Privacy-Preserving Localisation Using Private Linear-Combination Aggregation

Marko Ristic, Benjamin Noack, *Member, IEEE,* and Uwe D. Hanebeck, *Fellow, IEEE*

***Abstract*—Distributed state estimation and localisation methods have become increasingly popular with the rise of ubiquitous computing, and have led naturally to an increased concern regarding data and estimation privacy. Traditional distributed sensor navigation methods involve the leakage of sensor or navigator information during localisation protocols, thus not preserving participants' data privacy. Existing approaches which provide such guarantees fail to address sensor and navigator privacy in some common non-linear measurement model applications and forfeit broad applicability. We define the novel notion of linear-combination aggregation encryption and provide a cryptographically secure instance that is applied to the Extended Information Filter with range-sensor measurements. Our method keeps navigator location, sensor locations and sensor measurements private during navigation, and is implemented and run on simulations to evaluate its accuracy and performance. The novel computationally plausible and provably secure range-based localisation filter has direct applications to environments where nodes are not trusted or data is considered sensitive.**

***Index Terms*—State Estimation, Data Privacy, Sensor Fusion, Kalman Filters.**

I. Introduction

LOCALISATION methods in distributed sensor environments have long been an active topic of research [1], [2]. Ongoing advancements in portable computing power and sensor capabilities have led to the development of various forms of localisation methods, from range-free and range-based signal strength measurements [3], [4], acoustic range estimates [5] and magnetic field measurements [6] to the TDOA and TOA pseudo-range multilateration methods for aircraft [2]. Range-based localisation methods, including the use of radar measurements, GPS clocks and signal strength measurements, have found particularly large application due to their robustness and accuracy. These methods use distance measurements, or computed estimates, to produce location estimates and typically involve gathering sensor details such as measurement noise, geographical location and observations centrally. Broadcasting or gathering information in this way, however, implies a trust between participants within a trusted communication network to ensure shared data may not be used for individual malicious gain. With the advancements in cloud and ubiquitous computing, changes in possible use-cases involving distributed sensors have made the requirements of information security and user privacy more apparent than before [7], [8]. For example, an aircraft localisation scenario in the presence of privately operated measurement stations may imply measurements and locations of stations should be kept private from navigating vehicles. Or in autonomous vehicle state estimation, benefit from neighbouring vehicles' information may be desired while no participant wishes to share information about their hardware. Additionally, the dispatching of any expensive intermediate computations to service-providing cloud operators may require ensuring the privacy of information involved in the computations. All of these scenarios, and the one we will consider, require methods for computing specific operations while ensuring the privacy of some or all of the input data. Our contributions in this work will be presented in two parts. We first propose a novel aggregation-based encryption scheme which allows private aggregation of weighted sums, before we develop a localisation filter using this scheme such that sensor and navigator privacy is preserved.

The remainder of the paper is structured as follows. In Section III we introduce the localisation problem, the relevant aggregation opertation for its solution and the restrictions on involved parties. Sections IV and V will summarise cryptographic preliminaries and introduce a novel security notion for our required aggregation before defining and proving an encryption scheme which satisfies it. Sections VI and VII introduce localisation preliminaries and our privacy-preserving localisation method using the private aggregation scheme. We show and discuss simulation results in Section VIII and conclude our work in Section IX.

A. Notation

Throughout this work, we will use the following notation. Lowercase underlined characters are vectors, $\underline{a}$, and random variables are bold, $\boldsymbol{a}$. Uppercase bold characters are matrices, $\mathbf{A}$, with $\mathbf{A}^{-1}$ denoting the inverse and $\mathbf{A}^{\top}$ the transpose. $|a|$ is the absolute value of a and $\|\underline{a}\|$ the vector normal. Subscripts $\boldsymbol{x}_{k|j}$ will denote an estimate of $\boldsymbol{x}_k$ using measurements from times up to and including j. $\mathrm{E}\{\cdot\}$ and $\mathrm{Var}\{\cdot\}$ are used for the expected value and variance, respectively, and encryption and decryption with key k are denoted $\mathcal{E}_k(\cdot)$ and $\mathcal{D}_k(\cdot)$. Additionally, key k may be ommited when inferrable from context. $\lfloor\cdot\rceil$ rounds to the nearest integer, $a\|b$ stands for a bitwise concatenation of a and b and sets will be written as $\{\dots\}$ while ordered sequences as $\langle\dots\rangle$.

II. Literature Overview

In this section, we will discuss methods for providing security guarantees in signal processing tasks, and their applications to localisation and estimation problems.

M. Ristic, B. Noack and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory, Institute for Anthropomatics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. e-mail: marko.ristic@kit.edu

A. Encryption Schemes in Signal Processing

Of the methods used to preserve privacy during distributed signal processing, it is common for the information exchanging portion of the algorithm to be reduced to a form of aggregation of involved parties' data. The most common methods for ensuring privacy during aggregation required for signal processing tasks have been described below.

Differential Privacy These techniques formalise preserving the privacy of individual datum within an aggregation operation by introducing known distributions of noise to the resulting aggregation. To handle the cases with no trusted aggregator, local differential privacy is achieved by introducing noise to individual inputs [9], [10], [11]. Although differential privacy has strong computational benefits due to its implementational simplicity, it provides relatively weak security guarantees for multi-party networks, with no guarantees in the presence of multiple colluding corrupted parties. Additionally, noisy aggregation results hinder its use in scenarios where high precision is a desired property such as aircraft localisation.

Aggregator Oblivious Encryption This method of encryption formalises the privacy of *all* aggregation data from honest parties [12], [13], [14]. Encryptions of aggregation data are proven to be computationally indistinguishable and guarantee that only their sum can be learned by an aggregator, whether or not they are trusted. The originally proposed method [12] is restricted to a small input domain which has been expanded and computationally simplified in later works [14], [15]. The aggregation operation in these schemes is, however, strictly limited to the sum operation and restricts its applicability to simple processing tasks such as smart-meter sum or mean calculation [13], [15].

In addition to the private aggregation of data, it may also be the case that private data needs to be modified by an external party before being returned, as with privacy-preserving cloud computing. These methods have been broadly grouped into two domains and described below.

Fully Homomorphic Encryption These schemes allow all algebraic computations to be performed on encrypted data without the computing party learning any information about inputs, intermediate values or results [16], [17]. Although theoretically ideal for many data-sensitive tasks, current implementations are still computationally impractical for real-time or large scale tasks [18].

Partially Homomorphic Encryption A simplification of fully homomorphic schemes; these schemes provide only a subset, typically one, of algebraic operations on encrypted data [19], [20], [21]. Reduced computational requirements and implementation simplicity have made partially homomorphic encryption schemes commonly used in privacy-preserving signal processing tasks, including private matrix multiplication [22], set intersection computation [23] and control input aggregation [24]. However, these works are relatively restricted in application due to the limited operations provided.

Lastly, recent developments in encryption schemes have presented some novel methods also suitable for privacy-preserving computation in distributed environments.

Multi-Client Function Encryption These schemes allow the computing of a specific function given encrypted inputs [25], [26]. It is suitable for data-sensitive distributed processing as encryptions must be combined before a result is obtained, and can be considered a generalisation of the aggregation schemes described earlier. While general schemes that allow computing arbitrary functions are computationally expensive and complicate security definitions in distributed settings [27], partial schemes supporting only some functions, such as vector dot products [28], exist with formal security proofs, but have yet to find applications in more complicated signal processing tasks.

Of the aforementioned encryption schemes, some have found uses in privacy-preserving localisation and estimation tasks and we summarise these in the next section.

B. Encrypted Localisation and Estimation

Localisation methods exist for a variety of purposes requiring different levels of accuracy and supporting various forms of sensors. We aim to provide an accurate localisation filter in the presence of range-only sensors.

Relevant estimation methods include both Bayesian estimation methods which make assumptions about target trajectories to form their location estimates, typically based on the Kalman Filter and later derivatives [29], [30], and time-independent methods where location estimates are made independently of each other [31], [32]. The work in [31] uses partially homomorphic encryption to compute time-independent location estimates while preserving the privacy of sensor measurements and locations. The work proposes two approaches, using either polygon intersections suffering from geometric dilution of precision, or alternating projection requiring repeated interactive comparison protocols. Neither method considers measurement uncertainty, and no estimate errors are provided.

Bayesian approaches can achieve better accuracy due to their exploitation of process dynamic knowledge. In [33], partially homomorphic encryption is used to encrypt measurement information and produce location estimates at a navigator. While producing estimate errors and requiring only unidirectional communication, this method exclusively supports linear measurement models and is therefore not suitable for range-only measurements. Encrypted model predictive control methods [34] also introduce some privacy-preserving methods relevant to estimation. [24] proposes secure distributed control input aggregation using a novel weighted sum aggregation scheme, but requires repeated key distribution among all involved parties. In [35], a secure cloud-based control aggregator using partially homomorphic encryption is proposed, but is suitable only for untrusted service-providing cloud architectures.

While we have focused on the application of encryption schemes to localisation algorithms, homomorphic encryption schemes also require suitable number encoding when used

with real numbers. As is the case with localisation sensors, real number measurements need to be encoded as integers with operations consistent under scheme-provided homomorphic operations. Google's Encrypted BigQuery Client [36] provides an encoding which supports both addition and unencrypted multiplication when used with an additively homomorphic encryption scheme. It, however, leaks the exponent information of encrypted numbers. In [35], a simpler alternative for encoding is proposed, which leaks no information about encrypted numbers, but allows only a single unencrypted multiplication between encodings. We will discuss and justify our choice of encoding scheme when introducing the localisation filter.

III. Problem Statement

The localisation scenario we consider in this work is that of model-based self-navigation using range-only sensors. We consider localisation in the two-dimensional case for simplicity but will derive methods suitable for an extension to the three-dimensional equivalent.

The navigator state is defined as

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{dx} & \boldsymbol{y} & \boldsymbol{dy} \end{bmatrix}^\top . \tag{1}$$

A known process model is followed, which at time k is given by

$$\underline{\boldsymbol{x}}_k = \underline{f}(\underline{\boldsymbol{x}}_{k-1}) + \underline{\boldsymbol{w}}_k \,, \tag{2}$$

with zero-mean Gaussian process noise $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q})$. The measurement model is dependant on sensor i and given by

$$\boldsymbol{z}_{k,i} = h_i(\underline{\boldsymbol{x}}_k) + \boldsymbol{v}_{k,i} \,, \tag{3}$$

with noise $\boldsymbol{v}_{k,i} \sim \mathcal{N}(0, r_i)$, and measurement function h_i for sensor i at location

$$\underline{s}_i = \begin{bmatrix} s_{x,i} & s_{y,i} \end{bmatrix}^\top , \tag{4}$$

defined as

$$\begin{aligned} h_i(\underline{\boldsymbol{x}}) &= \left\| \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix}^\top - \underline{s}_i \right\| \\ &= \sqrt{(\boldsymbol{x} - s_{x,i})^2 + (\boldsymbol{y} - s_{y,i})^2} \,. \end{aligned} \tag{5}$$

We wish to run a state estimation filter with models (2) and (3) such that all involved sensors $1 \leq i \leq n$ do not learn state estimates of $\underline{\boldsymbol{x}}_k$ and the navigator does not learn sensor locations $\underline{s}_i$, $1 \leq i \leq n$, measurement variances r_i or their measurements $\boldsymbol{z}_{k,i}$ at any time k. We motivate these goals with the example of aircraft navigation in the presence of privately-operated range-measuring stations. It is reasonable to assume that the current state of an aircraft may not wish to be disclosed to unknown parties and, similarly, that tower locations may wish to be kept private from unidentified navigating aircraft. However, the additional safety to passengers provided by accurate aircraft localisation, may be a goal of all those involved justifying their cooperation.

A. Problem Decomposition

The recursive nature of the problem definition above complicates defining and proving of security requirements for the localisation scenario. As with existing estimation methods [15], [24] we will reduce the communication requirements, of sensors and the navigator in our case, to an aggregation operation which will make security more easily definable.

