimization problems and it is easy to directly apply or customize our approach to other problems, e.g., linear and quadratic programming problems, and linear control problems, in which a third party exists.

6.1. Problem formulation

In this subsection, we first present a distributed gradient-based algorithm and identify its privacy issues. After that, we introduce the adopted attacker model and privacy notions.

6.1.1. Gradient-based distributed optimization and its privacy issues

Consider a set of agents $\mathcal{V}=\{1,\cdots,N\}$ and a system operator (SO). The agents aim to solve a distributed optimization problem by the projected gradient method, in which each agent $i\in\mathcal{V}$ performs the following update rule:

$$x_i(k+1)=\mathbb{P}_{X_i}[x_i(k)-\gamma(k)\Phi_i(x(k))]. \tag{4}$$

In (4), k denotes the discrete step index; $x_i=[x_{i\ell}]$ is agent i's state, $X_i\subseteq\mathbb{R}^{n_i}$ is the feasible set of x_i, and $x=[x_i]_{i\in\mathcal{V}}\in\mathbb{R}^n$ with $n=\sum_{i\in\mathcal{V}}n_i$; $\gamma(k)>0$ is the step size at step k; $\Phi_i=[\Phi_{i\ell}]:\mathbb{R}^n\to\mathbb{R}^{n_i}$ is the first-order gradient of certain component functions of the underlying optimization problem with respect to x_i. In this section, we assume that there is an undirected communication link between each agent i and the system operator. Moreover, we assume that each function $\Phi_{i\ell}$ is a polynomial of x. The algebra of polynomials can approximate any continuous function over a compact domain arbitrarily well (DeVore & Lorentz, 1993). Denote by C_Φ the set of coefficients of the polynomial functions $(\Phi_1,\cdots,\Phi_N)$. For each $i\in\mathcal{V}\cup\{\mathrm{SO}\}$, participant i holds a subset of coefficients of C_Φ, denoted by C_Φ^i. We have $\bigcup_{i\in\mathcal{V}\cup\{\mathrm{SO}\}}C_\Phi^i=C_\Phi$. For ease of presentation, we assume that each coefficient is only held by one participant, i.e., $C_\Phi^i\cap C_\Phi^j=\emptyset$ for any $i,j\in\mathcal{V}\cup\{\mathrm{SO}\}$ and $i\neq j$. For each $i\in\mathcal{V}\cup\{\mathrm{SO}\}$, let $m_i=|C_\Phi^i|$ and let $c_i=[c_{i\ell}]\in\mathbb{R}^{m_i}$ be the vector of the elements of C_Φ^i. Let $c=[c_i]_{i\in\mathcal{V}\cup\{\mathrm{SO}\}}\in\mathbb{R}^m$ with $m=\sum_{i\in\mathcal{V}\cup\mathrm{SO}}m_i$ and let $c_\mathcal{V}=[c_i]_{i\in\mathcal{V}}$.

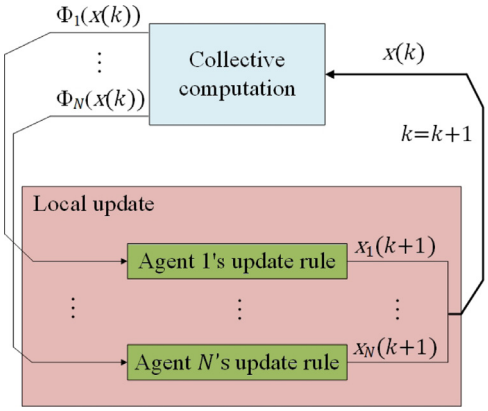

Fig. 1. Illustration of the execution of (4).

We next identify the privacy issues in the execution of (4). In distributed optimization, the state and feasible set of an agent could expose much information about the agent's behaviors and thus should not be leaked to other entities. For example, a demand response problem involves a set of end-users that aim to achieve optimal power loads such that the total cost induced by disutility and load charge is minimized and the load benefit is maximized. This problem can be formulated as a distributed optimization problem in which each end-user's state is its power load (Li, Chen, & Low, 2011). It has been shown that power load profiles at a granularity of 15 minutes may reveal whether a child is left alone at home and at a finer granularity may reveal the daily routines of end-users (Gong, Cai, Guo, & Fang, 2016). Moreover, the coefficients of a distributed optimization problem could be confidential and must be kept private to unauthorized entities. For example, an optimal power flow problem involves a set of power generators that aim to find optimal mechanical power and phase angles such that the operating cost is minimized. This problem can be formulated as a distributed optimization problem in which the coefficients of the flow balance equality constraints are line-dependent parameters, e.g., tie-line stiffness coefficient, of the power system (Wood & Wollenberg, 1996). It has been pointed out in Borden et al. (2013, 2012) that leakage of line-dependent parameters could be financially damaging and even cause potential threat to national security.

With the above discussions, we define $\{\{x_i(k)\},X_i,C_\Phi^i\}$ as private data of agent $i\in\mathcal{V}$. For the system operator, its private data only includes C_Φ^{SO}. We assume that the system operator knows the algebraic structure of $[\Phi_i]_{i\in\mathcal{V}}$ but is unaware of the values of x and the coefficients $C_\Phi\backslash C_\Phi^{\mathrm{SO}}$. In (4), for each $i\in\mathcal{V}$, notice that the computation of $\Phi_i(x(k))$ depends on private data $(x_{-i}(k),C_\Phi^{-i})$ of other participants and thus requires data exchange between the participants. Once agent i obtains the value of $\Phi_i(x(k))$, it can locally update $x_i(k)$. Hence, we decompose the execution of (4) into two parts as shown by Fig. 1: the collective computation part where all participants collectively compute $\Phi_i(x(k))$ for all $i\in\mathcal{V}$ and the local update part where each agent i updates its state $x_i(k)$ given $\Phi_i(x(k))$.

6.1.2. Attacker model and privacy notions

In this section, we are concerned with semi-honest adversaries, that is, any adversary $i\in\mathcal{V}\cup\{\mathrm{SO}\}$ correctly follows the algorithm but attempts to use its received messages throughout the execution of the algorithm to infer other participants' private data ((Hazay & Lindell, 2010), pp-20). This attacker model is broadly used in SMC (Cramer et al., 2015; Hazay & Lindell, 2010) and has been adopted in, e.g., linear programming, dataset process

and consensus (Dreier & Kerschbaum, 2011; Freedman, Nissim, & Pinkas, 2004; Huang et al., 2012). We assume that the adversaries do not collaborate with each other. Instead, if multiple adversaries collaborate, they are viewed as a single adversary.

In this section, we adopt computational security given by Definition 2.5 and the notion of plaintexts not efficiently solvable given in Section 2.5.

6.2. Private key secure computation algorithm

This section introduces a secure computation algorithm for the update rule (4) which is based on a private key fully homomorphic encryption scheme.

6.2.1. Preliminaries

For each $i \in \mathcal{V} \cup \{\text{SO}\}$, since the coefficient vector c_i is only known to participant i, we view Φ_j's for $j \in \mathcal{V}$ as functions of both x and c. Denote by $\Phi^s_{i\ell} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ the function with the same algebraic structure of $\Phi_{i\ell}$ but also takes c also as variables and let $y = [y_i] \in \mathbb{R}^m$ be the variables corresponding to c, that is, $\Phi_{i\ell}(x) = \Phi^s_{i\ell}(x, y)|_{y=c}$. Assume that each $\Phi^s_{i\ell}$ is written in the form of sum of monomials as $\Phi^s_{i\ell}(x, y) = \sum_{\nu=1}^{\kappa_{i\ell}} Q^\nu_{i\ell}(x, y)$, where $\kappa_{i\ell}$ is the number of monomials and $Q^\nu_{i\ell} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the νth monomial of $\Phi^s_{i\ell}$.

