Privacy-Preserving Set-Based Estimation Using Partially Homomorphic Encryption

Amr Alanwar[a,*], Victor Gaßmann[b], Xingkang He[c], Hazem Said[d], Henrik Sandberg[c], Karl Henrik Johansson[c], Matthias Althoff[b]

[a] *School of Computer Science and Engineering, Constructor University, Germany*
[b] *Department of Computer Engineering, Technical University of Munich, Germany*
[c] *Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden*
[d] *Department of Computer Engineering, Ain Shams University, Egypt*

Abstract

The set-based estimation has gained a lot of attention due to its ability to guarantee state enclosures for safety-critical systems. However, collecting measurements from distributed sensors often requires outsourcing the set-based operations to an aggregator node, raising many privacy concerns. To address this problem, we present set-based estimation protocols using partially homomorphic encryption that preserve the privacy of the measurements and sets bounding the estimates. We consider a linear discrete-time dynamical system with bounded modeling and measurement uncertainties. Sets are represented by zonotopes and constrained zonotopes as they can compactly represent high-dimensional sets and are closed under linear maps and Minkowski addition. By selectively encrypting parameters of the set representations, we establish the notion of encrypted sets and intersect sets in the encrypted domain, which enables guaranteed state estimation while ensuring privacy. In particular, we show that our protocols achieve computational privacy using the cryptographic notion of computational indistinguishability. We demonstrate the efficiency of our approach by localizing a real mobile quadcopter using ultra-wideband wireless devices.

Keywords: Set-based estimation, homomorphic encryption, zonotopes, constrained zonotopes.

1. Introduction

State estimation from noisy measurements is of great importance in many areas, such as navigation, communication, and remote sensing. Many of these applications are based on prior knowledge of noise distributions. However, assumed noise distributions are not always sufficiently accurate or even unknown. Furthermore, safety-critical applications require guaranteed state inclusion in a bounded set to provably avoid unsafe sets. This motivates the need for set-based estimation, which estimates the set of all possible system states when input disturbances and observation errors are unknown but belong to given bounded sets [1]. Set-based estimators are used in many applications, such as underwater robotics [2], fault detection [3, 4], leader-follower problems [5], and localization [6]. We refer the reader to [7] and references therein for more related work on set-based estimation.

Some state estimation algorithms require measurements made by a set of spatially distributed sensors. For instance, cellular signals from distributed mobile devices can be measured by base stations to estimate the targeted device location [8]. Situational awareness in safe autonomous driving requires collecting measurements from distributed vehicles and infrastructure nodes [9]. These computations require cloud-based services that aggregate and process gathered information to provide estimates with guarantees. However, this often requires that clients disclose sensitive information to the cloud to receive appropriate control decisions. This causes security vulnerabilities [10, 11], especially when sensors do not belong to the same trust zone in which members of the same organization trust each other. For this reason, we focus on set-based estimation in the cloud with estimation and privacy guarantees.

1.1. Related Work

There exist three types of set-based observers: strip-based observers, set-propagation observers, and interval observers [7]. Since we will use strip-based observers in this work, we focus our literature review on this observer type and refer the interested reader to [7] for the other observer types. Strip-based observers intersect the propagated set of states with the set of states consistent with the next measurement to obtain the next set of possible states. The set representation is essential to obtain a good computational complexity ratio and the estimated sets' achieved tightness. Ellipsoids are explored in [1, 12, 13], where the computations are generally efficient but not exact for the Minkowski sums. A new geometric method based on the Minkowski sum is proposed in [14] to produce a distributed ellipsoidal estimation. Zonotopes [15] are a special class of polytopes for which one can efficiently compute linear maps and Minkowski sums – both are important operations for set-based observers. Set-membership using zonotopes is explored in [16]. A novel zonotope intersection method and a new distributed set-based estimator were proposed in [17]. A distributed zonotopic and Gaussian Kalman filter is proposed in [18], where each

*Corresponding author.
Email addresses: aalanwar@constructor.university (Amr Alanwar), victor.gassmann@tum.de (Victor Gaßmann), xingkang@kth.se (Xingkang He), hazem.said@eng.asu.edu.eg (Hazem Said), hsan@kth.se (Henrik Sandberg), kallej@kth.se (Karl Henrik Johansson), althoff@tum.de (Matthias Althoff)

network node implements a local state estimator using zonotopes and Gaussian noise mergers. Polytopes [19] and orthotopes [20, 21] were explored as well.

Related work on set-based estimation does not provide privacy guarantees. Homomorphic encryption allows processing over encrypted data and has been used as a countermeasure for cloud-side information leakage, enabling useful tasks to be accomplished while keeping the data confidential from untrusted parties. Over the past few years, a significant effort in the form of a homomorphic library [22] has been made to make fully homomorphic encryption practical. Homomorphic encryption has been used for computationally expensive tasks over genome data [23] and classification over encrypted data [24]. However, fully homomorphic encryption remains impractical for real-time estimation [25, Section 2.10.1]. That said, partial homomorphic encryption methods are more promising and have been used for encrypted control [26, 27], image processing [28], estimation [8, 29], deep learning [30], optimization [31], and ride-sharing [32].

A related technique to our work is differential privacy [33, 34], which relies on the addition of structured noise to the data before sharing it, which preserves privacy. Variants of this scheme, such as local differential privacy [35, 36] and geo-indistinguishability [37], have been designed to ensure differential privacy for location data. However, the privacy guarantees of these methods are often achieved at the expense of accuracy [38]. In other words, the added structured noise results in a loss of estimation accuracy, making it unsuitable for use in safety-critical systems. To overcome the addition of excessive noise, a combination of homomorphic encryption with distributed noise has been proposed in [39], where each estimator generated its share of the aggregated noise required for differential privacy [40] and sent encrypted and obfuscated data to the aggregator.

1.2. Contributions

To the best of our knowledge, for the first time, we leverage a partially homomorphic cryptosystem to calculate encrypted sets that enclose states based on encrypted measurements and estimates from sensors or sensor groups. This work introduces two protocols providing state inclusion and privacy guarantees. In particular, we show that our protocols achieve computational privacy using computational indistinguishability against different coalitions of participated entities. We leverage state-of-art state estimation techniques in combination with homomorphic encryption to provide privacy-preserving set-based estimation protocols with security guarantees. Our entire code and data are available online[1].

More specifically, we make the following contributions:

- We encrypt a set of states using a partially homomorphic cryptosystem with different levels of privacy based on selective encryption and geometric features of the chosen set representation.
- We present two set-based estimation protocols which preserve privacy between sensor and sensor groups.
- We prove security guarantees of the two protocols against different coalitions, using formal cryptographic definitions of computational indistinguishability for protecting the estimated set position (Theorems 1 and 3) and protecting the estimated set position and shape (Theorems 2 and 4).

[1] https://github.com/aalanwar/Encrypted-set-based-estimation

1.3. Outline

The paper is organized as follows: In Section 2, we provide the necessary preliminaries. We formulate the problem and set our privacy goals in Section 3. After proposing the notion of encrypted sets in Section 4, we introduce protocols to privately bound the state among distributed sensors in Section 5 and then among sensor groups in Section 6. Finally, we evaluate the proposed protocols in Section 7 and conclude this paper with Section 8.

1.4. Notation

Vectors and scalars are denoted by lowercase letters, matrices are denoted by uppercase letters, the real and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$. We denote the set of positive real and positive natural numbers by $\mathbb{R}^+$ and $\mathbb{N}^+$, respectively, and all other continuous sets are denoted by calligraphic letters. For a given matrix $M \in \mathbb{R}^{o\times k}$, its Frobenius norm is given by $\|M\|_F = \sqrt{\text{tr}(M^T M)}$. For two sets $\mathcal{M}_1 \subseteq \mathbb{R}^q$, and $\mathcal{M}_2 \subseteq \mathbb{R}^q$, the Minkowski sum and the intersection are denoted by $\mathcal{M}_1 \boxplus \mathcal{M}_2$ and $\mathcal{M}_1 \cap \mathcal{M}_2$, respectively. For a set $\mathcal{M} \subseteq \mathbb{R}^q$, its linear map is denoted by $L\mathcal{M}$, where $L \in \mathbb{R}^{v\times q}$. For a given matrix M (can also be a vector or scalar), we denote with $[\![M]\!]$ the encrypted value of M. For given vectors a_1 and a_2 of same dimension, we denote with $[\![a_1]\!] \oplus [\![a_2]\!]$ and $[\![a_1]\!] \ominus [\![a_2]\!]$ the sum and difference over the encrypted values of a_1 and a_2, respectively. For two real scalars a and b, we denote with $a \otimes [\![b]\!]$ the multiplication of the encrypted scalar b with the unencrypted scalar a. We denote probability of an event E by $\Pr[E]$. The cardinality of a set $\mathcal{M}$ is denoted by $|\mathcal{M}|$. For a given vector $x \in \mathbb{R}^p$, the i-th component of x is denoted by $x_{[i]} \in \mathbb{R}$. We denote the reduce operator returning an over-approximative zonotope with q generators by $\downarrow_q$.

2. Preliminaries

In this section, we review the required preliminaries.

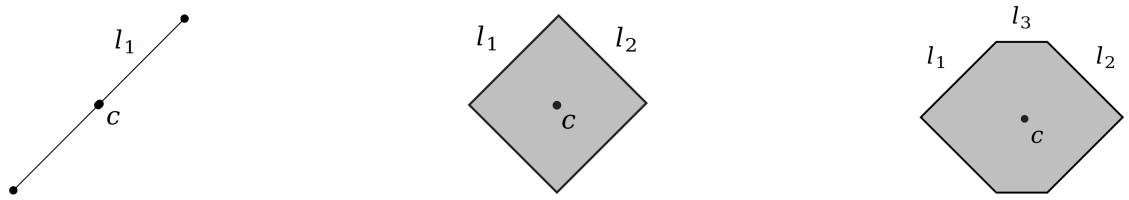
2.1. Set Representations and Set-Based Estimation

We define the following set representations:

Definition 1. (Zonotope) ([15]) An n-dimensional zonotope $\mathcal{Z}$ is defined as

$$\mathcal{Z} = \left\{ x \in \mathbb{R}^n \;\middle|\; x = c + G\beta,\ \|\beta\|_\infty \leq 1 \right\}, \tag{1}$$

where $c \in \mathbb{R}^n$ is the center, $G \in \mathbb{R}^{n\times e}$ is the generator matrix of the zonotope, and $\beta \in \mathbb{R}^e$ is the vector of zonotope factors.



(a) $c \boxplus l_1$ (b) $c \boxplus l_1 \boxplus l_2$ (c) $c \boxplus l_1 \boxplus l_2 \boxplus l_3$

Figure 1: Construction of a zonotope.

For later use, we represent the zonotope by $\mathcal{Z} = \langle c, G \rangle$. Note that G consists of e generators $g^{(i)} \in \mathbb{R}^n$, $i = 1, .., e$, such that $G = [g^{(1)}, ..., g^{(e)}]$.

This definition can be interpreted as the Minkowski sum of a finite set of line segments $l_i = \left\{ g^{(i)}\beta_i \in \mathbb{R}^n \;\middle|\; |\beta_i| \leq 1 \right\}$, where $i \in \{1, ..., e\}$ (see Figure 1). For a zonotope $\mathcal{Z} = \langle c, G \rangle$, we denote its F-radius as $\|G\|_F$. Given two zonotopes $\mathcal{Z}_1 = \langle c_1, G_1 \rangle$ and $\mathcal{Z}_2 = \langle c_2, G_2 \rangle$, the Minkowski sum is computed as $\mathcal{Z}_1 \boxplus \mathcal{Z}_2 = \langle c_1 + c_2, [G_1, G_2] \rangle$, whereas the linear map is computed as $L\mathcal{Z}_1 = \langle Lc_1, LG_1 \rangle$ [41].

Definition 2. (Constrained Zonotope) ([42, Prop. 1]) An n-dimensional constrained zonotope is defined as

$$\mathcal{C} = \left\{ x \in \mathbb{R}^n \;\middle|\; x = c + G\beta,\ A\beta = b,\ \|\beta\|_\infty \leq 1 \right\}, \tag{2}$$

where $c \in \mathbb{R}^n$ is the center, $G \in \mathbb{R}^{n \times n_g}$ is the generator matrix, $\beta \in \mathbb{R}^{n_g}$, and $A \in \mathbb{R}^{n_c \times n_g}$ and $b \in \mathbb{R}^{n_c}$ constitute the constraints. In short, we write $\mathcal{C} = \langle c, G, A, b \rangle$.

2.2. Paillier Homomorphic Cryptosystem and Privacy Definitions

A homomorphic cryptosystem supports computation over encrypted data. Our protocols heavily rely on Paillier additive homomorphic cryptosystems [43], which is a probabilistic public key cryptography scheme. The Paillier cryptosystem supports

$$\textsc{Decrypt}_{\text{sk}}(\llbracket a \rrbracket \oplus \llbracket b \rrbracket) = a + b, \tag{3}$$

$$\textsc{Decrypt}_{\text{sk}}(a \otimes \llbracket b \rrbracket) = a \cdot b, \tag{4}$$

where sk is the private key associated with the public key pk used for encryption. We will omit the symbol $\otimes$ when the type of multiplication can be inferred from the context. Our proposed protocols can utilize different homomorphic encryption schemes instead of the Paillier cryptosystem as long as it supports the same functionality.

