

# Privacy Guarantees for Cloud-based State Estimation using Partially Homomorphic Encryption

Sawsan Emad<sup>1</sup>, Amr Alanwar<sup>2</sup>, Yousra Alkabani<sup>3</sup>, M. Watheq El-Kharashi<sup>1</sup>,  
Henrik Sandberg<sup>4</sup>, and Karl Henrik Johansson<sup>4</sup>

**Abstract**—The privacy aspect of state estimation algorithms has been drawing high research attention due to the necessity for a trustworthy private environment in cyber-physical systems. These systems usually engage cloud-computing platforms to aggregate essential information from spatially distributed nodes and produce desired estimates. The exchange of sensitive data among semi-honest parties raises privacy concerns, especially when there are coalitions between parties. We propose two privacy-preserving protocols using Kalman filter and partially homomorphic encryption of the measurements and estimates while exposing the covariances and other model parameters. We prove that the proposed protocols achieve satisfying computational privacy guarantees against various coalitions based on formal cryptographic definitions of indistinguishability. We evaluate the proposed protocols to demonstrate their efficiency using data from a real testbed.

**Index Terms**—Kalman filter, estimation, computational privacy.

## I. INTRODUCTION

Cyber-physical systems (CPSs) have emerged as the new paradigm for the modern global technology industry, representing a highly interactive generation of intelligent systems with tight integration between computer resources and physical processes [1]. Some states of these systems aren't directly perceptible by sensors; sensors may be unable to sense data from the area of interest or can only sense physical variables relevant to the variables of interest, or measurements may be inaccurate or subject to noises [2]. To maintain robustness against measurement noise and modeling uncertainty, optimal state estimation algorithms that implement multisensor data fusion [3] are employed to find the best estimates for hidden states with minimal estimation error [4].

Typically, the estimator (aggregator) aggregates essential information from spatially distributed sensors, applies an estimation algorithm to produce the required estimates, and then sends them to the interested party who initiated the inquiry. Thus, estimators are usually outsourced to cloud-computing platforms like in [5], [6] and can also be centralized or distributed among multiple nodes [7], [8]. Kalman

filters [9] are widely-used optimal estimation algorithms that can fuse measurements and estimates [10], [11] within centralized or distributed implementations [12], [13] and provide accurate and precise estimates of hidden states considering process and measurement uncertainties.

Because cloud-based estimations use open computation and communication architectures, they might suffer from adversarial physical faults or cyber-attacks. Therefore, researchers proposed several approaches to perform computations on sensitive data while keeping the data confidential from untrustworthy parties, such as differential privacy [14], [15], obfuscation [16], [17], algebraic transformation [18], [19] and homomorphic encryption [20], [21]. Paillier encryption, which is partially homomorphic encryption (PHE), was employed with several estimation algorithms to preserve data privacy as in [5], [6], [22]. Kalman filters can operate in an encrypted domain while retaining their natural effectiveness. A secure state estimation using Kalman filter with the adoption of a hybrid homomorphic encryption scheme was proposed in [23]. Authors in [24] presented a multi-party dynamic state estimation using the Kalman filter and PHE, while [25] introduced a secure distributed Kalman filter using PHE. However, no work to date has provided a computational privacy investigation for estimation algorithms that use Kalman filters along with PHE, considering that other problems have undergone similar computational privacy analysis, such as set-based estimation in [5] and quadratic optimization in [21].

We focus on the privacy of multi-party cloud-based state estimation of a linear discrete time-invariant (LTI) system where the involved parties communicate over end-to-end encrypted networks. We consider semi-honest parties that follow protocols properly but keep a record of all their intermediate computations and may collude with other parties to reveal private information of non-colluding ones. In short, we make the following contributions:

- We propose two privacy-preserving estimation protocols using Kalman filters and Paillier cryptosystem by encrypting the measurements and estimates while revealing their covariances and model parameters.
- We provide computational privacy guarantees for the proposed protocols against various coalitions of semi-honest parties using formal cryptographic definitions of computational indistinguishability.

The remainder of this paper is organized as follows. We demonstrate two problem setups in Section II and follow

<sup>1</sup>The authors are with Computer and Systems Department, Ain Shams University. {sawsan.emad, watheq.elkharashi}@eng.asu.edu.eg. <sup>2</sup>The author is with Jacobs University, Bremen. a.alanwar@jacobs-university.de. <sup>3</sup>The author is with Halmstad University. yousra.alkabani@hh.se. <sup>4</sup>The authors are with KTH Royal Institute of Technology. {hsan, kallej}@kth.se.

This work was supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation, the Democritus project on Decision-making in Critical Societal Infrastructures by Digital Futures, and the European Unions Horizon 2020 Research and Innovation program under the CONCORDIA cyber security project (GA No. 830927).

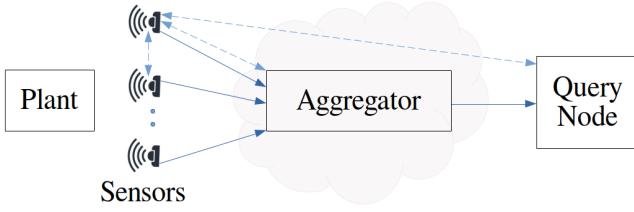


Fig. 1. Problem 1 setup where the bold arrows represent information communication and the dashed arrows represent coalitions.

them with privacy definitions and preliminaries in Section III. We propose two privacy-preserving protocols and summarize their privacy guarantees in Sections IV and V. Then, we discuss the protocols' privacy guarantees in Section VI. Finally, we evaluate the proposed protocols in Section VII and conclude this paper in Section VIII.

## II. PROBLEM SETUP

We consider two common problem setups similar to [5]. The first setup is in Fig. 1, and it involves:

- **Plant  $T$** : A passive entity whose states need to be estimated. We consider the state estimation of a plant modeled as a linear discrete time-invariant (LTI) dynamic system whose state-space model of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{n}_k, \quad (1)$$

$$\mathbf{y}_{i,k} = \mathbf{H}_i\mathbf{x}_k + \mathbf{v}_{i,k}, \quad (2)$$

where  $\mathbf{x}_k \in \mathbb{R}^n$  is the system state at time step  $k \in \mathbb{N}$ ,  $\mathbf{y}_{i,k} \in \mathbb{R}^p$  the measurements of sensor  $i \in 1, \dots, I$ ,  $\mathbf{F} \in \mathbb{R}^{n \times n}$  the process matrix,  $\mathbf{H} \in \mathbb{R}^{p \times n}$  the measurement matrix,  $\mathbf{n}_k \in \mathbb{R}^n$  the modeling noise and  $\mathbf{v}_{i,k} \in \mathbb{R}^p$  the measurement noise and both are independent zero-mean Gaussian white noises with covariances  $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$  and  $\mathbf{R}_{i,k} \in \mathbb{R}^{p \times p}$  respectively.

- **Sensor  $S_i$** : An entity with index  $i$  that provides measurements containing sensitive information that should not be revealed to other parties.
- **Aggregator  $A$  (or Cloud)**: An untrusted party has reasonable computational power that is needed to implement the estimation protocols. It collects encrypted data synchronously from other parties and operates in an encrypted domain to provide the query node with encrypted estimates of the plant  $T$  states.
- **Query Node  $Q$** : An untrusted party inquires about private states of plant  $T$ , which no other party has the right to know. Besides, it owns the encryption keys and shares the public key  $pk$  with others while keeping the private key  $sk$  hidden. The query node can be any entity other than the aggregator  $A$ , including the plant  $T$ .