Most existing homomorphic encryption schemes only work for integers. However, the variables and coefficients in distributed optimization problems are usually real numbers. This necessitates a mechanism for transformation between real numbers and integers. Throughout this section, the accuracy level is set by a parameter $\sigma \in \mathbb{N}$, which means that for any real number, σ decimal fraction digits are kept while the remaining decimal fraction digits are dropped. We assume that σ is known to the system operator and all the agents. Given a real number r, it is transformed into an integer z_r by $z_r = 10^\sigma r$. Given $\sigma \in \mathbb{N}$ and an odd positive integer w, an integer $z \in \mathbb{Z}_w$ is transformed into a signed real number by the following function parameterized by σ and w:

$$T_{\sigma,w}(z) = \begin{cases} z/10^\sigma, \text{ if } 0 \le z \le (w-1)/2, \\ (z-w)/10^\sigma, \text{ if } (w+1)/2 \le z < w. \end{cases} \tag{5}$$

The following property holds for the transformation (5).

Lemma 6.1. *Given an odd positive integer w, for any $r \in \mathbb{R}$ with σ decimal fraction digits such that $|10^\sigma r| \le (w-1)/2$, it holds that $T_{\sigma,w}(10^\sigma r \bmod w) = r$.*

6.2.2. Algorithm design and analysis

The private key secure computation algorithm for the update rule (4) is presented by Algorithm 1. This algorithm is based on the SingleMod encryption, which is an efficiently implementable private key fully homomorphic encryption scheme for integers (Dijk et al., 2010; Phatak et al., 2014). The algorithm is informally stated as follows.

At line 2, all the agents agree on a key w and keep it private to the system operator. One case where such key agreement is possible is that the sub-communication graph between the agents is connected and all its communication links are secure. For the sake of security, w needs to be a very large number, e.g., in the magnitude of 2^{2000} (Giry, 2017).

At line 3, each agent i encrypts its coefficient c_i as $\hat{y}_i$ via the SingleMod encryption using the key w and a random integer vector u_{y_i}. Notice that $10^\sigma c_i$ is an integer vector. Since c_i is fixed throughout the computing process, it only needs to be encrypted once.

At line 5, each agent i encrypts its current state $x_i(k)$ as $\hat{x}_i(k)$ via the SingleMod encryption using the key w and a random integer vector $u_{x_i}(k)$. At each iteration, the encryption of $x_i(k)$ uses a fresh randomness $u_{x_i}(k)$.

Algorithm 1 Private key secure computation algorithm.

1. Initialization: Each agent i chooses any $x_i(0) \in X_i$;
2. Key agreement: All agents agree on a large odd positive integer w and keep w secret from SO;
3. Coefficient encryption: Each agent i chooses any $u_{y_i} \in \mathbb{Z}^{m_i}_{>0}$ and sends to SO $\hat{y}_i$ computed as $\hat{y}_i = u_{y_i} w + 10^\sigma c_i$; SO forms $\hat{y} = [\hat{y}_j]_{j \in \mathcal{V} \cup \{\text{SO}\}}$ with $\hat{y}_{\text{SO}} = 10^\sigma c_{\text{SO}}$;
4. while $k \ge 0$
5. State encryption: Each agent i chooses any $u_{x_i}(k) \in \mathbb{Z}^{n_i}_{>0}$ and sends to SO $\hat{x}_i(k)$ computed as $\hat{x}_i(k) = u_{x_i}(k) w + 10^\sigma x_i(k)$; SO forms $\hat{x}(k) = [\hat{x}_j(k)]_{j \in \mathcal{V}}$;
6. Computation over ciphertexts: For each $i \in \mathcal{V}$, for each $\ell = 1, \cdots, n_i$, SO sends $\bar{\Phi}^s_{i\ell}(k)$ to agent i computed as:

$$\bar{\Phi}^s_{i\ell}(k) = \sum_{\nu=1}^{\kappa_{i\ell}} [10^{(\deg(\Phi^s_{i\ell}) - \deg(Q^\nu_{i\ell}))\sigma} Q^\nu_{i\ell}(\hat{x}(k), \hat{y})];$$

7. Decryption: For each $i \in \mathcal{V}$, for each $\ell = 1, \cdots, n_i$, agent i computes $\hat{\Phi}^s_{i\ell}(k) = T_{\deg(\Phi^s_{i\ell})\sigma, w}(\bar{\Phi}^s_{i\ell}(k) \mod w)$;
8. Local update: Each agent i forms $\hat{\Phi}^s_i(k) = [\hat{\Phi}^s_{i\ell}(k)]$ and computes $x_i(k+1) = \mathbb{P}_{X_i}[x_i(k) - \gamma(k) \hat{\Phi}^s_i(k)]$;
9. Set $k \leftarrow k+1$;
10. end while

At line 6, the system operator evaluates each function $\Phi^s_{i\ell}$ over $(\hat{x}(k), \hat{y})$. Notice that each monomial $Q^\nu_{i\ell}(\hat{x}(k), \hat{y})$ is multiplied by $10^{(\deg(\Phi^s_{i\ell}) - \deg(Q^\nu_{i\ell}))\sigma}$. This is to have each monomial scaled by the same times so that the sum operation can be performed. In particular, without the $u_{x_i}(k)w$ (resp. $u_{y_i}w$) part, which will be eliminated by modulo operation in the decryption step, each $x_i(k)$ (resp. c_i) is scaled by 10^σ times in the integer transformation operation at line 5 (resp. 3). Each $Q^\nu_{i\ell}(\hat{x}(k), \hat{y})$ is then scaled by $10^{\deg(Q^\nu_{i\ell})\sigma}$ times. By multiplying $10^{(\deg(\Phi^s_{i\ell}) - \deg(Q^\nu_{i\ell}))\sigma}$, each $10^{(\deg(\Phi^s_{i\ell}) - \deg(Q^\nu_{i\ell}))\sigma} Q^\nu_{i\ell}(\hat{x}(k), \hat{y})$ is scaled by the same $10^{\deg(\Phi^s_{i\ell})\sigma}$ times. Hence, all the monomials of $\Phi^s_{i\ell}$ are scaled by the same times and can be summed up.

Line 7 is the decryption step. Each agent i first performs modulo operation over the encrypted function value $\bar{\Phi}^s_{i\ell}(k)$ by the key w. All the terms having w as a factor are eliminated from $\bar{\Phi}^s_{i\ell}(k)$. Then, agent i transforms the remainder into a signed real number by (5). To guarantee the correctness of decryption, the value of w cannot be too small. Roughly speaking, w must be larger than all possible plaintexts of computing results. This is captured by the following assumption. One sufficient condition for this assumption is that x lives in a compact set whose bound is known to the agents. Even if this sufficient condition does not hold, since w is chosen very large for the sake of security (e.g., in the magnitude of 2^{2000}), this assumption is usually automatically satisfied.

Assumption 6.1. The key w is chosen large enough such that, for any step k, it holds that $w \ge 1 + 2 \max_{i,\ell} 10^{\deg(\Phi^s_{i\ell})\sigma} |\Phi_{i\ell}(x(k))|$. □

At line 8, each agent i locally updates $x_i(k)$ by (4) using $\hat{\Phi}^s_i(k)$ as $\Phi_i(x(k))$.

The following assumption is required to guarantee privacy against the system operator.

Assumption 6.2. The system operator can only perform temporarily independent attacks, i.e., at each step k, it uses the data received at step k to infer $x(k)$, but does not use past data to collectively infer the sequence $x(0), \cdots, x(k)$. □

One scenario where Assumption 6.2 holds is that the system operator is not fully aware of the update rule of $x(k)$ and views the sequence $\{x(k)\}$ as a temporally independent time series (Shi, Chan,

FxPal, Chow, & Song, 2011). This scenario has been widely considered in database privacy and many works in the field are based on the temporal independence assumption. For example, the work of Bhaskar, Bhowmick, Goyal, Laxman, and Thakurta (2011) studied noiseless database privacy under the assumption that the entries in the database are uncorrelated. Moreover, as pointed out by Dwork et al. (2010) and Chapter 14 of Zhu, Li, Zhou, and Yu (2017), most existing differential privacy works assume that the dataset consists of independent records, despite the fact that records in real world applications are often correlated.