Homomorphic encryption does not support float numbers. The naive solution is multiplying the float number by 10^f where f is the number of floating digits [44, 45, 46]. However, the recursive execution of the estimator or the controller generally requires recursive multiplication with fractional numbers. This approach requires truncating the significance of the state from time to time to avoid overflow. Such truncation might lead to computation errors and fast overflow. We can not use a solution that introduces computation errors because we provide safety and set containment guarantees. To overcome this limitation, we represent float numbers by a positive integer exponent and an integer mantissa, as we did in our previous work [28]. This representation provides exact computations. However, it still suffers from overflows after some iterations.

We define $\{0, 1\}^\star$ as a sequence of bits of unspecified length. An ensemble $X = \{X_o\}_{o \in \mathbb{N}}$ is a sequence of random variables X_o ranging over strings of bits of polynomial length in o. We need the following definitions in our privacy proofs.

Definition 3. (Computationally Indistinguishable) ([47, p.105]) The ensembles $X = \{X_o\}_{o \in \mathbb{N}}$ and $Y = \{Y_o\}_{o \in \mathbb{N}}$ are computationally indistinguishable, denoted $X \overset{c}{\equiv} Y$, if for every probabilistic polynomial-time algorithm D, every positive polynomial $p : \mathbb{N}^+ \mapsto \mathbb{R}^+$, and all sufficiently large o, it holds that

$$\left| \Pr[D(X_o) = 1] - \Pr[D(Y_o) = 1] \right| < \frac{1}{p(o)}. \tag{5}$$

In other words, given an algorithm D, we consider the probability that D outputs 1 given an ensemble taken from the two random variables X_o and Y_o as input. Then, we say $X \overset{c}{\equiv} Y$ if no efficient algorithm can tell the difference between them except with small probability $\frac{1}{p(o)}$.

Definition 4. (Execution View) Let $f : \mathbb{R}^o \mapsto \mathbb{R}^o$ be a deterministic polynomial-time function and Π a multi-party protocol computing $f(\bar{x})$, where $\bar{x} \in \mathbb{R}^o$. The view of the i^{th} party during an execution of Π on $\bar{x}$, denoted by V_i^Π, is $(x_i, coins, M_i)$, where $coins$ represents the outcome of the i^{th} party's internal coin toss, and M_i represents the set of messages it has received. For coalition $I = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, o\}$ of parties, the view $V_I^\Pi(\bar{x})$ of the coalition during an execution of Π is defined as

$$V_I^\Pi(\bar{x}) = (I, V_{i_1}^\Pi(\bar{x}), \ldots, V_{i_t}^\Pi(\bar{x})). \tag{6}$$

This means that the view of the party is all its accessible information and the view $V_I^\Pi(\bar{x})$ of the coalition I is the union of all the views of coalition parties.

Definition 5. (Multi-party Privacy w.r.t. Semi-honest Behavior) Let $f : \mathbb{R}^o \mapsto \mathbb{R}^o$ be a deterministic polynomial-time function and Π a multi-party protocol computing $f(\bar{x})$, where $\bar{x} \in \mathbb{R}^o$. For a coalition $I = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, o\}$ of parties, we have $\bar{x}_I = (x_{i_1}, \ldots, x_{i_t})$ and $f_I(\bar{x}) = (f_{i_1}(\bar{x}), \ldots, f_{i_t}(\bar{x}))$. We say that Π computes $f(\bar{x})$ privately if

- there exists a probabilistic polynomial time algorithm, denoted by simulator S, such that for every $I \subseteq \{1, \ldots, o\}$ [47, p.696]

$$S(\bar{x}_I, f_I(\bar{x})) \overset{c}{\equiv} V_I^\Pi(\bar{x}), \tag{7}$$

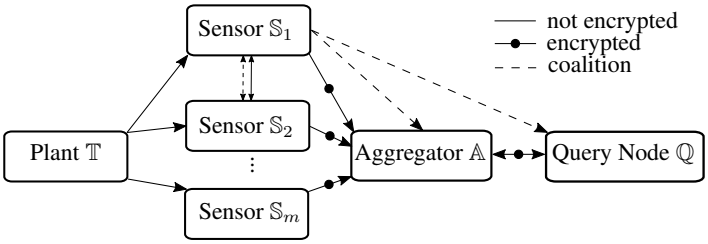


Figure 2: Diagram for the considered setup in Problem 1.

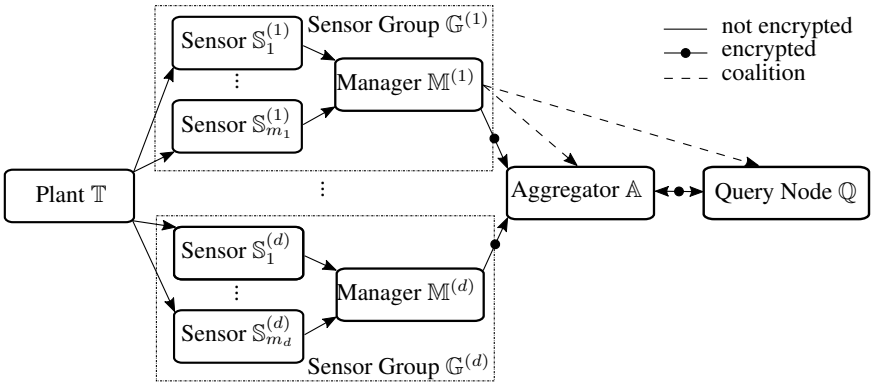


Figure 3: Diagram for the considered setup in Problem 2.

- the input and output of the coalition cannot be used to infer extra private information.

Put differently, a protocol privately computes $f(\bar{x})$ if whatever can be obtained from a party's view of a (semi-honest) execution could be essentially obtained from the input and output available to that party [47, p. 620]. Also, the inputs and outputs of the coalition cannot be used to infer extra private information. Thus, our privacy proofs will always consist of the two parts of Definition 5.

3. Problem Setup

Next, let us introduce some entities for our problem setups visualized in Figures 2 and 3.

- **Plant $\mathbb{T}$:** A passive entity whose set of possible states needs to be estimated. We consider discrete-time linear systems with bounded noise, specifically

$$\begin{aligned} x_{k+1} &= F x_k + n_k \\ y_{i,k} &= H_{i,k} x_k + v_{i,k}, \end{aligned} \tag{8}$$

  where $x_k \in \mathbb{R}^n$ is the state at time $k \in \mathbb{N}$, $y_{i,k} \in \mathbb{R}^p$ denotes the measurement observed at sensor i, F is the process matrix, $H_{i,k}$ stands for the measurement matrix, $n_k \in \mathcal{Q}_k$ is the process noise bounded by process noise zonotope $\mathcal{Q}_k = \langle 0, Q_k \rangle$, and $v_{i,k} \in \mathcal{R}_k$ is the measurement noise bounded by measurement noise zonotope $\mathcal{R}_k = \langle 0, \mathrm{diag}([r_{1,k}, \ldots, r_{m,k}]) \rangle$. All vectors and matrices are real-valued and have proper dimensions.

- **Sensor $\mathbb{S}_i$:** Entity with index i that provides private measurements. Its owner does not trust other active entities.

- **Aggregator $\mathbb{A}$ (or Cloud):** An untrusted party which has reasonable computational power. It executes the proposed private set-based estimation protocols over encrypted sensor information.

- **Query Node $\mathbb{Q}$:** An untrusted party that has a known public key pk and a hidden private key sk. The query node is the only node that is entitled to know the set of states of the plant $\mathbb{T}$. It might be the plant $\mathbb{T}$, but can also be any other entity other than the aggregator $\mathbb{A}$ (in order to preserve privacy).

- **Manager $\mathbb{M}^{(j)}$:** Entity with index j which estimates the state for a group of sensors and handles communication with other entities.

- **Sensor Group $\mathbb{G}^{(j)}$:** Entity with index j which consists of m_j sensors $\mathbb{S}_i^{(j)}$, $i \in \{1, ..., m_j\}$, and one manager $\mathbb{M}^{(j)}$ owned by one organization. All sensors within a group trust each other and do not trust other entities. Each sensor group aims to keep its measurements and estimates private from other groups and parties.

We provide the following definitions which are essential for set-based estimation:

Definition 6. (Set-based Estimator) Given system (8) with initial state $x_0 \in \langle c_0, G_0 \rangle$, the set-based estimator aims to find the corrected state set $\bar{\mathcal{S}}_{i,k}$ with state containment guarantees at each time step k, i.e., $\forall k : x_k \in \bar{\mathcal{S}}_{i,k}$.

With $x_0 \in \langle c_0, G_0 \rangle$, the predicted state set $\hat{\mathcal{S}}_{i,k}$, i.e., the set of all possible state values, is, according to (8), given by

$$\hat{\mathcal{S}}_{i,k} = F \bar{\mathcal{S}}_{i,k-1} \boxplus \mathcal{Q}_i. \tag{9}$$

For a given measurement $y_{i,k}$, the measurement state set $\mathcal{P}_{i,k}$ is the set of all possible state values satisfying the strip equation, i.e.,

$$\mathcal{P}_{i,k} = \left\{ x \;\middle|\; |H_{i,k}x - y_{i,k}| \leq r_{i,k} \right\}. \tag{10}$$

Where convenient, we will use the shorthand $\mathcal{P}_{i,k} = \langle y_{i,k}, H_{i,k}, r_{i,k} \rangle$ for a strip. The corrected state set $\bar{\mathcal{S}}_{i,k}$ is then the over-approximation of the intersection between $\hat{\mathcal{S}}_{i,k}$ and $\mathcal{P}_{i,k}$, specifically

$$\bar{\mathcal{S}}_{i,k} \supseteq (\hat{\mathcal{S}}_{i,k} \cap \mathcal{P}_{i,k}). \tag{11}$$

We aim to find solutions for the following two problems:

Problem 1. We want to estimate the set of possible state values of plant $\mathbb{T}$ while ensuring that measurements are private to the sensor nodes $\mathbb{S}_1, \ldots, \mathbb{S}_m$, $m \in \mathbb{N}^+$, and the estimated set is private to the query node $\mathbb{Q}$.

Problem 2. We want to estimate a set of all possible state values of the plant $\mathbb{T}$ while ensuring that measurements and internally estimated sets are private to the sensor groups $\mathbb{G}_1, \ldots, \mathbb{G}_d$, $d \in \mathbb{N}^+$, and the estimated set is private to the query node $\mathbb{Q}$.

To illustrate the practical relevance of Problems 1 and 2, consider the following scenario:

Example 1. To avoid collisions between traffic participants in a typical highway scenario (see Figure 4), each participant aims to perceive and comprehend a traffic situation by predicting the intent of vehicles and road users. This can be done by computing and sharing the reachable sets of all other traffic participants, known as shared situation awareness [9]. However, computing these sets is not always possible due to computational constraints or having a participant in an occluded area from the perspective of others (see the pedestrian in Figure 4). The different entities are the following: The plant is the combination of different car dynamics communicated in an initial phase, the sensors measure the distance between the traffic participants, the cloud is the aggregator, and the street management unit that guarantees participants' safety is the query node, which aims to compute the estimated set of the position of each participant. A possible solution hereby is to let the cloud compute reachable sets (and possible intersections thereof) while preserving the privacy of each participant (Problem 1). Specific future scenarios may contain a car platoon trusting its participants but not other platoons (Problem 2).

For both problems it is required to guarantee computational security during the estimation process. The query node $\mathbb{Q}$ is interested in finding the set of all possible state values of plant $\mathbb{T}$ in both problems. We should note that the group manager locally estimates over unencrypted data in Problem 2, which is not the case for Problem 1 (no group manager). If we consider the group of one sensor, there is still a need for a group manager to perform the local estimation over unencrypted data, so that Problem 1 is not a special case of Problem 2.

To set our privacy goals, we must first define the following coalitions that the attacker can perform for Problem 1:

Definition 7. (Sensor Coalition) A sensor colludes with up to $t-1$ other sensors in Problem 1 by exchanging their private measurements and cryptographic private keys, constituting a sensor coalition. The coalition aims to retrieve the private information of the non-participating sensors and the query node.

Definition 8. (Cloud Coalition) The aggregator $\mathbb{A}$ colludes with up to t sensors in Problem 1 by exchanging their private values, cryptographic private keys, and intermediate results, constituting the aggregator coalition. The coalition aims to retrieve the private information of the non-participating sensors and query node.

Definition 9. (Query Coalition) The query node $\mathbb{Q}$ colludes with up to t sensors in Problem 1 by exchanging their private values, cryptographic private keys, and the final decrypted outcome of the estimation protocol, constituting the query coalition. The coalition aims to retrieve the private information of the non-participating sensors.

The same definitions hold for Problem 2 by considering sensor groups instead of sensors. We consider semi-honest parties [8] following the protocol properly, with the exception that they keep a record of all its intermediate computations to infer extra information. This paper aims to solve Problems 1 and 2 by proposing multiple secure multi-party computation protocols. These protocols should guarantee computational privacy against the aforementioned coalitions. The privacy goals are based on the concept of computational indistinguishability, which is presented next, along with the formal definition of multi-party privacy with respect to semi-honest behavior while considering coalitions.

We state and summarize the assumptions of this work subsequently:

Assumption 1. We assume that both process noise n_k and measurement noise $v_{i,k}$ are bounded by a zonotope, i.e., $n_k \in \mathcal{Q}_k = \langle 0, Q_k \rangle$, and $v_{i,k} \in \mathcal{R}_k = \langle 0, \mathrm{diag}([r_{1,k}, \ldots, r_{m,k}]) \rangle$.

Furthermore, we make the following assumption for the attacker's ability:

Assumption 2. The attacker can form sensor, aggregator, or query coalitions (see Definitions 7, 8 and 9).

It is worth mentioning that we exclude an aggregator-query coalition, which is a common assumption in homomorphic encryption; see [44].