To support multiple aggregations at each time-step k, as is required for the aggregation of multi-dimensional data, aggregation instances have been denoted by t. The desired bi-direction communication protocol at instance t consists of of the computation of linear combinations of navigator weights by sensors, before their privacy-preserving aggregation by the navigator. That is, m weights $\omega_j^{(t)}, 1 \leq j \leq m$ are broadcast by the navigator, linear combinations $l_i^{(t)} = \sum_{j=1}^{m} a_{j,i}^{(t)} \omega_i^{(t)}$ are computed by each sensor $1 \leq i \leq n$, and aggregation is computed back at the navigator. This has been summarised in Figure 1.

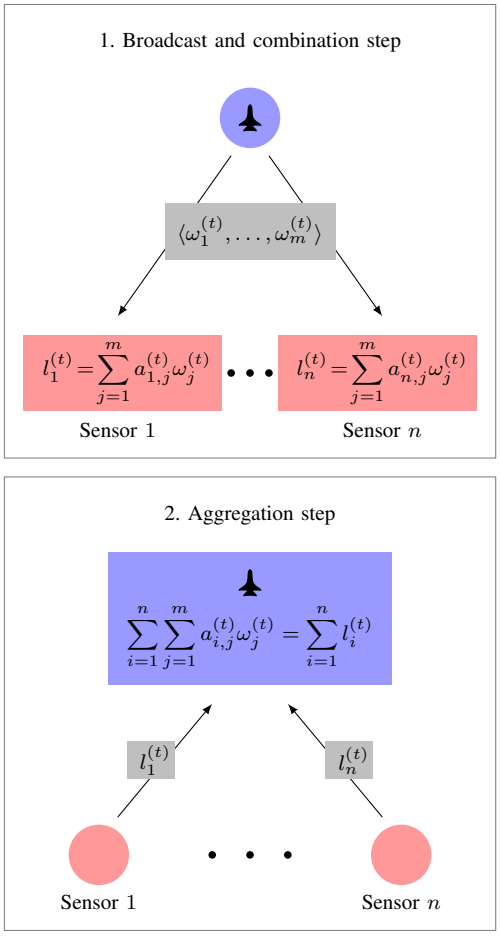

Fig. 1. Required linear-combination aggregation steps at instance t.

In Section VII we will show how this protocol is sufficient to compute the required estimation information at the sensors required to update the navigator's estimation filter.

Remark. From the aggregation definition above and the distributive property of real numbers, it is clear that an alterna-

tive method to the bi-directional communication requirement would be computing $\sum_{j=1}^{m} \omega_j^{(t)} \sum_{i=1}^{n} a_{i,j}^{(t)}$ by the navigator, requiring only uni-directional communication by sending values $a_{i,j}^{(t)}, 1 \leq j \leq m$ from each sensor i. We justify the use of bi-directional communication by reducing communication costs when the number of weights is larger than the number of sensors, $m > n$, and by sending fewer weights in the presence of repeats, as will be shown to be the case in Section VII-D.

B. Participant Capabilities

Before we can define concrete security requirements and a filtering protocol, we require some assumptions on the capabilities of the navigator and sensors in our problem scenario.

Global navigator broadcast From the decomposition defined above, we know that bi-directional communication between navigator and sensors is required. We make the assumption that broadcast information from the navigator is received by *all* sensors involved in the protocol.
A problem caused by this requirement is apparent in the context of aircraft navigation. It may be the case that some sensors are not within the range of a navigator broadcast. We loosely propose that this scenario can be handled with pre-defined sensor subsets during an initial step by a trusted party. An example has been shown in Figure 2, which reduces to two separate localisation instances with global broadcasts.

Consistent navigator broadcast Further, we make the assumption that broadcast information from the navigator is received equal by all sensors. This means the navigator may not send different information to individual sensors during a single time-step.
In the context of wireless communication, the widespread use of cheap non-directional antennas make this assumption reasonable.

Honest-but-curious sensors We adopt the honest-but-curious attacker model for all involved sensors, meaning they are assumed to follow the localisation procedure correctly but may store or use any gained sensitive information.
Misbehaving sensors are a known problem in estimation theory, often requiring inconsistency or outlier detection mechanisms [32], [37]. As privacy-preserving implementations of such mechanisms require additional complications, we will not consider the scenario in this work.

Computational capabilities The computational requirements for computing encryptions and homomorphic operations are typically more costly than their plaintext equivalents. We make the implicit assumption that all involved parties are computationally capable of computing the required encryption and filter operations.

C. Inherent Leakage

Leakage of information will be discussed in more detail in Section VII-E, however, we stress that in any estimation problem where process and measurement models are known, knowledge of sequential state estimates as is the case at the navigator, or the knowledge of one or more measurements as is the case by one or more malicious sensors, naturally leads to the leakage of some information. For this reason, we accept that average sensor information may be leaked to the navigator and that less accurate estimates of location may be leaked to sensors. We will Instead show that individual sensor information remains private to sensors and that the most precise localisation estimate, requiring all measurements, remains private to the navigator.

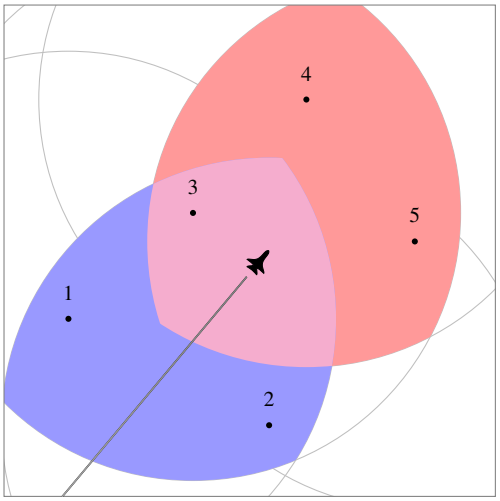

Fig. 2. An example of sensor subsets. Subsets $\{1, 2, 3\}$ and $\{3, 4, 5\}$ can be used independently for localisation when the navigator is within their ranges.

IV. Cryptography Preliminaries

When defining our aggregation security requirements and encryption scheme, we will reference some existing cryptographic security notions, the additively homomorphic Paillier encryption scheme and the Joye-Libert private aggregation scheme.

A. Security Notions

The security of a cryptographic scheme is typically defined by a security *game*, which captures both the desired privacy guarantees, as well as the capabilities of attackers [38]. The typical security notion for a homomorphic encryption scheme is Indistinguishability under Chosen Plaintext Attack (IND-CPA) [39].

Definition IV.1. An encryption scheme meets IND-CPA security if an attacker who can choose plaintext messages to be encrypted at will, gains no additional information about an unknown plaintext message when they learn only its encryption.

The formal security game for IND-CPA has been given in Appendix A.

Private aggregation schemes aim for the security notion of Aggregator Obliviousness (AO) [12].

Definition IV.2. An encryption scheme meets AO security if no colluding subset of participants *excluding* the aggregator gains additional information about the remaining aggregation

values given only their encryptions, while any colluding subset *including* the aggregator learns only their sum.

The formal security game for AO has been given in Appendix B.

B. Paillier Encryption Scheme

The Paillier encryption scheme [20] is an additively homomorphic encryption scheme which bases its security on the decisional composite residuosity assumption (DCRA) and meets the security notion of IND-CPA. Key generation of the Paillier scheme is performed by choosing two sufficiently large primes p and q, and computing $N = pq$. A generator g is also required for encryption, which is often set to $g = N+1$ when p and q are of equal bit length [38]. The public key is defined by (N, g) and secret key by (p, q).

Encryption of a plaintext message $a \in \mathbb{Z}_N$, producing ciphertext $c \in \mathbb{Z}^*_{N^2}$, is computed by

$$c = g^a \rho^N \pmod{N^2} \tag{6}$$

for a randomly chosen $\rho \in \mathbb{Z}_N$. ρ^N can be considered the noise term which hides the value $g^a \pmod{N^2}$, which due to the scheme construction, is an easily computable discrete logarithm. The decryption of a ciphertext is computed by

$$a = \frac{L(c^\lambda \pmod{N^2})}{L(g^\lambda \pmod{N^2})} \pmod{N} \tag{7}$$

where $\lambda = \mathsf{lcm}(p-1, q-1)$ and $L(u) = \frac{u-1}{N}$.

In addition to encryption and decryption, the following homomorphic functions are provided by the Paillier scheme. $\forall a_1, a_2 \in \mathbb{Z}_N$,

$$\mathcal{D}(\mathcal{E}(a_1)\mathcal{E}(a_2) \pmod{N^2}) = a_1 + a_2 \pmod{N} \tag{8}$$

$$\mathcal{D}(\mathcal{E}(a_1) g^{a_2} \pmod{N^2}) = a_1 + a_2 \pmod{N} \tag{9}$$

$$\mathcal{D}(\mathcal{E}(a_1)^{a_2} \pmod{N^2}) = a_1 a_2 \pmod{N}. \tag{10}$$

C. Joye-Libert Private Aggregation Scheme

The Joye-Libert private aggregation scheme [14] is a scheme defined on time-series data, that we refer to as instances for consistency, and meets the security notion of AO. Similarly to the Paillier scheme, it bases its security on the DCRA. A notable difference to a public-key encryption scheme is the need for a trusted party to perform an initial key generation and distribution step.

Key generation is computed by choosing two equal length and sufficiently large primes p and q, and computing $N = pq$. Additionally, hash function $H : \mathbb{Z} \to \mathbb{Z}^*_{N^2}$ is defined, and the public key is set to (N, H). n private keys are generated by choosing sk_i, $1 \leq i \leq n$ uniformly from $\mathbb{Z}_{N^2}$ and distributing them to all n users (whose values are to be aggregated), while the last key is set as

$$sk_0 = -\sum_{i=1}^{n} sk_i \pmod{N^2}, \tag{11}$$

and sent to the aggregator.

Encryption of plaintext $a_i^{(t)} \in \mathbb{Z}_N$ to ciphertext $c_i^{(t)} \in \mathbb{Z}_{N^2}$ at instance t is computed by user i as

$$c_i^{(t)} = (N+1)^{a_i^{(t)}} H(t)^{sk_i} \pmod{N^2}, \tag{12}$$

where the $H(t)^{sk_i}$ can be considered the noise term which hides the again easily computable discrete logarithm $g^{a_i^{(t)}} \pmod{N^2}$, where $g = N+1$.

When all encryptions $c_i^{(t)}$, $1 \leq i \leq n$ are sent to the aggregator, summation and decryption are computed by the functions

$$c^{(t)} = H(t)^{sk_0} \prod_{i=1}^{n} c_i^{(t)} \pmod{N^2} \tag{13}$$

and

$$\sum_{i=1}^{n} a_i^{(t)} = \frac{c^{(t)} - 1}{N}. \tag{14}$$

Correctness follows from $\sum_{i=0}^{n} sk_i = 0$, and thus

$$\begin{aligned}
&H(t)^{sk_0} \prod_{i=1}^{n} c_{i,t} \pmod{N^2} \\
\equiv &H(t)^{sk_0} \prod_{i=1}^{n} (N+1)^{a_{i,t}} H(t)^{sk_i} \pmod{N^2} \\
\equiv &H(t)^{\sum_{j=0}^{n} sk_j} \prod_{i=1}^{n} g^{a_{i,t}} \pmod{N^2} \\
\equiv &(N+1)^{\sum_{i=1}^{n} a_{i,t}} \pmod{N^2}
\end{aligned}$$

removing all noise terms.

V. Private Linear-Combination Aggregation

As stated in Section III-A, we decompose the localisation method such that the multi-party aggregation protocol, given in Figure 1, captures all communication between the navigator and sensors. To perform this protocol in a secure manner, we must first describe the security properties we aim to achieve during aggregation. Broadly speaking, we want a cryptographic scheme which allows instances of homomorphically computed linear combinations of encrypted weights, to be summed by a private aggregation scheme. That is, we do not want sensors to learn the navigator weights, while we do not want the navigator to learn individual weighted linear combinations. This can be summarised by the two informal security notions:

Indistinguishable Weights No colluding subset of sensors gains any additional knowledge about the navigator weights ω_j, $1 \leq j \leq m$ from receiving only their encryptions from the current and previous instances, and the ability to encrypt plaintexts of their choice.

Private Linear-Combination Aggregation No colluding subset *excluding* the aggregator gains additional information about the remaining sensor values to be weighted $a_{i,j}^{(t)}$, $1 \leq j \leq m$, where sensor i is not colluding, given only encryptions of their linear combinations l_i from the current and previous instances. Any colluding subset *including* the aggregator learns only

the sum of all linear combinations weighted by weights of their choice, $\sum_{i=1}^{n} l_i^{(t)} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}$.

