2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Observability Blocking for Functional Privacy of Linear Dynamic Networks

Yuan Zhang, Ranbo Cheng, and Yuanqing Xia

***Abstract*—This paper addresses the problem of determining the minimum set of state variables in a network that need to be blocked from direct measurements in order to protect functional privacy with respect to *any* output matrices. More precisely, the goal is to prevent adversarial observers or eavesdroppers from inferring a linear functional of states, either vector-wise or entry-wise. We relate the considered functional privacy to the concept of functional observability. Building on a PBH-like criterion for functional observability, we prove that both problems are NP-hard. However, by assuming a reasonable constant bound on the geometric multiplicities of the system's eigenvalues, we present an exact algorithm with polynomial time complexity for the vector-wise functional privacy protection problem. Based on this algorithm, we then provide a greedy algorithm for the entry-wise privacy protection problem. Finally, we provide an example to demonstrate the effectiveness of our proposed approach.**

***Index Terms*—Observability blocking, network privacy preservation, functional observability, algorithms**

I. Introduction

In recent years, research on the privacy preservation of control systems has gained increasing attention due to the growing use of networked control systems, cyber-physical systems and the increasing concerns over the privacy and security of these systems [1–3].

One area of research has focused on developing privacy-preserving control algorithms that can achieve the desired control objectives while protecting sensitive information about the system's states and inputs [4, 5]. Another area of research has explored the use of differential privacy techniques to protect the privacy of data collected from control systems [1]. Recently, encryption techniques, including homomorphic and nonhomomorphic encryptions, have also been adopted to preserve data privacy in the process of network transmissions and third-party computations [2, 3, 6]. Additionally, researchers have also investigated the impact of cyber-attacks on privacy and proposed secure and privacy-preserving communication protocols to mitigate these attacks [7].

Apart from the above research perspectives, there has been a natural relationship between the privacy preservation of control systems and system observability [8–13]. Observability refers to the ability to estimate the state of a system based on its output. In the context of privacy preservation, the idea is to make it difficult for an adversary to infer sensitive information about the system by limiting their ability to observe the system's state. In this line, privacy preservation through system design has attracted much interest. To name a few, [8] considered the problem of designing communication networks so that the average consensus is achieved while the observable subspace each individual agent can infer from shared information of its neighbors is as small as possible. The design of local state-feedback control systems in dynamical networks to block observability at remote nodes was studied in [9]. Leveraging the notion of non-strong observability, [10] considered adding perturbations to system inputs and outputs to protect partial entries of the initial states and inputs. The problem of blocking a minimum set of state variables from being measured by existing sensors to destroy observability was considered in [13]. The basic idea in these works to achieve privacy preservation is to design control systems with reduced observability. This means that the system state cannot be easily inferred from its output, making it more difficult for an adversary to infer sensitive information. Notably, it has been recently found that the system observability has a strong connection with differential privacy [11, 12].

In this paper, we take a step further in the direction of protecting network privacy by considering functional privacy, i.e., linear functionals of states that need to be kept confidential to adversarial observers. The goal is to identify the minimum set of state variables (nodes) in a network that need to be blocked from direct measurements to prevent the inference of a given functional privacy with respect to *any* output matrices. We consider two different privacy protection levels: vector-wise protection and entry-wise protection, meaning that the functional privacy in the vector form cannot be inferred as a whole and that every component (entry) of it cannot be inferred, respectively.

By relating functional privacy protection to the notion of functional observability and leveraging a PBH-like criterion for functional observability [14–17], we make the following contributions. First, we prove that both functional privacy protection problems are NP-hard. Second, assuming a reasonable constant bound on the geometric multiplicities of the system's eigenvalues, we provide an exact algorithm with polynomial time complexity for the vector-wise problem. Third, we provide a greedy algorithm for the entry-wise problem. Our results reveal the role of node measurements in protecting the functional privacy of linear dynamic networks and enable us to identify which set of nodes can be protected at a lower cost to preserve functional privacy more efficiently. These nodes can be regarded as "critical nodes" that may leak confidential information and require specific protection measures.

The rest is organized as follows. Section II presents the problem formulation, and Section III provides preliminaries on functional observability. The complexity of the considered problems is given in the next section. Section V presents algorithms for these problems, followed by an illustrative

This work was supported in part by the National Natural Science Foundation of China under Grants 62373059 and 62003042. The authors are with the School of Automation, Beijing Institute of Technology, Beijing, China. (email: zhangyuan14@bit.edu.cn, chengranbo123@163.com, xia_yuanqing@bit.edu.cn).

example in Section VI. The last section concludes this paper.

Notations: For a set, $|\cdot|$ denotes its cardinality. The symbol $[n] = \{1, 2, ..., n\}$. A matrix L is also denoted by $L = [l_{ij}]$ or $L = [L_{ij}]$, which means l_{ij} or L_{ij} is the entry in the ith row and jth column of L. By $\mathrm{eig}(M)$ we denote the set of eigenvalues of the square matrix M. Let $\mathbf{diag}\{X_i|_{i=1}^n\}$ be the block diagonal matrix whose ith diagonal block is X_i, and $\mathbf{col}\{X_i|_{i=1}^n\}$ be the matrix stacked by $X_i|_{i=1}^n$. I_n denotes the n dimensional identify matrix, where the subscript n may be omitted if it can be inferred from the context. By e_i we denote the ith column of I_n, and $\mathbf{1}_{m\times n}$ the $m \times n$ matrix with entries all being one. Given an $m \times n$ matrix M and a set $\mathcal{S} \subseteq [n]$, $M_{\mathcal{S}}$ denotes the sub-matrix of M formed by *rows* indexed by $\mathcal{S}$, and $M^{\mathcal{S}}$ denotes the matrix obtained from M by preserving its columns indexed with $\mathcal{S}$ and zeroing the rest. If $M = I$, $M_{\mathcal{S}}$ and $M^{\mathcal{S}}$ coincide.