Briefly, we seek to solve the following first problem:

**Problem 1.** *How to ensure privacy is preserved while estimating the plant  $T$  states by a remote aggregator  $A$  using measurements of spatially distributed sensors? It is required to ensure that measurements are private to the sensor nodes  $S_1, \dots, S_I$  and the estimated states are private to the query node  $Q$ , and to guarantee computational security during the estimation process as well.*

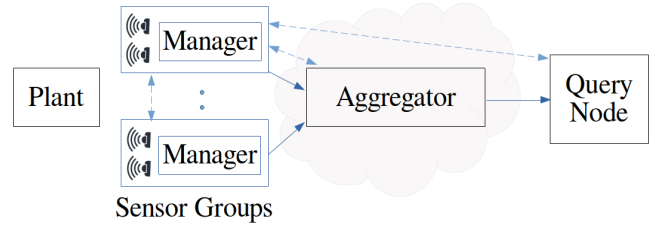


Fig. 2. Problem 2 setup where the bold arrows represent information communication and the dashed arrows represent coalitions.

The second problem setup is in Fig. 2, which includes the following entities in addition to the predefined entities:

- **Manager  $M_j$** : An entity with index  $j$  produces local estimates of plant  $T$  states using synchronously collected measurements from sensors within the group and handles communication with entities outside the group.
- **Sensor Group  $G_j$** : An entity with index  $j$  includes one manager  $M_j$  and  $I_{g_j}$  sensors, all owned by one organization. All group members trust each other while each group aims to keep its measurements and estimates private from other groups/parties.

The problem statement of the second setup is:

**Problem 2.** *How to ensure privacy is preserved in a multi-party cloud-based estimation? It is required to ensure that measurements and local estimates are private to sensor groups  $G_1, \dots, G_J$  and that global estimates are private to query node  $Q$  and sensor groups, and to guarantee computational security during the estimation process as well.*

The global estimate is computed by the aggregator node using local estimates provided by the sensor groups. We should note that Problem 1 is not a special case of Problem 2 because of the existence of group managers [5]. In sections IV and V, we present two protocols to provide private solutions for the above problems, and the purpose of this paper is to ensure that the proposed protocols preserve privacy against the following coalitions [5]:

**Definition 1 (Sensor coalition).** *A number of sensors/sensor groups  $t$  collude together by exchanging their private measurements to infer private information of the query node and non-colluding sensors/sensor groups.*

**Definition 2 (Cloud coalition).** *Aggregator  $A$  colludes with up to  $t$  sensors/sensor groups by exchanging private information and intermediate results to infer private information of the query node and non-colluding sensors/sensor groups.*

**Definition 3 (Query coalition).** *Query node  $Q$  colludes with up to  $t$  sensors/sensor groups by exchanging their private information, private cryptographic keys and the decrypted results to infer private information of non-colluding sensors/sensor groups.*

These coalitions are visualized by dashed lines in Fig. 1 and Fig. 2. We assume that only one type of the above coalitions can occur at a time, and there is always at least one sensor/sensor group that does not participate in the coalition.

In the next section, we outline the privacy goals we rely on to demonstrate that our protocols preserve privacy against the above coalitions.

### III. PRIVACY GOALS AND PRELIMINARIES

We start by defining our privacy goals.

#### A. Privacy Goals

To preserve data privacy, our protocols must guarantee computational security against predefined coalitions. In other words, privacy is preserved if all that a coalition of parties can obtain from keeping records of the intermediate computations can essentially be obtained from these parties' inputs and outputs only [26, p. 620]. Moreover, the coalition's inputs and outputs and recorded information cannot be exploited to infer further private information [5]. These privacy goals are based on the following definitions.

Let  $\{0, 1\}^*$  be a sequence of bits with indefinite length. Then, an ensemble  $X = \{X_n\}_{n \in \mathbb{N}}$  is a sequence of random variables  $X_n$  that ranges over strings of bits with a length polynomial in  $n$ .

**Definition 4.** ([27, p.105]) (**Computationally Indistinguishable**) *The ensembles  $X = \{X_n\}_{n \in \mathbb{N}}$  and  $Y = \{Y_n\}_{n \in \mathbb{N}}$  are computationally indistinguishable, denoted as  $X \stackrel{c}{\equiv} Y$ , if for every probabilistic polynomial-time algorithm  $D$ , each positive polynomial  $p(\cdot)$  with all sufficiently large  $n$ 's, it follows*

$$|\Pr[D(X_n) = 1] - \Pr[D(Y_n) = 1]| < \frac{1}{p(n)}. \quad (3)$$

**Definition 5.** ([26, p.620]) (**Execution View**) *Let  $f(\bar{x}) = (f_1(\bar{x}), \dots, f_n(\bar{x}))$  be a deterministic polynomial-time function and  $\Pi$  a multi-party protocol that computes  $f(\bar{x})$  with the input  $\bar{x} = (x_1, \dots, x_n)$ . The view of the  $i^{\text{th}}$  party during an execution of  $\Pi$  using  $\bar{x}$ , is defined as*

$$V_i^\Pi(\bar{x}) = (x_i, \text{coins}, M_i), \quad (4)$$

where *coins* are the outcome of the party's internal coin toss, and  $M_i$  is the set of messages it has received. For coalition  $I = i_1, \dots, i_t \subseteq \{1, \dots, n\}$  of parties, the coalition view  $V_I^\Pi(\bar{x})$  during an execution of  $\Pi$  is defined as [26, p.696]

$$V_I^\Pi(\bar{x}) = (V_{i_1}^\Pi(\bar{x}), \dots, V_{i_t}^\Pi(\bar{x})). \quad (5)$$

**Definition 6.** ([5]) (**Multi-Party Privacy w.r.t. Semi-Honest Behavior**) *Considering the coalition of parties  $I = \{i_1, \dots, i_t\} \subseteq \{1, \dots, n\}$ , we have  $\bar{x}_I = (x_{i_1}, \dots, x_{i_t})$  and  $f_I(\bar{x}) = (f_{i_1}(\bar{x}), \dots, f_{i_t}(\bar{x}))$ , where  $f(\bar{x})$  is a deterministic polynomial-time function. We say that the multi-party protocol  $\Pi$  computes  $f(\bar{x})$  privately if*

- there exists a probabilistic polynomial time algorithm, denoted by simulator  $S$ , such that for every  $I \subseteq \{1, \dots, n\}$  [26, p.696]:

$$S(\bar{x}_I, f_I(\bar{x})) \stackrel{c}{\equiv} V_I^\Pi(\bar{x}), \quad (6)$$

- the inputs and outputs of the coalition cannot be exploited to infer further private information.

Thus, our proofs in Appendix A and B will consist of proving these two points of Definition 6.

#### B. Paillier Homomorphic Cryptosystem

The homomorphic cryptosystem is a cryptographic primitive that allows computation over encrypted data. Proposed protocols use Paillier additive homomorphic cryptosystems [28], which is a probabilistic public-key cryptography scheme that provides two basic operations, namely (i) addition and subtraction of two encrypted values denoted by  $\oplus$  and  $\ominus$ , respectively, and (ii) multiplication of an encrypted value by a plaintext value denoted by  $\otimes$ . That is, if we denote the encryption of  $a$  using the public key  $pk$  by  $\llbracket a \rrbracket$ , then the Paillier cryptosystem supports

$$\text{DECRYPT}_{sk}(\llbracket a \rrbracket \oplus \llbracket b \rrbracket) = a + b, \quad (7)$$

$$\text{DECRYPT}_{sk}(a \otimes \llbracket b \rrbracket) = a \times b, \quad (8)$$

where  $sk$  is the private key associated with public key  $pk$ . We will omit the symbol  $\otimes$  when the type of multiplication can be inferred from the context. The security guarantees of the Paillier cryptosystem rely on the standard cryptographic assumption named decisional composite residuosity assumption (DCRA) [28]. We use the floats encoding mechanism presented in [29] that represents float numbers by a positive exponent and a mantissa, which are both integers and thus can be used with the aforementioned cryptosystems [30].