Remark. The notion of a leakage function including parameters from the aggregator requires extra care to be taken when giving its definition. Since an attacker may compromise the aggregator, they have control over the choice of these parameters, and therefore over the leakage function. We note that in the leakage function above, $\sum_{i=n}^{n}\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}$, an individual sum weighted by the same weight may be learned by the colluding subset, *e.g.* $\sum_{j=1}^{m} a_{i,j}^{(t)}$ given weights $(1, 0, \ldots, 0)$, but that individual sensor values $a_{i,j}^{(t)}$ remain private due to the requirement that all sensors receive the same weights.

From the informal definitions above, it is clear that weights encrypted by an IND-CPA secure encryption scheme are sufficient for the first requirement, while a scheme satisfying AO is not sufficient for the second. To formalise the second requirement, we define a novel encryption type "Linear-Combination Aggregator Oblivious Encryption" and an accompanying security game, which capture the additional weights and modified leakage of AO.

A. Linear-Combination Aggregator Oblivious Encryption

We let a linear-combination aggregator oblivious encryption scheme be defined as a tuple of the four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{CombEnc}, \mathsf{AggDec})$. These are used by a trusted third party, an aggregator and value producing users i, $1 \leq i \leq n$. They are defined as follows

$\mathsf{Setup}(\kappa)$ On input of security paramater κ, generate public parameters pub, the number of weights m, the aggregator's public and private keys pk_0 and sk_0, and the user private keys sk_i, $1 \leq i \leq n$.

$\mathsf{Enc}(pk_0, x)$ The aggregator and users can encrypt any value x with the aggregator public key pk_0, and obtain the encryption $\mathcal{E}_{pk_0}(x)$.

$\mathsf{CombEnc}(t, pk_0, sk_i, \mathcal{E}_{pk_0}(\omega_1^{(t)}),..., \mathcal{E}_{pk_0}(\omega_m^{(t)}), a_{i,1}^{(t)},..., a_{i,m}^{(t)})$ At instance t, user i computes and obtains the encrypted linear combination $l_i^{(t)} = \mathcal{E}_{pk_0,sk_i}(\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)})$ using its secret key sk_i.

$\mathsf{AggDec}(t, pk_0, sk_0, l_1^{(t)},..., l_n^{(t)})$ At instance t, the aggregator computes the aggregation of linear combinations $\sum_{i=1}^{n} l_i^{(t)} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}$ using its public and private keys pk_0, sk_0.

We formalise the security notion of Linear-Combination Aggregator Obliviousness (LCAO) as the following game between attacker and challenger:

Setup The challenger chooses security parameter κ, runs the $\mathsf{Setup}(\kappa)$ algorithm and gives pub, m and pk_0 to the attacker

Queries The attacker can now perform encryptions or submit queries that are answered by the challenger. The types of actions are:

1) *Encryption:* The attacker chooses a value x and computes an encryption of x under the aggregator's public key pk_0, obtaining $\mathcal{E}_{pk_0}(x)$.
2) *Weight Queries:* The attacker chooses an instance t and receives the weights for that instance encrypted with the aggregator's public key, $\mathcal{E}_{pk_0}(\omega_j^{(t)})$, $1 \leq j \leq m$.
3) *Combine Queries:* The attacker chooses a tuple $(i, t, a_{i,1}^{(t)}, \ldots, a_{i,m}^{(t)})$ such that for any two chosen combine query tuples $(i, t, a_{i,1}^{(t)}, \ldots, a_{i,m}^{(t)})$ and $(i', t', a_{i',1}^{\prime(t')}, \ldots, a_{i',m}^{\prime(t')})$, the following condition holds:

$$i = i' \wedge t = t' \implies a_{i,j}^{(t)} = a_{i',j}^{\prime(t')},\ 1 \leq j \leq m\,.$$

The attacker is then given back the encryption of the linear combination $\mathcal{E}_{pk_0,sk_i}(\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)})$ encrypted under both the aggregator public key pk_0 and the secret key sk_i.

4) *Compromise queries:* The attacker chooses i and receives the secret key sk_i. The aggregator's secret key may also be compromised (when choosing $i = 0$).

Challenge Next, the attacker chooses an instance t^*, and a subset of users $S \subseteq U$ where U is the complete set of users for which no combine queries, for the instance t^*, and no compromise queries, are made for the duration of the game. The attacker then chooses two series of tuples

$$\left\langle \left(i, t^*, a_{i,1}^{(t^*)(0)}, \ldots, a_{i,m}^{(t^*)(0)}\right) \,\middle|\, i \in S \right\rangle$$

and

$$\left\langle \left(i, t^*, a_{i,1}^{(t^*)(1)}, \ldots, a_{i,m}^{(t^*)(1)}\right) \,\middle|\, i \in S \right\rangle\,,$$

and gives them to the challenger. In the case that $0 \in S$ (*i.e.* the aggregator is compromised) and $S = U$, it is additionally required that

$$\sum_{i\in S}\sum_{j=1}^{m} a_{i,j}^{(t^*)(0)}\omega_j^{(t^*)} = \sum_{i\in S}\sum_{j=1}^{m} a_{i,j}^{(t^*)(1)}\omega_j^{(t^*)}\,,$$

for weights $\omega_j^{(t^*)}$, $1 \leq j \leq m$ returned by a *Weight Query* with chosen instance t^*. The challenger then chooses a random bit $b \in \{1, 0\}$ and returns encryptions

$$\left\langle \mathcal{E}_{pk_0,sk_i}\left(\sum_{j=1}^{m} a_{i,j}^{(t^*)(b)}\omega_j^{(t^*)}\right) \,\middle|\, i \in S \right\rangle\,.$$

More Queries The attacker can now perform more encryptions and submit queries, so long as the queries do not break the requirements in the Challenge stage. That is, $S \subseteq U$.

Guess At the end, the attacker outputs a bit b' and wins the game if and only if $b' = b$. The advantage of an attacker $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{LCAO}(\mathcal{A}) \coloneqq \left|\mathrm{P}[b' = b] - \frac{1}{2}\right|\,.$$

Definition V.1. An encryption scheme meets LCAO security if no adversary, running in probabilistic-time with respect to security parameter κ, has more than a negligible advantage in winning the above security game. Probabilities are taken over randomness introduced by $\mathcal{A}$, and in Setup, Enc and CombEnc.

In the next section, we will give a solution to an encryption scheme meeting LCAO security, with IND-CPA secure weight encryption, and give a cryptographic proof for its security.

B. Our Scheme

Our scheme is based on the Paillier and Joye-Libert schemes introduced in Sections IV-B and IV-C, and similarly bases its security on the DCRA. As with Joye-Libert's private aggregation scheme, a trusted party is required for the initial distribution of user secret keys. Below, we give definitions for the four algorithms comprising the linear-combination aggregation encryption scheme.

$\mathsf{Setup}(\kappa)$ On input parameter κ, generate two equal length, sufficiently large, primes p and q, and compute $N = pq$. Define a hash function $H : \mathbb{Z} \to \mathbb{Z}^*_{N^2}$, choose an $m > 1$ as the number of weights to combine, and set public parameter $\mathsf{pub} = H$, aggregator public key $pk_0 = N$ and aggregator private key $sk_0 = (p, q)$. The remaining user secret keys are generated by choosing sk_i, $1 \leq i \leq n-1$ uniformly from $\mathbb{Z}_{N^2}$ and setting the last key as $sk_n = -\sum_{i=1}^{n-1} sk_i \pmod{N^2}$.

$\mathsf{Enc}(pk_0, x)$ Encryption of values is computed as a Paillier encryption with implicit generator $g = N + 1$. This is given by

$$\mathcal{E}_{pk_0}(a) = (N+1)^x \rho^N \pmod{N^2}, \tag{15}$$

for a randomly chosen $\rho \in \mathbb{Z}_N$.

$\mathsf{CombEnc}(t, pk_0, sk_i, \mathcal{E}_{pk_0}(\omega_1^{(t)}),..., \mathcal{E}_{pk_0}(\omega_m^{(t)}), a_{i,1}^{(t)},..., a_{i,m}^{(t)})$ The linear combination encryption step at instance t is computed as

$$l_i^{(t)} = H(t)^{sk_i} \prod_{j=1}^{m} \mathcal{E}_{pk_0}(\omega_j^{(t)})^{a_{i,j}^{(t)}} \pmod{N^2}, \tag{16}$$

and makes use of the homomorphic property (10). Correctness follows from

$$\begin{aligned} l_i^{(t)} &= H(t)^{sk_i} \prod_{j=1}^{m} \mathcal{E}_{pk_0}(\omega_j^{(t)})^{a_{i,j}^{(t)}} \pmod{N^2} \\ &= H(t)^{sk_i} \prod_{j=1}^{m} \mathcal{E}_{pk_0}(a_{i,j}^{(t)}\omega_j^{(t)}) \pmod{N^2} \\ &= H(t)^{sk_i} \prod_{j=1}^{m} (N+1)^{a_{i,j}^{(t)}\omega_j^{(t)}} \rho_j^N \pmod{N^2} \\ &= H(t)^{sk_i} (N+1)^{\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}} \rho_i^N \pmod{N^2}, \end{aligned}$$

for some values $\rho_i, \rho_j \in \mathbb{Z}_N, 1 \leq j \leq m$. Here, ρ_i^N and $H(t)^{sk_i}$ can be considered the noise terms corresponding to the two levels of encryption from pk_0 and sk_i, respectively.

$\mathsf{AggDec}(t, pk_0, sk_0, l_1^{(t)},..., l_n^{(t)})$ Aggregation is computed as $l^{(t)} = \prod_{i=1}^{n} l_i^{(t)} \pmod{N^2}$, removing aggregation noise terms, and is followed by Paillier decryption

$$\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)} = \frac{L((l^{(t)})^\lambda \pmod{N^2})}{L((N+1)^\lambda \pmod{N^2})} \pmod{N}, \tag{17}$$

with $\lambda = \mathsf{lcm}(p-1, q-1)$ and $L(u) = \frac{u-1}{N}$. The correctness of aggregation can be seen from

$$\begin{aligned} l^{(t)} &= \prod_{i=1}^{n} H(t)^{sk_i} (N+1)^{\sum_{j=1}^{m} a_{i,j}\omega_j} \rho_i^N \pmod{N^2} \\ &= H(t)^{\sum_{i=1}^{n} sk_i} \cdot \\ &\qquad \prod_{i=1}^{n} (N+1)^{\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}} \rho_i^N \pmod{N^2} \\ &= (N+1)^{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}^{(t)}\omega_j^{(t)}} \rho'^N \pmod{N^2}, \end{aligned}$$

for some values $\rho_i, \rho' \in \mathbb{Z}_N$, $1 \leq i \leq n$.

Additionally, we note that in the above construction, all weights $\omega_j^{(t)}$ and values $a_{i,j}^{(t)}$ are integers, and that resulting linear combinations and summations are computed $\pmod{N}$.

Remark. The construction of this scheme additionally supports the linear combination of the implicit weight, $\omega_j^{(t)} = 1$, by replacing

$$\mathcal{E}_{pk_0}(1)^{a_{i,j}^{(t)}} = (N+1)^1 \rho_j^N \pmod{N^2} \tag{18}$$

in (16) by

$$(N+1)^{a_{i,j}^{(t)}} \pmod{N^2}, \tag{19}$$

due to the removal of ρ_j^N terms during decryption. This can be used to reduce the broadcast communication cost by one weight.

C. Security Proof

To prove the security of our introduced scheme, we recall the desired security properties of an LCAO secure scheme with IND-CPA secure encrypted weights. From the definition above, weights encrypted with public key pk_0 are identical to encryptions of the Paillier scheme and therefore meet security notion IND-CPA. We omit this proof here and refer readers to the security proof of the Paillier encryption scheme [20] instead.