II. Problem Formulation

Consider a network of n nodes. The ith node evolves according to the following dynamics

$$\dot{x}_i(t) = a_{ii}x_i(t) + \sum_{j\in[n]\setminus\{i\}} a_{ij}x_j(t) + \sum_{j=1}^{m} b_{ij}u_j(t), \quad (1)$$

where $x_i(t) \in \mathbb{R}$ is the state variable of the ith node, $u_j(t) \in \mathbb{R}$ is the jth input, $a_{ii} \in \mathbb{R}$ is the self-damping coefficient, $a_{ij} \in \mathbb{R}$ is the coupling strength from node j to node i, and $b_{ij} \in \mathbb{R}$ stands for the affection from the jth input to the ith node. The topology of this network can be described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., n\}$ denotes the node set and $\mathcal{E} = \{(i, j) : a_{ji} \neq 0\}$ denotes the edge set. Let $x(t) = [x_1(t), \cdots, x_n(t)]^\intercal$, $u(t) = [u_1(t), \cdots, u_m(t)]^\intercal$, $A = [a_{ij}]$ and $B = [b_{ij}]$. The lumped form of (1) can be written as

$$\dot{x}(t) = Ax(t) + Bu(t). \quad (2)$$

System (1) can describe the dynamics of a linear time-invariant network system. A typical example is the multi-agent system where each agent is a single integrator running the consensus protocol [18]. In this case, $a_{ii} = -\sum_{j\in[n]\setminus\{i\}} a_{ij}$, meaning that $-A$ is the weighted Laplacian matrix of $\mathcal{G}$. Other examples include interacted liquid tanks [19], synchronizing networks of linear oscillators [20], opinion propagation in social networks [21], etc.

The output of system (2) is given by

$$y(t) = Cx(t), \quad (3)$$

with $C \in \mathbb{R}^{p\times n}$ the output matrix. Let $y(x, u, t)$ be the output signal of system (2)-(2) at time t generated from the initial state x by the input u. Each nonzero row of C corresponds to a sensor. If there is only one nonzero entry in a row of C, we call the sensor associated with this row a dedicated sensor, meaning that this sensor measures only one state variable.

Suppose an adversarial observer or eavesdropper intends to infer the information

$$z(t) = Fx(t), \quad (4)$$

where $F \in \mathbb{R}^{r\times n}$. Write F as $F = \mathbf{col}\{f_i|_{i=1}^r\}$, $f_i \in \mathbb{R}^{1\times n}$. As commonly assumed, the observer knows system parameters (A, C, F) and has access to the signals $u(t)$ and $y(t)$ for a sufficiently long time horizon [4, 8, 10]. The vector $z(t)$ can be some private or confidential information that the network wants to protect from being inferred by the observer, and is called *functional privacy* since it is a linear combination of the state $x(t)$. Typical examples of $z(t)$ include:

- the full state $x(t)$ when $F = I_n$;
- the vector formed by a set of target states indexed by $\mathcal{S} \subseteq [n]$ when $F = \mathbf{col}\{e_i^\intercal|_{i\in\mathcal{S}}\}$;
- the average of all states when $F = \frac{1}{n}\mathbf{1}_{1\times n}$;
- the vector consisting of averages of states of clusters indexed by $\mathcal{S}_1, ..., \mathcal{S}_l \subseteq [n]$ when $F = \mathbf{col}\{\frac{1}{|\mathcal{S}_i|}\mathbf{1}^{\mathcal{S}_i}|_{i=1}^l\}$.

We remark that designing observers to infer the average state or average cluster states has been considered in [22].

There are typically several ways to protect the functional privacy $z(t)$, for example, adding noise to the output $y(t)$ in the spirit of differential privacy [23], or using certain encryption techniques to encrypt $y(t)$ [6]. In this paper, however, we consider the structure requirement on the output matrix C, under which adversarial observers cannot infer the functional privacy $z(t)$. The advantage of doing so is that by designing an appropriate measurement structure we can preserve the functional privacy *without* using any privacy-preserving techniques.

***Definition** 1 (Vector-wise functional privacy protection):* The functional privacy $z(t) = Fx(t)$ is inferable, if for any initial state $x(0)$ and input $u(t)$, there exists a finite time t_f such that the initial value of the function $Fx(0)$ can be uniquely determined from the observation $y(t)$ and input $u(t)$, $0 \leq t \leq t_f$. The functional privacy $z(t)$ is said to be (vector-wisely) protected if it is not inferable.

***Definition** 2 (Entry-wise functional privacy protection):* The functional privacy $z(t) = Fx(t)$ is said to be entry-wisely protected if every component of $z(t)$, i.e., $z_i(t) = f_ix(t)$, is protected for $i = 1, ..., r$.