In the next two sections, we present our privacy-preserving state estimation protocols using synchronously collected measurements. We consider that all parties communicate over end-to-end encrypted channels. In our protocols, we propose using the Paillier cryptosystem to implement further encryption only for measurements and estimates, without covariances and model parameters due to the nature of the PHE. We don't use the fully homomorphic encryption FHE as PHE is more efficient and good enough since knowing only covariances doesn't reveal measurements or estimates. For simplicity, we will denote only paillier encryption by  $\llbracket \cdot \rrbracket$ .

### IV. PRIVATE ESTIMATION AMONG DISTRIBUTED SENSORS

We propose a protocol that solves Problem 1 to estimate the states of a system within the first pre-stated setup in Section II while preserving information privacy.

Initially, the query node generates the Paillier public key  $pk$  and the private key  $sk$  and shares the public key with other parties, then sends the initial estimates to the aggregator after encrypting its state vector  $\llbracket \mathbf{x}_{q,0} \rrbracket$  while revealing its covariance matrix  $\mathbf{P}_{q,0}$ . At every time step  $k$ , each sensor  $i$  also encrypts its measurement vector  $\llbracket \mathbf{y}_{i,k} \rrbracket$  and reveals its covariance matrix  $\mathbf{R}_{i,k}$ , and then sends them to the aggregator. The aggregator, in turn, aggregates all received measurements and applies the estimation algorithm steps [31, p. 190]. During the time update, the Kalman filter produces a predicted estimate for the system states in the current time step. During the measurement update, the predicted estimate is updated by processing all measurements in parallel. All measurement vectors are combined to form a new measurement vector  $\llbracket \mathcal{Y}_k \rrbracket \in \mathbb{R}^{pI}$  where  $I$  is the total number of sensors. Similarly,  $\mathcal{H} \in \mathbb{R}^{pI \times n}$  is the new observation matrix, and assuming that measurement noises are uncorrelated, the

---

**Protocol 1** Private Estimation Among Distributed Sensors
 

---

The query node  $Q$  encrypts the initial state  $\llbracket \mathbf{x}_{q,0} \rrbracket$  and sends it with the initial error covariance  $\mathbf{P}_{q,0}$  (not encrypted) to the aggregator node  $A$  to have  $\llbracket \hat{\mathbf{x}}_{a,0} \rrbracket = \llbracket \mathbf{x}_{q,0} \rrbracket$  and  $\mathbf{P}_{a,0} = \mathbf{P}_{q,0}$ ,

At every time step  $k$ , every sensor node shares its encrypted measurements  $\llbracket \mathbf{y}_{i,k} \rrbracket$  and the measurement noise covariance matrix  $\mathbf{R}_{i,k}$  (not encrypted) with the aggregator.

In order to find the required estimates, the aggregator uses Kalman filter estimation algorithm as following:

**Step 1:** Time update at the aggregator:

$$\llbracket \hat{\mathbf{x}}_{a,k}^- \rrbracket = \mathbf{F} \llbracket \hat{\mathbf{x}}_{a,k-1} \rrbracket, \quad (9)$$

$$\mathbf{P}_{a,k}^- = \mathbf{F} \mathbf{P}_{a,k-1} \mathbf{F}^T + \mathbf{Q}_k. \quad (10)$$

**Step 2:** Measurement update at the aggregator:

$$\mathbf{P}_{a,k} = \left( (\mathbf{P}_{a,k}^-)^{-1} + \mathcal{H}_k^T \mathcal{R}_k^\dagger \mathcal{H}_k \right)^{-1}, \quad (11)$$

$$\mathcal{K}_k = \mathbf{P}_{a,k} \mathcal{H}_k^T \mathcal{R}_k^\dagger, \quad (12)$$

$$\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket = \llbracket \hat{\mathbf{x}}_{a,k}^- \rrbracket \oplus \mathcal{K}_k \left( \llbracket \mathcal{Y}_k \rrbracket \ominus \mathcal{H}_k \llbracket \hat{\mathbf{x}}_{a,k}^- \rrbracket \right). \quad (13)$$

where

$$\llbracket \mathcal{Y}_k \rrbracket = \llbracket \llbracket \mathbf{y}_{1,k}^T \rrbracket, \dots, \llbracket \mathbf{y}_{I,k}^T \rrbracket \rrbracket^T,$$

$$\mathcal{H}_k = \llbracket \mathbf{H}_1^T, \dots, \mathbf{H}_I^T \rrbracket^T,$$

$$\mathcal{R}_k = \text{diag} \{ \mathbf{R}_{1,k}, \dots, \mathbf{R}_{I,k} \},$$

$\dagger$  represents the Moore–Penrose pseudoinverse,

$$\mathcal{K}_k = \llbracket \mathbf{K}_{1,k}, \dots, \mathbf{K}_{I,k} \rrbracket \text{ and } \mathbf{K}_{i,k} = \mathbf{P}_{a,k} \mathbf{H}_i^T \mathbf{R}_{i,k}^{-1}.$$

**Step 3:** The aggregator sends the outputs  $\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$  and  $\mathbf{P}_{a,k}$  to the query node to have  $\llbracket \hat{\mathbf{x}}_{q,k} \rrbracket = \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$  and  $\mathbf{P}_{q,k} = \mathbf{P}_{a,k}$ . In turn, query node decrypts  $\llbracket \hat{\mathbf{x}}_{q,k} \rrbracket$  and gets the desired estimate  $\hat{\mathbf{x}}_{q,k}$ .

---

covariance of  $\llbracket \mathcal{Y}_k \rrbracket$  is the diagonal matrix  $\mathcal{R}_k \in \mathbb{R}^{pI \times pI}$ . A new gain matrix  $\mathcal{K}_k \in \mathbb{R}^{n \times pI}$  is computed and then used to find the optimal estimate that the query node inquires. As the query node has the private key  $sk$ , it decrypts the received estimates and retrieves the desired state estimate  $\mathbf{x}_{q,k}$  for each time step. All the computational steps are detailed in Protocol 1. In the following theorem, we summarize the privacy guarantees of the protocol against coalitions predefined in Definitions 1, 2, and 3:

**Theorem 1.** *Protocol 1 solves Problem 1 while preserving computational privacy against*

- *Sensor coalitions,*
- *Cloud coalitions,*
- *Query coalitions if  $pm_r > n$ , where  $p$  is the measurement size,  $m_r$  the number of non colluding sensors and  $n$  the state size.*

Proof of Theorem 1 is detailed in Appendix A.

## V. PRIVATE ESTIMATION AMONG SENSOR GROUPS

In this section, we present a protocol to solve Problem 2 using a diffusion Kalman filter algorithm. Unlike Protocol 1, where both estimates and measurements are encrypted, now only estimates are homomorphically encrypted. There is no need to encrypt measurements within the same group as each sensor trusts all other sensors within the same group. Initially, the query node shares the initial estimates only with sensor groups, while it shares the Paillier public key with all parties after generating a pair of keys.