Theorem 6.1. *Suppose that Assumptions 6.1 and 6.2 hold. By Algorithm 1, the following claims hold:*

1) Correctness: $\hat{\Phi}_i^s(k) = \Phi_i(x(k))$ *for any* k *and* $i \in \mathcal{V}$.

2) Security: Algorithm 1 is computationally secure against each agent, and at each step k, *Algorithm 1 is as hard as the approximate greatest common divisor (GCD) problem* [1] *against the system operator.* □

6.2.3. Discussion and extension

Algorithm 1 has two major limitations: (i) it requires the agents to agree on a private key; (ii) its security level against the system operator is not computational security, and as a consequence, it can only resist temporarily independent attacks against the system operator.

The reason of the second limitation is that the encryption scheme of Algorithm 1 leverages the SingleMod encryption, which is not computationally secure (Phatak et al., 2014). Instead, the privacy of the SingleMod encryption is claimed in the sense of Plaintexts not efficiently solvable, which is weaker than computational security (Yi et al., 2014). On the other hand, the weaker security level renders that Algorithm 1 is fully homomorphic and efficiently implementable. As mentioned in Section 2.5, this sense of privacy claim has been adopted in many widely used encryption schemes, e.g., RSA, DES and AES. To the best of our knowledge, there does not exist a fully homomorphic encryption scheme which is both efficiently implementable and computationally secure.

In Section 4 of our paper (Lu & Zhu, 2018b), we consider the special case where $\Phi_{i\ell}$'s are affine functions and propose a public key secure computation algorithm which addresses the above two limitations. In particular, the encryption scheme of the public key secure computation algorithm leverages the Paillier encryption scheme (Paillier, 1999); please see Example 2.2. To compute Φ_i, agent i generates a set of Paillier keys, and publicizes the encryption keys while keeping the decryption keys private to itself. By knowing the encryption keys, any other participant can perform encryption. While, without knowing the decryption keys, no one but agent i can decrypt the ciphertexts. In the above procedure, since the encryption keys can be publicized, the key distribution problem is solved. Moreover, since the Paillier encryption scheme is computationally secure, our algorithm is also computationally secure against the system operator and can resist causal attacks against the system operator.

7. Privacy preserving dynamic data release

In this section, we use our recent work (Lu & Zhu, 2018a) on privacy preserving data release of linear dynamic networks to exemplify how to achieve private data release in the context of CPS. We choose to use this work as an example based on the following reasons. First, we would like to mention that the branch of differential privacy-based works is the most fruitful one among other private data release-related works in CPS privacy. However, as the works of Cortes et al. (2016) and Han and Pappas (2018) have already provided excellent reviews on differential privacy in various control applications, here we want to avoid repetition. Moreover, our work (Lu & Zhu, 2018a) is representative in developing control-theoretic privacy preserving techniques. The privacy notion of our work is observability-inspired and thus is intrinsically suitable for dynamic systems. Furthermore, to the best of our knowledge, our approach for the first time used state-feedback perturbations, rather than random perturbations in differential privacy. As a result, our approach is superior in reducing impacts of perturbations on control system performance, because other system utilities, e.g., stability or controllability, can be simultaneously taken into account in the perturbation design. Preliminary results of Lu and Zhu (2018a) were published in Zhu and Lu (2015).

7.1. Problem statement

In this subsection, we introduce the network model, the attacker model and the privacy notion adopted in this section.

7.1.1. Network model

Consider an interconnected dynamic network of $\mathcal{V} = \{1, \cdots, N\}$. Each agent $i \in \mathcal{V}$ has the following linear discrete-time physical dynamics:

$$\begin{aligned} x_i(k+1) &= \bar{A}_{ii}x_i(k) + \sum_{j\in\mathcal{N}_i} \bar{A}_{ij}x_j(k) + \bar{B}_i u_i(k) \\ y_i'(k) &= \bar{G}_i' x_i(k) + \bar{H}_i' u_i(k). \end{aligned} \tag{6}$$

In (6), $x_i(k) \in \mathbb{R}^{n_i}$, $u_i(k) \in \mathbb{R}^{p_i}$ and $y_i'(k) \in \mathbb{R}^{l_i}$ are the state, input and output of agent i at time instant k, respectively, and $\mathcal{N}_i \subseteq \mathcal{V} \setminus \{i\}$ is the set of agents whose states affect x_i. The collection system (6) for all the agents can be compactly written as follows:

$$x(k+1) = \bar{A}x(k) + \bar{B}u(k) \tag{7}$$

$$y'(k) = \bar{G}'x(k) + \bar{H}'u(k) \tag{8}$$

where $x(k) = [x_i(k)] \in \mathbb{R}^n$, $u(k) = [u_i(k)] \in \mathbb{R}^p$ and $y'(k) = [y_i'(k)] \in \mathbb{R}^l$, with $n = \sum_{i\in\mathcal{V}} n_i$, $p = \sum_{i\in\mathcal{V}} p_i$ and $l = \sum_{i\in\mathcal{V}} l_i$. The matrices $\bar{A}$, $\bar{B}$, $\bar{G}'$ and $\bar{H}'$ are system parameters known to the agents. The communication topology between the agents is defined by a digraph $\mathcal{G}^C = (\mathcal{V}, E^C)$, where $(i, j) \in E^C$ if agent j can send messages to agent i. Let S^x (resp. S^u) be the set of (i, ℓ) such that $x_{i\ell}$ (resp. $u_{i\ell}$) can be measured by agent i. Hence, S^x and S^u represent the sets of sensor locations and $\bar{G}'$ and $\bar{H}'$ in (8) are restricted by S^x and S^u, respectively.

7.1.2. Attacker model

There is an external data requester who requests the following linear combinations of the agents' individual outputs in (8):

$$y(k) = \Pi y'(k) = \bar{G}x(k) + \bar{H}u(k) \tag{9}$$

where $\Pi \in \mathbb{R}^{q\times l}$, $\bar{G} = \Pi\bar{G}' \in \mathbb{R}^{q\times n}$ and $\bar{H} = \Pi\bar{H}' \in \mathbb{R}^{q\times p}$. The data requester determines the query matrix Π and tells the agents its valuation. The agents release the requested output in (9) to the data requester but is unaware of how the released data will be used. In the rest of this section, we use (A, B, G, H) to represent arbitrary system matrices. In particular, $(\bar{A}, \bar{B}, \bar{G}, \bar{H})$ are specific valuations of (A, B, G, H) that represent system (7) and (9).

The data requester is assumed to be semi-honest and aims to exploit $y(k)$ to infer some entries of $x(0)$ and $\{u(k)\}$. This attacker model is motivated by several practical scenarios, e.g., smart building (Lisovich et al., 2010) and load monitoring in smart grid (McLaughlin, McDaniel, & Aiello, 2011). We assume that the data requester knows the matrices $\bar{A}$, $\bar{B}$, $\bar{G}'$ and $\bar{H}'$. This knowledge models the auxiliary/side information of the adversary.

[1] Approximate GCD: Given polynomially many integers randomly chosen close to multiples of a large integer p, i.e., in the form of $a_i = pq_i + r_i$, where p, q_i's and r_i's are all integers, find the "common near divisor" p. This problem is widely believed to be NP-hard (Dijk et al., 2010; Phatak et al., 2014).

7.1.3. Privacy notion

Next we introduce the privacy notion adopted in this section. The data requester aims to infer the values of partial (potentially all) entries of the initial state $x(0)$ and the input sequence $\{u(k)\}$, called target entries. The remaining entries of $x(0)$ and $\{u(k)\}$ are called nontarget entries. Denote by $x^t(0)$ and $x^n(0)$ (resp. u^t and u^n) the column vectors of the target and nontarget entries of $x(0)$ (resp. u), respectively. Let d_x^t, d_u^t, d_x^n and d_u^n be the dimensions of $x^t(0)$, u^t, $x^n(0)$ and u^n, respectively. We have $d_x^t + d_x^n = n$ and $d_u^t + d_u^n = p$. Denote by $x_\ell^t(0)$ (resp. u_ℓ^t, $x_\ell^n(0)$ and u_ℓ^n) the ℓth entry of $x_\ell^t(0)$ (resp. u^t, $x^n(0)$ and u^n). A target entry u_ℓ^t is said to be protected if and only if $u_\ell^t(k)$ is protected for any $k \in \mathbb{N}$. In other words, if the value of $u_\ell^t(k)$ for one time instant k is disclosed to the data requester, then the privacy of u_ℓ^t is compromised.