To show our scheme meets the security notion of LCAO, we prove by contrapositive that for an adversary $\mathcal{A}$ playing against a challenger using *our scheme*, we can create an adversary $\mathcal{A}'$ playing against a challenger $\mathcal{C}$ using the *Joye-Libert scheme*, such that

$$\mathsf{Adv}^{LCAO}(\mathcal{A}) > \eta_1(\kappa) \implies \mathsf{Adv}^{AO}(\mathcal{A}') > \eta_2(\kappa),$$

for some negligible functions η_1, η_2 and security parameter κ. (*i.e.* if we assume our scheme is not LCAO secure, then the Joye-Libert scheme is not AO secure.) Given the Joye-Libert AO proof in [14], we know our scheme must be LCAO secure and will thus conclude our proof. The proof overview has been given in Figure 3. Additionally, the function H used by our scheme is treated as a *random oracle* in the Joye-Libert AO proof and will, therefore, prove our scheme secure in the random oracle model as well.

Proof. Consider adversary $\mathcal{A}$ playing the LCAO game defined in Section V-A. The following is a construction of an adversary $\mathcal{A}'$ playing the AO game in Appendix B against a challenger $\mathcal{C}$ using the Joye-Libert aggregation scheme from Section IV-C.

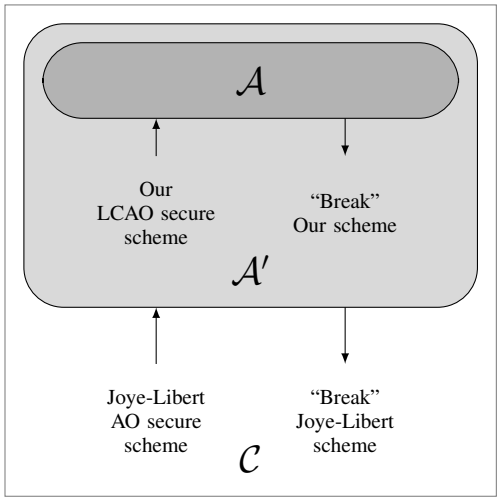

Fig. 3. High-level overview of our scheme's LCAO security proof.

Setup When receiving N and H as public parameters from $\mathcal{C}$, choose an $m > 1$ and give public parameter H, number of weights m, and $pk_0 = N$ to $\mathcal{A}$.

Queries Handle queries from $\mathcal{A}$:

Weight Query When $\mathcal{A}$ submits a weight query t, choose weights $\omega_j^{(t)}$, $1 \leq j \leq m$ and random values $\rho_j \in \mathbb{Z}_N$, $1 \leq j \leq m$, and return encryptions

$$(N+1)^{\omega_j^{(t)}} \rho_j^N \pmod{N^2}, 1 \leq j \leq m$$

to $\mathcal{A}$.

Combine Query When $\mathcal{A}$ submits combine query $(i, t, a_{i,1}^{(t)}, \ldots, a_{i,m}^{(t)})$, choose weights $\omega_j^{(t)}, 1 \leq j \leq m$ if not already chosen for the instance t, and make an AO encryption query $(i, t, \sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)})$ to $\mathcal{C}$. The received response will be of the form $(N+1)^{\sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}} H(t)^{sk_i}$; multiply it by ρ^N for a random $\rho \in \mathbb{Z}_N$ and return

$$(N+1)^{\sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}} \rho^N H(t)^{sk_i} \pmod{N^2}$$

to $\mathcal{A}$.

Compromise Query When $\mathcal{A}$ submits compromise query i, make the same compromise query i to $\mathcal{C}$, and return the recieved secret key sk_i to $\mathcal{A}$.

Challenge When $\mathcal{A}$ submits challenge series

$$\left\langle \left(i, t^*, a_{i,1}^{(t^*)(0)}, \ldots, a_{i,m}^{(t^*)(0)} \right) \,\middle|\, i \in S \right\rangle$$

and

$$\left\langle \left(i, t^*, a_{i,1}^{(t^*)(1)}, \ldots, a_{i,m}^{(t^*)(1)} \right) \,\middle|\, i \in S \right\rangle ,$$

choose weights $\omega_j^{(t^*)}, 1 \leq j \leq m$ for instance t^* and submit AO challenge series

$$\left\langle \left(i, t^*, \sum_{j=1}^m a_{i,j}^{(t^*)(0)} \omega_j^{(t^*)} \right) \,\middle|\, i \in S \right\rangle$$

and

$$\left\langle \left(i, t^*, \sum_{j=1}^m a_{i,j}^{(t^*)(1)} \omega_j^{(t^*)} \right) \,\middle|\, i \in S \right\rangle ,$$

to $\mathcal{C}$. The received response will be of the form

$$\left\langle (N+1)^{\sum_{j=1}^m a_{i,j}^{(t^*)(b)} \omega_j^{(t^*)}} H(t^*)^{sk_i} \,\middle|\, i \in U \right\rangle ,$$

for an unknown $b \in \{0, 1\}$. Multiply series elements by ρ_i^N, $1 \leq i \leq n$ for randomly chosen $\rho_i \in \mathbb{Z}_N$ and return

$$\left\langle (N+1)^{\sum_{j=1}^m a_{i,j}^{(t^*)(b)} \omega_j^{(t^*)}} \rho_i^N H(t^*)^{sk_i} \,\middle|\, i \in U \right\rangle$$

to $\mathcal{A}$.

Guess When $\mathcal{A}$ makes guess b', make the same guess b' to $\mathcal{C}$.

In the above construction, $\mathcal{C}$ follows the Joye-Libert scheme from Section IV-C exactly, and to $\mathcal{A}$, $\mathcal{A}'$ follows our scheme in Section V-B exactly. Since $\mathcal{A}'$ runs in polynomial-time to security parameter when $\mathcal{A}$ does, and no non-neglibile advantage adversary to $\mathcal{C}$ exists [14], we conclude that no non-negligible advantage adversary $\mathcal{A}$ exists. That is, there exists a negligible function η, such that

$$\mathsf{Adv}^{LCAO}(\mathcal{A}) \leq \eta(\kappa)$$

for security parameter κ. □

VI. Encrypted Localisation Preliminaries

Our next goal is to apply the LCAO secure encryption scheme introduced to a Bayesian localisation filter. The filter we introduce requires the use of real-valued inputs and functions, and relies on a non-linear measurement model. We will make use of a real-number encoding scheme supporting our required homomorphic operations as well as a reformulated Extended Kalman Filter which reduces the filter update step to use only these operations.

A. Integer Encoding for Real Numbers

In the encryption scheme introduced, weights and values are restricted to integers and all operations are computed $(\mathrm{mod}\ N)$, thus bounding meaningful inputs to $\{a \,|\, a \in \mathbb{Z}_N\}$. For this reason, a quantisation and integer mapping method for real numbers is required for their encryption and homomorphic processing. We quantise with a generalised Q number format [40] due to implementation simplicity and applicability.

We define a subset of rational numbers in terms of a range $M \in \mathbb{N}$ and fractional precision $\phi \in \mathbb{N}$. This contrasts with the common definition given in terms of total bits and fractional bits [40], [41], [35], but allows for a direct mapping to integer ranges which are not powers of two. Rational subset $\mathbb{Q}_{M,\phi}$ is given by

$$\mathbb{Q}_{M,\phi} = \left\{ q \,\middle|\, \phi q \in \mathbb{N} \wedge -\left\lfloor \frac{M}{2} \right\rfloor \leq q < \left\lfloor \frac{M}{2} \right\rfloor \right\} , \qquad (20)$$

and we quantise any real number a by taking the nearest rational $q \in \mathbb{Q}_{M,\phi}$. That is, $\arg\min_{q \in \mathbb{Q}_{M,\phi}} |a - q|$. In this form, mapping rationals $\mathbb{Q}_{M,\phi}$ to the encryption scheme range $\mathbb{Z}_N$ is achieved by choosing $M = N$, and handling negatives with

modulo arithmetic. In addition, we note that the Q number format requires a precision factor ϕ to be removed after each encoded multiplication, which is not supported by our encryption scheme. This is captured by a third parameter d; the number of multiplication factors to add or remove from encodings.

The function for combined quantisation and encoding $\mathsf{E}_{M,\phi,d}(a)$, of a given a real number $a \in \mathbb{R}$, integer range $\mathbb{Z}_M$, and the desired scaling for d prior encoded multiplications, is given by

$$\mathsf{E}_{M,\phi,d}(a) = \left\lfloor \phi^{d+1} a \right\rceil \pmod{M}. \tag{21}$$

Decoding of an integer $u \in \mathbb{Z}_M$, is given by

$$\mathsf{E}^{-1}_{M,\phi,d}(u) = \begin{cases} \dfrac{u \,(\mathrm{mod}\ M)}{\phi^{d+1}}, & u \,(\mathrm{mod}\ M) \leq \left\lfloor \dfrac{M}{2} \right\rfloor \\ -\dfrac{M - u \,(\mathrm{mod}\ M)}{\phi^{d+1}}, & \text{otherwise} \end{cases}. \tag{22}$$

This encoding scheme provides the following homomorphic operations,

$$\mathsf{E}_{M,\phi,d}(a_1) + \mathsf{E}_{M,\phi,d}(a_2) \pmod{M} = \mathsf{E}_{M,\phi,d}(a_1 + a_2) \tag{23}$$

and

$$\mathsf{E}_{M,\phi,d}(a_1)\mathsf{E}_{M,\phi,d}(a_2) \pmod{M} = \mathsf{E}_{M,\phi,d+1}(a_1 a_2), \tag{24}$$

noting that when $M = N$, the modulus corresponds with the encrypted homomorphic operations in (9), (10) and therefore (16).

The choice of a high fractional precision ϕ may reduce quantisation errors introduced in (21), however, risks an overflow following too many multiplications. Given the largest number of expected multiplications d_{max}, and the largest expected decoded value a, the parameter should be chosen such that the following condition holds

$$\left|\phi^{d_{max}+1} a\right| < \left\lfloor \frac{M}{2} \right\rfloor. \tag{25}$$

In practice, N is typically very large ($N > 2^{1024}$) and this condition can be ignored when $M = N$.

B. Extended Information Filter

The Extended Information Filter (EIF) [42] is a reformulation of the Extended Kalman Filter (EKF), such that the update equations are simplified to sums of measurement information from each sensor. This requires the conversion of predicted state estimate $\underline{\boldsymbol{x}}_{k|k-1}$ and estimate covariance $\mathbf{P}_{k|k-1}$ to the information vector and matrix

$$\underline{\boldsymbol{y}}_{k|k-1} = \mathbf{P}^{-1}_{k|k-1}\underline{\boldsymbol{x}}_{k|k-1} \quad \text{and} \quad \mathbf{Y}_{k|k-1} = \mathbf{P}^{-1}_{k|k-1}, \tag{26}$$

respectively. In this form, the update equations for n measurements at time k, given process and measurement models (2) and (3), are given by

$$\underline{\boldsymbol{y}}_{k|k} = \underline{\boldsymbol{y}}_{k|k-1} + \sum_{i=1}^{n} \mathbf{H}^{\top}_{k,i} r_i^{-1} (z_{k,i} - h_i(\underline{\boldsymbol{x}}_{k|k-1}) + \mathbf{H}_{k,i}\underline{\boldsymbol{x}}_{k|k-1}) \tag{27}$$

and

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{i=1}^{n} \mathbf{H}^{\top}_{k,i} r_i^{-1} \mathbf{H}_{k,i}, \tag{28}$$

with Jacobians

$$\mathbf{H}_{k,i} = \left.\frac{\partial h_i}{\partial \underline{x}}\right|_{\underline{\boldsymbol{x}}_{k|k-1}} \tag{29}$$

for sensors $1 \leq i \leq n$. The filter's prediction step can be computed by converting information vector $\underline{\boldsymbol{y}}_{k|k}$ and matrix $\mathbf{Y}_{k|k}$ back to state estimate and covariance

$$\underline{\boldsymbol{x}}_{k|k} = \mathbf{Y}^{-1}_{k|k}\underline{\boldsymbol{y}}_{k|k} \quad \text{and} \quad \mathbf{P}_{k|k} = \mathbf{Y}^{-1}_{k|k}, \tag{30}$$

before using the normal EKF prediction equations

$$\underline{\boldsymbol{x}}_{k+1|k} = \underline{f}(\underline{\boldsymbol{x}}_{k|k}) \tag{31}$$

and

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}^{\top}_k, \tag{32}$$

with Jacobian

$$\mathbf{F}_k = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{\boldsymbol{x}}_{k|k}}. \tag{33}$$

VII. Privacy-Preserving Localisation

With the relevant preliminaries, we now apply the LCAO secure scheme to our original localisation problem. Recall we are considering a navigator in the presence of n range sensors. The navigator runs a local filter, computing its update step with equations (27) and (28), where received information from each sensor i consists of the measurement vector

$$\underline{\boldsymbol{i}}_{k,i} = \mathbf{H}^{\top}_{k,i} r_i^{-1} (\boldsymbol{z}_{k,i} - h_i(\underline{\boldsymbol{x}}_{k|k-1}) + \mathbf{H}_{k,i}\underline{\boldsymbol{x}}_{k|k-1}) \tag{34}$$

and the measurement matrix

$$\mathbf{I}_{k,i} = \mathbf{H}^{\top}_{k,i} r_i^{-1} \mathbf{H}_{k,i}, \tag{35}$$

at each time-step k. In this form, all sensitive sensor information is exclusively found in $\underline{\boldsymbol{i}}_{k,i}$ and $\mathbf{I}_{k,i}$. Namely, their measurements $\boldsymbol{z}_{k,i}$, measurement variances r_i and locations $\underline{s}_i$; captured in measurement functions h_i and Jacobians $\mathbf{H}_{k,i}$.