At every time step  $k$ , each sensor  $i$  within group  $j$  participates with its measurements  $\mathbf{y}_{i,k}$ . All measurements within-group are collected and used along with the previously estimated state  $\hat{\mathbf{x}}_{q,k-1}$  to compute a new prior estimate  $\hat{\mathbf{x}}_{g_j,k}^-$ . Then, the owner of each sensor group encrypts its prior estimate  $\llbracket \hat{\mathbf{x}}_{g_j,k}^- \rrbracket$  homomorphically and sends it to the aggregator. The aggregator, in turn, performs the diffusion update and computes a weighted average of all received encrypted estimates based on their uncertainties assuming that they are independent from each other which implies zero cross-covariances [32]. Next, the aggregator calculates the time update to find and submit the optimal estimate for the current time step  $\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$ . Finally, the query node decrypts the received result to find the desired estimated state  $\hat{\mathbf{x}}_{q,k}$  and then sends it to all sensor groups for use in the next iteration.

Unlike Protocol 1, which performs both Kalman filter steps (measurement update and time update) at the aggregator in the encrypted domain, Protocol 2 performs the measurement update at the sensor group level in the plaintext domain and implements a diffusion update step along with the time update step at the aggregator in the encrypted domain. For the diffusion update, there are several methods in the literature [13], and we chose the weighted average method since it suits homomorphic computations. In the following theorem, we summarize the privacy guarantees of Protocol 2 against coalitions predefined in Definitions 1, 2, and 3:

**Theorem 2.** *Protocol 2 solves Problem 2 while preserving computational privacy against*

- *Sensor coalitions,*
- *Cloud coalitions,*
- *Query coalitions if  $d_r > 1$ , where  $d_r$  is the number of non-colluding groups.*

Proof of Theorem 2 is detailed in Appendix B.

## VI. PRIVACY DISCUSSION

This paper demonstrates that Protocol 1 solves Problem 1 with reasonable privacy guarantees summarized in Theorem 1, which illustrates that the protocol preserves privacy against all sensor and cloud coalitions. In the case of query coalitions, it is necessary to constantly make sure that the number of non-colluding sensors  $m_r$  is greater than the ratio of the state size to the measurement size  $n/p$  to preserve privacy. Besides, we can suggest a minor change by keeping the measurement matrix  $\mathbf{H}_i$  or covariance  $\mathbf{R}_{i,k}$  private to the sensors and aggregator to overcome this information leakage.

## Protocol 2 Private Estimation among Sensor Groups

Query node  $Q$  sends the initial state to all sensor groups. At every time instant  $k$ , following steps are executed:

**Step 1:** Measurement update at each sensor group  $j$ :

$$\mathbf{P}_{g_j,k}^- = \left( (\mathbf{P}_{q,k-1})^{-1} + \sum_{i=1}^{I_j} \mathbf{H}_i^T \mathbf{R}_{i,k}^{-1} \mathbf{H}_i \right)^{-1}, \quad (14)$$

$$\mathbf{K}_{i,k} = \mathbf{P}_{g_j,k}^- \mathbf{H}_i^T \mathbf{R}_{i,k}^{-1}, \quad (15)$$

$$\hat{\mathbf{x}}_{g_j,k}^- = \hat{\mathbf{x}}_{q,k-1} + \sum_{i=1}^{I_j} \mathbf{K}_{i,k} (\mathbf{y}_{i,k} - \mathbf{H}_i \hat{\mathbf{x}}_{q,k-1}). \quad (16)$$

**Step 2:** Each sensor group  $j$  encrypts its estimate  $\llbracket \hat{\mathbf{x}}_{g_j,k}^- \rrbracket$  and sends its encrypted state to the aggregator.

**Step 3:** Diffusion update at the aggregator:

$$\mathbf{P}_{a,k}^- = \left( \sum_{j=1}^J (\mathbf{P}_{g_j,k}^-)^{-1} \right)^{-1}, \quad (17)$$

$$\llbracket \hat{\mathbf{x}}_{a,k}^- \rrbracket = \mathbf{P}_{a,k}^- \sum_{j=1}^J (\mathbf{P}_{g_j,k}^-)^{-1} \llbracket \hat{\mathbf{x}}_{g_j,k}^- \rrbracket. \quad (18)$$

**Step 4:** Time update at the aggregator:

$$\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket = \mathbf{F} \llbracket \hat{\mathbf{x}}_{a,k}^- \rrbracket, \quad (19)$$

$$\mathbf{P}_{a,k} = \mathbf{F} \mathbf{P}_{a,k}^- \mathbf{F}^T + \mathbf{Q}_k. \quad (20)$$

**Step 5:** The aggregator sends the encrypted state  $\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$ ,  $\mathbf{P}_{a,k}$  to the query node which decrypts and sends the results to the sensor groups.

Similarly, Protocol 2 solves Problem 2 while obtaining satisfying privacy guarantees as summarized in Theorem 2, which states that privacy is preserved against all coalitions unless the query node succeeds in colluding with all but one of the groups in an attempt to reveal the estimates of that group. Therefore, we must ensure that the number of non-colluding groups  $d_r$  is always more than one. To overcome the information leakage in the case of the query coalitions, we propose a slight modification by keeping the process matrix  $\mathbf{F}$  or the modeling noise covariance  $\mathbf{Q}_k$  private to the aggregator.

**Remark 1.** *There is an analogy between our protocols and two privacy-preserving set-based estimation protocols proposed in [5] using Zonotopes. Zonotope  $\mathcal{Z} = \langle c, G \rangle$  is a centrally symmetric set representation, where  $c$  is its center and  $G$  is its generator matrix. Assuming that the modeling and measurement noises are unknown but bounded by zonotopes:  $\mathbf{n}_k \in \mathcal{Z}_{\mathbf{Q},k} = \langle 0, \mathbf{Q}_k^z \rangle$  and  $\mathbf{v}_{i,k} \in \mathcal{Z}_{\mathbf{R},k} = \langle 0, \mathbf{R}_k^z \rangle$  respectively, and having  $\hat{\mathbf{x}}_k \equiv \hat{\mathbf{c}}_k$ ,  $\mathbf{P}_k \equiv \hat{\mathbf{G}}_k \hat{\mathbf{G}}_k^T$ ,  $\mathbf{R}_{i,k} \equiv \mathbf{R}_{i,k}^z \mathbf{R}_{i,k}^{zT}$  [33], we found that our first protocol achieves privacy guarantees similar to the privacy guarantees of the set-based estimation protocol among distributed sensors [5, Theorem 6.2], while privacy guarantees of our second protocol are similar to their peers of the set-based estimation among sensor groups [5, Theorem 7.2].*

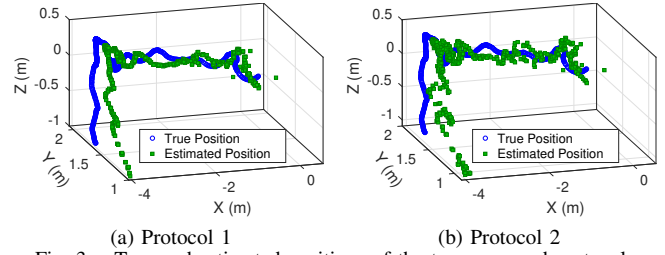


Fig. 3. True and estimated positions of the two proposed protocols.

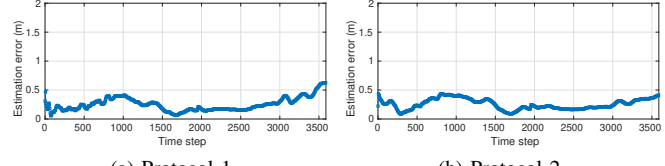


Fig. 4. Estimation error of the two proposed protocols.