Given system matrices (A, B, G, H), by Section 4.2.2 of Chen (1999), the output $y(k)$ at each time instant k can be expressed as a linear combination of the entries of $x(0)$ and $u_{[0,k]}$:

$$y(k) = GA^k x(0) + \sum_{m=0}^{k-1} GA^{k-1-m} Bu(m) + Hu(k). \tag{10}$$

Given system matrices (A, B, G, H) and time instant $\kappa \in \mathbb{N}$, for any feasible output sequence $y_{[0,\kappa]}$, we define a set $\Delta_{A,B,G,H}(y_{[0,\kappa]})$ as:

$$\begin{aligned}&\Delta_{A,B,G,H}(y_{[0,\kappa]}) = \{x^t(0), u_{[0,\kappa]}^t : \exists x^n(0), u_{[0,\kappa]}^n,\\ &\text{s.t. } y(k) = \text{right-hand-side of (10)}, \forall k = 0, \cdots, \kappa,\\ &\text{with } x(0) \text{ the composition of } x^t(0) \text{ and } x^n(0) \text{ and } u(k)\\ &\text{the composition of } u^t(k) \text{ and } u^n(k), \forall k = 0, \cdots, \kappa\}.\end{aligned}$$

The set $\Delta_{A,B,G,H}(y_{[0,\kappa]})$ includes all possible valuations of $\{x^t(0), u_{[0,\kappa]}^t\}$ that can generate $y_{[0,\kappa]}$ in (10). We define the diameter of $\Delta_{A,B,G,H}(y_{[0,\kappa]})$ as:

$$\text{Diam}_{A,B,G,H}(y_{[0,\kappa]}) = \sup_{w,w' \in \Delta_{A,B,G,H}(y_{[0,\kappa]})} \|w - w'\|_{\min}.$$

Notice that $\|w - w'\|_{\min}$ is the smallest distance along all the target entries between w and w'. Also notice that all the elements in $\Delta_{A,B,G,H}(y_{[0,\kappa]})$ can produce the same output sequence $y_{[0,\kappa]}$. Hence, a larger diameter $\text{Diam}_{A,B,G,H}(y_{[0,\kappa]})$ implies a larger uncertainty on all the target entries, and an infinite diameter achieves the largest possible uncertainties of the target entries. This observation leads to the following privacy definition.

Definition 7.1. Given system matrices (A, B, G, H), the privacy of $x^t(0)$ and u^t is said to be protected if, for any $\kappa \in \mathbb{N}$, $\text{Diam}_{A,B,G,H}(y_{[0,\kappa]}) = \infty$ for any feasible output sequence $y_{[0,\kappa]}$. •

Definition 7.1 is extended from the notion of ℓ-diversity; please see Section 3.2. Recall that possessing ℓ-diversity essentially means that there are at least ℓ different values for each sensitive attribute of the dataset in the released table. A larger diversity indicates a larger uncertainty and thus the notion of diversity can be viewed as a measure of uncertainty. In ℓ-diversity, the diversity of discrete-valued sensitive attributes is defined by the number of different valuations for the attributes. In contrast, the target entries $x^t(0)$ and u^t in our problem are continuous-valued and uncountable. This requires a new measure to quantify the diversity/uncertainty. We propose to measure the diversity/uncertainty by the diameter of the set $\Delta_{\hat{A},\hat{B},\hat{G},\hat{H}}(y_{[0,\kappa]})$. Hence, Definition 7.1 extends ℓ-diversity from the discrete-valued setting to the continuous-valued setting.

7.1.4. Privacy preserving data release

To protect privacy, we propose to perturb the inputs and outputs such that the data requester cannot infer the target entries. However, the perturbations should maintain certain system utilities, e.g., system controllability. Throughout this section, we assume that the original system $(\bar{A}, \bar{B})$ is controllable and aim to maintain controllability of the perturbed system. These partially conflicting sub-objectives define the problem of *privacy preserving data release*.

7.2. Intentional input-output perturbations

In this subsection, we first formulate intentional input-output perturbations by a class of optimization problems. After that, we analyze the computational complexity of the formulated optimization problem.

7.2.1. Optimization formulation

To protect privacy, each agent i intentionally perturbs its own input $u_i(k)$ and output $y_i'(k)$ by adding signals $\mu_i^u(k) \in \mathbb{R}^{p_i}$ and $\mu_i^y(k) \in \mathbb{R}^{l_i}$, respectively. The perturbations $\mu_i^u(k)$ and $\mu_i^y(k)$ are linear combinations of system states and inputs and given by:

$$\begin{aligned}\mu_i^u(k) &= \sum_{j\in\mathcal{V}} K_{ij}^{SS} x_j(k) + \sum_{j\in\mathcal{V}} K_{ij}^{SI} u_j(k)\\ \mu_i^y(k) &= \sum_{j\in\mathcal{V}} K_{ij}^{OS} x_j(k) + \sum_{j\in\mathcal{V}} K_{ij}^{OI} u_j(k).\end{aligned} \tag{11}$$

The superscript SI means a perturbation from an input to a state. Other superscripts are defined analogously, with O denoting output. By adding the perturbations $\mu^u(k) = [\mu_i^u(k)]$ and $\mu^y(k) = [\mu_i^y(k)]$ into (7) and (9), we obtain the following perturbed system:

$$x(k+1) = \bar{A}x(k) + \bar{B}(u(k) + \mu^u(k)) = \hat{A}x(k) + \hat{B}u(k) \tag{12}$$

$$y(k) = \Pi(\bar{G}'x(k) + \bar{H}'(u(k) + \mu^u(k)) + \mu^y(k)) = \hat{G}x(k) + \hat{H}u(k) \tag{13}$$

where $\hat{A} = \bar{A} + \bar{B}K_{SS}$, $\hat{B} = \bar{B}(I_p + K_{SI})$, $\hat{G} = \bar{G} + \bar{H}K_{SS} + \Pi K_{OS}$ and $\hat{H} = \bar{H} + \bar{H}K_{SI} + \Pi K_{OI}$, with $K_{SS} = [K_{ij}^{SS}] \in \mathbb{R}^{p\times n}$, $K_{SI} = [K_{ij}^{SI}] \in \mathbb{R}^{p\times p}$, $K_{OS} = [K_{ij}^{OS}] \in \mathbb{R}^{l\times n}$ and $K_{OI} = [K_{ij}^{OI}] \in \mathbb{R}^{l\times p}$. Let $K = \begin{bmatrix} K_{SS} & K_{SI} \\ K_{OS} & K_{OI}\end{bmatrix} \in \mathbb{R}^{(p+l)\times(n+p)}$. We assume that K is known to the data requester. This is another piece of auxiliary information available to the data requester. In the rest of this section, we use $(\hat{A}, \hat{B}, \hat{G}, \hat{H})$ to specifically represent the perturbed system (12) and (13). The perturbation matrix K is subject to the following three constraints:

(i) The perturbation positions are restricted by the sensing and communication capabilities specified by E^C, S^x and S^u.

(ii) The perturbed system $(\hat{A}, \hat{B})$ needs to remain controllable.

(iii) The data requester cannot infer the target entries from the outputs (13).

The first constraint is captured by a set $\mathbb{K} \subseteq \mathbb{R}^{(p+l)\times(n+p)}$, which specifies the zero-nonzero structure of K. For example, the specification could be that $K \in \mathbb{K}$ if and only if K is in the form of $\begin{bmatrix} 0 & * & * \\ & 0 & * \end{bmatrix}$, where $*$ indicates that the corresponding entry could be any real number, while 0 means that the corresponding entry must be 0 because no sensor and/or communication link can be installed at this position.