The reason for using a linear-combination aggregation scheme in this scenario is due to computing $\underline{\boldsymbol{i}}_{k,i}$ and $\mathbf{I}_{k,i}$ at the sensors requiring the navigator to disclose some location information in $\underline{\boldsymbol{x}}_{k|k-1}$, for the computing of Jacobians $\mathbf{H}_{k,i}$. Instead, we want to reformulate $\underline{\boldsymbol{i}}_{k,i}$ and $\mathbf{I}_{k,i}$ such that the sensors can use the LCAO secure scheme, with location information in $\underline{\boldsymbol{x}}_{k|k-1}$ as weights from the navigator. As was shown in Section V-B, this guarantees that sensors cannot learn navigator location information $\underline{\boldsymbol{x}}_{k|k-1}$, and that the navigator cannot learn individual sensor values required for computing $\underline{\boldsymbol{i}}_{k,i}$ and $\mathbf{I}_{k,i}$. From (34) and (35) we see that such a linear reformulation will set requirements on the possible measurement functions h_i.

A. Measurement Model Requirements

From the definition of matrix multiplication, we know that matrices whose elements are linear combinations of weights can be multiplied to produce elements of new linear combinations, with new weights as functions dependant only on weights from the original matrices. For example, consider dimension $D \times D$ matrices $\mathbf{A}$ and $\mathbf{B}$, where element (v, w) in matrix $\mathbf{A}$ is of the form $\sum_{i=0}^{W} a_{i,v,w} x_{i,v,w}$ and similarly for matrix $\mathbf{B}$, the form $\sum_{i=0}^{W} b_{i,v,w} y_{i,v,w}$. If we consider $a_{i,v,w}$ and $b_{i,v,w}$ as the weight terms, we can write element (v, w) of a product matrix $\mathbf{AB}$, as a linear combination of functions dependant only on these weights, as

$$\begin{aligned}
&\sum_{d=0}^{D} \left(\left(\sum_{i=0}^{W} a_{i,v,d} x_{i,v,d} \right) \left(\sum_{i=0}^{W} b_{i,d,w} y_{i,d,w} \right) \right) \\
&\quad = \sum_{d=0}^{N} \left(\sum_{i=0}^{W} \sum_{j=0}^{W} a_{i,v,d} x_{i,v,d} b_{j,d,w} y_{j,d,w} \right) \\
&\quad = \sum_{d=0}^{N} \sum_{i=0}^{W} \sum_{j=0}^{W} (a_{i,v,d} b_{j,d,w}) x_{i,v,d} y_{j,d,w} \,.
\end{aligned} \tag{36}$$

Similarly, scalar multiplication also holds this property. It is, therefore, sufficient for h_i and $\mathbf{H}_{k,i}$ to be linear combinations of information private to the navigator, for $\underline{\boldsymbol{i}}_{k,i}$ and $\mathbf{I}_{k,i}$ to be as well. Next, we will show how this can be achieved for the specific case of two-dimensional range-only localisation.

B. Localisation Measurement Model

Recalling the state definition (1) and two-dimensional measurement model (3), we can see that h_i *cannot* be rearranged to a linear combination of elements private to the navigator. In addition, the Jacobian of h_i at $\underline{\boldsymbol{x}}_{k|k-1}$,

$$\mathbf{H}_{k,i} = \begin{bmatrix} \frac{\boldsymbol{x}_{k|k-1} - s_{x,i}}{\sqrt{(\boldsymbol{x}_{k|k-1} - s_{x,i})^2 + (\boldsymbol{y}_{k|k-1} - s_{y,i})^2}} \\ \frac{\boldsymbol{y}_{k|k-1} - s_{y,i}}{\sqrt{(\boldsymbol{x}_{k|k-1} - s_{x,i})^2 + (\boldsymbol{y}_{k|k-1} - s_{y,i})^2}} \end{bmatrix} , \tag{37}$$

cannot be either. Instead, we consider the modified measurement functions

$$h'_i(\underline{\boldsymbol{x}}) = h_i(\underline{\boldsymbol{x}})^2 \,. \tag{38}$$

A measurement function in this form allows rearrangement to a linear combination of navigator information, and thus of the corresponding modified measurement vector $\underline{\boldsymbol{i}}'_{k,i}$ and measurement matrix $\mathbf{I}'_{k,i}$ as well. The modified measurement functions h'_i can be written as

$$\begin{aligned}
h'_i(\underline{\boldsymbol{x}}) &= \left\| \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix}^\top - \underline{s}_i \right\|^2 \\
&= (\boldsymbol{x} - s_{x,i})^2 + (\boldsymbol{y} - s_{y,i})^2 \\
&= \boldsymbol{x}^2 + \boldsymbol{y}^2 - 2 s_{x,i} \boldsymbol{x} - 2 s_{y,i} \boldsymbol{y} + s_{x,i}^2 + s_{y,i}^2
\end{aligned} , \tag{39}$$

and the corresponding Jacobians $\mathbf{H}'_{k,i}$ as

$$\mathbf{H}'_{k,i} = \begin{bmatrix} 2\boldsymbol{x}_{k|k-1} - 2s_{x,i} \\ 2\boldsymbol{y}_{k|k-1} - 2s_{y,i} \end{bmatrix} . \tag{40}$$

From the above we see that $h'_i(\underline{\boldsymbol{x}}_{k|k-1})$ and $\mathbf{H}'_{k,i}$ can be rearranged as linear combinations of the weights $\boldsymbol{x}^2_{k|k-1}$, $\boldsymbol{y}^2_{k|k-1}$, $\boldsymbol{x}_{k|k-1}$ and $\boldsymbol{y}_{k|k-1}$, which capture all the required information that is private to the navigator.

To show how $\underline{\boldsymbol{i}}'_{k,i}$ and $\mathbf{I}'_{k,i}$ can be formulated in a similar manner using (36), we require the existance of a modified measurement model of the form

$$\boldsymbol{z}'_{k,i} = h'_i(\underline{\boldsymbol{x}}_k) + \boldsymbol{v}'_{k,i} \,, \tag{41}$$

where $\boldsymbol{z}'_{k,i}$ is the modified measurement, and noise $\boldsymbol{v}'_{k,i}$ is zero-mean with known variance. The approximation of $\boldsymbol{z}'_{k,i}$ and its noise variance $\boldsymbol{r}_{k,i}$ from original measurements and variances $\boldsymbol{z}_{k,i}$ and r_i will be shown in Section VII-C.

With the existence of (41), measurement vector and measurement matrix linear combinations can be given by

$$\begin{aligned}
\underline{\boldsymbol{i}}'_{k,i} &= \mathbf{H}'^{\top}_{k,i} \boldsymbol{r}^{-1}_{k,i} (\boldsymbol{z}'_{k,i} - h'_i(\underline{\boldsymbol{x}}_{k|k-1}) + \mathbf{H}'_{k,i} \underline{\boldsymbol{x}}_{k|k-1}) \\
&= \begin{bmatrix}
(2\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{x}^3_{k|k-1} + (2\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{x}_{k|k-1} \boldsymbol{y}^2_{k|k-1} \\
\quad + (-\boldsymbol{r}^{-1}_{k,i} s_{x,i}) \boldsymbol{x}^2_{k|k-1} + (-2\boldsymbol{r}^{-1}_{k,i} s_{x,i}) \boldsymbol{y}^2_{k|k-1} \\
\quad + (2\boldsymbol{r}^{-1}_{k,i} \boldsymbol{z}'_{k,i}) \boldsymbol{x}_{k|k-1} + (-2\boldsymbol{r}^{-1}_{k,i} s^2_{x,i}) \boldsymbol{x}_{k|k-1} \\
\quad + (-2\boldsymbol{r}^{-1}_{k,i} s^2_{y,i}) \boldsymbol{x}_{k|k-1} + (2\boldsymbol{r}^{-1}_{k,i} s^3_{x,i}) \\
\quad + (2\boldsymbol{r}^{-1}_{k,i} s_{x,i} s^2_{y,i}) + (-2\boldsymbol{r}^{-1}_{k,i} s_{x,i} \boldsymbol{z}'_{k,i}) \\
(2\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{y}^3_{k|k-1} + (2\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{x}^2_{k|k-1} \boldsymbol{y}_{k|k-1} \\
\quad + (-2\boldsymbol{r}^{-1}_{k,i} s_{y,i}) \boldsymbol{x}^2_{k|k-1} + (-2\boldsymbol{r}^{-1}_{k,i} s_{y,i}) \boldsymbol{y}^2_{k|k-1} \\
\quad + (2\boldsymbol{r}^{-1}_{k,i} \boldsymbol{z}'_{k,i}) \boldsymbol{y}_{k|k-1} + (-2\boldsymbol{r}^{-1}_{k,i} s^2_{x,i}) \boldsymbol{y}_{k|k-1} \\
\quad + (-2\boldsymbol{r}^{-1}_{k,i} s^2_{y,i}) \boldsymbol{y}_{k|k-1} + (2\boldsymbol{r}^{-1}_{k,i} s_{y,i} s^2_{x,i}) \\
\quad + (2\boldsymbol{r}^{-1}_{k,i} s^3_{y,i}) + (-2\boldsymbol{r}^{-1}_{k,i} s_{y,i} \boldsymbol{z}'_{k,i})
\end{bmatrix}
\end{aligned} \tag{42}$$

and

$$\begin{aligned}
\mathbf{I}'_{k,i} &= \mathbf{H}'^{\top}_{k,i} \boldsymbol{r}^{-1}_{k,i} \mathbf{H}'_{k,i} \\
&= \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} ,
\end{aligned} \tag{43}$$

with

$$\begin{aligned}
\beta_{11} &= (4\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{x}^2_{k|k-1} + (-8\boldsymbol{r}^{-1}_{k,i} s_{x,i}) \boldsymbol{x}_{k|k-1} + (4\boldsymbol{r}^{-1}_{k,i} s^2_{x,i}) \\
\beta_{12} &= (4\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{x}_{k|k-1} \boldsymbol{y}_{k|k-1} + (-4\boldsymbol{r}^{-1}_{k,i} s_{y,i}) \boldsymbol{x}_{k|k-1} \\
&\quad + (-4\boldsymbol{r}^{-1}_{k,i} s_{x,i}) \boldsymbol{y}_{k|k-1} + (4\boldsymbol{r}^{-1}_{k,i} s_{x,i} s_{y,i}) \\
\beta_{21} &= \beta_{12} \\
\beta_{22} &= (4\boldsymbol{r}^{-1}_{k,i}) \boldsymbol{y}^2_{k|k-1} + (-8\boldsymbol{r}^{-1}_{k,i} s_{y,i}) \boldsymbol{y}_{k|k-1} + (4\boldsymbol{r}^{-1}_{k,i} s^2_{y,i}) \,.
\end{aligned}$$

The above rearrangements result in $\underline{\boldsymbol{i}}'_{k,i}$ and $\mathbf{I}'_{k,i}$ as linear comabinations of the weights

$$\begin{aligned}
&\langle \boldsymbol{x}^3_{k|k-1},\ \boldsymbol{y}^3_{k|k-1},\ \boldsymbol{x}^2_{k|k-1} \boldsymbol{y}_{k|k-1},\ \boldsymbol{x}_{k|k-1} \boldsymbol{y}^2_{k|k-1}, \\
&\quad \boldsymbol{x}^2_{k|k-1},\ \boldsymbol{y}^2_{k|k-1},\ \boldsymbol{x}_{k|k-1} \boldsymbol{y}_{k|k-1},\ \boldsymbol{x}_{k|k-1},\ \boldsymbol{y}_{k|k-1} \rangle \,,
\end{aligned} \tag{44}$$

which again capture all the required information private to the navigator.