***Remark** 1:* The case that $z(t)$ is not inferable does not imply that every component of it is not inferable. There may exist scenarios where $z(t)$ is not inferable but its partial components are. Therefore, the entry-wise protection is stricter than the vector-wise protection. From Definition 1, with the knowledge of (A, C, F) and $u(t), y(t)$, $t \in [0, t_f]$, if an observer can infer $z(0)$, then it can infer $z(t)$ for $t \in [0, t_f]$.

In the network context, the structure of C can be dominantly determined by the state variables (nodes) it directly measures. We say a set of state variables indexed by $\mathcal{S} \subseteq [n]$ is blocked from direct measurement with respect to the output matrix C (blocked w.r.t. C for short) if the columns of C indexed by $\mathcal{S}$ are turned to zeros (i.e., C is turned to $C^{[n]\setminus\mathcal{S}}$). To see how many state variables should be blocked w.r.t. whatever output matrices C to protect the functional privacy, we consider the following two problems:

***Problem** 1:* How can we select the minimum set of state variables to be blocked w.r.t. *any* output matrix C such that $z(t)$ is protected?

***Problem** 2:* How can we select the minimum set of state variables to be blocked w.r.t. *any* output matrix C such that *every* component of $z(t)$ is protected?

A trivial solution to Problems 1 and 2 is the full state set, under which $y(t)$ cannot convey any information of $z(t)$. This

implies these two problems are well-defined. Problems 1 and 2 enable a better understanding of the role of nodal measurements in the preservation of the given functional privacy $z(t)$. Note the problems do not depend on a specific output matrix C, meaning that the solutions are properties of the network system matrix A and the functional privacy characterized by F. The solutions to these problems can somehow tell us the set of nodes that we can protect with less cost to preserve functional privacy more efficiently (since blocking more nodes from direct measurement means that more energy or additional effort is needed). System (2)-(4) may be represented by the pair (A, F) when Problems 1 and 2 are considered.

It is obvious that in Problems 1 and 2, it suffices to consider the case with dedicated sensors, i.e., $C = I_n$. This is because, for any output matrix $C \in \mathbb{R}^{m\times n}$ and a given $\mathcal{S} \subseteq [n]$, even $m > n$, upon defining $\bar{\mathcal{S}} = [n]\backslash\mathcal{S}$, the new output matrix $C^{\bar{\mathcal{S}}}$ after blocking state variables $\mathcal{S}$ means that at most $n - |\mathcal{S}|$ state variables are measured so that we can extract no more than $n - |\mathcal{S}|$ individual states indexed by $\bar{\mathcal{S}}$.

III. Functional observability

In this section, we investigate Problems 1 and 2 in the spirit of functional observability, resulting in reformulations for them.

We shall denote the system described by (2)-(4) as the triple (A, C, F). Before introducing the definition of functional observability, we give an equivalent definition of being inferable as follows.

Lemma 1: The functional privacy $z(t) = Fx(t)$ is inferable (by an adversarial observer), if and only if for any initial states x_1 and x_2 and input u, $y(x_1, u, t) = y(x_2, u, t)$ for all $t \geq 0$ implies that $Fx_1 = Fx_2$.

Proof: The sufficiency is obvious since this condition implies for any output $y(t)$ and input $u(t)$, there is a unique initial function $Fx(0)$ that obeys the system dynamics. For the necessity, suppose there exist two initial states x_1, x_2 and input u such that $y(x_1, u, t) = y(x_2, u, t)$ for all $t \geq 0$ but $Fx_1 \neq Fx_2$. Then, the initial function $Fx(0)$ cannot be determined uniquely. □

Definition 3 (Observability): [24] System (2)-(3) is said to be observable, if for any initial states x_1, x_2 and the zero input, $y(x_1, 0, t) = y(x_2, 0, t)$ for all $t \geq 0$ implies that $x_1 = x_2$.

Definition 4 (Functional observability): [15, 16] System (2)-(4) is said to be functionally observable, if for any initial states x_1, x_2 and input u, $y(x_1, u, t) = y(x_2, u, t)$ for all $t \geq 0$ implies that $Fx_1 = Fx_2$.

In other words, functional observability is the ability to infer linear functions of states $Fx(t)$ from the knowledge of external inputs $u(t)$ and outputs $y(t)$ of a system [14]. When $F = I_n$, functional observability collapses to conventional observability. Based on functional observability and Lemma 1, the functional privacy $z(t) = Fx(t)$ is inferable for a system (A, C, F), if and only if (A, C, F) is functionally observable. In addition, $Fx(t)$ is entry-wisely protected, if and only if (A, C, f_i) is not functionally observable for $i = 1, ..., r$, recalling f_i is the ith row of F. Therefore, Problems 1 and 2 can be equivalently formulated as

Problem 1 :
$\min_{\mathcal{S}\subseteq[n]} \ |\mathcal{S}|$
s.t. $(A, I_{[n]\backslash\mathcal{S}}, F)$ not functionally observable.

Problem 2 :
$\min_{\mathcal{S}\subseteq[n]} \ |\mathcal{S}|$
s.t. $(A, I_{[n]\backslash\mathcal{S}}, f_i)$ not functionally observable, $\forall i \in [r]$.

The following lemma revises [15, Theo. 4] and [16, Theo.2], which gives a necessary and sufficient condition for functional observability under the diagonalization assumption on A.