Meanwhile, the agents aim to minimize the additional sensing and communication costs induced by the perturbations. If one entry of K_{ji}^{SS} or K_{ji}^{OS} (resp. K_{ji}^{SI} or K_{ji}^{OI}) is nonzero, then agent i needs to sense the corresponding component of $x_i(k)$ (resp. $u_i(k)$) and send it to agent j. Notice that E^C, S^x and S^u characterize the existing communication and sensing capacities, respectively. Define matrix $\mathcal{L} \in \{0,1\}^{(p+l)\times(n+p)}$ such that $\mathcal{L}_{\ell\ell'} = 0$ if there exist both a communication link and a sensor at position (ℓ, ℓ') in the original system and $\mathcal{L}_{\ell\ell'} = 1$ otherwise. If $\mathcal{L}_{\ell\ell'} = 0$, adding perturbation at position (ℓ, ℓ') generates zero additional cost. If $\mathcal{L}_{\ell\ell'} = 1$, adding perturbation at position (ℓ, ℓ') generates one additional cost (unit

cost). The minimization of the additional sensing and communication costs caused by the perturbations can be realized by maximizing the sparsity of K and equivalently minimizing the ℓ_0 norm of K over $\mathcal{L}$, i.e., $\min \|K\|_{0,\mathcal{L}}$, where $\|K\|_{0,\mathcal{L}} \triangleq \sum_{\ell,\ell'} 1_{[T_{\ell\ell'}\neq 0]}$, with $T = [T_{\ell\ell'}] = K \circ \mathcal{L}$.

All the above objectives are encoded in the following optimization problem:

$$\begin{aligned} &\min_{K\in\mathbb{K}} \|K\|_{0,\mathcal{L}} \\ &\text{s.t. } \mathrm{Diam}_{\hat{A},\hat{B},\hat{C},\hat{H}}(y_{[0,\kappa]}) = \infty, \forall \kappa \in \mathbb{N} \text{ and feasible } y_{[0,\kappa]}, \\ &\quad (\hat{A},\hat{B}) \text{ is controllable.} \end{aligned} \tag{14}$$

Problem (14) is referred to as the structured ℓ_0 minimization and denoted by $\mathbb{P}_{0,\mathcal{L}}$. Different from differentially private schemes, the perturbations given by (11) are added in a closed-loop fashion and diminishing as the system is stabilized. Furthermore, data privacy has a fundamental utility-privacy tradeoff: disclosing fully accurate information maximizes data utility (i.e., quality of data analysis) but minimizes data privacy, while disclosing random noises achieves the opposite (Li & Li, 2009). Besides data utility, our optimization formulation (14) allows us to take into account dynamic system utilities, e.g., controllability.

7.2.2. Relaxation of problem (14)

The first constraint of (14) has a clear privacy interpretation, but is not analytically tractable. In this subsection, we relax this privacy constraint by a rank constraint.

Given system matrices (A, B, G, H), for any $z \in \mathbb{C}$, define matrix pencil $D_{A,B,G,H}(z) \triangleq \begin{bmatrix} zI_n - A & -B \\ G & H \end{bmatrix}$. For any $v \in \mathbb{R}^{n+p}$, we write $v = [v_1^T, v_2^T]^T$ with $v_1 \in \mathbb{R}^n$ and $v_2 \in \mathbb{R}^p$. Let $v_1^t = [v_{1\ell}^t]$ (resp. $v_2^t = [v_{2\ell}^t]$) be the sub-vector of v_1 (resp. v_2) corresponding to $x^t(0)$ (resp. u^t). The following lemma provides a sufficient condition for the privacy of $x^t(0)$ and u^t.

Lemma 7.1. *Given a linear system (A, B, G, H), the privacy of $x^t(0)$ and u^t is protected if there exists a pair of $z \in \mathbb{C}$ and $v \in \mathbb{R}^{n+p}\backslash\{\mathbf{0}_{n+p}\}$ satisfying $D_{A,B,G,H}(z)v = \mathbf{0}_{n+q}$ such that the following two conditions are satisfied simultaneously:*

(1) if $d_x^t \neq 0$, then $v_{1\ell}^t \neq 0$ for any $\ell \in \{1, \cdots, d_x^t\}$;

(2) if $d_u^t \neq 0$, then $z \neq 0$ and $v_{2\ell}^t \neq 0$ for any $\ell \in \{1, \cdots, d_u^t\}$.•

An intuition of Lemma 7.1 is that one can protect more entries of $x(0)$ and u by reducing the rank of $D_{A,B,G,H}(z)$. This is verified by the following lemma.

Lemma 7.2. *Given (A, B, G, H), if there exists $z \neq 0$ such that $D_{A,B,G,H}(z)$ has column rank r, then at least $n + p - r$ entries of $x(0)$ and u are protected.•*

In the rest of this section, let $\bar{D}(z) = D_{\bar{A},\bar{B},\bar{C},\bar{H}}(z)$ and $\hat{D}(z) = D_{\hat{A},\hat{B},\hat{C},\hat{H}}(z)$. We can derive that $\hat{D}(z) = \bar{D}(z) + FK$ with $F = \begin{bmatrix} -\bar{B} & \mathbf{0}_{n\times l} \\ \bar{H} & \Pi \end{bmatrix}$. By Lemma 7.2, by reducing the rank of $\hat{D}(z)$, it is more likely that one can protect more entries of $x^t(0)$ and u^t. By this observation, we relax problem (14) as follows:

$$\begin{aligned} &\min_{K\in\mathbb{K}, z\in\mathbb{C}} \|K\|_{0,\mathcal{L}} \\ &\text{s.t. } \mathrm{rank}(\bar{D}(z) + FK) < \rho, \\ &\quad (\hat{A},\hat{B}) \text{ is controllable} \end{aligned} \tag{15}$$

where $\rho \in [1, \min\{n + p, n + q\}]$ is a constant integer. Denote problem (15) by $\tilde{\mathbb{P}}_{0,\mathcal{L}}$. Given an optimal solution (K, z) of problem (15), if the null space of $\bar{D}(z) + FK$ admits a vector v such that $v_{1\ell}^t \neq 0$, then the ℓth entry of $x^t(0)$ is protected; if $z \neq 0$ and the null space of $\bar{D}(z) + FK$ admits a vector v such that $v_{2\ell}^t \neq 0$, then the ℓth entry of u^t is protected. If some entries of $x^t(0)$ and u^t are not protected, we then decrease ρ and re-solve problem (15). Our objective is to protect all the entries of $x^t(0)$ and u^t with the largest possible ρ. The remaining issue is how to solve problem (15) under a given ρ. The next subsection shows that this problem could be NP-hard.

7.2.3. Computational intractability

This subsection analyzes the computational complexity of problem (15). The next theorem shows the non-convexity of the constraint set of problem (15), indicating that the problem might be hard to solve.

Theorem 7.1. *Assume $\mathbb{K} = \mathbb{R}^{(p+l)\times(n+p)}$. The constraint set of problem (15) is non-convex.•*

Consider the following problem derived by fixing z and dropping the controllability constraint of problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$:

$$\min_{K\in\mathbb{K}} \|K\|_{0,\mathcal{L}} \quad \text{s.t. } \mathrm{rank}(\bar{D}(z) + FK) < \rho. \tag{16}$$

Denote problem (16) by $\hat{\mathbb{P}}_{0,\mathcal{L}}(z)$. By fixing z, the dimension of the decision variables is reduced. By Theorem 7.1, the controllability constraint of problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$ is non-convex. Hence, intuitively, problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$ might be harder to solve than problem $\hat{\mathbb{P}}_{0,\mathcal{L}}(z)$. The following theorem states that problem $\hat{\mathbb{P}}_{0,\mathcal{L}}(z)$ is NP-hard due to the non-convexity of its objective function. This provides an implication that problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$ might also be NP-hard.