4. Encrypted Set of States

The aforementioned set representations are deliberately chosen such that they can be used with the Paillier cryptosystem and do not reveal critical information about the measurements and estimates. More specifically, we propose using zonotopes, constrained zonotopes, and strips as set representations in privacy-preserving set-based estimation:

1. Zonotopes: We homomorphically encrypt the center and reveal the generator matrix, thus hiding the position of the zonotope and only revealing the estimation uncertainty (zonotope shape) described by the generator matrix.

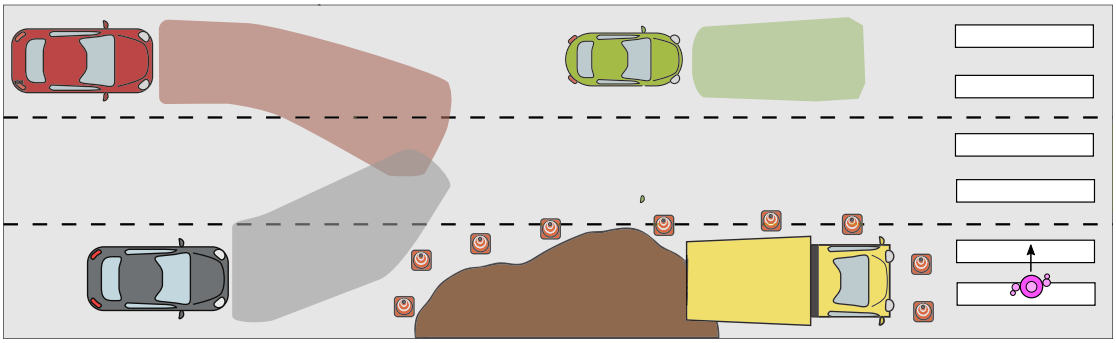
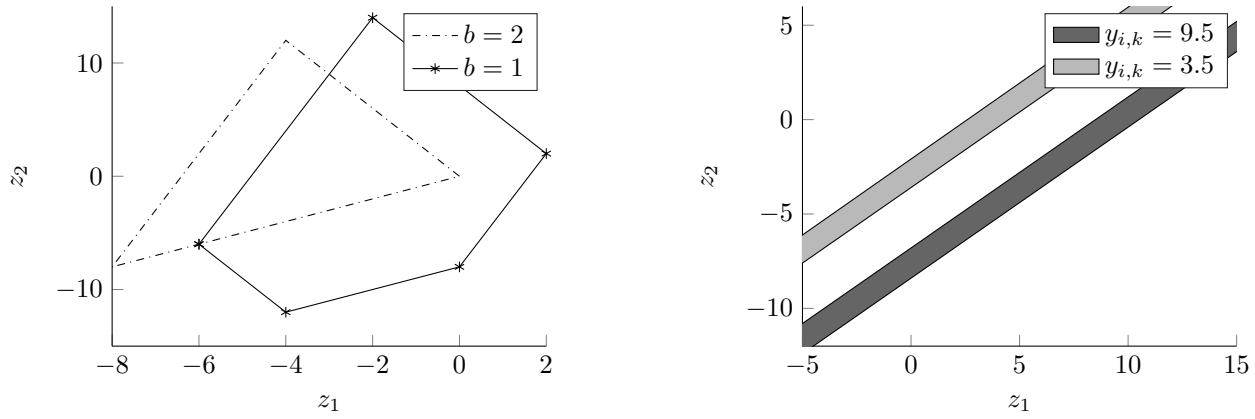


Figure 4: Highway scenario with construction work as possible application for Problems 1 and 2. Reachable sets for each car are shown in their respective, transparent color [48, adapted].



(a) Two constrained zonotopes with different b values.

(b) Two strips with different $y_{i,k}$ values.

Figure 5: Changing the effective parameter of constrained zonotope and strips.

2. Constrained zonotopes: To protect the estimation uncertainty (more privacy) while introducing extra computation overhead, we use constrained zonotopes instead of zonotopes. Changing b of a constrained zonotope $\langle c, G, A, b\rangle$ (see Definition 2) changes both its position and shape as shown in Figure 5a. We propose to encrypt the vectors c and b and thus encrypt both position and shape of the set.
3. Strips: For a strip given by (10), we encrypt $y_{i,k}$, and reveal $H_{i,k}$ and $r_{i,k}$. Encrypting $y_{i,k}$ preserves the privacy of the strip position as shown in Figure 5b for two strips with $H_{i,k} = [-1.25, 1]$, $r_{i,k} = 1$, and $y_{i,k} \in \{3.5, 9.5\}$.

The chosen selective encryption will allow us to decouple the computation of public information from private information, as we will show later. We propose two protocols to solve Problems 1 and 2 while preserving the mentioned privacy goals. Two variants of the proposed protocols solve the problems using zonotopes while revealing the estimation uncertainty. The other two variants solve the problems using constrained zonotopes while preserving the uncertainty around the estimates. We start by discussing the protocol solving Problem 1.

5. Private Estimation Using Distributed Sensors

In this section, we introduce a protocol for estimating the set of possible states using zonotopes and constrained zonotopes while achieving our privacy goals. We first describe both protocols using a general set representation and then specify the required operations for zonotopes and constrained zonotopes. The query node $\mathbb{Q}$ generates the Paillier public key pk and private key sk and shares the public key with other parties. It then chooses a large enough initial set of possible states, enclosing the true state according to the public information. The initial set $\hat{S}_{q,0}$ is encrypted by the query node. We add the subscript q to the set notation to indicate that the set computation is done at the query node. The initial encrypted set $[\![\hat{S}_{q,0}]\!]$ is sent to the aggregator.

Our proposed privacy-preserving approach consists of three steps: the measurement update, the time update, and sharing of the results in a continuous loop, as presented in Protocol 1. More specifically, during the measurement update, the aggregator collects an encrypted strip $[\![\mathcal{P}_{i,k}]\!]$ from each sensor node i at step k, as shown in Figure 6. The family of encrypted strips (measurements) is intersected in the encrypted domain with the predicted reachable set at the aggregator (indicated by subscript a) – initially, it is the initial encrypted set $[\![\hat{S}_{q,0}]\!]$ sent by the query node – to obtain the encrypted corrected set $[\![\bar{\mathcal{S}}_{a,k}]\!]$, shown in Figure 6. Finally, the aggregator performs the time update and sends the encrypted corrected set $[\![\hat{\mathcal{S}}_{a,k}]\!]$, after decreasing its order, to the query node, which decrypts the result for each time step k.

We start by describing the required operations for the zonotopic case. The intersection between zonotopes and a family of

Protocol 1 Private Estimation using Distributed Sensors

The query node $\mathbb{Q}$ encrypts the initial set $[\![\hat{\mathcal{S}}_{q,0}]\!]$ and sends it to the aggregator node to have $[\![\hat{\mathcal{S}}_{a,0}]\!] = [\![\hat{\mathcal{S}}_{q,0}]\!]$. At every time instant k, every sensor node shares an encrypted strip $[\![\mathcal{P}_{i,k}]\!] = \langle [\![y_{i,k}]\!], H_k, r_{i,k}\rangle$ with the aggregator which executes the following steps:

Step 1: Measurement update at the aggregator:

$$[\![\bar{\mathcal{S}}_{a,k}]\!] = [\![\hat{\mathcal{S}}_{a,k-1}]\!] \cap [\![\mathcal{P}_{1,k}]\!] \cap \cdots \cap [\![\mathcal{P}_{m,k}]\!] \tag{12}$$

Step 2: Time update at the aggregator:

$$[\![\tilde{\mathcal{S}}_{a,k}]\!] = F[\![\bar{\mathcal{S}}_{a,k}]\!] \boxplus \mathcal{Q}_k \tag{13}$$

$$[\![\hat{\mathcal{S}}_{a,k}]\!] = \downarrow_q [\![\tilde{\mathcal{S}}_{a,k}]\!] \tag{14}$$

Step 3: The aggregator sends the encrypted set $[\![\hat{\mathcal{S}}_{a,k}]\!]$ to the query node which decrypts the result for each time step k.

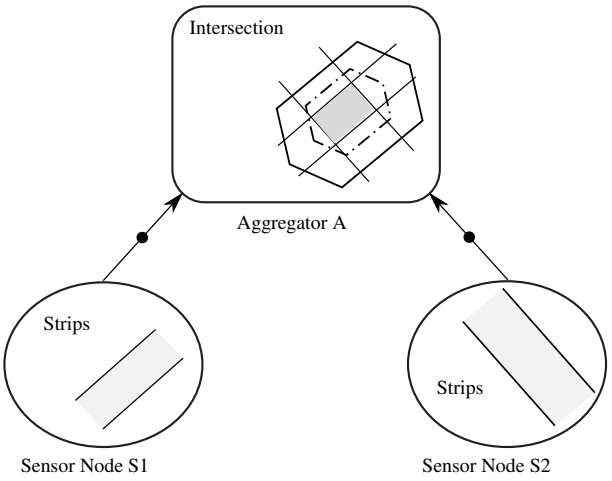


Figure 6: Overview of Protocol (1) where every sensor shares an encrypted strip with the aggregator, which then privately intersects these encrypted strips with an encrypted zonotope.

strips can be performed according to [49] and is summarized in the following lemma:

Lemma 1. ([49, Prop.1]) The intersection $\hat{\mathcal{Z}}_{k-1} \cap \mathcal{P}_{1,k} \cap \cdots \cap \mathcal{P}_{m,k}$ of a zonotope $\hat{\mathcal{Z}}_{k-1} = \langle \hat{c}_{k-1}, \hat{G}_{k-1}\rangle$ and the family of m strips $\mathcal{P}_{j,k} = \langle y_{j,k}, H_{j,k}, r_{j,k}\rangle$ in (10), $\forall j \in \mathcal{N}$, $|\mathcal{N}| = m$, is overapproximated by a zonotope $\bar{\mathcal{Z}}_k = \langle \bar{c}_k, \bar{G}_k\rangle$, where $\lambda_{j,k} \in \mathbb{R}^{n\times p}$ is the design parameter, and

$$\bar{c}_k = \hat{c}_{k-1} + \sum_{j\in\mathcal{N}} \lambda_{j,k}(y_{j,k} - H_{j,k}\hat{c}_{k-1}), \tag{15}$$

$$\bar{G}_k = \Big[(I - \sum_{j\in\mathcal{N}} \lambda_{j,k}H_{j,k})\hat{G}_{k-1}, \lambda_{1,k}r_{1,k}, \ldots, \lambda_{m,k}r_{m,k}\Big]. \tag{16}$$

The factor $\lambda_{j,k} \in \mathbb{R}^{n\times p}$ is a degree of freedom in Lemma 1 which we use to maximize the tightness of the intersection overapproximation. Thus, we want to find $\Lambda_k = \big[\lambda_{1,k}, \ldots, \lambda_{m,k}\big]$ that decreases the uncertainty around the estimates. We achieve this by computing the Λ_k that decreases the Frobenius norm of the generator matrix $\bar{G}_k$ in (16) [50].

During the time update step, the aggregator computes the time evolution of the estimated encrypted zonotope according to (13) and (14), i.e.,

$$\hat{c}_{a,k} = F\bar{c}_{a,k}, \tag{17}$$

$$\tilde{G}_{a,k} = [F\bar{G}_{a,k}, Q_k], \quad \hat{G}_{a,k} = \downarrow_q \tilde{G}_{a,k}. \tag{18}$$

Decreasing the order of the generator matrix, denoted by $\downarrow_q$, is done according to [51], which can be done over encrypted set as the generator matrix is revealed.

The generators do not participate in determining the position of the zonotope. Thus, it is sufficient to process over encrypted zonotope centers, as clarified in Section 4. Given the nature of the intersection between the strips and a zonotope in Lemma 1, the operations in the encrypted domain are decoupled from the plaintext domain computations. Note that the Paillier properties in (3) and (4) allow one to process (15) over the encrypted center $[\![\bar{c}_k]\!]$ and measurement $[\![y_{j,k}]\!]$ in Protocol 1. This protocol computes (15) in the encrypted domain and (16) in the unencrypted domain, where we also compute Λ_k. More specifically, we operate over encrypted centers and measurements as follows:

$$[\![\bar{c}_k]\!] = [\![\hat{c}_{k-1}]\!] \oplus \sum_{j\in\mathcal{N}} \lambda_{j,k}([\![y_{j,k}]\!] \ominus H_{j,k}[\![\hat{c}_{k-1}]\!]), \tag{19}$$

$$[\![\hat{c}_{a,k}]\!] = F[\![\bar{c}_{a,k}]\!]. \tag{20}$$

The generators are in the unencrypted domain in all our algorithms. The Minkowski sum in (13) is computed over encrypted centers and unencrypted generators.

The next theorem summarizes the privacy of Protocol 1 against different coalitions in Definitions 7, 9, and 8 when we use zonotopes and strips as sets.

Theorem 1. Protocol 1 solves Problem 1 using encrypted zonotopes while revealing the shape of the estimated zonotope and achieving privacy against

- sensor coalitions,
- cloud coalitions,
- query coalitions if $m_r p > n$, where m_r is the number of non-colluding sensors, p the measurement size, n the size of the state.

The proof is detailed in the Appendix. To overcome the information leakage in case of query coalitions when $m_r p \leq n$, we propose a slight modification by keeping the strip parameter $r_{i,k}$ private to the sensor and aggregator. Then, the aggregator swaps the columns of the generator matrix $\hat{G}_{a,k}$ before sending it to the query node. Swapping the generator columns produces the same estimated zonotope, but preserves privacy by preventing the coalition from computing Λ_k and thus also prevents the extraction of the center $[\![\hat{c}_{a,k}]\!]$.