Lemma 2: [17, Coro. 2] Suppose that A is diagonalizable. The triple (A, C, F) is functionally observable if and only if

$$\operatorname{rank}\begin{bmatrix} A-\lambda I_n \\ C \\ F \end{bmatrix} = \operatorname{rank}\begin{bmatrix} A-\lambda I_n \\ C \end{bmatrix}, \forall \lambda \in \mathbb{C}. \quad (5)$$

It can be seen that when $F = I_n$, the above condition collapses to the PBH test for conventional observability. When (A, C, F) is functionally observable, one can find a matrix F_0 satisfying two additional conditions (see [14, Theo. 2]), based on which a functional observer with arbitrary poles can be constructed to estimate $z(t)$ asymptotically; see [14] for details.

IV. Complexity Analysis

In this section, we prove that both Problems 1 and 2 are NP-hard. When F is a row vector, Problem 1 reduces to Problem 2. This indicates to show the NP-hardness of Problem 1, it suffices to show the NP-hardness of Problem 2 with scalar functional privacy.

Definition 5 (Linear degeneracy problem, [25]): Given an $n \times k$ ($k < n$) matrix W, the linear degeneracy problem is to determine whether there exist k rows of W that are linearly dependent, i.e., whether a set $\mathcal{S} \subseteq [n]$ with $|\mathcal{S}| = k$ exists such that $\det W_{\mathcal{S}} = 0$.

Theorem 1: Both Problems 1 and 2 are NP-hard.

Proof: We shall present a reduction from the linear degeneracy problem to Problem 2. For space limitation, please refer to [26]. □

Remark 2: We remark that the NP-hardness of Problem 1 can also be obtained from [13, Theo. 1], which established the NP-hardness of determining the minimum number of sensors whose removal can destroy system observability with dedicated sensors (i.e., $C = I_n$). This means Problem 1 is NP-hard with $F = I_n$, under which circumstance the functional observability collapses to the conventional observability. However, the technique in [13, Theo. 1] is not sufficient to prove the NP-hardness of Problem 2.

V. Algorithms

In this section, we give exact algorithms for Problem 1 under a reasonable assumption that the eigenvalue geometric multiplicities of A are bounded by a constant. We also present a greedy algorithm for Problem 2.

The following assumption on the computational availability of eigenvalues and eigenvectors of A is adopted.

Assumption 1: Suppose that the eigenvalues and eigenvectors of A are computationally available. Moreover, suppose there are q distinct eigenvalues in $\text{eig}(A)$, the ith one denoted by λ_i, and $X_i \in \mathbb{C}^{n\times k_i}$ consists of the maximum number of linearly independent eigenvectors (i.e., X_i is the eigenbasis) associated with λ_i.

Remark 3: It is worth mentioning that, while eigenvalues and eigenvectors of a matrix can be computed to any prescribed precision in theory, the practical limitations of numerical methods and the conditioning of the matrix may sometimes make it difficult to achieve high precision in practice [27]. On the other hand, the eigenvalues and eigenvectors of adjacency or Laplacian matrices of large-scale sparse graphs have been extensively studied [18], and there are specialized algorithms for computing eigenvalues of sparse matrices more efficiently and accurately than general-purpose algorithms [28]. This makes Assumption 1 reasonable for studying sparsely-connected networks.

Assumption 2: The state matrix A is diagonalizable.

Remark 4: Equivalently, this assumption requires that $\sum_{i=1}^{q} k_i = n$. Diagonalizable matrices are quite common in system modeling and control. For example, all symmetric matrices, naturally arising in Laplacian matrices and adjacency matrices of undirected graphs, are diagonalizable. Moreover, the weighted Laplacian matrices of strongly connected directed graphs and adjacency matrices of random networks are mostly diagonalizable [18, 29].

Lemma 3: [30, Chap. 0.2.7] Let $M = \mathbf{col}\{M_1, M_2\}$ be a composite matrix and $M_1^\perp$ (if exists) consist of a set of linearly independent column vectors spanning the null space of M_1 ($M_1^\perp$ is called a basis matrix). Then, M is of full column rank, if and only if $M_2 M_1^\perp$ is of full column rank.

A. Algorithms for Problem 1

According to Lemma 2, under Assumption 2, a natural idea to find a minimum set (i.e., a set with the minimum cardinality) $\mathcal{S} \subseteq [n]$ such that $(A, I^{[n]\backslash\mathcal{S}}, F)$ fails to be functionally observable is to determine the minimum set $\mathcal{S}_i$ for each eigenvalue $\lambda_i \in \text{eig}(A)$ such that

$$\text{rank}\begin{bmatrix} A-\lambda_i I_n \\ I^{[n]\backslash\mathcal{S}_i} \\ F \end{bmatrix} > \text{rank}\begin{bmatrix} A-\lambda_i I_n \\ I^{[n]\backslash\mathcal{S}_i} \end{bmatrix}, \tag{6}$$

and then find the minimum $|\mathcal{S}_i|$ over $i \in [q]$. In the following, we characterize the minimum set $\mathcal{S}_i$ that satisfies (6).