## VII. EVALUATION

To evaluate the proposed protocols, we use measurements collected from the real testbed used in [34], [35], which includes a motion capture system that provides 3D rigid body position measurements. We apply our protocols to estimate the location of a quadrotor while preserving our previously mentioned privacy goals. Fig. 3 shows the true and estimated values of the 3D positions for each protocol. The estimation error of the two protocols is presented in Fig. 4. A comparison between the average execution time of each party is presented in Table I. The cloud time in Protocol 2 is shorter and depends on the diffusion method, and the encryption time is longer in sensor groups based on the sizes of state and measurement.

## VIII. CONCLUSIONS

This paper presented two privacy-preserving estimation protocols for both a typical sensor setup and another setup in which trustworthy sensors are grouped into sensor groups. We proved the privacy guarantees of each protocol using the computational indistinguishability concept. We demonstrated the possibility of encrypting only sensitive information rather than all while maintaining an acceptable level of privacy. Finally, we evaluated our protocols using actual data from a physical testbed and verified that they offer satisfying results while guaranteeing privacy. Our protocols have several practical applications, which we leave as future work.

## REFERENCES

- [1] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27–40, 2017.
- [2] X. Guan, B. Yang, C. Chen, W. Dai, and Y. Wang, "A comprehensive overview of cyber-physical systems: From perspective of feedback system," *Journal of Automatica Sinica*, vol. 3, no. 1, pp. 1–14, 2016.
- [3] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information fusion*, vol. 14, no. 1, pp. 28–44, 2013.
- [4] X.-B. Jin, R. J. RobertJeremiah, T.-L. Su, Y.-T. Bai, and J.-L. Kong, "The new trend of state estimation: from model-driven to hybrid-driven methods," *Sensors*, vol. 21, no. 6, p. 2085, 2021.
- [5] A. Alanwar, V. Gassmann, X. He, H. Said, H. Sandberg, K. H. Johansson, and M. Althoff, "Privacy preserving set-based estimation using partially homomorphic encryption," *arXiv preprint arXiv:2010.11097*, 2020.

TABLE I  
EXECUTION TIME IN ms.

	Entities		
	Sensor/Sensor group	Aggregator	Query
Protocol 1	7.75	4.6	1.77
Protocol 2	23.67	2.31	1.77

[6] J. Wang, D. Shi, J. Chen, and C.-C. Liu, "Privacy-preserving hierarchical state estimation in untrustworthy cloud environments," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1541–1551, 2020.

[7] C. Ierardi, L. Orihuela, and I. Jurado, "Distributed estimation techniques for cyber-physical systems: a systematic review," *Sensors*, vol. 19, no. 21, p. 4720, 2019.

[8] S. Huang, Y. Li *et al.*, "Distributed state estimation for linear time-invariant dynamical systems: A review of theories and algorithms," *Chinese Journal of Aeronautics*, 2021.

[9] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[10] J. Gao and C. J. Harris, "Some remarks on kalman filters for the multisensor fusion," *Information Fusion*, vol. 3, no. 3, pp. 191–201, 2002.

[11] D. Willner, C. Chang, and K. Dunn, "Kalman filter algorithms for a multi-sensor system," in *1976 conference on decision and control including the 15th symposium on adaptive processes*, 1976, pp. 570–574.

[12] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *8th IEEE International Conference on Intelligent Networks and Intelligent Systems*, 2015, pp. 74–77.

[13] M. S. Mahmoud and H. M. Khalid, "Distributed kalman filtering: a bibliographic review," *IET Control Theory & Applications*, vol. 7, no. 4, pp. 483–501, 2013.

[14] J. Le Ny, "Differentially private kalman filtering," in *Differential Privacy for Dynamic Data*. Springer, 2020, pp. 55–75.

[15] K. Yazdani and M. Hale, "Error bounds and guidelines for privacy calibration in differentially private kalman filtering," in *IEEE American Control Conference*, 2020, pp. 4423–4428.

[16] M. Mowbray, S. Pearson, and Y. Shen, "Enhancing privacy in cloud computing via policy-based obfuscation," *The Journal of Supercomputing*, vol. 61, no. 2, pp. 267–291, 2012.

[17] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," *arXiv preprint arXiv:1807.01860*, 2018.

[18] Y. Song, C. X. Wang, and W. P. Tay, "Privacy-aware kalman filtering," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4434–4438.

[19] A. Sultangazin, S. Diggavi, and P. Tabuada, "Protecting the privacy of networked multi-agent systems controlled over the cloud," in *IEEE 27th International Conference on Computer Communication and Networks*, 2018, pp. 1–7.

[20] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *IEEE 55th Conference on Decision and Control*, 2016, pp. 5053–5058.

[21] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based quadratic optimization with partially homomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2357–2364, 2020.

[22] M. Zamani, L. Sadeghikhorrani, A. A. Safavi, and F. Farokhi, "Private state estimation for cyber-physical systems using semi-homomorphic encryption," in *Proceedings of the 23rd International Symposium on Mathematical Theory of Networks and Systems*, 2018, pp. 399–404.

[23] Z. Zhang, P. Cheng, J. Wu, and J. Chen, "Secure state estimation using hybrid homomorphic encryption scheme," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1704–1720, 2020.

[24] Y. Ni, J. Wu, L. Li, and L. Shi, "Multi-party dynamic state estimation that preserves data and model privacy," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2288–2299, 2021.

[25] L. Sadeghikhorrani and A. A. Safavi, "Secure distributed kalman filter using partially homomorphic encryption," *Journal of the Franklin Institute*, vol. 358, no. 5, pp. 2801–2825, 2021.

[26] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[27] —, *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.

[28] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, 1999, pp. 223–238.

[29] A. A. M. A. Abdelhafez, "Localization of cyber-physical systems: privacy, security and efficiency," Ph.D. dissertation, Technische Universität München, 2020.

[30] M. T. I. Ziad, A. Alanwar, M. Alzantot, and M. Srivastava, "CryptoImg: privacy preserving processing over encrypted images," in *IEEE Conference on Communications and Network Security*, Oct 2016, pp. 570–575.

[31] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.

[32] F. Govaers, "Distributed kalman filter," *Kalman Filters: Theory for Advanced Applications*, p. 253, 2018.

[33] A. Alanwar, J. J. Rath, H. Said, K. H. Johansson, and M. Althoff, "Distributed set-based observers using diffusion strategy," *arXiv preprint arXiv:2003.10347*, 2020.

[34] H. Ferraz, A. Alanwar, M. Srivastava, and J. P. Hespanha, "Node localization based on distributed constrained optimization using jacobi's method," in *IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 3380–3385.

[35] A. Alanwar, H. Said, A. Mehta, and M. Althoff, "Event-triggered diffusion kalman filters," in *ACM/IEEE 11th International Conference on Cyber-Physical Systems*, 2020, pp. 206–215.

[36] J. R. Schott, *Matrix analysis for statistics*. John Wiley & Sons, 2016.

## APPENDIX

### A. Theorem 1 Proof

The proof is along similar lines of [5]. To prove that the privacy is preserved, we need to show that the coalitions views and simulators are computationally indistinguishable and that the coalition inputs and outputs do not leak extra private information according to Definition 6. The quantities denoted by  $\widetilde{(\cdot)}$  are those obtained by the simulator and they differ from the quantities of the views but follow the same distribution. "Coins" are random numbers used for the encryption process and key generation.  $[\Gamma_X]$  for coalition  $X$  represents any information that is transferred between other parties over an encrypted channel that may use double encryption with different keys from the homomorphic encryption keys.