Theorem 7.2. *Problem $\hat{\mathbb{P}}_{0,\mathcal{L}}(z)$ is NP-hard.•*

Please refer to Garey and Johnson (1979) and Leeuwen (1990) for a thorough introduction of the complexity theory. The proof of Theorem 7.2 is established by showing that problem $\hat{\mathbb{P}}_{0,\mathcal{L}}(z)$ is as hard as finding a sparsest null vector of a matrix with more columns than rows, which has been proven to be NP-hard (Coleman & Pothen, 1986). Please refer to the proof of Theorem 3.2 of our paper (Lu & Zhu, 2018a).

7.3. Relaxation of problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$

In this subsection, we provide a convex relaxation for problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$.

The first source of non-convexity is the ℓ_0 norm $\|\cdot\|_{0,\mathcal{L}}$ in problem (15). We relax the objective function $\|K\|_{0,\mathcal{L}}$ by the ℓ_1 norm heuristic as $\|\mathrm{vec}(K \circ \mathcal{L})\|_1$. Such a relaxation technique is commonplace in compressed sensing (Candes & Tao, 2005; Donoho, 2006). It is proven that the ℓ_1 norm heuristic returns the sparest solution under certain conditions, e.g., restricted isometry property (RIP) (Candes & Tao, 2005). Experiments have shown that in many cases the ℓ_1 norm heuristic can return sparse solutions even RIP does not hold (Yang & Zhang, 2011).

The second source of non-convexity is the rank constraint $\mathrm{rank}(\bar{D}(z) + FK) < \rho$. In general, rank constraint/minimization problems are hard to solve, both in theory and practice. We relax the rank constraint by the nuclear norm heuristic as $\min_{z\in\mathbb{C}, K\in\mathbb{K}} \|\bar{D}(z) + FK\|_*$. Notice that this relaxation turns the hard rank constraint into a soft constraint. The work of Fazel, Hindi, and Boyd (2004) showed that $\|M\|_*$ is the convex envelop of the function $\mathrm{rank}(M)$ on the set $\{M \in \mathbb{R}^{m\times n} | \|M\|_2 \leq 1\}$. The work of Recht, Fazel, and Parrilo (2010) further showed that, under certain conditions, e.g., RIP, the nuclear norm heuristic can return minimum-rank solutions.

The third source of non-convexity is the controllability constraint. The following lemma states that the invertibility of $I_p + K_{SI}$ is a sufficient condition for the controllability of $(\hat{A}, \hat{B})$.

Lemma 7.3. *Assume that $(\bar{A}, \bar{B})$ is controllable. If $I_p + K_{SI}$ is invertible, then $(\hat{A}, \hat{B})$ is controllable.•*

The invertibility of $I_p + K_{SI}$ is equivalent to that its determinant is non-zero. However, the determinant of $I_p + K_{SI}$ is a polynomial of

the entries of K_{SI} and is still non-convex. We then further relax the invertibility of $I_p + K_{SI}$ by the condition that K_{SI} is symmetric and $I_p + K_{SI} \succ 0$. The strict positive definite condition is usually difficult to handle and may lead to infeasibility of the problem. We relax this by a semidefinite condition as $I_p + K_{SI} - \varepsilon I_p \succeq 0$, where $\varepsilon > 0$ is a tuning parameter. It is easy to see that $(1-\varepsilon)I_p + K_{SI} \succeq 0$ is a sufficient condition for the invertibility of $I_p + K_{SI}$.

With the above relaxations, problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$ is relaxed as follows:

$$\begin{aligned} &\min_{z\in\mathbb{C},K\in\mathbb{K},K_{SI}\in\mathbb{S}^p} \|\mathrm{vec}(K\circ\mathcal{L})\|_1 + c\|\bar{D}(z)+FK\|_* \\ &\text{s.t. } (1-\varepsilon)I_p + K_{SI} \succeq 0 \end{aligned} \tag{17}$$

where $c > 0$ and $\varepsilon > 0$ are tuning parameters.

By the linear program (LP) characterization of ℓ_1 norm (page 294 of Boyd & Vandenberghe (2004)), $\min_{K\in\mathbb{K}} \|\mathrm{vec}(K\circ\mathcal{L})\|_1$ can be cast as: $\min_{K\in\mathbb{K},t\in\mathbb{R}^m} \sum_{\ell=1}^m t_\ell$, s.t. $-t \le \mathrm{vec}(K\circ\mathcal{L}) \le t$. By the semidefinite program (SDP) characterization of nuclear norm (Recht et al., 2010), $\min_{z\in\mathbb{C},K\in\mathbb{K}} c\|\bar{D}(z)+FK\|_*$ can be cast as:

$$\begin{aligned} &\min_{z\in\mathbb{C},K\in\mathbb{K},W_1\in\mathbb{S}^{n+l},W_2\in\mathbb{S}^{n+p}} c(\mathrm{Tr}(W_1)+\mathrm{Tr}(W_2)) \\ &\text{s.t. } \begin{bmatrix} W_1 & \bar{D}(z)+FK \\ (\bar{D}(z)+FK)^T & W_2 \end{bmatrix} \succeq 0. \end{aligned}$$

By the above LP and SDP characterizations, problem (17) can be equivalently cast as the following SDP:

$$\begin{aligned} &\min_{\substack{z\in\mathbb{C},K\in\mathbb{K},t\in\mathbb{R}^m,K_{SI}\in\mathbb{S}^p \\ W_1\in\mathbb{S}^{n+l},W_2\in\mathbb{S}^{n+p}}} \sum_{\ell=1}^m t_\ell + c(\mathrm{Tr}(W_1)+\mathrm{Tr}(W_2)) \\ &\text{s.t. } \quad -t \le \mathrm{vec}(K\circ\mathcal{L}) \le t, \quad (1-\varepsilon)I_p + K_{SI} \succeq 0 \\ &\qquad \begin{bmatrix} W_1 & \bar{D}(z)+FK \\ (\bar{D}(z)+FK)^T & W_2 \end{bmatrix} \succeq 0. \end{aligned} \tag{18}$$

In the above, we have relaxed problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$ into the SDP (18). There are several types of efficient algorithms for solving SDPs, e.g., interior point methods and bundle method (Vandenberghe & Boyd, 1996). These methods are implemented in commercial solvers such as Mosek, SeDuMi and CVX, and can output the value of the SDP up to an additive error ϵ in time that is polynomial in the program size and $\log 1/\epsilon$.

7.4. Discussion and extension

In the last subsections, we study the case where the agents aim to minimize the additional sensing and communication costs induced by the perturbations, i.e., $\min \|K\|_{0,\mathcal{L}}$. Another representative case is to minimize the data disutility caused by the perturbations. This can be realized by minimizing the ℓ_2 norm of the perturbation matrix, i.e., $\min \|K\|_2$. The relaxation approach in the last subsection can be applied to the ℓ_2 norm minimization via replacing the ℓ_1 norm heuristic by the SDP characterization of ℓ_2 norm (page 170 of Boyd & Vandenberghe (2004)) in (18). However, one drawback of this approach is that the parameter c can only be tuned empirically and it is challenging to estimate the total time of tuning *a priori*. For each valuation of c, one needs to numerically solve the SDP of problem (18). In Section 5 of our paper (Lu & Zhu, 2018a), we propose an alternative approach for a subclass of the ℓ_2 minimization such that one can analytically construct a feasible perturbation matrix K that satisfies the constraint $\mathrm{rank}(\bar{D}(z)+FK) < \rho$. Roughly speaking, we first fix z and analytically construct a feasible solution of perturbation matrix $K(z)$, and then minimize $\|K(z)\|_2$ over $z \in \mathbb{C}$ and analytically derive a suboptimal solution of z. This approach is more systematic as one can determine the largest possible tuning time of ρ *a priori*. Moreover, this approach is computationally more efficient than numerically solving the SDP of problem (18). Some crucial steps of this approach rely on certain properties of singular value decomposition (SVD) that only hold for ℓ_2 norm, but do not hold for ℓ_0 norm or ℓ_1 norm. Hence, this approach is not suitable for problem $\tilde{\mathbb{P}}_{0,\mathcal{L}}$.