Proposition 1: Let $\mathcal{S}_i^*$ be a set with the minimum cardinality that satisfies (6). Then, it holds that

$$\text{rank}\begin{bmatrix} A-\lambda_i I_n \\ I^{[n]\backslash\mathcal{S}_i^*} \\ F \end{bmatrix} = n, \tag{7}$$

$$\text{rank}\begin{bmatrix} A-\lambda_i I_n \\ I^{[n]\backslash\mathcal{S}_i^*} \end{bmatrix} = n-1. \tag{8}$$

Moreover, $\mathcal{S}_i^*$ (if exists) must be a *minimal set* that satisfies (8) (by 'minimal set' we mean $\mathcal{S}_i^*$ satisfies (8), but any proper subset $\bar{\mathcal{S}}_i^* \subseteq \mathcal{S}_i^*$ cannot satisfy (8)).

Proof: The proof can be found in [26]. □

By Proposition 1, to determine $\mathcal{S}_i^*$ for each $\lambda_i \in \text{eig}(A)$, one can determine all minimal sets that satisfy (8), and then find out those that satisfy (7). Then, $\mathcal{S}_i^*$ is such a set with the minimum cardinality. Note that $\mathcal{S}_i^*$ may be empty for some $\lambda_i \in \text{eig}(A)$. But with Assumption 2, there is at least one $\lambda_i \in \text{eig}(A)$ such that $\mathcal{S}_i^*$ is not empty. Indeed, the worst-case solution to Problem 1 is the full state set $[n]$. After determining $\mathcal{S}_i^*$ for each $\lambda_i \in \text{eig}(A)$, the optimal solution is the set $\mathcal{S}_i^*$ with the minimum cardinality over $i \in [q]$.

1) Simple dynamic case: We first consider the case where A has no repeated eigenvalues, i.e., the simple dynamic case. In this situation, $k_i = 1\ \forall i$ and $q = n$. If a set $\mathcal{S}_i$ satisfies (8), by Lemma 3, it holds that

$$[X_i]_{[n]\backslash\mathcal{S}_i} = 0.$$

Hence, the minimal set satisfying (8) is unique, which is the support of X_i, given by

$$\bar{\mathcal{S}}_i^* = \text{supp}X_i \doteq \{j \in [n] : [X_i]_j \neq 0\}.$$

Let $\mathcal{S}_i^*$ be such $\bar{\mathcal{S}}_i^*$ that satisfies (7). For ease of description, if $\bar{\mathcal{S}}_i^*$ does not satisfy (7), we assign $\mathcal{S}_i^* = [n]$. As a result, the optimal solution to Problem 1 (denoted by $\mathcal{S}_{P_1}^*$) is

$$\mathcal{S}_{P_1}^* = \arg_{i\in[n]} \min |\mathcal{S}_i^*|.$$

2) Bounded eigenvalue geometric multiplicity case: We now generalize the simple dynamic case to systems with bounded eigenvalue geometric multiplicities. More precisely, we consider systems satisfying the following assumption:

Assumption 3: The geometric multiplicities of eigenvalues of A are bounded by some fixed constant k_c, i.e., $k_i \leq k_c$ $\forall i \in [q]$ as n increases.

The above assumption can be satisfied by most practical systems. The simple dynamic case is one such with $k_c = 1$. Besides, when modeling networks of coupled oscillators, power grids, diffusively couple networks, epidemiological networks using graphs, a common setting is that each node has a self-loop [18–21], under which these networks can be controllable using some constant number of inputs regardless of the network size [31]. This indicates the above assumption is satisfied for these networks, since the minimum number of inputs for achieving controllability equals the maximum eigenvalue geometric multiplicities of system state matrices [32].

From the previous analysis, the key step for solving Problem 1 is to determine the collection Ω_i of all minimal sets that satisfy (8) for a given $\lambda_i \in \text{eig}(A)$. We provide Algorithm 1 for this purpose. In this algorithm, $\mathcal{W}_j'$ is a *maximal* set satisfying $\text{rank}[X_i]_{\mathcal{W}_j'} = k_i - 1$, meaning that adding additional rows to $[X_i]_{\mathcal{W}_j'}$ will increase its rank. Therefore, by Lemma 3, $[n]\backslash\mathcal{W}_j'$ is a minimal set satisfying (8). With Assumption 3, $|\Omega_i| \leq \binom{n}{k_i} \leq n^{k_c}$. After the determination of Ω_i for $i \in [q]$, the rest is similar to the simple dynamic case. We collect the whole procedure in Algorithm 2, and state the following result.

Theorem 2: Under Assumptions 1-3, Algorithm 2 is able to find an optimal solution to Problem 1 in time $O(n^{k_c+2}k_c^3)$.

Proof: The reason why Algorithm 2 returns an optimal solution has been explained in Proposition 1 and the main contexts. Here we just need to justify the computational

complexity. Step 1 incurs $O(n^3)$ time [27]. For each eigenbasis X_i, checking whether its k_i-1 rows are linearly independent can use the singular value decomposition (SVD), which takes $O(k_i^3)$ time [27]. For each $\mathcal{W}_j$ in Algorithm 1, it takes $O(nk_i^3)$ time to get $\mathcal{W}_j'$ via SVD. For each member $\mathcal{S}_{ij}\in\Omega_i$, it takes $O(k_i^2(n+r))$ time to get $\bar{\mathcal{S}}_{ij}$. Since $|\Omega_i|\leq n^{k_i}$, determining $\mathcal{S}_i^*$ incurs time $O(n^{k_i+1}k_i^3)$. As $q\leq n$ and $k_i\leq k_c$, the total time complexity is at most $O(n^{k_c+2}k_c^3)$. □

***Remark** 5:* Theorem 2 makes it clear that the computation cost of Algorithm 2 scales exponentially with k_c. This indicates it is the eigenvalue geometric multiplicities of A that cause the computational intractability of Problem 1.