The final step required for the application of our LCAO secure encryption scheme to the linear combinations (42) and (43) is the handling of the scheme's instance variable t in (16). As six linear-combination aggregations occur at each time-step k (four elements in $\mathbf{I}'_{k,i}$ and two in $\underline{\boldsymbol{i}}'_{k,i}$), it is required for the security of the scheme that t be unique for each aggregation at each time. This is handled by setting t to the concatenation

$$t = k \parallel v \parallel w \parallel \tau \tag{45}$$

for aggregation in row v and column w, with $\tau = 0$ for aggregations in $\underline{\boldsymbol{i}}'_{k,i}$ and $\tau = 1$ otherwise.

Remark. The solutions (42) and (43) have been derived for two-dimensional localisation, but can be similarly extended to the three-dimensional case. We note however, the additional cost with increasing dimension. The number of weights is increased in the rearranged functions h'_i, and therefore combinatorially increased in $\underline{\boldsymbol{i}}'_{k,i}$ and $\mathbf{I}'_{k,i}$ by (36). This results in a combinatorial increase in weights, and therefore communication, with respect to the number of state parameters required for computing functions h_i.

C. Range Measurement Modification

In Section VII-B, we assumed the existance of a measurement model (41) where modified sensor measurements $\boldsymbol{z}'_{k,i}$ had zero-mean noises with variances $\boldsymbol{r}_{k,i}$. In practice, conversion of range measurements $\boldsymbol{z}_{k,i}$ to $\boldsymbol{z}'_{k,i}$ is complicated by the noise term $\boldsymbol{v}_{k,i} \sim \mathcal{N}(0, r_i)$ in (3). Squaring the range measurement produces

$$\begin{aligned} \boldsymbol{z}^2_{k,i} &= (h_i(\underline{\boldsymbol{x}}_k) + \boldsymbol{v}_{k,i})^2 \\ &= h_i(\underline{\boldsymbol{x}}_k)^2 + 2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i} \\ &= h'_i(\underline{\boldsymbol{x}}_k) + 2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i}\,, \end{aligned} \tag{46}$$

with new noise term, $2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i}$, now dependant on the measurement function h_i and no longer zero-mean or Gaussian. We can compute the mean of the new noise term (a function of the Gaussian term $\boldsymbol{v}_{k,i}$) as

$$\mathrm{E}\left\{2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i}\right\} = r_i \tag{47}$$

and define modified measurements as

$$\begin{aligned} \boldsymbol{z}'_{k,i} &= \boldsymbol{z}^2_{k,i} - r_i \\ &= h_i(\underline{\boldsymbol{x}}_k)^2 + 2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i} - r_i \\ &= h'_i(\underline{\boldsymbol{x}}_k) + \boldsymbol{v}'_{k,i}\,, \end{aligned} \tag{48}$$

with now zero-mean noise $\boldsymbol{v}'_{k,i} = 2h_i(\underline{\boldsymbol{x}}_k)\boldsymbol{v}_{k,i} + \boldsymbol{v}^2_{k,i} - r_i$. The noise in this case (again a function of white Gaussian term $\boldsymbol{v}_{k,i}$) has variance

$$\mathrm{Var}\left\{\boldsymbol{v}'_{k,i}\right\} = 4h_i(\underline{\boldsymbol{x}}_k)^2 r_i + 2r_i^2 \tag{49}$$

and is dependant on $h_i(\underline{\boldsymbol{x}}_k)$. To use the modified measurement function (48) with an EIF, we require an estimate of its noise variance using only the values available to the sensor, $\boldsymbol{z}_{k,i}$ and r_i. A conservative estimate of the variance (*i.e.* a larger estimate resulting in less confidence in measurements) is desirable to avoid increasing the likeliness of filter divergence. While replacing $h_i(\underline{\boldsymbol{x}}_k)$ with $\boldsymbol{z}_{k,i}$ only provides a conservative estimate when

$$\begin{aligned} & \boldsymbol{z}_{k,i} > h_i(\underline{\boldsymbol{x}}_k) \\ \implies & \boldsymbol{z}_{k,i} - h_i(\underline{\boldsymbol{x}}_k) > 0 \\ \implies & \boldsymbol{v}_{k,i} > 0\,, \end{aligned} \tag{50}$$

and cannot be guaranteed, we can instead provide a conservative estimate with 95% confidence, by shifting measurement $\boldsymbol{z}_{k,i}$ by two standard deviations $\sqrt{r_i}$. Thus, the modified measurement variance for sensor i at time k is conservatively approximated by

$$\boldsymbol{r}_{k,i} = 4(\boldsymbol{z}_{k,i} + 2\sqrt{r_i})^2 r_i + 2r_i^2\,. \tag{51}$$

Together, the model and variance, (48) and (51), provide a suitable substitute for the required measurement model (41) and can be used to achieve the desired linear properties using only the available range sensors.

D. Algorithm

In this section, we piece together our LCAO secure encryption scheme in Section V-B with the modified measurement model in Sections VII-B and VII-C and define our privacy-preserving localisation algorithm.

The privacy-preserving localisation filter can be described by the following steps.

Setup The Setup algorithm from Section V-B is executed by a trusted third party, N and H are made known to the navigator and all sensors, and the navigator and sensor secret keys, $sk_0 = \lambda = \mathsf{lcm}(p-1, q-1)$ and sk_i, $1 \leq i \leq n$ respectively, are distributed accordingly. Additionally, we assume that fractional precision ϕ from Section VI-A is also a public parameter and made known to the navigator and sensors. We will thus simplify the encoding notation $\mathsf{E}_{N,\phi,d}(\cdot)$ to $\mathsf{E}_d(\cdot)$.

Prediction The navigator computes the typical EKF prediction equations before encrypting state variables and broadcasting to sensors. This is given by Algorithm 1.

Measurement Sensors modify their measurements before homomorphically computing measurement vectors and matrices $\underline{\boldsymbol{i}}'_{k,i}$ and $\mathbf{I}'_{k,i}$, encrypting them for aggregation, and sending them back to the navigator. Care must be taken when encoding during homomorphic operations to ensure fractional factor d satisfies (23), in particular when using the property from Remark V-B. This is described by Algorithm 2.

Update The navigator aggregates measurement vectors and matrices before decrypting them. The typical EIF update equations can then be computed. This is shown in Algorithm 3.

Algorithm 1 Navigator Prediction

procedure PREDICTION($\underline{\boldsymbol{x}}_{k-1|k-1}$, $\mathbf{P}_{k-1|k-1}$, $\underline{f}$, $\mathbf{Q}$, N)
  Compute $\mathbf{F}_k$ by (33)
  $\underline{\boldsymbol{x}}_{k|k-1} \leftarrow \underline{f}(\underline{\boldsymbol{x}}_{k-1|k-1})$
  $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}$
  Compute $\mathsf{E}_0(\boldsymbol{x}^3_{k|k-1})$ by (21)
  Compute $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^3_{k|k-1}))$ by (15)
  Broadcast $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^3_{k|k-1}))$ to sensors
  for Remaining weights in (44) **do**
    Broadcast weight in the form above
  end for
  return $\underline{\boldsymbol{x}}_{k|k-1}, \mathbf{P}_{k|k-1}$
end procedure

Algorithms 1, 2 and 3 have been summarised graphically in Figure 4, where encryptions and encodings of vectors and

Algorithm 2 Measurement at Sensor i

procedure MEASUREMENT(i, $s_{x,i}$, $s_{y,i}$, r_i, N, H)
  Measure $\boldsymbol{z}_{k,i}$
  Compute $\boldsymbol{z}'_{k,i}$ by (48)
  Compute $\boldsymbol{r}_{k,i}$ by (51)
  Recieve $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^3_{k|k-1}))$
  for Remaining weights in (44) **do**
    Recieve weight in the form above
  end for
  Let $\alpha_v^{(i)}$ represent an encryption of element v in $\underline{\boldsymbol{i}}'_{k,i}$ from (42)
  $\alpha_1^{(i)} \leftarrow \mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^3_{k|k-1}))^{\mathsf{E}_0(2\boldsymbol{r}^{-1}_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}_{k|k-1}\boldsymbol{y}^2_{k|k-1}))^{\mathsf{E}_0(2\boldsymbol{r}^{-1}_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^2_{k|k-1}))^{\mathsf{E}_0(-\boldsymbol{r}^{-1}_{k,i}s_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{y}^2_{k|k-1}))^{\mathsf{E}_0(-2\boldsymbol{r}^{-1}_{k,i}s_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}_{k|k-1}))^{\mathsf{E}_0(2\boldsymbol{r}^{-1}_{k,i}\boldsymbol{z}'_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}_{k|k-1}))^{\mathsf{E}_0(-2\boldsymbol{r}^{-1}_{k,i}s^2_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}_{k|k-1}))^{\mathsf{E}_0(-2\boldsymbol{r}^{-1}_{k,i}s^2_{y,i})}\cdot$
  $(N+1)^{\mathsf{E}_1(2\boldsymbol{r}^{-1}_{k,i}s^3_{x,i})}(N+1)^{\mathsf{E}_1(2\boldsymbol{r}^{-1}_{k,i}s_{x,i}s^2_{y,i})}\cdot$
  $(N+1)^{\mathsf{E}_1(-2\boldsymbol{r}^{-1}_{k,i}s_{x,i}\boldsymbol{z}'_{k,i})}H(k \,\|\, 1 \,\|\, 1 \,\|\, 0) \pmod{N^2}$
  Compute $\alpha_2^{(i)}$ using (42), (16) and Remark V-B in the form above
  for $v \leftarrow 1$ to 2 **do**
    Send α_v to the navigator
  end for
  Let $\beta_{vw}^{(i)}$ represent an encryption of element (v, w) in $\mathbf{I}'_{k,i}$ from (43)
  $\beta_{11}^{(i)} \leftarrow \mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}^2_{k|k-1}))^{\mathsf{E}_0(4\boldsymbol{r}^{-1}_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\boldsymbol{x}_{k|k-1}))^{\mathsf{E}_0(-8\boldsymbol{r}^{-1}_{k,i}s_{x,i})}\cdot$
  $(N+1)^{\mathsf{E}_1(4\boldsymbol{r}^{-1}_{k,i}s^2_{x,i})}\cdot$
  $H(k \,\|\, 1 \,\|\, 1 \,\|\, 1) \pmod{N^2}$
  Compute remaining $\beta_{vw}^{(i)}$ using (43), (16) and Remark V-B in the form above
  for $v \leftarrow 1$ to 2 **do**
    for $w \leftarrow 1$ to 2 **do**
      Send $\beta_{vw}^{(i)}$ to the navigator
    end for
  end for
end procedure

Algorithm 3 Navigator Update

procedure UPDATE($\underline{\boldsymbol{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, N, λ)
  for $v \leftarrow 1$ to 2 **do**
    Receive $\alpha_v^{(i)}$ from each sensor $1 \le i \le n$
  end for
  for $v \leftarrow 1$ to 2 **do**
    for $w \leftarrow 1$ to 2 **do**
      Receive $\beta_{vw}^{(i)}$ from each sensor $1 \le i \le n$
    end for
  end for
  Let α_v represent an encryption of element v in $\sum_{i=1}^n \underline{\boldsymbol{i}}'_{k,i}$
  for $v \leftarrow 1$ to 2 **do**
    $\alpha_v \leftarrow \prod_{i=1}^n \alpha_v^{(i)}$
    Compute $\mathcal{D}_{sk_0}(\alpha_v)$ with λ by (17)
    Compute $\mathsf{E}_1^{-1}(\mathcal{D}_{sk_0}(\alpha_v))$ by (22)
  end for
  Construct $\sum_{i=1}^n \underline{\boldsymbol{i}}'_{k,i}$ from decoded decryptions above
  Let β_{vw} represent an encryption of element (v, w) in $\sum_{i=1}^n \mathbf{I}'_{k,i}$
  for $v \leftarrow 1$ to 2 **do**
    for $w \leftarrow 1$ to 2 **do**
      $\beta_{vw} \leftarrow \prod_{i=1}^n \beta_{vw}^{(i)}$
      Compute $\mathcal{D}_{sk_0}(\beta_{vw})$ with λ by (17)
      Compute $\mathsf{E}_1^{-1}(\mathcal{D}_{sk_0}(\beta_{vw}))$ by (22)
    end for
  end for
  Construct $\sum_{i=1}^n \mathbf{I}'_{k,i}$ from decoded decryptions above
  $\underline{\boldsymbol{y}}_{k|k} \leftarrow \mathbf{P}^{-1}_{k|k-1}\underline{\boldsymbol{x}}_{k|k-1} + \sum_{i=1}^n \underline{\boldsymbol{i}}'_{k,i}$
  $\mathbf{Y}_{k|k} \leftarrow \mathbf{P}^{-1}_{k|k-1} + \sum_{i=1}^n \mathbf{I}'_{k,i}$
  $\underline{\boldsymbol{x}}_{k|k} \leftarrow \mathbf{Y}^{-1}_{k|k}\underline{\boldsymbol{y}}_{k|k}$
  $\mathbf{P}_{k|k} \leftarrow \mathbf{Y}^{-1}_{k|k}$
  return $\underline{\boldsymbol{x}}_{k|k}, \mathbf{P}_{k|k}$
end procedure

matrices have been denoted with $\mathcal{E}_{pk_0,sk_i}(\cdot)$ and $\mathsf{E}_m(\cdot)$ for brevity, and represent element-wise operations with the same parameters.