8. Conclusion

In this paper, we first provide a tutorial on existing privacy notions for data privacy of ICT systems introduced in the computer science community. Second, we provide a review on recent works in CPS privacy and compare existing works from the perspectives of privacy notions, privacy-enhancing techniques, attacker models, computation requirements, and impacts on control system performance. Third, we use two recent representative works to exemplify how to address new issues in CPS privacy.

Conflict of interest

None.

References

Altafini, C. (2019). A dynamical approach to privacy preserving average consensus, arXiv:1808.08085.pdf

Bhaskar, R., Bhowmick, A., Goyal, V., Laxman, S., & Thakurta, A. (2011). Noiseless database privacy. In *Proceedings of the international conference on the theory and application of cryptology and information security* (pp. 215–232).

Borden, A. R., Molzahn, D. K., Lesieutre, B. C., & Ramanathan, P. (2013). Power system structure and confidentiality preserving transformation of optimal power flow problem. In *Fifty-first annual allerton conference* (pp. 1021–1028).

Borden, A. R., Molzahn, D. K., Ramanathan, P., & Lesieutre, B. C. (2012). Confidentiality-preserving optimal power flow for cloud computing. In *Fiftieth annual allerton conference* (pp. 1300–1307).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

California Public Utilities Commission (2010). CA senate bill 1476. *Technical report*.

Candes, E., & Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, (12), 4203–4215.

Cavoukian, A. (2012). Smart meters in Europe: Privacy by design at its best. *Technical report*. Information and Privacy Commissioner, Ontario, Canada.

Chase, M. (2007). Multi-authority attribute based encryption. In *Proceedings of the 4th conference on theory of cryptography* (pp. 515–534).

Chen, C.-T. (1999). *Linear system theory and design*. Oxford University Press.

Coleman, T., & Pothen, A. (1986). The null space problem. I. Complexity. *SIAM Journal of Algebraic Discrete Methods, 7*(4), 527–537.

Cortes, J., Dullerud, G. E., Han, S., Ny, J. L., Mitra, S., & Pappas, G. J. (2016). Differential privacy in control and network systems. In *Proceedings of the 2016 ieee 55th conference on decision and control* (pp. 4252–4272).

Cottrill, C., & Thakuriah, P. (2011). Privacy in context: An evaluation of policy-based approaches to location privacy protection. *Journal of the Transportation Research Board*, (2215), 67–74.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. John Wiley and Sons Ltd.

Cramer, R., Damgård, I., & Nielsen, J. B. (2015). *Secure multiparty computation and secret sharing*. Cambridge University Press.

Dent, A., & Price, G. (2005). Certificate management using distributed trusted third parties. In C. Mitchell (Ed.), *Trusted computing, chapter 9. the Institution of engineering and technology* (pp. 251–270).

DeVore, R. A., & Lorentz, G. G. (1993). *Constructive approximation*. Springer-Verlag.

Dijk, M. V., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In *Proceedings of EUROCRYPT* (pp. 24–43).

Dong, R., Krichene, W., Bayen, A. M., & Sastry, S. S. (2015). Differential privacy of populations in routing games. In *2015 ieee 54th annual conference on decision and control* (pp. 2798–2803).

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, (4), 1289–1306.

Dreier, J., & Kerschbaum, F. (2011). Practical privacy-preserving multiparty linear programming based on problem transformation. In *2011 IEEE third international conference on privacy, security, risk and trust (PASSAT)* (pp. 916–924).

Dubreil, J., Darondeau, P., & Marchand, H. (2010). Supervisory control for opacity. *IEEE Transactions on Automatic Control, 55*(5), 1089–1100.

Dwork, C. (2006). Differential privacy. In *3rd international colloquium on automata, languages and programming* (pp. 1–12).

Dwork, C., & Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the 41st annual ACM symposium on theory of computing* (pp. 371–380).

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. *Lecture Notes in Computer Science, 3876*, 265–284.

Dwork, C., Naor, M., Pitassi, T., & Rothblum, G. N. (2010). Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium of theory of computing* (pp. 715–724).

Dwork, C., & Roth, A. (2014a). The algorithm foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science, 9*(3–4), 211–407.

Dwork, C., & Roth, A. (2014b). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science, 9*(3–4), 211–407.

Evans, P. C., & Annunziata, M. (2012). Industrial internet: Pushing the boundaries of minds and machines. *General Electric*. https://www.ge.com/docs/chapters/Industrial_Internet.pdf

Farokhi, F., Shames, I., & Batterham, N. (2017). Secure and private control using semi-homomorphic encryption. *Control Engineering Practice, 67*, 13–20.

Fazel, M., Hindi, H., & Boyd, S. (2004). Rank minimization and applications in system theory. In *Proc. of American control conference* (pp. 3273–3278).

Freedman, M. J., Nissim, K., & Pinkas, B. (2004). Efficient private matching and set intersection. In *Proceedings of advances in cryptology, EUROCRYPT 2004* (pp. 1–19).

Freris, N. M., & Patrinos, P. (2016). Distributed computing over encrypted data. In *Proceedings of 2016 54th annual Allerton conference on communication, control, and computing (Allerton)* (pp. 1116–1122).

Gao, H., Zhang, C., Ahmand, M., & Wang, Y. (2018). Privacy-preserving average consensus on directed graphs using push-sum. In *Proceedings of the 2018 IEEE conference on communications and network security*.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.

Geng, Q., & Viswanath, P. (2014). The optimal mechanism in differential privacy. In *2014 IEEE international symposium on information theory* (pp. 2371–2375).

Gentry, C., Halevi, S., & Smart, N. P. (2012). Homomorphic evaluation of the AES circuit. In *Proceedings of the 32nd annual cryptology conference* (pp. 850–867).

Giry, D. 2017. Cryptographic key length recommendation. Technical report, BlueKrypt, https://www.keylength.com/en/8/,.

Goldreich, O. (2004). *Foundations of cryptography: Volume 2-basic applications*. Cambridge University Press.

Goldwasser, S., & Micali, S. (1982). Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14th symposium on theory of computing* (pp. 365–377).

Gong, Y., Cai, Y., Guo, Y., & Fang, Y. (2016). A privacy-preserving scheme for incentive-based demand response in the smart grid. *IEEE Transactions on Smart Grid, 7*(3), 1304–1313.

Hale, M. T., & Egerstedt, M. (2015). Differentially private cloud-based multiagent optimization with constraints. In *American control conference* (pp. 1235–1240).

Hale, M. T., & Egerstedt, M. (2018). Cloud-enabled differentially private multiagent optimization with constraints. *IEEE Transactions on Control of Network Systems, 5*(4), 1693–1706.

Han, S., & Pappas, G. J. (2018). Privacy in control and dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems, 1*, 309–332.

Han, S., Topcu, U., & Pappas, G. J. (2016). Event-based information-theoretic privacy: A case study of smart meters. In *Proc. of American control conference* (pp. 2074–2079).

Han, S., Topcu, U., & Pappas, G. J. (2017). Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control, 62*(1), 50–64.

Hazay, C., & Lindell, Y. (2010). *Efficient secure two-party protocols–techniques and constructions*. Springer.

Huang, Z., Mitra, S., & Dullerud, G. (2012). Differentially private iterative synchronous consensus. In *Proceedings of the 2012 ACM workshop on privacy in the electronic society* (pp. 81–90).

Jacob, R., Lesage, J.-J., & Faure, J.-M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control, 41*, 135–146.

Ji, Y., Wu, Y.-C., & Lafortune, S. (2018). Enforcement of opacity by public and private insertion functions. *Automatica, 93*(7), 369–378.

Jia, R., Dong, R., Sastry, S. S., & Spanos, C. J. (2017). Privacy-enhanced architecture for occupancy-based HVAC control. In *Proceedings of the 8th international conference on cyber-physical systems* (pp. 177–186).

Kogiso, K., & Fujita, T. (2015). Cyber-security enhancement of networked control systems using homomorphic encryption. In *Proceedings of 2015 ieee 54th annual conference on decision and control (CDC)* (pp. 6836–6843).