Algorithm 1 : Enumerating all minimal sets satisfying (8)

Input: The eigenbasis X_i of $\lambda_i\in\mathrm{eig}(A)$
Output: The collection Ω_i of the minimal sets satisfying (8)
1: Determine all sets $\{\mathcal{W}_j\}_{j=1}^{N_1}$ that contain k_i-1 linearly independent rows of X_i (N_1 is the number of such sets).
2: **for** $j=1$ to N_1 **do**
3: $\quad\mathcal{W}_j'=\mathcal{W}_j\cup\{k:\mathrm{rank}[X_i]_{\mathcal{W}_j\cup\{k\}}=k_i-1\}$
4: $\quad\mathcal{S}_{ij}=[n]\backslash\mathcal{W}_j'$
5: **end for**
6: Return $\Omega_i=\{\mathcal{S}_{ij}\}_{j=1}^{N_1}$.

Algorithm 2 : Algorithm for Problem 1

Input: Parameters (A,F) satisfying Assumption 1-3
Output: The optimal solution $\mathcal{S}_{P_1}^*$ to Problem 1
1: Calculate the eigenbases $\{X_i|_{i=1}^q\}$ of A.
2: **for** $i=1$ to q **do**
3: $\quad$Determine the collection Ω_i of the minimal sets satisfying (8) using Algorithm 1.
4: $\quad$For each member $\mathcal{S}_{ij}\in\Omega_i$, check whether it satisfies (7), if yes, let $\bar{\mathcal{S}}_{ij}=\mathcal{S}_{ij}$; otherwise, let $\bar{\mathcal{S}}_{ij}=[n]$. Find $\mathcal{S}_i^*=\arg\min|\bar{\mathcal{S}}_{ij}|$.
5: **end for**
6: Return $\mathcal{S}_{P_1}^*=\arg_{i\in[q]}\min|\mathcal{S}_i^*|$.

B. Greedy algorithm for Problem 2

By the definition of functional observability, if we obtain the optimal solution $\mathcal{S}_{P_1i}^*$ to Problem 1 associated with (A,f_i) for each row f_i of F, then $\bigcup_{i=1}^r\mathcal{S}_{P_1i}^*$ is a feasible solution to Problem 2. However, such a solution ignores the possible overlaps among different $f_i's$, which may be far from the optimal one. In what follows, we provide a greedy algorithm for Problem 2. This algorithm is based on the following result, which generalizes Proposition 1.

***Proposition** 2:* With Assumption 2, suppose for one $\lambda_i\in\mathrm{eig}(A)$, some $\mathcal{T}_k\subseteq[n]$, and the jth row f_j of F, it holds

$$\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k}\\ f_j\end{bmatrix}=\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k}\end{bmatrix}=n_{ij}\leq n. \quad (9)$$

If $\Delta_{kj}^*\subseteq\mathcal{T}_k$ is a set with the minimum cardinality satisfying

$$\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k\backslash\Delta_{kj}^*}\\ f_j\end{bmatrix}>\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k\backslash\Delta_{kj}^*}\end{bmatrix}, \quad (10)$$

it must hold that

$$\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k\backslash\Delta_{kj}^*}\\ f_j\end{bmatrix}=n_{ij},\mathrm{rank}\begin{bmatrix}A-\lambda_iI_n\\ I^{\mathcal{T}_k\backslash\Delta_{kj}^*}\end{bmatrix}=n_{ij}-1, \quad (11)$$

and Δ_{kj}^* is a minimal set satisfying the second equality of (11).

Proof: The proof is similar to that of Proposition 1, thus omitted. □

It is clear that if $\mathcal{T}_k=[n]$, $n_{ij}=n$, then Proposition 2 collapses to Proposition 1. As a result, when (9) holds for every $\lambda_i\in\mathrm{eig}(A)$, we can use a similar manner to Algorithm 2 to find a set Δ_{kj}^* with the minimum cardinality such that (10) is true for some $\lambda_i\in\mathrm{eig}(A)$. Denote such a process by $\Delta_{kj}^*\leftarrow\mathrm{Alg2}[A,f_j,\mathcal{T}_k]$. That is, $\mathrm{Alg2}[A,f_j,\mathcal{T}_k]$ finds the minimum set Δ_{kj}^* from $\mathcal{T}_k$ such that $(A,I^{\mathcal{T}_k\backslash\Delta_{kj}^*},f_j)$ becomes functionally unobservable (Δ_{kj}^* will be empty if $(A,I^{\mathcal{T}_k},f_j)$ is already functionally unobservable). We formulate the greedy algorithm for Problem 2 as Algorithm 3.