Next, we present the required operations using constrained zonotopes. The following theorem shows the intersection in the unencrypted domain.

Lemma 2. The intersection $\hat{\mathcal{C}}_k \cap \mathcal{P}_{1,k} \cap \cdots \cap \mathcal{P}_{m,k}$ of a constrained zonotope $\hat{\mathcal{C}}_k = \langle \hat{c}_k, \hat{G}_k, \hat{A}_k, \hat{b}_k \rangle$ and the family of m strips $\mathcal{P}_{j,k} = \langle y_{i,k}, H_{i,k}, r_{i,k} \rangle$ in (10), $\forall j \in \mathcal{N}$, $|\mathcal{N}| = m$, is a constrained zonotope $\bar{\mathcal{C}}_k = \langle \bar{c}_k, \bar{G}_k, \bar{A}_k, \bar{b}_k \rangle$ where $\lambda_{j,k} \in \mathbb{R}^{n \times p}$ is a degree of freedom and

$$\bar{c}_k = \hat{c}_k + \sum_{j \in \mathcal{N}} \lambda_{j,k}(y_{j,k} - H_{j,k}\hat{c}_k), \tag{21}$$

$$\bar{G}_k = \Big[(I - \sum_{j \in \mathcal{N}} \lambda_{j,k} H_{j,k})\hat{G}_k, \lambda_{1,k} r_{1,k}, \ldots, \lambda_{m,k} r_{m,k}\Big], \tag{22}$$

$$\bar{A}_k = \begin{bmatrix} \hat{A}_k & 0 & 0 & \ldots & 0 \\ H_{1,k}\hat{G}_k & -r_{1,k} & 0 & \ldots & 0 \\ H_{2,k}\hat{G}_k & 0 & -r_{2,k} & \ldots & 0 \\ \vdots & \vdots & & \ddots & \\ H_{m,k}\hat{G}_k & 0 & 0 & \ldots & -r_{m,k} \end{bmatrix}, \tag{23}$$

$$\bar{b}_k = \begin{bmatrix} \hat{b}_k \\ y_{1,k} - H_{1,k}\hat{c}_k \\ y_{2,k} - H_2\hat{c}_k \\ \vdots \\ y_{m,k} - H_{m,k}\hat{c}_k \end{bmatrix}. \tag{24}$$

Proof. The results can be obtained by applying an intersection from [42, Prop. 1] and then adding a degree of freedom from [42, Prop. 5]. □

During the measurement update in Protocol 1 when using constrained zonotopes, the aggregator performs the proposed intersection over the encrypted center $[\![\hat{c}_{a,k-1}]\!]$ and the encrypted constraint shift $[\![\hat{b}_{a,k-1}]\!]$. Note that various combinations of zonotopes and constraints can represent the same constrained zonotope. Here, we exploit the additional degree of freedom in Λ_k and choose it at random in our protocol, which improves privacy, as discussed in the Appendix. Next, the aggregator propagates the sets forward in time according to (8), and then reduces the order of the given set [42], i.e.,

$$\hat{c}_{a,k} = F\bar{c}_{a,k}, \;\; \tilde{G}_{a,k} = [F\bar{G}_{a,k}, Q_k], \;\; \hat{b}_{a,k} = \bar{b}_{a,k}, \tag{25}$$

$$\{\hat{G}_{a,k}, \hat{A}_{a,k}\} = \downarrow_q \{\tilde{G}_{a,k}, \bar{A}_{a,k}\}. \tag{26}$$

The privacy of Protocol 1 against different coalitions in Definitions 7, 8, and 9 is summarized in the following theorem.

Theorem 2. Protocol 1 solves Problem 1 using encrypted constrained zonotopes while protecting the shape of the estimated set and achieves privacy against

- sensor coalitions,
- cloud coalitions,
- query coalitions.

The proof is detailed in the Appendix. After presenting our constrained zonotopic privacy-preserving protocol for Problem 1, we move on to the privacy-preserving protocol for Problem 2.

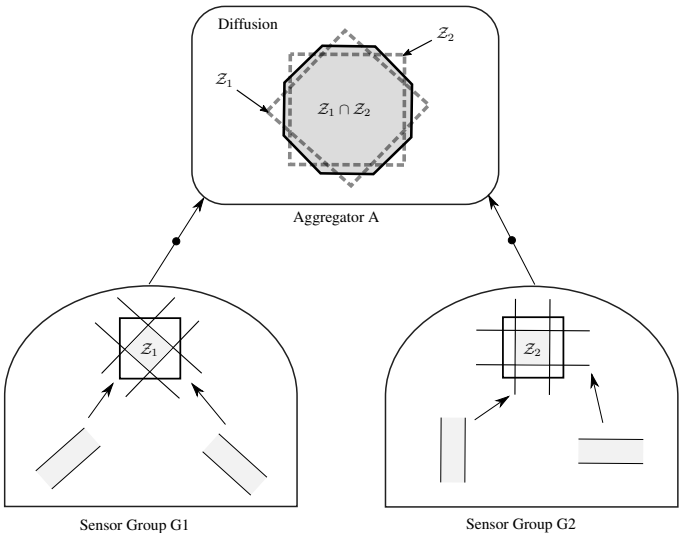


Figure 7: Overview of Protocol 2 where every sensor group computes the intersection between a zonotope and its strips. The aggregator then computes the intersection over the encrypted zonotopes.

6. Private Estimation Using Sensor Groups

We provide a privacy-preserving protocol for Problem 2 in Protocol 2, which is represented graphically in Figure 7 for the zonotopic case. Each sensor j within group i is participating with a strip (measurement) set $\mathcal{P}^{(i)}_{j,k}$ at each time step k. All strips are collected within the group i and are then intersected with the previously estimated set $\hat{\mathcal{S}}_{a,k-1}$ in (29) in the unencrypted domain, as the group participants trust each other and the plaintext execution is faster than the encrypted domain execution. The owner of each sensor group encrypts the resulting set $\bar{\mathcal{S}}^{(i)}_k$ and sends it to the aggregator, which in turn computes the intersection of all received encrypted sets in the encrypted domain in (30). Next, the aggregator performs the time update. Finally, the aggregator submits $[\![\hat{\mathcal{S}}_{a,k}]\!]$ to the query node, which decrypts the result and sends it to each sensor group.

We first describe the required operations for zonotopes and then proceed with constrained zonotopes. In the measurement update step, the intersection between a zonotope and a family of strips is performed as described in Lemma 1. Unlike Protocol 1, where we perform the aforementioned intersection at the aggregator in the encrypted domain, we now intersect at the sensor group level in the unencrypted domain since each sensor trusts all other sensors from the same group. Different methods exist in the literature for the zonotope intersection required during the diffusion update. Here, we picked our previously proposed intersection method described in [17], which fits the homomorphic computations, as summarized in the following lemma.

Lemma 3. ([17, Th.2]) The intersection $\bar{\mathcal{Z}}_{1,k} \cap \cdots \cap \bar{\mathcal{Z}}_{d,k}$ between d zonotopes $\bar{\mathcal{Z}}_{i,k} = \langle \bar{c}_{i,k}, \bar{G}_{i,k} \rangle$, $i \in \{1, \ldots, d\}$, can be over-approximated using the zonotope $\grave{\mathcal{Z}}_k = \langle \grave{c}_k, \grave{G}_k \rangle$ given by

$$\grave{c}_k = \frac{1}{\sum_{i=1}^{d} w_{i,k}} \sum_{i=1}^{d} w_{i,k}\bar{c}_{i,k}, \tag{27}$$

Protocol 2 Private Estimation using Sensor Groups

The query node $\mathbb{Q}$ sends the initial set to each sensor group $i \in \{1, ..., d\}$, and the aggregator node $\mathbb{A}$. For each sensor group i, m_i strips $[\![\mathcal{P}^{(i)}_{j,k}]\!] = \langle [\![y^{(i)}_{j,k}]\!], H_k, r^{(i)}_{j,k} \rangle$, $j \in \left\{1, ..., m_i\right\}$, are available. At every time instant k, the following steps are executed:

Step 1: Measurement update at each sensor group i:

$$\bar{\mathcal{S}}^{(i)}_k = \hat{\mathcal{S}}_{a,k-1} \cap \mathcal{P}^{(i)}_{1,k} \cap \cdots \cap \mathcal{P}^{(i)}_{m_i,k} \tag{29}$$

Step 2: Diffusion update at the aggregator:

$$[\![\grave{\mathcal{S}}_{a,k}]\!] = [\![\bar{\mathcal{S}}^{(1)}_k]\!] \cap \cdots \cap [\![\bar{\mathcal{S}}^{(d)}_k]\!] \tag{30}$$

Step 3: Time update at the aggregator:

$$[\![\tilde{\mathcal{S}}_{a,k}]\!] = F[\![\grave{\mathcal{S}}_{a,k}]\!] \boxplus \mathcal{Q}_k \tag{31}$$

$$[\![\hat{\mathcal{S}}_{a,k}]\!] = \downarrow_q [\![\tilde{\mathcal{S}}_{a,k}]\!] \tag{32}$$

Step 4: The aggregator sends the encrypted set $[\![\hat{\mathcal{S}}_{a,k}]\!]$ to the query node which decrypts and sends the results to the sensor groups.

$$\grave{G}_k = \frac{1}{\sum_{i=1}^{d} w_{i,k}} [w_{1,k}\bar{G}_{1,k}, ..., w_{d,k}\bar{G}_{d,k}], \tag{28}$$

where the weights $w_{i,k}$ are chosen such that $\sum_{i=1}^{d} w_{i,k} \neq 0$.

Let $\mathfrak{w}_k = [w_{1,k}, \ldots, w_{d,k}]$, where d is the number of sensor groups. Ideally, $\mathfrak{w}_k$ is chosen such that the size of the zonotope $\grave{\mathcal{Z}}_k = \langle \grave{c}_k, \grave{G}_k \rangle$ is minimized. The size of the zonotope appears in the unencrypted generator matrix due to the selective encryption, and can be replaced by the Frobenius norm of the generator matrix. The next theorem summarizes the privacy features of the protocol against different coalitions in Definitions 7, 9, and 8.

Theorem 3. Protocol 2 solves Problem 2 using encrypted zonotopes while revealing the shape of the estimated set and achieving privacy against

- sensor coalitions,
- cloud coalitions,
- query coalitions if ($d_r > 1$), where d_r is the number of non-colluding sensor groups.

The proof is detailed in the Appendix. In order to solve Problem 2 without revealing the shape of the estimated set as in Theorem 3, we again use constrained zonotopes. The intersection between the constrained zonotopes and strips during the measurement update of Protocol 1 is done according to Lemma 2 in the unencrypted domain. Then, the intersection between the constrained zonotopes is performed, which we describe next.

Lemma 4. The intersection $\bar{\mathcal{C}}_{1,k} \cap \cdots \cap \bar{\mathcal{C}}_{d,k}$ between d constrained zonotopes $\bar{\mathcal{C}}_{j,k} = \left\langle \bar{c}_{j,k}, \bar{G}_{j,k}, \bar{A}_{j,k}, \bar{b}_{j,k} \right\rangle$ is a constrained zonotope $\grave{\mathcal{Z}}_k = \langle \grave{c}_k, \grave{G}_k, \grave{A}_k, \grave{b}_k \rangle$, where

$$\grave{c}_k = \bar{c}_{1,k}, \quad \grave{G}_k = \left[\bar{G}_{1,k}, 0, \ldots, 0\right], \tag{33}$$

$$\grave{A}_k = \begin{bmatrix} \bar{A}_{1,k} & 0 & \ldots & 0 \\ 0 & \bar{A}_{2,k} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \bar{A}_{d,k} \\ \bar{G}_{1,k} & -\bar{G}_{2,k} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{G}_{1,k} & 0 & \ldots & -\bar{G}_{d,k} \end{bmatrix}, \quad \grave{b}_k = \begin{bmatrix} \bar{b}_{1,k} \\ \bar{b}_{2,k} \\ \vdots \\ \bar{b}_{d,k} \\ \bar{c}_{2,k} - \bar{c}_{1,k} \\ \vdots \\ \bar{c}_{d,k} - \bar{c}_{1,k} \end{bmatrix}. \tag{34}$$

Proof. The lemma is the multi-strip intersection of [42, Prop. 1]. □

Unlike for zonotopes, Lemma 4 computes the intersection exactly. Then, the time update is done according to (25) and (26). The privacy of the protocol against different coalitions in Definitions 7, 9, and 8 is summarized in the following theorem.

Theorem 4. Protocol 2 solves Problem 2 using encrypted constrained zonotopes while protecting the shape of the estimated set and achieves privacy against

- sensor coalitions,
- cloud coalitions,
- query coalitions.

The proof is detailed in the Appendix. In the next section, we will evaluate the presented protocols.

7. Evaluation

In this section, we evaluate the proposed protocols using data from a real-world testbed. We first describe our testbed in detail and then evaluate the proposed protocols. The protocols are evaluated on a custom ultra-wideband (UWB) RF testbed based on the DecaWave DW1000 IR-UWB radio[2]. The overall setup is the same as in [52]. The main components of the considered testbed can be summarized as follows:

1. The motion capture system consists of eight cameras capable of performing 3D rigid body position measurements with an accuracy of less than 0.5 mm.
2. The fixed nodes each consist of a custom-built circuit board equipped with a ARM Cortex M4 processor with 196 MHz (Figure 8), powered over Ethernet and communicating via a Decawave DW1000 ultra-wideband radio (Figure 9).
3. The battery-powered mobile node is a modified CrazyFlie 2.0 helicopter[3] (Figure 10) and is equipped with the same DW1000 radio and ARM Cortex M4 processor.