E. Leakage

With the algorithm defined, we can analyse the localisation leakage when the strictly defined LCAO leakage is taken into account. We recall the assumptions on participant capabilities made in Section III-B. Consistent broadcasting is in line with the LCAO definition and required to guarantee the scheme's leakage. Honest-but-curious sensors are not required for scheme privacy, but avoid the possibility of malicious sensors gaining prior knowledge about remaining sensor measurements. This is possibly achieved by detecting changes in real measurements after reporting false measurements to the navigator. As stated in Section III-B, misbehaving sensors are a known problem which complicates homomorphic operations and active malicious sensors are not considered in the scope of this work.

The aggregation operation of our encryption scheme leaks the sums of sensor measurement vectors and matrices $\sum_{i=1}^n \underline{\boldsymbol{i}}'_{k,i}$ and $\sum_{i=1}^n \mathbf{I}'_{k,i}$. However, as stated in Remark V, weights chosen by a corrupted navigator mean that individual sums of coefficients can be leaked as well. That is, sums of coefficients in (42) and (43) (*i.e.* $\sum_{i=1}^n 2\boldsymbol{r}^{-1}_{k,i}$, $\sum_{i=1}^n -\boldsymbol{r}^{-1}_{k,i}s_{x,i}$, $\sum_{i=1}^n -2\boldsymbol{r}^{-1}_{k,i}s_{x,i}$, $\sum_{i=1}^n 2\boldsymbol{r}^{-1}_{k,i}\boldsymbol{z}'_{k,i}$, …) are leaked to a corrupted navigator.

From the leaked coefficient sums, we see that all information private to the sensors and broadcast to the navigator, namely their modified measurements $\boldsymbol{z}'_{k,i}$, variance estimates $\boldsymbol{r}_{k,i}$ and locations $\underline{s}_i$, are present only in the sums

$$\sum_{i=1}^n \boldsymbol{z}'_{k,i}\,,\ \sum_{i=1}^n \boldsymbol{r}_{k,i}\,,\ \sum_{i=1}^n s_{x,i} \text{ and } \sum_{i=1}^n s_{y,i}\,. \tag{52}$$

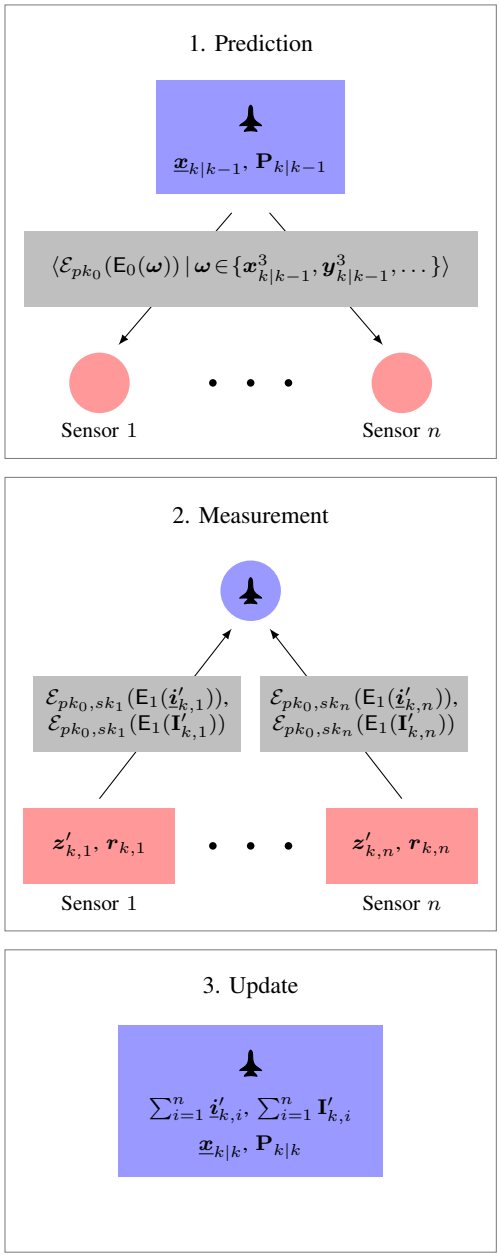

Fig. 4. The summary of steps involved in our proposed privacy-preserving EIF.

From these sums and the definitions (48) and (51), we conclude that the navigator can *at most* learn the sums of sensor private data, which in practice corresponds to the average sensor modified measurements, estimated variances and locations.

Theorem VII.1. In the context of our localisation scheme, leakage of the incorporated LCAO secure encryption scheme corresponds to the leakage of average sensor private information, given by the sums in (52).

VIII. Simulation and Results

To demonstrate our proposed approach, we have implemented the algorithm in Section VII-D and simulated the measurement and navigation of an object following a linear, time-invariant, process model in two-dimensional space.

Code was written in the C programming language, using the MPI library [43] to support simultaneous computations by sensors and navigator as different processes. The OpenSSL library's [44] mask generation function MGF1 and hash function SHA256 were used to implement the required hash function H, while the Libpaillier library [45] was used for the implementation of the Paillier encryption scheme. Additionally, GNU libraries GSL [46] and GMP [47] were used for algebraic operations and the handling of multiple-precision integers, respectively. Simulation execution was performed on a 3.00GHz Intel i7-9700 CPU, and run on the Windows Subsystem for Linux (WSL), where no real-time kernel was ensured but sufficiently consistent runtimes were observed for the purposes of our simulation.

Recalling the dependence of the modified measurement noise on the true measurement in (48), the noise distribution of the model changes with varying sizes of measurements. We considered four layouts with varying sensor distances when running our simulations to observe differences in estimation error when measurements are of different sizes. The considered layouts have been plotted and named in Figure 5. To evaluate the accuracy of our method, we have run 100 simulations, all with 50 filter iterations, for each of the sensors layouts with sensor measurement variances set to $r_i = 5$ simulation distance units. Due to the large encoding modulus N, no notable advantage is gained when choosing a low fractional precision factor ϕ, thus, $\phi = 2^{32}$ was chosen to ensure minimal loss of precision when compared to the floating-point computations in the unencrypted EIF. The plotted average RMSE of our proposed method as well as that of the unencrypted EIF filter can be seen in Figure 6. From the results, it can be seen that sensor distance has little effect on the performance of the filter and that the measurement modification from Section VII-C has a negligible effect when compared to a normal EIF filter. The robustness of the privacy-preserving method can be attributed to the conservativeness of variance estimates in (51) and the sufficiently large fractional precision parameter.

In addition to accuracy, computational requirements are an important factor in the adoption of novel cryptographic methods. Figure 7 shows the average simulation runtimes when encryption key bit lengths (*i.e.* bit lengths of N) and the number of sensors are varied. Every plotted time is again the average of 100 simulations, each running for 50 filter iterations. As sensor and navigator computations run in parallel, increases in runtime from additional sensors are primarily due to the longer aggregation operation at the navigator. The predominant computational cost is dependant on the encryption scheme key size, which must be chosen such that sufficient security is provided by the scheme. The current recommendation for a secure implementation relying on prime factorisation (difficulty of factorising N) is to use

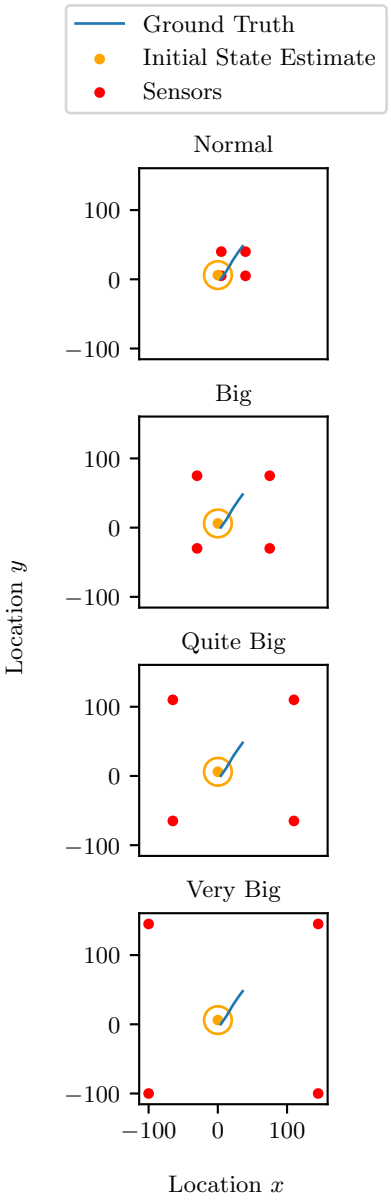

Fig. 5. The considered layouts.

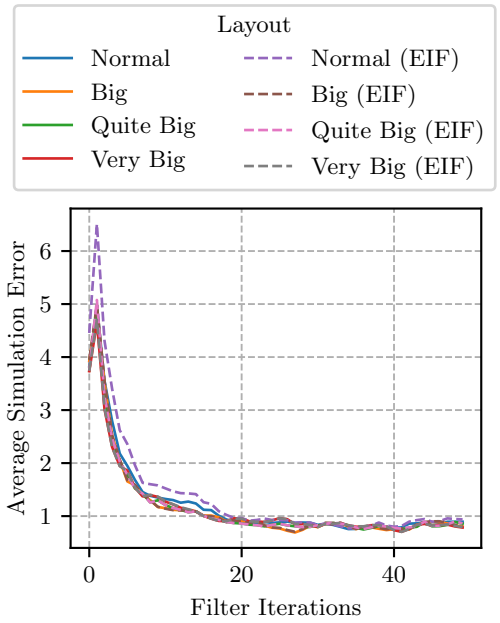

Fig. 6. Filter error for different layouts.

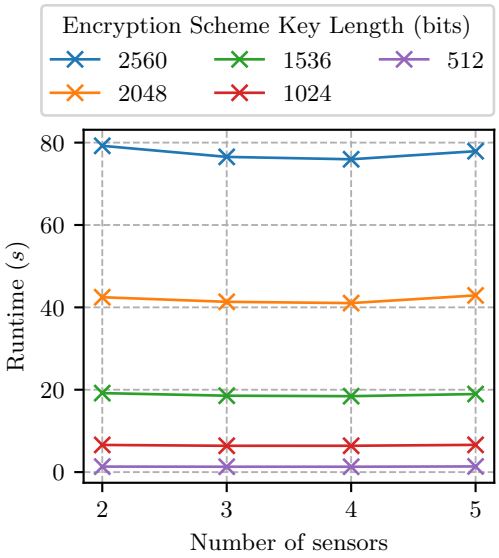

Fig. 7. Runtimes for varying key sizes and numbers of sensors.

2048 bit-long keys [48].