Algorithm 3 : Greedy Algorithm for Problem 2

Input: Parameters (A,F) satisfying Assumption 1-3
1: Calculate the eigenbases $\{X_i|_{i=1}^q\}$ of A.
2: Initialize $\mathcal{T}_0=[n],\mathcal{F}=[r],k=0$
3: **while** $|\mathcal{T}_k|>0$ and $|\mathcal{F}|>0$ **do**
4: $\quad$**for** $j\in\mathcal{F}$ **do**
5: $\quad\quad\Delta_{kj}^*\leftarrow\mathrm{Alg2}[A,f_j,\mathcal{T}_k]$.
6: $\quad$**end for**
7: $\quad\Delta_{kj^*}^*\leftarrow\arg\min_{j\in\mathcal{F}}|\Delta_{kj}^*|$
8: $\quad$Update $\mathcal{T}_{k+1}\leftarrow\mathcal{T}_k\backslash\Delta_{kj^*}^*$, $\mathcal{F}\leftarrow\mathcal{F}\backslash\{j^*\}$, and $k\leftarrow k+1$.
9: **end while**
10: Return a solution $[n]\backslash\mathcal{T}_k$

Compared to the naive method mentioned at the beginning of this subsection, the advantage of Algorithm 3 lies in that, it not only guarantees to protect one scalar functional privacy $f_ix(t)$ per step but also accounts for the relations between different f_i's. The computation time of Algorithm 3 is dominated by the subroutine $\mathrm{Alg2}[A,f_j,\mathcal{T}_k]$, which runs at most r^2 times. Therefore, the time complexity of Algorithm 3 is $O(r^2n^{k_c+2}k_c^3)$. When k_c is large, this is a huge computational burden.

***Remark** 6:* Based on Proposition 2, both Algorithms 2 and 3 can be trivially extended to the case where the adversarial observers or eavesdroppers have access only to a restricted set of full states. In this case, we just need to change the full state set $[n]$ to the aforementioned restricted set ($\mathcal{T}_k$ alike).

VI. Illustrative Example

Consider a network system with

$$A=\begin{bmatrix}1&0&0&0&0&0\\3&5&2&0&0&0\\4&0&4&0&0&0\\2&0&0&2&0&0\\0&2&1&3&6&0\\0&0&0&5&4&9\end{bmatrix},$$

whose topology is given in Fig. 1. It follows that $\mathrm{eig}(A)=\{1,5,4,2,6,9\}$. The corresponding eigenvectors have supports respectively as $\bar{\mathcal{S}}_1^*=\{1,\cdots,6\}$, $\bar{\mathcal{S}}_2^*=\{2,5,6\}$, $\bar{\mathcal{S}}_3^*=\{2,3,5,6\}$, $\bar{\mathcal{S}}_4^*=\{4,5,6\}$, $\bar{\mathcal{S}}_5^*=\{5,6\}$, and $\bar{\mathcal{S}}_6^*=\{6\}$.

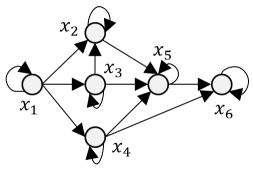

Fig. 1. Network topology of the system in Section VI.

First, let us consider Problem 1 with $F_1 = I_6$. Algorithm 2 returns a solution $\{6\}$. This means blocking state x_6 only is enough to protect the full state vector-wisely. Now, let $F_2 = [0, 1, 1, 1, 0, 0]/3$, meaning that the functional privacy is the average state of cluster $\{x_2, x_3, x_4\}$. Two solutions, $\{2, 5, 6\}$ or $\{4, 5, 6\}$, are found via Algorithm 2. It can be validated that both are optimal.

Next, consider $F_3 = \mathbf{col}\{e_3^\intercal, e_4^\intercal, e_5^\intercal\}$ ($n = 6$), i.e., protecting states $\{x_3, x_4, x_5\}$. Using Algorithm 2, we get an optimal solution with 2 states $\{5, 6\}$, implying that at least two states need to be blocked. Comparing these solutions, it turns out that although blocking state x_6 is enough to protect the full state vector-wisely, it cannot protect states $\{x_3, x_4, x_5\}$. Finally, suppose we are to protect state variables $\{x_3, x_4, x_5\}$ entry-wisely. Implementing Algorithm 3 on (A, F_3), we get $\mathcal{T}_1 = \{1, 2, 3, 4\}$ and $\mathcal{F}_1 = \{3, 4\}$, $\mathcal{T}_2 = \{1, 2, 3\}$ and $\mathcal{F}_2 = \{3\}$, and $\mathcal{T}_3 = \{1\}$ and $\mathcal{F}_3 = \emptyset$, which means the solution is $\{2, 3, 4, 5, 6\}$. By exhaustive search, it can be validated that this solution is optimal.

This above example shows that the minimum set of states needed to be blocked varies drastically with the functional privacy to be protected; and even for the same functional privacy, the vector-wise protection and entry-wise protection can lead to drastically different solutions.

VII. Conclusion

This paper addressed and investigated the problem of protecting functional privacy in a network by blocking the minimum set of state variables from direct measurements. We have related the considered functional privacy to the concept of functional observability. By leveraging a PBH-like criterion for functional observability, we have proven that both the vector-wise and entry-wise functional privacy protection problems are NP-hard, but presented an exact algorithm with polynomial time complexity for the vector-wise problem by assuming a reasonable constant bound on the system eigenvalue geometric multiplicities. A greedy algorithm for the entry-wise problem is further provided. The effectiveness of the proposed approach is demonstrated through an example. In the future, we plan to extend our study to a structured system model [17].