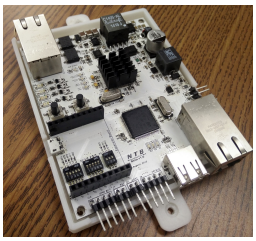


Figure 8: Custom anchor with ARM Cortex M4 processor and UWB slot.

Figure 9: Ceiling-mounted anchor with UWB radio in 3D-printed enclosure.

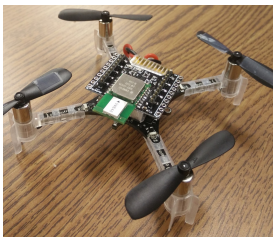


Figure 10: CrazyFlie 2.0 quadrotor helicopter with UWB expansion.

For the sake of a fair evaluation between the four variants of our two protocols, we used a collected data from the testbed and ran the four variants on the same set of measurements. We aim to estimate the set that encloses the location of the quadrotor while preserving our aforementioned privacy goals. We start with an initial set of size (8×8 m^2) covering the whole localization area at the initial point (time step $k = 0$). This set is then iteratively shrunk by using the received measurements and performing geometric intersections to correct the estimated set. Figure 11 shows the true values, upper bounds, and lower bounds of the three-dimensional estimated location of the four variants of Protocol 1. We omit the results of Protocol 2 as they are close to the results of Protocol 1. The upper bounds and lower bounds are obtained by converting the zonotopes and constrained zonotopes to intervals. It is worth mentioning that the result using the zonotopic case of Protocol 1 is tighter than the result using the zonotopic case of Protocol 2. This is because Protocol 2 requires two over-approximations, namely, the intersection between every zonotope and the family of strips and the intersection of the family of zonotopes.

We consider the center of the estimated set to be the single-point estimate in the zonotopic case. Thus, we report the localization error with respect to the center of the zonotope. For constrained zonotopes, the reported center in the representation is the center of the original zonotope without constraints and hence can be outside of the constrained zonotope. Therefore, we compute the Chebychev center of the polytope in the constrained zonotope factor space [53]. The estimation error of the four variants is presented in Figure 12.

There is a trade-off between the provided privacy, the computation overhead, and the exactness of set operations. Constrained zonotopes provide more privacy, more computation overhead, and less conservatism due to the exact set operations. On the other hand, zonotopes provide less privacy due to revealing the shape of the sets, less computation overhead, and more conservative sets. The trade-off between the provided privacy and the execution time is presented in Table 1. Keeping the shape of the estimated set private by using constrained zonotopes instead of zonotopes increases the required execution time. All computations were run on a single thread of an Intel(R) Core(TM) i7-8750 with 16 GB RAM with 1024 key size. The comparison between the size of the reachable sets appears in Figure 11.

[2]Decawave DW1000: `http://www.decawave.com/products/dw1000`
[3]Bitcraze CrazyFlie 2.0: `https://www.bitcraze.io/`

8. Conclusions

We proposed the first privacy-preserving, set-based observers using homomorphic encryption. We considered both a traditional sensor setup as well as a scenario where trusting sensors are grouped into sensor groups. We showed that by choosing zonotopes and constrained zonotopes to represent our sets, it is possible to selectively encrypt only the critical set parameters while achieving the desired level of privacy. To prove that privacy for each protocol, the concept of computational indistinguishability was used. Finally, we evaluated our algorithms on real data from a physical test bed, which showed that the proposed protocols achieve satisfactory results while guaranteeing privacy.

One main drawback of guaranteeing privacy using homomorphic encryption is the overflow problem after a sequence of operations in the encrypted domain. To overcome the overflow limitation, we send the encrypted set to the query node each time, which decrypts the estimated set and sends the re-encrypted set back to the aggregator. This solves the overflow problem at the cost of computation and communication overhead. However, after the encrypted estimated set is sent from the aggregator to the query node, decrypting said set is not regarded as overhead since the query node is interested in the estimated set after each time step by assumption, and thus decryption is required anyway. That said, solving the overflow problem in a more efficient way is an open research problem that we leave for future work.

Acknowledgements

We gratefully acknowledge partial financial support by the project justITSELF funded by the European Research Council (ERC) under grant agreement No 817629, the project interACT under grant agreement No 723395, and the CONCORDIA cyber security project No. 830927; these projects are funded within the EU Horizon 2020 program. This work is also supported by the Knut and Alice Wallenberg Foundation, the Swedish Foundation for Strategic Research and the Swedish Research Council.

Appendix A. Theorems' Proofs

We need to show in the following proofs that the views and simulators of the coalitions are computationally indistinguishable and that the input and output of the coalition do not leak

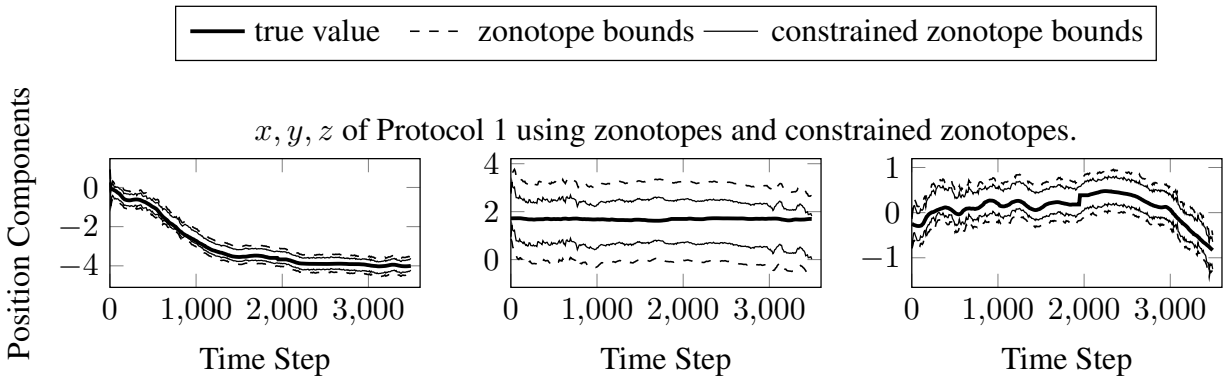


Figure 11: Ground truth, upper and lower bounds on the position components of the three-dimensional estimated states in meters of Protocol 1.

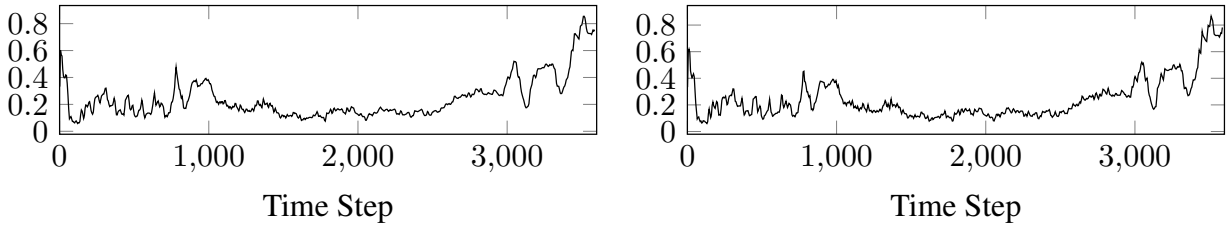


Figure 12: Estimation error for Protocol 1 using zonotopes and constrained zonotopes.

extra private information according to Definition 5. This will be done for each type of coalition. Showing the computational indistinguishability is done by building the views of each coalition then proving that there is an equivalent simulator that could be obtained from the input and output available to the coalition.

Remark 1. The view V and simulator S are computationally indistinguishable $V \overset{c}{\equiv} S$, if they have the same list of values or have values that are generated according to the same distribution and independent from other parameters [44].

We denote the quantities obtained by the simulator by $\widetilde{()}$, which follow the same distribution but are otherwise different from the quantities of the views. We may omit the time index k from the views and simulators for simplicity. Here, the coins are random numbers that are used for the encryption process and key generation. Further information that is exchanged between other parties over an encrypted channel is denoted by $[\![\Gamma_X]\!]$ for coalition X. Note that the encrypted channel uses extra keys different from the homomorphic encryption keys and uses double encryption to protect privacy.

Appendix A.1. Proof of Theorem 1

The proof consists of three types of coalitions as described bellow.

Appendix A.1.1. Coalition of sensors s

The strip information is considered as the input to the sensor and appears as part of the view and the simulator of the coalition. We denote the view of coalition s consisting of the set of sensors $s = \{s_1, \ldots, s_t\}$ by V_s^{Π}, which is defined as the combination of every sensor view and given by

$$V_s^{\Pi} = (V_{s_1}^{\Pi}, \ldots, V_{s_t}^{\Pi}) = (H_{s,k}, y_{s,k}, r_{s,k}, [\![y_{s,k}]\!], \text{coins}_s, pk, [\![\Gamma_s]\!]), \tag{A.1}$$

where the subscript s on $H_{s,k}, y_{s,k}, r_{s,k}$ denotes an array of strip information of the coalition. The sensors only submit their encrypted data to the aggregator. Hence, a simulator, denoted by S_s, consists of the input and output and by generating $\widetilde{[\![\Gamma_s]\!]}$, $\widetilde{[\![y_{s,k}]\!]}$ and $\widetilde{\text{coins}}_s$, i.e.

$$S_s = (pk, H_{s,k}, y_{s,k}, r_{s,k}, \widetilde{[\![y_{s,k}]\!]}, \widetilde{\text{coins}}_s, \widetilde{[\![\Gamma_s]\!]}). \tag{A.2}$$

The $\widetilde{\text{coins}}_s$ are generated according to the same distribution of coins_s and are independent from other parameters, where the same is true for $\widetilde{[\![\Gamma_s]\!]}$ and $[\![\Gamma_s]\!]$ as well as $\widetilde{[\![y_{s,k}]\!]}$ and $[\![y_{s,k}]\!]$. Therefore, we conclude that $S_s \overset{c}{\equiv} V_s^{\Pi}$.

Moreover, the information contained in each strip is independent from all others. Thus, the coalition strips cannot be used to infer new information about other strips. The information in each iteration is different from other iterations. That is why we considered only a single step in the previous proof. In the following two subsections, we will prove that the view of each coalition after $K \in \mathbb{N}^+$ iterations of the protocol is computationally indistinguishable from the view of a simulator that executes K iterations.

Appendix A.1.2. Coalition of sensors s and aggregator

The view of the aggregator is denoted by V_a^{Π}. We denote the view of a coalition consisting of a set of sensors by $s = \{s_1, \ldots, s_t\}$ and the aggregator by V_{sa}^{Π} which is defined by

$$V_{sa}^{\Pi} = (V_s^{\Pi}, V_a^{\Pi}) = (V_s^{\Pi,K}, V_a^{\Pi,K}), \tag{A.3}$$

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

Table 1: Execution Time in ms.

	Entities		
	Sensor/Sensor group	Aggregator	Query
Protocol 1 using zonotopes	2.371	3.195	0.550
Protocol 1 using constrained zonotopes	2.371	6.632	14.529
Protocol 2 using zonotopes	8.389	5.968	0.550
Protocol 2 using constrained zonotopes	81.426	9.787	14.529

are given by

$$V_s^{\Pi,k+1} = (V_s^{\Pi,k}, I_s^{k+1}), \quad V_a^{\Pi,k+1} = (V_a^{\Pi,k}, I_a^{k+1}) \tag{A.4}$$

$\forall k = 0, 1, \ldots, K-1$, where I_s^k and I_a^k are the newly added data points at the k-th iteration for the coalition of sensors $V_s^{\Pi,0} = I_s^0$ and aggregator $V_a^{\Pi,0} = I_a^0$. The view of the aggregator contains encrypted strips $\langle H_{s,k}, [\![y_{s,k}]\!], r_{s,k}\rangle$ from the sensors, initial set $\langle [\![\hat{c}_{q,0}]\!], \hat{G}_{q,0}\rangle$ from the query node, and the estimated set $\langle [\![\hat{c}_{a,k}]\!], \hat{G}_{a,k}\rangle$ at k-th iteration. Let us denote the strip information of the sensors at the k-th iteration, which are not part of the coalition by subscript r, i.e., $\langle H_{r,k}, [\![y_{r,k}]\!], r_{r,k}\rangle$, $k = 0, 1, \ldots, K-1$. Then, I_a^k and I_s^k are

$$\begin{aligned} I_a^k = (&H_{s,k}, [\![y_{s,k}]\!], r_{s,k}, H_{r,k}, [\![y_{r,k}]\!], r_{r,k}, [\![\hat{c}_{q,0}]\!], \\ &\hat{G}_{q,0}, [\![\hat{c}_{a,k}]\!], \hat{G}_{a,k}, \mathrm{coins}_a, pk, q, F, Q_k), \end{aligned} \tag{A.5}$$

$$I_s^k = (H_{s,k}, y_{s,k}, r_{s,k}, [\![y_{s,k}]\!], \mathrm{coins}_s, pk, [\![\Gamma_s]\!]), \tag{A.6}$$

where $\mathcal{Z}_{q,0} = \langle \hat{c}_{q,0}, \hat{G}_{q,0}\rangle$ is the initial zonotope at the query node and $\mathcal{Z}_{a,k} = \langle \hat{c}_{a,k}, \hat{G}_{a,k}\rangle$ is the estimated zonotope on the aggregator side at time step k. The view of the coalition V_{sa}^{Π} is constructed from (A.3)–(A.6). Let the simulator of the coalition 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 iteratively constructed by combining the values obtained at each time step k as follows:

$$S_{sa}^{k+1} = (S_{sa}^k, I_{sa}^{S,k+1}), \quad k = 0, 1, \ldots, K-1, \tag{A.7}$$

where $I^{S,k+1}$ is the portion of the simulator generated at iteration $k+1$, which is given by

$$\begin{aligned} I_{sa}^{S,k} = (&H_{s,k}, \widetilde{[\![y_{s,k}]\!]}, r_{s,k}, H_{r,k}, \widetilde{[\![y_{r,k}]\!]}, r_{r,k}, \widetilde{[\![\hat{c}_{q,0}]\!]}, \hat{G}_{q,0}, \\ &\widetilde{[\![\hat{c}_{a,k}]\!]}, \hat{G}_{a,k}, \widetilde{\mathrm{coins}}_{sa}, q, F, Q_k, y_{s,k}, pk, \widetilde{[\![\Gamma_s]\!]}) \end{aligned}$$

and where the values are computed or generated as follows:

1. Generate $\widetilde{[\![\Gamma_s]\!]}$, $\widetilde{[\![\bar{c}_{q,0}]\!]}$, $\widetilde{[\![\hat{c}_{a,k}]\!]}$, $\widetilde{[\![y_{r,k}]\!]}$, and $\widetilde{[\![y_{s,k}]\!]}$ according to the same distribution of $[\![\Gamma_s]\!]$, $[\![\bar{c}_{q,0}]\!]$, $[\![\hat{c}_{a,k}]\!]$, $[\![y_{r,k}]\!]$ and $[\![y_{s,k}]\!]$, respectively.
2. Compute $\hat{G}_{a,k}$ according to (16).
3. Let the combination of all coins of the parties be $\mathrm{coins}_{sa} = (\mathrm{coins}_a, \mathrm{coins}_s)$. Generate $\widetilde{\mathrm{coins}}_{sa}$ according to the distribution of coins_{sa}.