*Proof.* In the following proof, we investigate the privacy of Protocol 1 against the following three coalitions:

#### 1) Coalition of sensors $s$

If we consider a set of sensors  $s = \{s_1, \dots, s_t\}$  participating in the coalition  $s$ , then  $V_s^\Pi$  represents the coalition view that can be defined as a combination of every sensor view and given by

$$V_s^\Pi = (V_{s_1}^\Pi, \dots, V_{s_t}^\Pi) = (\mathbf{H}_{s,k}, \mathbf{y}_{s,k}, \mathbf{R}_{s,k}, [\mathbf{y}_{s,k}], pk, \text{coins}_s, [\Gamma_s]), \quad (21)$$

and the coalition simulator  $S_s$  can be obtained by generating  $[\widetilde{\Gamma}_s]$ ,  $[\widetilde{y}_{s,k}]$  and  $\widetilde{\text{coins}}_s$ , i.e.,

$$S_s = (\mathbf{H}_{s,k}, \mathbf{y}_{s,k}, \mathbf{R}_{s,k}, [\widetilde{y}_{s,k}], pk, \widetilde{\text{coins}}_s, [\widetilde{\Gamma}_s]), \quad (22)$$

where  $\widetilde{\text{coins}}_s$  are generated according to the same distribution of  $\text{coins}_s$  and are independent from other parameters, where the same is true for  $[\widetilde{y}_{s,k}]$  and  $[\mathbf{y}_{s,k}]$  as well as  $[\widetilde{\Gamma}_s]$  and  $[\Gamma_s]$ . Therefore, we find that  $S_s \stackrel{c}{=} V_s^\Pi$ . Furthermore, each

measurement information is independent of all others, which makes the coalition unable to infer new information about the measurements of non-colluding sensors. For this coalition, we consider a single step in our proof since the information in each iteration differs from the other iterations.

For the next two coalitions, we will prove that each coalition view after  $K \in \mathbb{N}^+$  iterations of the protocol is computationally indistinguishable from the view of a simulator that runs the same  $K$  iterations.

## 2) Coalition of sensors and aggregator $sa$

Considering  $V_a^\Pi$  is the view of the aggregator, the view of a coalition consisting of a set of sensors  $s = \{s_1, \dots, s_t\}$  and the aggregator can be indicated by  $V_{sa}^\Pi$  where

$$V_{sa}^\Pi = (V_s^\Pi, V_a^\Pi) = (V_s^{\Pi,K}, V_a^{\Pi,K}), \quad (23)$$

where  $V_s^{\Pi,K}$  and  $V_a^{\Pi,K}$  are the views of the coalition of sensors and the aggregator respectively after executing  $K$  iterations, and

$$\begin{aligned} V_s^{\Pi,k+1} &= (V_s^{\Pi,k}, I_s^{k+1}), \\ V_a^{\Pi,k+1} &= (V_a^{\Pi,k}, I_a^{k+1}), \quad k = 0, 1, \dots, K-1, \end{aligned} \quad (24)$$

where  $I_s^k$  and  $I_a^k$  are the new data added at the  $k$ -th iteration for the sensors coalition  $V_s^{\Pi,0} = I_s^0$  and the aggregator  $V_a^{\Pi,0} = I_a^0$ . The view of the aggregator includes the sensors encrypted measurements and the initial estimates from the query node. The measurements at the  $k$ -th iteration from sensors that are not part of the coalition are marked by subscript  $r$  (i.e.,  $\mathbf{H}_{r,k}, \llbracket \mathbf{y}_{r,k} \rrbracket, \mathbf{R}_{r,k}, k = 0, 1, \dots, K-1$ ). Then, assuming that  $\mathbf{H}_{r,k}, \mathbf{R}_{r,k}$  are public,  $I_s^k$  and  $I_a^k$  are

$$I_s^k = (\mathbf{H}_{s,k}, \mathbf{y}_{s,k}, \mathbf{R}_{s,k}, pk, \llbracket \mathbf{y}_{s,k} \rrbracket, \text{coins}_s, \llbracket \Gamma_s \rrbracket), \quad (25)$$

$$\begin{aligned} I_a^k &= (\mathbf{H}_{s,k}, \llbracket \mathbf{y}_{s,k} \rrbracket, \mathbf{R}_{s,k}, \mathbf{H}_{r,k}, \llbracket \mathbf{y}_{r,k} \rrbracket, \mathbf{R}_{r,k}, \llbracket \hat{\mathbf{x}}_{q,0} \rrbracket, \\ &\quad \mathbf{P}_{q,0}, \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \mathbf{P}_{a,k}, \mathbf{F}, \mathbf{Q}_k, pk, \text{coins}_a), \end{aligned} \quad (26)$$

where  $\llbracket \hat{\mathbf{x}}_{q,0} \rrbracket, \mathbf{P}_{q,0}$  is the initial estimates from the query node and  $\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \mathbf{P}_{a,k}$  is the  $k$  estimates on the aggregator side. The view of the coalition  $V_{sa}^\Pi$  is constructed from (23)-(25). The simulator of the coalition can be denoted by  $S_{sa} = S_{sa}^K$ , where  $S_{sa}^K$  is the simulator after executing  $K$  iterations. The simulator  $S_{sa}$  can be formed using

$$S_{sa}^{k+1} = (S_{sa}^k, I_{sa}^{k+1}), \quad k = 0, 1, \dots, K-1, \quad (27)$$

where  $I_{sa}^{k+1}$  is the simulator portion generated at iteration  $k+1$ , that is given by

$$\begin{aligned} I_{sa}^{S,k} &= (\mathbf{H}_{s,k}, \llbracket \mathbf{y}_{s,k} \rrbracket, \mathbf{R}_{s,k}, \mathbf{H}_{r,k}, \llbracket \mathbf{y}_{r,k} \rrbracket, \mathbf{R}_{r,k}, \llbracket \hat{\mathbf{x}}_{q,0} \rrbracket, \mathbf{P}_{q,0}, \\ &\quad \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \mathbf{P}_{a,k}, \mathbf{F}, \mathbf{Q}_k, \mathbf{y}_{s,k}, pk, \widetilde{\text{coins}}_{sa}, \llbracket \Gamma_s \rrbracket), \end{aligned} \quad (28)$$

where its terms are generated or computed as follows:

- Generate  $\llbracket \mathbf{y}_{s,k} \rrbracket, \llbracket \mathbf{y}_{r,k} \rrbracket, \llbracket \hat{\mathbf{x}}_{q,0} \rrbracket, \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$  and  $\llbracket \Gamma_s \rrbracket$ , according to the same distribution of  $\llbracket \mathbf{y}_{s,k} \rrbracket, \llbracket \mathbf{y}_{r,k} \rrbracket, \llbracket \hat{\mathbf{x}}_{q,0} \rrbracket, \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket$  and  $\llbracket \Gamma_s \rrbracket$ , respectively.
- Compute  $\mathbf{P}_{a,k}$  according to (11).
- Suppose both coins are combined as  $\text{coins}_{sa} = (\text{coins}_a, \text{coins}_s)$ , then generate  $\widetilde{\text{coins}}_{sa}$  according to the same distribution.