IX. Conclusion

To develop a model-based localisation algorithm which preserves navigator and sensor privacy, we first defined the novel notion of an LCAO secure encryption scheme and gave a provably secure implementation based on the Joye-Libert aggregation scheme. This scheme was then used in our proposed privacy-preserving EIF, where sensors compute measurement information homomorphically such that it can be privately used by a navigator. Our privacy-preserving estimation method may find uses in a variety of untrusted distributed localisation environments including airspaces and autonomous vehicle networks as well as those where alternative measurement models satisfying our defined linearity requirements can be applied. Possible future work in this topic includes the expanding of the LCAO security notion to *ensure* that the same weights are broadcast to all sensors, the application of LCAO secure schemes to alternative measurement models and the exploring of implications behind an active sensor attacker model.

Appendix A
Indistinguishability under Chosen Plaintext Attack (IND-CPA)

A public-key encryption scheme is defined by the tuple of algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$, defined as

$\mathsf{Setup}(\kappa)$ On input of security paramater κ, generate public key pk and secret key sk

$\mathsf{Enc}(pk, a)$ Encryption of value a is computable using the public key pk, obtaining $\mathcal{E}_{pk}(a)$.

$\mathsf{Dec}(sk, \mathcal{E}_{pk}(a))$ Decryption of value a is computable using the secret key sk.

The security game between attacker and challenger for IND-CPA security is given by

Setup The challenger chooses security parameter κ, runs the $\mathsf{Setup}(\kappa)$ algorithm and gives public key pk to the attacker

Encryptions The attacker may compute encryptions using the public key pk.

Challenge Next, the attacker chooses two values

$$a^{(0)} \text{ and } a^{(1)}$$

and gives them to the challenger. The challenger then chooses a random bit $b \in \{1, 0\}$ and returns the encryption

$$\mathcal{E}_{pk}(a^{(b)}).$$

More Encryptions The attacker can now compute more encryptions with the public key pk.

Guess At the end, the attacker outputs a bit b' and wins the game if and only if $b' = b$. The advantage of an attacker $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{IND-CPA}(\mathcal{A}) \coloneqq \left| \mathrm{P}[b' = b] - \frac{1}{2} \right| .$$

Appendix B
Aggregator Obliviousness (AO)

An aggregator oblivious encryption scheme is defined by the tuple of algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{AggDec})$, defined as

$\mathsf{Setup}(\kappa)$ On input of security paramater κ, generate public parameters pub, the user private keys sk_i, $1 \leq i \leq n$, and the aggregator's private key $sk_0 = -\sum_{i=1}^{n} sk_i$.

$\mathsf{Enc}(t, sk_i, a_i^{(t)})$ At instance t, user i computes and obtains the encrypted value $l_i^{(t)} = \mathcal{E}_{sk_i}(a_i^{(t)})$ using its secret key sk_i.

$\mathsf{AggDec}(t, sk_0, l_1^{(t)}, ..., l_n^{(t)})$ At instance t, the aggregator computes the aggregation of values $\sum_{i=1}^{n} a_i^{(t)}$ using its private key sk_0.

The security game between attacker and challenger for AO security is given by

Setup The challenger chooses security parameter κ, runs the $\mathsf{Setup}(\kappa)$ algorithm and gives public parameters pub to the attacker

Queries The attacker can now submit queries that are answered by the challenger. The types of queries are:

1) *Combine Queries:* The attacker chooses a tuple $(i, t, a_i^{(t)}$ such that for any two chosen combine query tuples $(i, t, a_i^{(t)})$ and $(i', t', a_{i'}^{\prime(t')}$, the following condition holds:

$$i = i' \wedge t = t' \implies a_i^{(t)} = a_{i'}^{\prime(t')} .$$

The attacker is then given back the encryption of the value $\mathcal{E}_{sk_i}(a_i^{(t)})$ encrypted under the secret key sk_i.

2) *Compromise queries:* The attacker chooses i and receives the secret key sk_i. The aggregator's secret key may also be compromised (when choosing $i = 0$).

Challenge Next, the attacker chooses an instance t^*, and a subset of users $S \subseteq U$ where U is the complete set of users for which no combine queries, for the instance t^*, and no compromise queries, are made for the duration of the game. The attacker then chooses two series of tuples

$$\left\langle (i, t^*, a_i^{(t^*)(0)}) \,\middle|\, i \in S \right\rangle$$

and

$$\left\langle (i, t^*, a_i^{(t^*)(1)}) \,\middle|\, i \in S \right\rangle ,$$

and gives them to the challenger. In the case that $0 \in S$ (*i.e.* the aggregator is compromised) and $S = U$, it is additionally required that

$$\sum_{i \in S} a_i^{(t^*)(0)} = \sum_{i \in S} a_i^{(t^*)(1)} .$$

The challenger then chooses a random bit $b \in \{1, 0\}$ and returns encryptions

$$\left\langle \mathcal{E}_{sk_i}(a_i^{(t^*)(b)}) \,\middle|\, i \in S \right\rangle .$$

More Queries The attacker can now submit more queries, so long as the queries do not break the requirements in the Challenge stage. That is, $S \subseteq U$.

Guess At the end, the attacker outputs a bit b' and wins the game if and only if $b' = b$. The advantage of an attacker $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{AO}(\mathcal{A}) \coloneqq \left| \mathrm{P}[b' = b] - \frac{1}{2} \right| .$$

Acknowledgment

Partially supported by ZIM ...

References

[1] J. Pierce, "An Introduction to Loran," *Proceedings of the IRE*, vol. 34, no. 5, pp. 216–234, 1946.

[2] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, "Contributed Review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements," *Review of Scientific Instruments*, vol. 87, no. 4, pp. 921–960, 2016.

[3] Q. Wang, Z. Duan, X. R. Li, and U. D. Hanebeck, "Convex Combination for Source Localization Using Received Signal Strength Measurements," in *21st International Conference on Information Fusion (Fusion 2018)*. Cambridge, UK: IEEE, 2018, pp. 323–330.

[4] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," in *9th Annual International Conference on Mobile Computing and Networking*, 2003, pp. 81–95.

[5] F. Beutler and U. Hanebeck, "A New Nonlinear Filtering Technique for Source Localization," in *3rd IEEE Conference on Sensors (Sensors 2004)*, vol. 1, 2004, pp. 413–416.

[6] B. Siebler, S. Sand, and U. D. Hanebeck, "Localization with Magnetic Field Distortions and Simultaneous Magnetometer Calibration," *IEEE Sensors Journal*, pp. 1–1, 2020.

[7] M. Brenner, J. Wiebelitz, G. von Voigt, and M. Smith, "Secret Program Execution in the Cloud Applying Homomorphic Encryption," in *5th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2011, pp. 114–119.

[8] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[9] S. Han and G. J. Pappas, "Privacy in Control and Dynamical Systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 309–332, 2018.

[10] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*, ser. Lecture Notes in Computer Science. Springer, 2008, pp. 1–19.

[11] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-Indistinguishability: Differential Privacy for Location-Based Systems," in *ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 901–914.

[12] E. Shi, T.-H. H. Chan, and E. Rieffel, "Privacy-Preserving Aggregation of Time-Series Data," *Annual Network & Distributed System Security Symposium (NDSS)*, p. 17, 2011.

[13] T. H. H. Chan, E. Shi, and D. Song, "Privacy-Preserving Stream Aggregation with Fault Tolerance," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science. Springer, 2012, pp. 200–214.

[14] M. Joye and B. Libert, "A Scalable Scheme for Privacy-Preserving Aggregation of Time-Series Data," in *International Conference on Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science. Springer, 2013, pp. 111–125.

[15] F. Benhamouda, M. Joye, and B. Libert, "A New Framework for Privacy-Preserving Aggregation of Time-Series Data," *ACM Transactions on Information and System Security*, vol. 18, no. 3, pp. 10:1–10:21, 2016.

[16] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *41st ACM Symposium on Theory of Computing (STOC)*, 2009, pp. 169–178.

[17] D. Stehlé and R. Steinfeld, "Faster Fully Homomorphic Encryption," in *Advances in Cryptology (ASIACRYPT)*, ser. Lecture Notes in Computer Science, vol. 6477, 2010, pp. 377–394.

[18] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

[19] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[20] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology (EUROCRYPT)*. Springer, 1999, pp. 223–238.

[21] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Theory of Cryptography Conference*, ser. Lecture Notes in Computer Science. Springer, 2005, pp. 325–341.

[22] K. Kogiso and T. Fujita, "Cyber-Security Enhancement of Networked Control Systems Using Homomorphic Encryption," in *54th IEEE Conference on Decision and Control (CDC)*, vol. 54, 2015, pp. 6836–6843.

[23] F. Kerschbaum, "Outsourced Private Set Intersection Using Homomorphic Encryption," in *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2012, p. 85.

[24] A. B. Alexandru, M. S. Darup, and G. J. Pappas, "Encrypted Cooperative Control Revisited," in *58th IEEE Conference on Decision and Control (CDC)*, 2019, pp. 7196–7202.

[25] D. Boneh, A. Sahai, and B. Waters, "Functional Encryption: Definitions and Challenges," in *Theory of Cryptography Conference*, ser. Lecture Notes in Computer Science. Springer, 2011, pp. 253–273.

[31] A. Alanwar, Y. Shoukry, S. Chakraborty, P. Martin, P. Tabuada, and M. Srivastava, "PrOLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption," in *16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 41–52.

[26] S. Goldwasser, S. D. Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, and H.-S. Zhou, "Multi-Input Functional Encryption," in *Advances in Cryptology (EUROCRYPT)*, ser. Lecture Notes in Computer Science. Springer, 2014, pp. 578–602.

[27] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Functional Encryption: New Perspectives and Lower Bounds," in *Advances in Cryptology (CRYPTO)*, ser. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, 2013, pp. 500–518.

[28] J. Chotard, E. Dufour Sans, R. Gay, D. H. Phan, and D. Pointcheval, "Decentralized Multi-Client Functional Encryption for Inner Product," in *Advances in Cryptology (ASIACRYPT)*, ser. Lecture Notes in Computer Science. Springer, 2018, pp. 703–732.

[29] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[30] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC press, 1998.

[32] L. Lazos and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," in *ACM Workshop on Wireless Security (WiSe)*. Philadelphia, PA, USA: ACM, 2004, p. 21.

[33] M. Aristov, B. Noack, U. D. Hanebeck, and J. Müller-Quade, "Encrypted Multisensor Information Filtering," in *21st International Conference on Information Fusion (Fusion 2018)*, Cambridge, UK, 2018, pp. 1631–1637.

[34] F. Farokhi, Ed., *Privacy in Dynamical Systems*. Springer, 2020.

[35] F. Farokhi, I. Shames, and N. Batterham, "Secure and Private Control Using Semi-Homomorphic Encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, 2017.

[36] Google, "Encrypted-bigquery-client," https://github.com/google/encrypted-bigquery-client, 2015.

[37] I. Ben-Gal, "Outlier Detection," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2005, pp. 131–146.

[38] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall, 2008.

[39] M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, J. Hoffstein, K. Lauter, S. Lokam, D. Moody, T. Morrison, and A. Sahai, "Security of Homomorphic Encryption," *Technical Report, HomomorphicEncryption.org, Redmond WA, USA*, 2017.

[40] E. L. Oberstar, *Fixed-Point Representation and Fractional Math*. Oberstar Consulting, 2007.

[41] M. Schulze Darup, A. Redder, and D. E. Quevedo, "Encrypted Cooperative Control Based on Structured Feedback," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 37–42, 2019.

[42] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1982.

[43] The OpenMPI Project, "Open MPI," https://www.open-mpi.org/, 2020.

[44] The OpenSSL Project, "OpenSSL," https://www.openssl.org/, 2020.

[45] J. Bethencourt, "Libpaillier," http://acsc.cs.utexas.edu/libpaillier/, 2010.

[46] The GSL development team, "GSL - GNU Scientific Library," https://www.gnu.org/software/gsl/, 2019.

[47] T. Granlund and the GMP development team, "GMP - The GNU Multiple Precision Arithmetic Library," https://gmplib.org/, 2020.

[48] E. Barker, L. Chen, A. Roginsky, A. Vassilev, R. Davis, and S. Simon, "Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-56Br2, Mar. 2019.