Based on this generation scheme, the values $\widetilde{[\![\,]\!]}$ and $[\![\,]\!]$ are indistinguishable and all remaining variables in $I_{sa}^{S,k+1}$ are either public or feasible through the protocol steps. After all iteration steps, we end up with a simulator that satisfies $S_{sa} \overset{c}{\equiv} V_{sa}^{\Pi}$.

The second part of the proof is about inferring extra private information from the input and output. The coalition's target is to determine the private measurement of the remaining sensors $y_{r,k}$. Note that the relation between $[\![y_{s,k}]\!]$ and $[\![y_{r,k}]\!]$ is characterized by (A.8).

$$\begin{aligned} \sum_{j\in\mathcal{N}_r} \lambda_{j,k}[\![y_{j,k}]\!] = &\sum_{j\in\mathcal{N}} (\lambda_{j,k}H_{j,k} - 1)[\![\hat{c}_{a,k-1}]\!] \oplus [\![\bar{c}_{a,k}]\!] \\ &\ominus \underbrace{\sum_{j\in\mathcal{N}/r} \lambda_{j,k}[\![y_{j,k}]\!]}_{\text{known to the coalition in plaintext}}, \end{aligned} \tag{A.8}$$

where $\mathcal{N}_r$ is the set of the remaining sensors. Since the coalition does not have the private key and the query node sends the initial encrypted center $[\![\hat{c}_{a,0}]\!]$, we end up with an underdetermined system in (A.8).

Appendix A.1.3. Coalition of sensors s and query node

We denote the view of a coalition consisting of a set of sensors by $s = \{s_1, \ldots, s_t\}$ and define the query as

$$V_{sq}^{\Pi} = (V_s^{\Pi}, V_q^{\Pi}) = (V_s^{\Pi,K}, V_q^{\Pi,K}), \tag{A.9}$$

where

$$V_s^{\Pi,k+1} = (V_s^{\Pi,k}, I_s^{k+1}), \quad V_q^{\Pi,k+1} = (V_q^{\Pi,k}, I_q^{k+1}), \tag{A.10}$$

$\forall k = 0, 1, \ldots, K-1$, where I_s^k is given in (A.6), and I_q^k are the newly added data points from the k-th iteration for the query node with $V_q^{\Pi,0} = I_q^0$ such that

$$I_q^k = (\hat{c}_{q,0}, \hat{G}_{q,0}, \hat{c}_{a,k}, \hat{G}_{a,k}, [\![\hat{c}_{a,k}]\!], \mathrm{coins}_q, [\![\Gamma_{sq}]\!], pk, sk). \tag{A.11}$$

The view of the coalition V_{sq}^{Π} is constructed from (A.6), (A.10) and (A.11). The construction of the simulator is similar to Section Appendix A.1.2. Thus, we focus on the values added in the k-th iteration to the simulator. Let the combination of all coins of the parties be denoted by $\mathrm{coins}_{sq} = (\mathrm{coins}_q, \mathrm{coins}_s)$. The inputs and outputs to the coalition are $(pk, sk, H_{s,k}, y_{s,k}, r_{s,k}, \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{c}_{q,0}, \hat{G}_{q,0})$. Thus, the simulator S_{sq}^k can be easily generated by

$$S_{sq}^k = (pk, sk, H_{s,k}, y_{s,k}, r_{s,k}, \hat{c}_{q,0}, \hat{G}_{q,0}, \hat{c}_{a,k}, \hat{G}_{a,k}, \widetilde{[\![\hat{c}_{a,k}]\!]},$$

$$\widetilde{\text{coins}}_{sq}, \widetilde{[\![\Gamma_{sq}]\!]}, S_{sq}^{k-1}). \tag{A.12}$$

The tuples $\left([\![\hat{c}_{a,k}]\!], \text{coins}_{sq}, [\![\Gamma_{sq}]\!]\right)$ and $\left(\widetilde{[\![\hat{c}_{a,k}]\!]}, \widetilde{\text{coins}}_{sq}, \widetilde{[\![\Gamma_{sq}]\!]}\right)$ are generated according to the same distribution and are independent from other parameters. Therefore, $S_{sq}^{k} \overset{c}{\equiv} (I_s^k, I_q^k)$, which leads to

$$S_{sq} \overset{c}{\equiv} V_{sq}^{\Pi}. \tag{A.13}$$

In case of a coalition between s sensors and the query node, the aim would be to find the measurements of the remaining group, denoted by N_r with size m_r. Rewriting (A.8) after decryption – as the query has the Paillier private key sk – results in

$$\Lambda_{r,k} Y_{r,k} = z_{s,k}, \tag{A.14}$$

where

$$\begin{aligned} z_{s,k} &= \sum_{j\in\mathcal{N}} (\lambda_{j,k} H_{j,k}) \hat{c}_{a,k-1} + \bar{c}_{a,k} - \sum_{j\in\mathcal{N}/r} \lambda_{j,k} y_{j,k}, \\ \Lambda_{r,k} &= [\lambda_{j_1,k}, \lambda_{j_2,k}, \ldots, \lambda_{j_{m_r},k}] \in \mathbb{R}^{n\times pm_r}, \\ Y_{r,k} &= [y_{j_1,k}^T, y_{j_2,k}^T, \ldots, y_{j_{m_r},k}^T]^T \in \mathbb{R}^{pm_r}, \end{aligned}$$

where $z_{s,k}$ is known to the coalition given that $\lambda_{j,k}$ is computed based on the generator matrix. To find the conditions at which the privacy of $Y_{r,k}$ is ensured, we show that there is no unique retrieval for $Y_{r,k}$. This non-unique retrieval requires that (A.14) has multiple solutions. According to [54, Theorem 6.4], $\tilde{Y}_{r,k}$ is a solution of (A.14) for any $X_r \in \mathbb{R}^{pm_r}$ with

$$\tilde{Y}_{r,k} = \Lambda_{r,k}^{-} z_{s,k} + (I_{pm_r} - \Lambda_{r,k}^{-} \Lambda_{r,k}) X_r, \tag{A.15}$$

where $\Lambda_{r,k}^{-}$ is any generalized inverse of $\Lambda_{r,k}$. For every solution $\tilde{Y}_{r,k}$ of (A.14) there is an X_r. For $I_{pm_r} - \Lambda_{r,k}^{-}\Lambda_{r,k} = 0$, the system is consistent and thus has one solution [54, Theorem 6.1]. Therefore, we aim to find conditions at which $I_{pm_r} - \Lambda_{r,k}^{-}\Lambda_{r,k} \neq 0$ to ensure privacy. We have $rank(\Lambda_{r,k}^{-}\Lambda_{r,k}) \leq \min\{pm_r, n\}$ according to [54, Theorem 2.8]. Thus, under the condition $pm_r > n$, we have $I_{pm_r} - \Lambda_{r,k}^{-}\Lambda_{r,k} \neq 0$ which ensures the privacy of $Y_{r,k}$. □

Appendix A.2. Proof of Theorem 2

In the following proof, we consider the view and simulation for one step (i.e., k-th step) for notational convenience. The proof for $K \in \mathbb{N}^+$ steps is similar to the proof of Theorem 1. We are going to prove the privacy against the three coalitions as follows:

Appendix A.2.1. Coalition of sensors s

The strips information is considered to be an input to the sensor and appears as part of the view and the simulator of the coalition. The strips information is exactly the same as for zonotopes. Thus, the proof is similar to section Appendix A.1.1 and is therefore omitted.

Appendix A.2.2. Coalition of sensors s and aggregator

The aggregator has encrypted strips $(H_{s,k}, [\![y_{s,k}]\!], R_{s,k}, H_{r,k}, [\![y_{r,k}]\!], R_{r,k})$ from the sensors, the initial constrained zonotope $\langle [\![\hat{c}_{q,0}]\!], \hat{G}_{q,0}, [\![\hat{b}_{q,0}]\!], \hat{A}_{q,0}\rangle$ from the query node, and estimated constrained zonotope $\langle [\![\hat{c}_{g,k}]\!], \hat{G}_{g,k}, [\![\hat{b}_{g,k}]\!], \hat{A}_{g,k}\rangle$ at each k-iteration. The view of the coalition is defined as

$$\begin{aligned} V_{sa}^{\Pi} &= (V_s^{\Pi}, V_a^{\Pi}) \\ &= (V_s^{\Pi}, H_{s,k}, [\![y_{s,k}]\!], R_{s,k}, H_{r,k}, [\![y_{r,k}]\!], R_{r,k}, [\![\hat{c}_{q,0}]\!], \hat{G}_{q,0}, [\![\hat{b}_{q,0}]\!], \\ &\quad \hat{A}_{q,0}, [\![\hat{c}_{g,k}]\!], \hat{G}_{g,k}, [\![\hat{b}_{g,k}]\!], \hat{A}_{g,k}, \text{coins}_{sa}, pk, q, F, Q_k) \\ &\overset{(A.1)}{=} (H_{s,k}, y_{s,k}, R_{s,k}, [\![y_{s,k}]\!], H_{r,k}, [\![y_{r,k}]\!], R_{r,k}, [\![\hat{c}_{q,0}]\!], \hat{G}_{q,0}, [\![\hat{b}_{q,0}]\!], \\ &\quad \hat{A}_{q,0}, [\![\hat{c}_{g,k}]\!], \hat{G}_{g,k}, [\![\hat{b}_{g,k}]\!], \hat{A}_{g,k}, \text{coins}_{sa}, pk, q, F, Q_k). \end{aligned} \tag{A.16}$$

The simulation is the same as in Section Appendix A.1.2, except for the additional information contained in a constrained zonotope, i.e., the constraints. This results in

$$\begin{aligned} S_{sa} = (&H_{s,k}, y_{s,k}, R_{s,k}, \widetilde{[\![y_{s,k}]\!]}, H_{r,k}, \widetilde{[\![y_{r,k}]\!]}, R_{r,k}, \widetilde{[\![\hat{c}_{q,0}]\!]}, \hat{G}_{q,0}, \\ &\widetilde{[\![\hat{b}_{q,0}]\!]}, \hat{A}_{q,0}, \widetilde{[\![\hat{c}_{g,k}]\!]}, \hat{G}_{g,k}, \widetilde{[\![\hat{b}_{g,k}]\!]}, \hat{A}_{g,k}, \widetilde{\text{coins}}_{sa}, pk, q, F \\ &, Q_k). \end{aligned} \tag{A.17}$$

We arrive at a simulator that satisfies $S_{sa} \overset{c}{\equiv} V_{sa}^{\Pi}$. Thus, similarly to Section Appendix A.1.2, the coalition cannot infer extra information from the input and the output.

Appendix A.2.3. Coalition of sensors s and query node

The view of the coalition consists of the view of the sensors V_s^{Π} and the view of the query node V_q^{Π} which consist of the initial estimated constrained zonotope $\langle \hat{c}_{q,0}, \hat{G}_{q,0}, \hat{A}_{q,0}, \hat{b}_{q,0}\rangle$ and resultant estimated set $\langle \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}\rangle$ at each k-iteration as follows:

$$\begin{aligned} V_{sq}^{\Pi} &= (V_s^{\Pi}, V_q^{\Pi}) \\ &= (V_s^{\Pi}, \hat{c}_{q,0}, \hat{G}_{q,0}, \hat{A}_{q,0}, \hat{b}_{q,0}, \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, \text{coins}_s, [\![\Gamma_s]\!], \\ &\quad pk, sk) \\ &\overset{(A.1)}{=} (H_{s,k}, y_{s,k}, R_{s,k}, [\![y_{s,k}]\!], H_{r,k}, [\![y_{r,k}]\!], R_{r,k}, \hat{c}_{q,0}, \hat{G}_{q,0}, \hat{A}_{q,0}, \hat{b}_{q,0}, \\ &\quad \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, \text{coins}_{sq}, [\![\Gamma_{sq}]\!], pk, sk). \end{aligned} \tag{A.18}$$

The simulator will be again similar to Section Appendix A.1.3 after adding and generating the constrained zonotope information, specifically

$$\begin{aligned} S_{sq} = (&H_{s,k}, y_{s,k}, R_{s,k}, \widetilde{[\![y_{s,k}]\!]}, H_{r,k}, \widetilde{[\![y_{r,k}]\!]}, R_{r,k}, \hat{c}_{q,0}, \hat{G}_{q,0}, \hat{A}_{q,0}, \hat{b}_{q,0}, \\ &\hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, \widetilde{\text{coins}}_{sq}, \widetilde{[\![\Gamma_{sq}]\!]}, pk, sk). \end{aligned} \tag{A.19}$$

As before, the generated values $\widetilde{[\![y_{s,k}]\!]}$, $\widetilde{[\![y_{r,k}]\!]}$, $\widetilde{\text{coins}}_{sq}$ and $\widetilde{[\![\Gamma_{sq}]\!]}$ are generated according to the distribution of the original values and are independent from other parameters. Therefore $S_{sq} \overset{c}{\equiv} V_{sq}^{\Pi}$.