Then, all the  $\llbracket \cdot \rrbracket$  and  $\llbracket \cdot \rrbracket$  values are indistinguishable and all other variables in  $I_{sa}^{S,k+1}$  are either public or attainable through the protocol steps. Hence, After all iteration steps, we end up with a simulator that achieves  $S_{sa} \stackrel{c}{\equiv} V_{sa}^\Pi$ . Now we need to ensure the coalition cannot infer further private information. The coalition's target is to find the private measurements of the non-colluding sensors  $\mathbf{y}_{r,k}$ . The relation between  $\llbracket \mathbf{y}_{s,k} \rrbracket$  and  $\llbracket \mathbf{y}_{r,k} \rrbracket$  can be derived from (13) as

$$\begin{aligned} \sum_{i \in \mathcal{N}_r} \mathbf{K}_{i,k} \llbracket \mathbf{y}_{i,k} \rrbracket &= \sum_{i \in \mathcal{N}_I} (\mathbf{K}_{i,k} \mathbf{H}_{i,k} - 1) \llbracket \hat{\mathbf{x}}_{a,k-1} \rrbracket \oplus \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket \\ &\quad \ominus \underbrace{\sum_{i \in \mathcal{N}/r} \mathbf{K}_{i,k} \llbracket \mathbf{y}_{i,k} \rrbracket}_{\text{known to the coalition in plaintext}}, \end{aligned} \quad (29)$$

where  $\mathcal{N}_r$  is the remaining sensors set. Since the coalition does not have the private key and the query node sends the initial encrypted estimate  $\llbracket \hat{\mathbf{x}}_{a,0} \rrbracket$ , As a result, we have a system that is undermined in (29).

## 3) Coalition of sensors and query node $sq$

Let  $V_q^\Pi$  be the view of the query node, then, the view of a coalition consisting of a set of sensors and the query is  $V_{sq}^\Pi$  that defined by

$$V_{sq}^\Pi = (V_s^\Pi, V_q^\Pi) = (V_s^{\Pi,K}, V_q^{\Pi,K}), \quad (30)$$

where

$$\begin{aligned} V_s^{\Pi,k+1} &= (V_s^{\Pi,k}, I_s^{k+1}), \\ V_q^{\Pi,k+1} &= (V_q^{\Pi,k}, I_q^{k+1}), \quad k = 0, 1, \dots, K-1, \end{aligned} \quad (31)$$

where  $I_s^k$  is given in (25), and  $I_q^k$  are the new data added from the  $k$ -th iteration for the query node with  $V_q^{\Pi,0} = I_q^0$  such that

$$I_q^k = (\hat{\mathbf{x}}_{q,0}, \mathbf{P}_{q,0}, \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \mathbf{P}_{a,k}, \hat{\mathbf{x}}_{a,k}, pk, sk, \text{coins}_q, \llbracket \Gamma_{sq} \rrbracket). \quad (32)$$

The coalition view  $V_{sq}^\Pi$  can be formed using (25), (31) and (32), where the simulator can be constructed as

$$\begin{aligned} S_{sq}^k &= (\mathbf{H}_{s,k}, \mathbf{y}_{s,k}, \mathbf{R}_{s,k}, \hat{\mathbf{x}}_{q,0}, \mathbf{P}_{q,0}, \hat{\mathbf{x}}_{a,k}, \mathbf{P}_{a,k}, \\ &\quad \llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \widetilde{\text{coins}}_{sq}, \llbracket \Gamma_{sq} \rrbracket, pk, sk, S_{sq}^{k-1}), \end{aligned} \quad (33)$$

where  $(\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \widetilde{\text{coins}}_{sq}, \llbracket \Gamma_{sq} \rrbracket)$  are generated according to the same distribution of  $(\llbracket \hat{\mathbf{x}}_{a,k} \rrbracket, \text{coins}_{sq}, \llbracket \Gamma_{sq} \rrbracket)$  and are independent from other parameters. Therefore,  $S_{sq}^k \stackrel{c}{\equiv} (I_s^k, I_q^k)$ , which proves that  $S_{sq} \stackrel{c}{\equiv} V_{sq}^\Pi$ . To complete our proof and similar to [5], it is important to examine whether the coalition can reveal the private information of the non-colluding  $m_r$  sensors. Since the query has the Paillier private key  $sk$ , we can rewrite (29) after decryption

$$\mathbf{K}_{r,k} \mathbf{Y}_{r,k} = \mathbf{z}_{s,k}, \quad (34)$$

where

$$\mathbf{z}_{s,k} = \sum_{i \in \mathcal{N}_I} (\mathbf{K}_{i,k} \mathbf{H}_{i,k}) \hat{\mathbf{x}}_{a,k-1} + \hat{\mathbf{x}}_{a,k} - \sum_{i \in \mathcal{N}/r} \mathbf{K}_{i,k} \mathbf{y}_{i,k}, \quad (35)$$

$$\mathbf{K}_{r,k} = [\mathbf{K}_{i_1,k}, \mathbf{K}_{i_2,k}, \dots, \mathbf{K}_{i_{m_r},k}] \in \mathbb{R}^{n \times pm_r},$$

$$\mathbf{Y}_{r,k} = [\mathbf{y}_{i_1,k}^T, \mathbf{y}_{i_2,k}^T, \dots, \mathbf{y}_{i_{m_r},k}^T]^T \in \mathbb{R}^{pm_r},$$

where  $\mathbf{z}_{s,k}$  is known to the coalition since each  $\mathbf{K}_{i,k}$  can be calculated using (12). We need to examine whether there is no unique retrieval for  $Y_{r,k}$  to preserve its privacy, which means that (34) has multiple solutions. According to [36, Theorem 6.4],  $\tilde{Y}_{r,k}$  is a solution of (34) for any  $X_r \in \mathbb{R}^{pI_r}$  with

$$\tilde{Y}_{r,k} = \mathbf{K}_{r,k}^- \mathbf{z}_{s,k} + (I_{pm_r} - \mathbf{K}_{r,k}^- \mathbf{K}_{r,k}) X_r, \quad (36)$$

where  $\mathbf{K}_{r,k}^-$  is any generalized inverse of  $\mathbf{K}_{r,k}$ . In order to ensure that (36) has multiple solutions, we need to keep  $(I_{pm_r} - \mathbf{K}_{r,k}^- \mathbf{K}_{r,k}) \neq 0$ . Having  $\mathbf{K}_{r,k}^- \mathbf{K}_{r,k} \neq I_{pm_r}$  implies that  $\text{rank}(\mathbf{K}_{r,k}) < pm_r$ , which is always true while  $n < pm_r$  according to [36, Theorem 5.23, Theorem 2.6]. Then, under the condition  $pm_r > n$ , system (36) has infinity solutions, which preserves the privacy of  $Y_{r,k}$ .  $\square$

### B. Theorem 2 Proof

*Proof.* Along the same line of the previous proof, we consider the view and simulator for one step (i.e.,  $k$ -th step). The proof for  $K \in \mathbb{N}^+$  steps is the same as previous proof. We prove again the privacy against the following three coalitions:

#### 1) Coalition of sensor groups $g$

Let  $V_g^\Pi$  denote the view of a coalition includes a set of sensor groups  $g = \{g_1, \dots, g_t\}$  and defined by

$$V_g^\Pi = (V_{g_1}^\Pi, \dots, V_{g_t}^\Pi) \quad (37)$$

$$= (\mathcal{H}_{g,k}, \mathcal{Y}_{g,k}, \mathcal{R}_{g,k}, \mathbf{P}_{q,k-1}, \hat{\mathbf{x}}_{q,k-1}, \mathbf{P}_{g,k}^-, \hat{\mathbf{x}}_{g,k}^-, pk, [\hat{\mathbf{x}}_{g,k}^-], \text{coins}_g, [\Gamma_g]), \quad (38)$$

where the subscript  $g$  denotes items owned by the coalition. Each sensor group submit only its encrypted prior estimate to the aggregator. Hence, the simulator  $S_g$  is defined by

$$S_g = (\mathcal{H}_{g,k}, \mathcal{Y}_{g,k}, \mathcal{R}_{g,k}, \mathbf{P}_{q,k-1}, \hat{\mathbf{x}}_{q,k-1}, \mathbf{P}_{g,k}^-, \hat{\mathbf{x}}_{g,k}^-, pk, [\hat{\mathbf{x}}_{g,k}^-], \text{coins}_g, [\Gamma_g]), \quad (39)$$

and  $[\hat{\mathbf{x}}_{g,k}^-]$ ,  $\text{coins}_g$  and  $[\Gamma_g]$  are generated with the same distribution of  $[\hat{\mathbf{x}}_{g,k}^-]$ ,  $\text{coins}_g$ ,  $[\Gamma_g]$  and are independent from other parameters. Therefore, we find that  $S_g \stackrel{c}{=} V_g^\Pi$ . Also, the coalition cannot infer any further information about other estimates of the non-colluding groups.