References

[1] S. Han, U. Topcu, G. J. Pappas, Differentially private distributed constrained optimization, IEEE Transactions on Automatic Control 62 (1) (2016) 50–64.

[2] M. S. Darup, A. B. Alexandru, D. E. Quevedo, G. J. Pappas, Encrypted control for networked systems: An illustrative introduction and current challenges, IEEE Control Systems Magazine 41 (3) (2021) 58–78.

[3] Y. Xia, Y. Zhang, L. Dai, Y. Zhan, Z. Guo, A brief survey on recent advances in cloud control systems, IEEE Transactions on Circuits and Systems II: Express Briefs 69 (7) (2022) 3108–3114.

[4] Y. Mo, R. M. Murray, Privacy preserving average consensus, IEEE Transactions on Automatic Control 62 (2) (2016) 753–765.

[5] Y. Wang, Privacy-preserving average consensus via state decomposition, IEEE Transactions on Automatic Control 64 (11) (2019) 4711–4716.

[6] Y. Lu, M. Zhu, Privacy preserving distributed optimization using homomorphic encryption, Automatica 96 (2018) 314–325.

[7] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, I. Khalil, An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems, IEEE Transactions on Sustainable Computing 6 (1) (2019) 66–79.

[8] S. Pequito, S. Kar, S. Sundaram, A. P. Aguiar, Design of communication networks for distributed computation with privacy guarantees, in: 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 1370–1376.

[9] A. Al Maruf, S. Roy, Observability-blocking controllers for network synchronization processes, in: 2019 American Control Conference (ACC), IEEE, 2019, pp. 2066–2071.

[10] Y. Lu, M. Zhu, On privacy preserving data release of linear dynamic networks, Automatica 115 (2020) 108839.

[11] Y. Kawano, M. Cao, Design of privacy-preserving dynamic controllers, IEEE Transactions on Automatic Control 65 (9) (2020) 3863–3878.

[12] L. Wang, I. R. Manchester, J. Trumpf, G. Shi, Differential initial-value privacy and observability of linear dynamical systems, Automatica 148 (2023) 110722.

[13] Y. Zhang, Y. Xia, K. Liu, Observability robustness under sensor failures: A computational perspective, IEEE Transactions on Automatic Control, DOI: 10.1109/TAC.2023.3295698, 2023.

[14] T. L. Fernando, H. M. Trinh, L. Jennings, Functional observability and the design of minimum order linear functional observers, IEEE Transactions on Automatic Control 55 (5) (2010) 1268–1273.

[15] L. S. Jennings, T. L. Fernando, H. M. Trinh, Existence conditions for functional observability from an eigenspace perspective, IEEE transactions on automatic control 56 (12) (2011) 2957–2961.

[16] T. Fernando, L. Jennings, H. Trinh, Functional observability, in: 2010 Fifth International Conference on Information and Automation for Sustainability, IEEE, 2010, pp. 421–423.

[17] Y. Zhang, T. Fernando, M. Darouach, Functional observability, structural functional observability and optimal sensor placement, arXiv preprint arXiv:2307.08923(2023).

[18] M. Mesbahi, M. Egerstedt, Graph Theoretic Methods in Multiagent Networks, Vol. 33, Princeton University Press, 2010.

[19] Y. Zhang, Y. Xia, D.-H. Zhai, Structural controllability of networked relative coupling systems, Automatica 128 (2021) 109547.

[20] L. Scardovi, R. Sepulchre, Synchronization in networks of identical linear systems, Automatica 45 (11) (2009) 2557–2562.

[21] M. Ye, J. Liu, B. D. Anderson, C. Yu, T. Başar, Evolution of social power in social networks with dynamic topology, IEEE transactions on automatic control 63 (11) (2018) 3793–3808.

[22] M. U. B. Niazi, C. Canudas-de Wit, A. Y. Kibangou, Average state estimation in large-scale clustered network systems, IEEE Transactions on Control of Network Systems 7 (4) (2020) 1736–1745.

[23] J. Le Ny, G. J. Pappas, Differentially private filtering, IEEE Transactions on Automatic Control 59 (2) (2013) 341–354.

[24] H. Trentelman, A. A. Stoorvogel, H. Hautus, Control Theory for Linear Systems, Springer Science and Business Media, 2012.

[25] L. Khachiyan, On the complexity of approximating extremal determinants in matrices, Journal of Complexity 11 (1) (1995) 138–153.

[26] Y. Zhang, R. Cheng, Y. Xia, Observability blocking for functional privacy of linear dynamic networks, arXiv preprint arXiv:2304.07928 (2023).

[27] G. H. Golub, C. F. Van Loan, Matrix Computations, JHU press, 2013.

[28] Y. Saad, Numerical Methods for Large Eigenvalue Problems: Revised Edition, SIAM, 2011.

[29] T. Tao, V. Vu, Random matrices have simple spectrum, Combinatorica 37 (3) (2017) 539–553.

[30] R. A. Horn, C. R. Johnson, Matrix Analysis, Cambridge University Press, 2013.

[31] N. J. Cowan, E. J. Chastain, D. A. Vilhena, et al, Nodal dynamics, not degree distributions, determine the structural controllability of complex networks, PloS One 7 (6).

[32] Y. Zhang, T. Zhou, Input matrix construction and approximation using a graphic approach, International Journal of Control 93 (7) (2020) 1577–1590.