The coalition aims to find the measurements of the remaining group, denoted by N_r with size m_r. Note that $\Lambda_{a,k}$ is chosen

by the aggregator and not known to the query; additionally, it is also chosen at random and not dependant on publicly shared generator matrix. Thus, computing the measurement $y_{r,k}$ according to (A.14) is not valid anymore. In contrast to Theorem 1, privacy can be guaranteed in all cases. □

Appendix A.3. Proof of Theorem 3

In the following proof, we consider the view and simulation for one step (i.e., k-th step) for notational convenience. The proof for $K \in \mathbb{N}^+$ steps is similar to the proof of Theorem 1. We prove again the privacy against the following three coalitions:

Appendix A.3.1. Coalition of sensors groups g:

We define the view of a coalition consisting of a of set of sensor groups $g = \{g_1, \dots, g_t\}$ by V_g^Π by

$$\begin{aligned} V_g^\Pi &= (V_{g_1}^\Pi, \dots, V_{g_t}^\Pi) \\ &= (pk, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \bar{c}_{g,k}, [\![\bar{c}_{g,k}]\!], [\![\Gamma_g]\!], coin_g), \end{aligned} \tag{A.20}$$

where the subscript g denotes the variables owned by the coalition. The sensor groups only submit their encrypted data to the aggregator. Hence, a simulator S_g, defined by

$$S_g = (pk, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \bar{c}_{g,k}, \widetilde{[\![\bar{c}_{g,k}]\!]}, \widetilde{[\![\Gamma_g]\!]}, \widetilde{coin}_g), \tag{A.21}$$

is obtained by generating $\widetilde{[\![\bar{c}_{g,k}]\!]}$, $\widetilde{[\![\Gamma_g]\!]}$ and $\widetilde{\text{coins}}_g$ according to the distribution of $([\![\bar{c}_{g,k}]\!], [\![\Gamma_g]\!], coin_g)$ and are independent from other parameters. Therefore, we conclude that $S_g \overset{c}{\equiv} V_g^\Pi$.

Moreover, the resulting zonotopes from the sensor groups are independent. As a result, the coalition zonotopes cannot be used to infer new information about other zonotopes.

Appendix A.3.2. Coalition of sensor groups g and the aggregator:

The view of the coalition is defined by

$$V_{ga}^\Pi = (V_g^\Pi, V_a^\Pi) \tag{A.22}$$

with

$$V_a^\Pi = ([\![\bar{c}_{g,k}]\!], \bar{G}_{g,k}, [\![\bar{c}_{r,k}]\!], \bar{G}_{r,k}, [\![\hat{c}_{a,k}]\!], \hat{G}_{a,k}, q, F, Q_k, \text{coins}_a, pk) \tag{A.23}$$

where $[\![\bar{c}_{r,k}]\!]$ and $\bar{G}_{r,k}$ represents the encrypted center and the generators of the remaining sensor groups which are not part of the coalition. The simulator, denoted by S_{ga}, can be constructed from the input and output $(H_{g,k}, R_{g,k}, F, pk, q, Q_k, y_{g,k})$. Specifically:

1. Add $H_{r,k}$ and $R_{r,k}$ as they are public information.
2. Compute $\bar{G}_{g,k}$ and $\bar{G}_{r,k}$ according to (16).
3. Compute $\hat{G}_{a,k}$ according to (18).
4. Generate $\widetilde{[\![\bar{c}_{g,k}]\!]}$, $\widetilde{[\![\bar{c}_{r,k}]\!]}$, $\widetilde{[\![\Gamma_g]\!]}$, and $\widetilde{[\![\hat{c}_{a,k}]\!]}$ according to the distributions of the original values.
5. Let the combination of coins of all parties be $\text{coins}_{ga} = (\text{coins}_a, \text{coins}_g)$. Generate $\widetilde{\text{coins}}_{ga}$ according to the distribution of coins_{ga}.
6. Compute $\tilde{c}_{g,k}$ according to (15).

We end up with the simulator

$$\begin{aligned} S_{ga} = (&pk, H_{r,k}, R_{r,k}, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \tilde{c}_{g,k}, \widetilde{[\![\bar{c}_{g,k}]\!]}, \widetilde{[\![\bar{c}_{r,k}]\!]}, \widetilde{[\![\Gamma_g]\!]}, \\ &\bar{G}_{r,k}, \widetilde{[\![\hat{c}_{a,k}]\!]}, \hat{G}_{a,k}, q, F, Q_k, \widetilde{\text{coins}}_{ga}, pk). \end{aligned} \tag{A.24}$$

Thus, we find that $S_{ga} \overset{c}{\equiv} V_{ga}^\Pi$. The target of this coalition is to get the zonotopes of the remaining groups, denoted by N_r with size d_r. The centers of the zonotopes are related by

$$\sum_{j \in \mathcal{N}_r} w_{a,k}^j [\![\bar{c}_{g_j,k}]\!] \oplus [\![\tilde{c}_{a,k}]\!] \sum_{j \in \mathcal{N}} w_{a,k}^j = \sum_{j \in \mathcal{N}/r} w_{a,k}^j [\![\bar{c}_{g_j,k}]\!]. \tag{A.25}$$

The right hand side of (A.25) is known to the coalition. However, since the coalition does not have the private key, the privacy of the centers of the remaining group can be guaranteed.

Appendix A.3.3. Coalition of sensor groups g and the query:

The view of the coalition is defined as V_{gq}^Π where $V_{gq}^\Pi = (V_g^\Pi, V_q^\Pi)$ with V_q^Π given by

$$V_q^\Pi = ([\![\hat{c}_{a,k}]\!], \hat{c}_{a,k}, \hat{G}_{a,k}, q, F, Q_k, \text{coins}_q, pk, sk, [\![\Gamma_q]\!]). \tag{A.26}$$

Constructing the simulator S_{gq} from the inputs and outputs of the coalition as done before results in

$$\begin{aligned} S_{gq} = (&pk, sk, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \bar{c}_{g,k}, \widetilde{[\![\Gamma_q]\!]}, \widetilde{[\![\hat{c}_{a,k}]\!]}, \hat{c}_{a,k}, \hat{G}_{a,k}, \\ &\widetilde{\text{coins}}_{gq}), \end{aligned} \tag{A.27}$$

which implies that $S_{gq} \overset{c}{\equiv} V_{gq}^\Pi$. The target of this coalition is to get the zonotopes of the remaining group, denoted as before by $\mathcal{N}_r$ with size d_r. Rewriting (A.25) after decryption – as the query has the Paillier private key sk – results in

$$W_{r,k} C_{r,k} = z_{g,k}, \tag{A.28}$$

with

$$z_{g,k} = \sum_{j \in \mathcal{N}/r} w_{a,k}^j \bar{c}_{g_j,k} - \tilde{c}_{a,k} \sum_{j \in \mathcal{N}} w_{a,k}^j, \tag{A.29}$$

$$W_{r,k} = [w_{a,k}^{j_1} I_n, w_{a,k}^{j_2} I_n, \dots, w_{a,k}^{j_{d_r}} I_n] \in \mathbb{R}^{n \times n d_r}, \tag{A.30}$$

$$C_{r,k} = [c_{j_1,k}^T, c_{j_2,k}^T, \dots, c_{j_{d_r},k}^T]^T \in \mathbb{R}^{n d_r}, \tag{A.31}$$

where $z_{g,k}$ is known to the coalition. Similarly to the proof of Theorem 1 and according to [54, Theorem 6.4], $\tilde{C}_{r,k}$ is a solution of (A.28) for any $X_r \in \mathbb{R}^{n d_r}$ where

$$\tilde{C}_{r,k} = W_{r,k}^- z_{g,k} + (I_{n d_r} - W_{r,k}^- W_{r,k}) X_r, \tag{A.32}$$

and where $W_{r,k}^-$ is any generalized inverse of $W_{r,k}$. For every solution $\tilde{C}_{r,k}$ of (A.14) there is a X_r. If $I_{n d_r} - W_{r,k}^- W_{r,k} = 0$, we end up with a consistent system with one solution [54, Theorem 6.1]. Thus, we aim to find conditions at which $I_{n d_r} - W_{r,k}^- W_{r,k} \neq 0$ to ensure privacy. We have $rank(W_{r,k}^- W_{r,k}) \leq \min\{n d_r, n\}$ according to [54, Theorem 2.8]. Thus, under the condition $d_r > 1$, it follows that $I_{p d_r} - W_{r,k}^- W_{r,k} \neq 0$ which ensures privacy of $C_{r,k}$.

Appendix A.4. Proof of Theorem 4

In the following proof, we consider the view and simulation for one step (i.e., k-th step) for notational convenience. The proof for $K \in \mathbb{N}^+$ steps is similar to the proof of Theorem 1. We are going to prove the privacy against the three coalitions as follows:

Appendix A.4.1. Coalition of sensor groups g :

The view of the coalition can be defined by

$$V_g^{\Pi} = (V_{g1}^{\Pi}, \ldots, V_{g_t}^{\Pi}) = (pk, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \bar{c}_{g,k}, [\![\bar{c}_{g,k}]\!], \bar{A}_{g,k}, \bar{b}_{g,k}, [\![\bar{b}_{g,k}]\!], [\![\Gamma_g]\!], coin_g). \tag{A.33}$$

Again, sensors only submit their encrypted data to the aggregator. Hence, a simulator S_s given by

$$S_g = (pk, H_{g,k}, y_{g,k}, R_{g,k}, \bar{G}_{g,k}, \bar{c}_{g,k}, \widetilde{[\![\bar{c}_{g,k}]\!]}, \bar{A}_{g,k}, \bar{b}_{g,k}, \widetilde{[\![\bar{b}_{g,k}]\!]}, \widetilde{[\![\Gamma_g]\!]}, \widetilde{coin}_g), \tag{A.34}$$

is obtained by generating $\widetilde{[\![\bar{c}_{g,k}]\!]}, \widetilde{[\![\Gamma_g]\!]}, \widetilde{[\![\bar{b}_{g,k}]\!]}$ and $\widetilde{\text{coins}}_g$. The generated and the original values are generated according to the same distribution and are independent from other parameters. Therefore, we conclude that $S_g \overset{c}{\equiv} V_g^{\Pi}$.

Moreover, the resulting constrained zonotopes from the sensor groups are independent. Thus, the coalition zonotopes cannot be used to infer new information about other zonotopes.

Appendix A.4.2. Coalition of sensor groups g and the aggregator:

The view of the coalition, denoted by V_{ga}^{Π}, is

$$\begin{aligned} V_{ga}^{\Pi} &= (V_g^{\Pi}, V_a^{\Pi}) \\ &= (V_g^{\Pi}, [\![\bar{c}_{g,k}]\!], \bar{G}_{g,k}, \bar{A}_{g,k}, [\![\bar{b}_{g,k}]\!], [\![\bar{c}_{r,k}]\!], \bar{G}_{r,k}, \bar{A}_{r,k}, [\![\bar{b}_{r,k}]\!], [\![\hat{c}_{a,k}]\!], \\ &\quad \hat{G}_{a,k}, \hat{A}_{a,k}, [\![\hat{b}_{a,k}]\!], q, F, Q_k, \text{coins}_a, pk) \\ &\overset{(A.33)}{=} (H_{r,k}, R_{r,k}, H_{g,k}, y_{g,k}, R_{g,k}, \bar{c}_{g,k}, \bar{G}_{g,k}, [\![\bar{c}_{g,k}]\!], \bar{A}_{g,k}, \bar{b}_{g,k}, [\![\bar{b}_{g,k}]\!], \\ &\quad [\![\bar{c}_{r,k}]\!], \bar{G}_{r,k}, \bar{A}_{r,k}, [\![\bar{b}_{r,k}]\!], [\![\hat{c}_{a,k}]\!], \hat{G}_{a,k}, \hat{A}_{a,k}, [\![\hat{b}_{a,k}]\!], q, F, Q_k, \\ &\quad \text{coins}_{ga}, pk) \end{aligned} \tag{A.35}$$

where $\langle [\![\bar{c}_{r,k}]\!], \bar{G}_{r,k}, \bar{A}_{r,k}, [\![\bar{b}_{r,k}]\!] \rangle$, represents the encrypted constrained zonotopes of the sensor groups which are not part of the coalition. The simulator, denoted by S_{ga}, can be constructed given the input and output $(H_{g,k}, R_{g,k}, F, pk, q, Q_k, y_{g,k}, \bar{b}_{g,k}, \bar{c}_{g,k})$ as follows:

1. Add $H_{r,k}$, $R_{r,k}$ as they are public information.
2. Compute $\bar{G}_{g,k}$ and $\bar{G}_{r,k}$ according to (22).
3. Compute $\bar{A}_{g,k}$ and $\bar{A}_{r,k}$ according to (23).
4. Compute $\hat{G}_{a,k}$ according to (28) and reduction operation similar to (26).
5. Compute $\hat{A}_{a,k}$ according to (34) and reduction operation similar to (26).
6. Generate $\widetilde{[\![\bar{c}_{g,k}]\!]}, \widetilde{[\![\bar{c}_{r,k}]\!]}$, and $\widetilde{[\![\hat{c}_{a,k}]\!]}$ according to the distributions of the original values.
7. Generate $\widetilde{[\![\bar{b}_{g,k}]\!]}, \widetilde{[\![\bar{b}_{r,k}]\!]}$, and $\widetilde{[\![\hat{b}_{a,k}]\!]}$ according to the distributions of the original values.
8. Let the combination of the coins of all parties be $\text{coins}_{ga} = (\text{coins}_a, \text{coins}_{g_1}, \ldots, \text{coins}_{g_t})$. Generate $\widetilde{\text{coins}}_{ga}$ according to the distribution and of coins_{ga}.