#### 2) Coalition of sensor groups and the aggregator $ga$

The coalition view is defined by

$$V_{ga}^\Pi = (V_g^\Pi, V_a^\Pi), \quad (40)$$

where

$$V_a^\Pi = ([\hat{\mathbf{x}}_{g,k}^-], \mathbf{P}_{g,k}^-, [\hat{\mathbf{x}}_{r,k}^-], \mathbf{P}_{r,k}^-, [\hat{\mathbf{x}}_{a,k}^-], \mathbf{P}_{a,k}, \mathbf{F}, \mathbf{Q}_k, \text{coins}_a, pk). \quad (41)$$

The simulator  $S_{ga}$  can be constructed by calculating  $\mathbf{P}_{g,k}^-$  and  $\mathbf{P}_{r,k}^-$  using (14),  $\mathbf{P}_{a,k}^-$  using (17), and generating

$[\hat{\mathbf{x}}_{g,k}^-]$ ,  $[\hat{\mathbf{x}}_{r,k}^-]$ ,  $[\hat{\mathbf{x}}_{a,k}^-]$ , and  $[\Gamma_g]$  as before.

$$S_{ga} = (\mathcal{H}_{r,k}, \mathcal{R}_{r,k}, \mathcal{H}_{g,k}, \mathcal{Y}_{g,k}, \mathcal{R}_{g,k}, \mathbf{P}_{q,k-1}, \hat{\mathbf{x}}_{q,k-1}, \mathbf{P}_{g,k}^-, \hat{\mathbf{x}}_{g,k}^-, \mathbf{P}_{r,k}^-, [\hat{\mathbf{x}}_{r,k}^-], [\hat{\mathbf{x}}_{g,k}^-], [\hat{\mathbf{x}}_{a,k}^-], \mathbf{P}_{a,k}, F, Q_k, pk, \text{coins}_{ga}, [\Gamma_g]). \quad (42)$$

Thus, we find that  $S_{ga} \stackrel{c}{=} V_{ga}^\Pi$ . To ensure that the coalition cannot find the remaining groups estimates, we use (18) to describe the relation between group estimates as

$$\sum_{j \in \mathcal{N}_r} (\mathbf{P}_{g_j,k}^-)^{-1} [\hat{\mathbf{x}}_{g_j,k}^-] = (\mathbf{P}_{a,k}^-)^{-1} [\hat{\mathbf{x}}_{a,k}^-] \oplus \underbrace{\sum_{j \in \mathcal{N}/r} (\mathbf{P}_{g_j,k}^-)^{-1} [\hat{\mathbf{x}}_{g_j,k}^-]}_{\text{known to the coalition in plaintext}}, \quad (43)$$

where  $\mathcal{N}_r$  is the set of non-colluding groups with size  $d_r$ . However, since the coalition does not know the private key, the privacy of the remaining groups' estimates can be guaranteed.

#### 3) Coalition of sensor groups and the query node $gq$

The coalition view  $V_{gq}^\Pi$  is defined as  $V_{gq}^\Pi = (V_g^\Pi, V_q^\Pi)$  where

$$V_q^\Pi = ([\hat{\mathbf{x}}_{a,k}^-], \hat{x}_{a,k}, \mathbf{P}_{a,k}, pk, sk, \text{coins}_q, [\Gamma_q]). \quad (44)$$

And the simulator  $S_{gq}$  can be constructed as before

$$S_{gq} = (\mathcal{H}_{g,k}, \mathcal{Y}_{g,k}, \mathcal{R}_{g,k}, \mathbf{P}_{g,k}^-, \hat{\mathbf{x}}_{g,k}^-, [\hat{\mathbf{x}}_{a,k}^-], \hat{x}_{a,k}, \mathbf{P}_{a,k}, pk, sk, \text{coins}_{gq}, [\Gamma_q]). \quad (45)$$

Thus we conclude that  $S_{gq} \stackrel{c}{=} V_{gq}^\Pi$ . Similar to [5] and to investigate the privacy of the non-colluding groups' estimates, we use (43) after decryption as the query has the private key  $sk$ .

$$\mathbf{P}_{r,k} \mathbf{X}_{r,k} = \mathbf{z}_{g,k}, \quad (46)$$

with

$$\mathbf{z}_{g,k} = (\mathbf{P}_{a,k}^-)^{-1} \hat{\mathbf{x}}_{a,k}^- - \sum_{j \in \mathcal{N}/r} (\mathbf{P}_{g_j,k}^-)^{-1} \hat{\mathbf{x}}_{g_j,k}^-, \quad (47)$$

$$\mathbf{P}_{r,k} = [(\mathbf{P}_{j_1,k}^-)^{-1}, (\mathbf{P}_{j_2,k}^-)^{-1}, \dots, (\mathbf{P}_{j_{d_r},k}^-)^{-1}] \in \mathbb{R}^{n \times nd_r},$$

$$\mathbf{X}_{r,k} = [\mathbf{x}_{j_1,k}^T, \mathbf{x}_{j_2,k}^T, \dots, \mathbf{x}_{j_{d_r},k}^T]^T \in \mathbb{R}^{nd_r},$$

where  $\mathbf{z}_{g,k}$  is known to the coalition as  $\hat{\mathbf{x}}_{a,k}^-$  and  $\mathbf{P}_{a,k}^-$  can be calculated using (19) and (20) assuming that  $\mathbf{F}$  and  $\mathbf{Q}_k$  are public and  $\mathbf{F}$  is invertible. If  $\mathbf{F}$  isn't invertible, then the privacy of the remaining groups will be guaranteed against all coalitions. Similarly to proof A and according to [36, Theorem 6.4],  $\tilde{X}_{r,k}$  is a solution of (46) for any  $X_r \in \mathbb{R}^{nd_r}$  where

$$\tilde{X}_{r,k} = \mathbf{P}_{r,k}^- \mathbf{z}_{g,k} + (I_{nd_r} - \mathbf{P}_{r,k}^- \mathbf{P}_{r,k}) X_r, \quad (48)$$

with  $\mathbf{P}_{r,k}^-$  is any generalized inverse of  $\mathbf{P}_{r,k}$ . We aim to find conditions at which  $I_{nd_r} - \mathbf{P}_{r,k}^- \mathbf{P}_{r,k} \neq 0$  to ensure privacy. Having  $\mathbf{P}_{r,k}^- \mathbf{P}_{r,k} \neq I_{nd_r}$  implies that  $\text{rank}(\mathbf{P}_{r,k}) < nd_r$ , which is always true while  $n < nd_r$  according to [36, Theorem 5.23, Theorem 2.6]. Thus, under the condition  $d_r > 1$ , the privacy of  $X_{r,k}$  is guaranteed.  $\square$