Krontiris, I., Freiling, F. C., & Dimitriou, T. (2010). Location privacy in urban sensing networks: Research challenges and directions. *IEEE Wireless Communications, 17*(5), 30–35.

Kuntze, N., Mahler, D., & Schmidt, A. U. (2006). Employing trusted computing for the forward pricing of pseudonyms in reputation systems. In *Proceedings of the 2nd international conference on automated production of cross media content for multi-channel distribution volume for workshops industrial and applications sessions* (pp. 145–149).

Leeuwen, J. V. (1990). *Handbook of theoretical computer science*. MIT Press.

Li, N., Chen, L., & Low, S. H. (2011). Optimal demand response based on utility maximization in power networks. In *Proceedings of IEEE power and energy society general meeting* (pp. 1–8).

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the 23rd international conference on data engineering* (pp. 106–115).

Li, T., & Li, N. (2009). On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 517–526).

Lindell, Y., & Pinkas, B. (2009). Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality, 1*(1), 59–98.

Lisovich, M., Mulligan, D., & Wicker, S. (2010). Inferring personal information from demand-response systems. *IEEE Security & Privacy, 8*(1), 11–20.

Lu, Y., & Zhu, M. (2015a). Game-theoretic distributed control with information-theoretic security guarantees. In *Proceedings of the 5th IFAC workshop on distributed estimation and control in networked systems: 48* (pp. 264–269). (22)

Lu, Y., & Zhu, M. (2015b). Secure cloud computing algorithms for discrete constrained potential games. In *Proceedings of the 5th IFAC workshop on distributed estimation and control in networked systems: 48* (pp. 180–185). (22)

Lu, Y., & Zhu, M. (2018a). On privacy preserving data release of linear dynamic networks. http://php.scripts.psu.edu/muz16/pdf/YL-MZ-Auto18.pdf.

Lu, Y., & Zhu, M. (2018b). Privacy preserving distributed optimization using homomorphic encryption. *Automatica, 96*(10), 314–325.

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). ℓ-Diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data, 1*(1), 1–52.

McDaniel, P., & McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *IEEE Security and Privacy, 7*(3), 75–77.

McLaughlin, S., McDaniel, P., & Aiello, W. (2011). Protecting consumer privacy from electric load monitoring. In *18th ACM conference on computer and communications security* (pp. 87–98).

McSherry, F., & Talwar, K. (2007). Mechanism design via differential privacy. In *Proceedings of the 48th annual IEEE symposium on foundations of computer science* (pp. 94–103).

Mo, Y., & Murray, R. M. (2017). Privacy preserving average consensus. *IEEE Transactions on Automatic Control, 62*(2), 753–765.

Nozari, E., Tallapragada, P., & Cortes, J. (2016). Differentially private distributed convex optimization via functional perturbation. *IEEE Transactions on Control of Network Systems, 5*(1), 395–408.

NSF Cyber-Physical Systems (CPS) Program. Technical report, http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286. 2019.

Ny, J. L., & Pappas, G. J. (2014). Differentially private filtering. *IEEE Transactions on Automatic Control, 59*(2), 341–354.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of advances in cryptology, EUROCRYPT 1999* (pp. 223–238).

Panackal, J. J., & Pillai, A. S. (2013). Privacy preserving data mining: An extensive survey. In *Proceedings of international conference on multimedia processing, communication and information technology* (pp. 297–304).

Phatak, D. S., Tang, Q., Sherman, A. T., Smith, W. D., Ryan, P., & Kalpakis, K. (2014). DoubleMod and SingleMod: Simple randomized secret-key encryption with bounded homomorphicity. Cryptology ePrint Archive, Report 2014/670. http://eprint.iacr.org/2014/670.

Recht, B., Fazel, M., & Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Journal of Algebraic Discrete Methods, 52*(3), 471–501.

Roy, S., Xue, M., & Das, S. K. (2012). Security and discoverability of spread dynamics in cyber-physical networks. *IEEE Transactions on Parallel and Distributed Systems, 23*(9), 1694–1707.

Ruan, M., Ahmand, M., & Wang, Y. (2017). Secure and privacy-preserving average consensus. In *Proceedings of the 2017 workshop on cyber-physical systems security and privacy* (pp. 123–129).

Ruan, M., Gao, H., & Wang, Y. Secure and privacy-preserving consensus. *IEEE Transactions on Automatic Control*, To appear.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision, 40*(2), 99–121.

Saboori, A., & Hadjicostis, C. N. (2012). Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Transactions on Automatic Control, 57*(5), 1155–1165.

Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical report, SRI-CSL-98-04*. SRI Computer Science Laboratory.

Sankar, L., Rajagopalan, S. R., Mohajer, S., & Poor, H. V. (2013). Smart meter privacy: A theoretical framework. *IEEE Transactions on Smart Grid, 4*(2), 837–846.

Sankar, L., Rajagopalan, S. R., & Poor, H. V. (2013). Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security, 8*(6), 838–852.

Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612–613.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell System Technical Journal, 28*(4), 656–715.

Shi, E., Chan, T.-H. H., FxPal, E. R., Chow, R., & Song, D. (2011). Privacy-preserving aggregation of time-series data. In *Proceedings of 18th annual network and distributed system security symposium*.

Shokri, R., Theodorakopoulos, G., Boudec, J.-Y. L., & Hubaux, J.-P. (2011). Quantifying location privacy. In *2011 IEEE symposium on security and privacy* (pp. 247–262).

Shoukry, Y., Gatsis, K., Alanwar, A., Pappas, G. J., Seshia, S. A., Srivastava, M., & Tabuada, P. (2016). Privacy-aware quadratic optimization using partially homomorphic encryption. In *Proceedings of the 2016 ieee 55th conference on decision and control* (pp. 5053–5058).

Tanaka, T., Skoglund, M., Sandberg, H., & Johansson, K. H. (2017). Directed information and privacy loss in cloud-based control. In *2017 American control conference* (pp. 1666–1672).

Vaghashia, H., & Ganatra, A. (2015). A survey: Privacy preservation techniques in data mining. *International Journal of Computer Applications, 119*(4), 20–26.

Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM Review*, (1), 49–95.

Venkitasubramaniam, P., Yao, J., & Pradhan, P. (2015). Information-theoretic security in stochastic control systems. *Proceedings of the IEEE, 103*(10), 1914–1931.

Wang, C., Ren, K., & Wang, J. (2016). Secure optimization computation outsourcing in cloud computing: A case study of linear programming. *IEEE Transactions on Computers, 65*(1), 216–229.

Wang, Y., Huang, Z., Mitra, S., & Dullerud, G. E. (2017). Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs. *IEEE Transactions on Control of Network Systems, 4*(1), 118–130.

Wood, A., & Wollenberg, B. F. (1996). *Power generation operation and control*. New York: Wiley.

Wu, Y.-C., & Lafortune, S. (2014). Synthesis of insertion functions for enforcement of opacity security properties. *Automatica, 50*(5), 1336–1348.

Xue, M., Wang, W., & Roy, S. (2014). Security concepts for the dynamics of communicating autonomous-unmanned-vehicle networks. *Automatica*.

Yang, J., & Zhang, Y. (2011). Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing, 33*(1), 250–278.

Yi, X., Paulet, R., & Bertino, E. (2014). *Homomorphic encryption and applications*. Springer.

Yin, X., & Lafortune, S. (2016). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control, 61*(8), 2140–2154.

Zhang, C., Ahmand, M., & Wang, Y. (2019). ADMM based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security, 14*(3), 565–580.

Zhang C., & Wang Y., Enabling privacy-preservation in decentralized optimization. *IEEE Transactions on Control Network Systems*, To appear.

Zhang, H., Shu, Y., Cheng, P., & Chen, J. (2016). Privacy and performance trade-off in cyber-physical systems. *IEEE Network, 30*(2), 62–66.

Zhu, M., & Lu, Y. (2015). On confidentiality preserving monitoring of dynamic networks against inference attacks. In *2015 American control conference* (pp. 359–364).

Zhu, T., Li, G., Zhou, W., & Yu, P. S. (2017). *Differential privacy and applications*. Springer.