We end up with the following simulator

$$S_{ga} = (H_{r,k}, R_{r,k}, H_{g,k}, y_{g,k}, R_{g,k}, \bar{c}_{g,k}, \bar{G}_{g,k}, \widetilde{[\![\bar{c}_{g,k}]\!]}, \bar{A}_{g,k}, \bar{b}_{g,k}, \widetilde{[\![\bar{b}_{g,k}]\!]}, \widetilde{[\![\bar{c}_{r,k}]\!]}, \bar{G}_{r,k}, \bar{A}_{r,k}, \widetilde{[\![\bar{b}_{r,k}]\!]}, \widetilde{[\![\hat{c}_{a,k}]\!]}, \hat{G}_{a,k}, \hat{A}_{a,k}, \widetilde{[\![\hat{b}_{a,k}]\!]}, q, F, Q_k, \widetilde{\text{coins}}_{ga}, pk). \tag{A.36}$$

Thus, we find that $S_{ga} \overset{c}{\equiv} V_{ga}^{\Pi}$. Similarly to Section Appendix A.3.2, the coalition is not be able to infer information about the constrained zonotopes of the remaining group.

Appendix A.4.3. Coalition of sensor groups g and the query:

The view of the coalition is defined by

$$\begin{aligned} V_{gq}^{\Pi} &= (V_g^{\Pi}, V_q^{\Pi}) = (V_g^{\Pi}, [\![\hat{c}_{a,k}]\!], [\![\hat{b}_{a,k}]\!], \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, q, F, \\ &\quad Q_k, \text{coins}_q, pk, \text{sk}, [\![\Gamma_q]\!]) \\ &\overset{(A.33)}{=} (H_{r,k}, R_{r,k}, H_{g,k}, y_{g,k}, R_{g,k}, \bar{c}_{g,k}, \bar{G}_{g,k}, [\![\bar{c}_{g,k}]\!], \bar{A}_{g,k}, \bar{b}_{g,k}, [\![\bar{b}_{g,k}]\!], \\ &\quad [\![\hat{c}_{a,k}]\!], [\![\hat{b}_{a,k}]\!], \hat{c}_{a,k}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, q, F, Q_k, \text{coins}_{gq}, pk, \text{sk}, \\ &\quad [\![\Gamma_{gq}]\!]). \end{aligned} \tag{A.37}$$

Constructing the simulator S_{gq} from the inputs and outputs of the coalition is done as before

$$S_{gq} = (H_{r,k}, R_{r,k}, H_{g,k}, y_{g,k}, R_{g,k}, \bar{c}_{g,k}, \bar{G}_{g,k}, \widetilde{[\![\bar{c}_{g,k}]\!]}, \bar{A}_{g,k}, \bar{b}_{g,k}, \widetilde{[\![\bar{b}_{g,k}]\!]}, \widetilde{[\![\hat{c}_{a,k}]\!]}, \widetilde{[\![\hat{b}_{a,k}]\!]}, \widetilde{\hat{c}_{a,k}}, \hat{G}_{a,k}, \hat{A}_{a,k}, \hat{b}_{a,k}, q, F, Q_k, \text{coins}_{gq}, pk, \text{sk}, \widetilde{[\![\Gamma_{gq}]\!]}), \tag{A.38}$$

which in turn implies that $S_{gq} \overset{c}{\equiv} V_{gq}^{\Pi}$. The target of this coalition is to get the constrained zonotopes of the remaining groups. As shown in (33), any center and generator of the coalition can determine the containing zonotope of the constrained zonotope. However, the remaining rows of the $[\![\grave{b}_{a,k}]\!]$ in (34), which belongs to the non-colluding sensor group, can not be inferred from the coalition.

References

[1] D. Bertsekas, I. Rhodes, Recursive state estimation for a set-membership description of uncertainty, IEEE Transactions on Automatic Control 16 (2) (1971) 117–128.

[2] L. Jaulin, Robust set-membership state estimation; application to underwater robotics, in: Automatica, Vol. 45, 2009, pp. 202–206.

[3] V. Puig, Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies, Vol. 20, 2010, pp. 619–635.

[4] C. Combastel, Merging Kalman filtering and zonotopic state bounding for robust fault detection under noisy environment, Vol. 48, Elsevier, 2015, pp. 289–295.

[5] X. Ge, Q.-L. Han, F. Yang, Event-based set-membership leader-following consensus of networked multi-agent systems subject to limited communication resources and unknown-but-bounded noise, in: IEEE Transactions on Industrial Electronics, Vol. 64, 2017, pp. 5045–5054.

[6] P. Bouron, D. Meizel, P. Bonnifait, Set-membership non-linear observers with application to vehicle localisation, in: European Control Conference, IEEE, 2001, pp. 1255–1260.

[7] M. Althoff, J. J. Rath, Comparison of guaranteed state estimators for linear time-invariant systems, Automatica 130, article no. 109662 (2021).

[8] A. Alanwar, Y. Shoukry, S. Chakraborty, P. Martin, P. Tabuada, M. Srivastava, Proloc: resilient localization with private observers using partial homomorphic encryption, in: Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2017, pp. 41–52.

[9] V. Narri, A. Alanwar, J. Martensson, C. Norén, L. Dal Col, K. H. Johansson, Set-membership estimation in shared situational awareness for automated vehicles in occluded scenarios, in: IEEE Intelligent Vehicles Symposium, 2021, pp. 385–392.

[10] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al., Comprehensive experimental analyses of automotive attack surfaces., in: USENIX Security Symposium, Vol. 4, San Francisco, 2011, pp. 447–462.

[11] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, A. Winnicki, Cyber-physical systems security: experimental analysis of a vinyl acetate monomer plant, in: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, 2015, pp. 1–12.

[12] Y. Liu, Y. Zhao, F. Wu, Ellipsoidal state-bounding-based set-membership estimation for linear system with unknown-but-bounded disturbances, IET Control Theory & Applications 10 (4) (2016) 431–442.

[13] N. Xia, F. Yang, Q.-L. Han, Distributed networked set-membership filtering with ellipsoidal state estimations, in: Information Sciences, Vol. 432, 2018, pp. 52 – 62.

[14] N. Xia, F. Yang, Q.-L. Han, Distributed networked set-membership filtering with ellipsoidal state estimations, Information Sciences 432 (2018) 52–62.

[15] W. Kühn, Rigorously computed orbits of dynamical systems without the wrapping effect, in: Computing, Vol. 61, 1998, pp. 47–67.

[16] R. García, L. Orihuela, P. Millán, F. Rubio, M. Ortega, Guaranteed estimation and distributed control of vehicle formations, International Journal of Control 93 (11) (2020) 2729–2742.

[17] A. Alanwar, J. J. Rath, H. Said, M. Althoff, Distributed set-based observers using diffusion strategy, arXiv preprint arXiv:2003.10347 (2020).

[18] C. Combastel, A. Zolghadri, A distributed kalman filter with symbolic zonotopes and unique symbols provider for robust state estimation in cps, International Journal of Control 93 (11) (2020) 2596–2612.

[19] J. Blesa, V. Puig, J. Saludes, Robust fault detection using polytope-based set-membership consistency test, IET Control Theory & Applications 6 (12) (2012) 1767–1777.

[20] G. Belforte, B. Bona, V. Cerone, Parameter estimation algorithms for a set-membership description of uncertainty, Automatica 26 (5) (1990) 887–898.

[21] S. E. Hamdi, M. Amairi, M. Aoun, Orthotopic set-membership parameter estimation of fractional order model, in: 24th Mediterranean Conference on Control and Automation, IEEE, 2016, pp. 634–639.

[22] C. Gentry, S. Halevi, Implementing Gentry's fully-homomorphic encryption scheme, in: Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology, 2011, pp. 129–148.

[23] K. Lauter, A. López-Alt, M. Naehrig, Private computation on encrypted genomic data, in: International Conference on Cryptology and Information Security in Latin America, 2014, pp. 3–27.

[24] R. Bost, R. A. Popa, S. Tu, S. Goldwasser, Machine learning classification over encrypted data, in: 22nd Annual Network and Distributed System Security Symposium, 2015.

[25] A. A. M. A. Abdelhafez, Localization of cyber-physical systems: privacy, security and efficiency, Dissertation, Technische Universität München, München (2020).

[26] A. B. Alexandru, G. J. Pappas, Secure multi-party computation for cloud-based control, in: Privacy in Dynamical Systems, 2020, pp. 179–207.

[27] K. Kogiso, Encrypted control using multiplicative homomorphic encryption, in: Privacy in Dynamical Systems, Springer, 2020, pp. 267–286.

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

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

[30] Y. Aono, T. Hayashi, L. Wang, S. Moriai, et al., Privacy-preserving deep learning via additively homomorphic encryption, IEEE Transactions on Information Forensics and Security 13 (5) (2017) 1333–1345.

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

[32] F. Farokhi, I. Shames, K. Johansson, Private routing and ride sharing using homomorphic encryption, 2020.

[33] H. Imtiaz, J. Mohammadi, A. D. Sarwate, Distributed differentially private computation of functions with correlated noise, 2019.

[34] T. Zhang, Q. Zhu, Dynamic differential privacy for admm-based distributed classification learning, IEEE Transactions on Information Forensics and Security 12 (1) (2016) 172–187.

[35] J. C. Duchi, M. I. Jordan, M. J. Wainwright, Privacy aware learning, in: Journal of ACM, Vol. 61, 2014, pp. 38:1–38:57.

[36] R. Dobbe, Y. Pu, J. Zhu, K. Ramchandran, C. Tomlin, Customized local differential privacy for multi-agent distributed optimization, 2018.

[37] N. E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Optimal geo-indistinguishable mechanisms for location privacy, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 251–262.

[38] T. T. Cai, Y. Wang, L. Zhang, The cost of privacy: optimal rates of convergence for parameter estimation with differential privacy, 2019.

[39] E. Shi, R. Chow, T. h. Hubert Chan, D. Song, E. Rieffel, Privacy-preserving aggregation of time-series data, in: Proceedings of the Network and Distributed System Security Symposium, 2011.

[40] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, Our data, ourselves: privacy via distributed noise generation, in: Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques, 2006, pp. 486–503.

[41] M. Althoff, Reachability analysis and its application to the safety assessment of autonomous cars, Ph.D. thesis, Technische Universität München (2010).

[42] J. K. Scott, D. M. Raimondo, G. R. Marseglia, R. D. Braatz, Constrained zonotopes: A new tool for set-based estimation and fault detection, Vol. 69, Elsevier, 2016, pp. 126–136.

[43] 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.

[44] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, G. J. Pappas, Cloud-based quadratic optimization with partially homomorphic encryption, IEEE Transactions on Automatic Control 66 (5) (2020) 2357–2364.

[45] C. Murguia, F. Farokhi, I. Shames, Secure and private implementation of dynamic controllers using semihomomorphic encryption, IEEE Transactions on Automatic Control 65 (9) (2020) 3950–3957.

[46] J. Kim, H. Shim, K. Han, Dynamic controller that operates over homomorphically encrypted data for infinite time horizon, IEEE Transactions on Automatic Control (2022).

[47] O. Goldreich, Foundations of cryptography: volume 1, basic tools, Cambridge university press, 2007.

[48] N. Kochdumper, F. Gruber, B. Schürmann, V. Gaßmann, M. Klischat, M. Althoff, AROC: A toolbox for automated reachset optimal controller synthesis, in: Proc. of the 24th ACM International Conference on Hybrid Systems: Computation and Control, 2021, article no. 23.

[49] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, D. Dumur, Zonotope-based set-membership estimation for multi-output uncertain systems, in: IEEE International Symposium on Intelligent Control, 2013, pp. 212–217.

[50] C. Combastel, Zonotopes and kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence, Automatica 55 (2015) 265–273.

[51] A. Girard, Reachability of uncertain linear systems using zonotopes, in: International Workshop on Hybrid Systems: Computation and Control,

2005, pp. 291–305.
[52] A. Alanwar, H. Said, A. Mehta, M. Althoff, Event-triggered diffusion kalman filters, in: ACM/IEEE 11th International Conference on Cyber-Physical Systems, 2020, pp. 206–215.
[53] M. Althoff, D. Grebenyuk, N. Kochdumper, Implementation of Taylor models in CORA 2018, in: Proceedings of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems, 2018, pp. 91–105.
[54] J. R. Schott, Matrix analysis for statistics, John Wiley & Sons, 2016.