

# Privacy-Preserving Localization Using Private Linear-Combination Aggregation

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

**Abstract**—Distributed state estimation and localization 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 localization protocols, thus not preserving participants’ data privacy. Existing approaches that do provide such guarantees fail to address sensor and navigator privacy in common non-linear measurement model applications and therefore forfeit broad applicability. We define a suitable notion of linear-combination aggregation encryption and provide a cryptographically secure instance that is applied to the Extended Kalman Filter with range-sensor measurements. Our method keeps navigator location, sensors’ locations, and sensors’ measurements private during navigation, and has been implemented to evaluate its accuracy and performance. The novel computationally plausible and provably secure range-based localization filter has direct applications to environments where participants are not trusted or data is considered sensitive.

**Index Terms**—State Estimation, Data Privacy, Sensor Fusion, Extended 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 localization 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 localization methods, including the use of radar measurements, GPS clocks, and signal strength measurements, have found particularly large applications due to their robustness and accuracy. These methods use distance measurements 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 localization 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 neighboring 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 the private aggregation of weighted sums, before developing a computationally plausible localization filter in the presence of range-only sensors using this scheme, such that sensor and navigator privacy is preserved.

The remainder of the paper is structured as follows. In Section we give an overview of the relevant literature to the topic and in Section we introduce the localization problem, the relevant aggregation operation for its solution and the restrictions on involved parties. Sections and will summarize cryptographic preliminaries and introduce a novel security notion for our required aggregation before defining and proving an encryption scheme which satisfies it. Sections and introduce localization preliminaries and our privacy-preserving localization method using the private aggregation scheme. We show and discuss simulation results in Section and conclude our work in Section .

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

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 formalize 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 col-

cluding corrupted parties. Additionally, noisy aggregation results hinder its use in scenarios where high precision is a desired property such as aircraft localization.

**Aggregator Oblivious Encryption** This method of encryption formalizes 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 aggregation (typically 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, limited to the sum operation and restricts its applicability to simple processing tasks such as smart-meter sums or mean calculations [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 of (typically one) 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, as with Aggregator Oblivious Encryption, 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 Functional Encryption** These schemes allow the computing of a specific function given encrypted inputs [25], [26]. They are suitable for data-sensitive distributed processing as encryptions must be combined before a result is obtained, and can be considered a generalization 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.

In the context of privacy-preserving localization and state estimation, some of the aforementioned schemes have found applications and some specialized encryption methods have been proposed. These are summarized in the next section.

Localization methods exist for a variety of purposes requiring different levels of accuracy and supporting various forms of sensors. We aim to provide an accurate localization filter in the presence of range-only sensors while providing uncertainties associated with estimates.

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, making them unsuitable for our desired solution. 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 uni-directional 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. In [24], [35], the novel security notions of private weighted sum aggregation with centralized or hidden weights (pWSAc and pWSAh, respectively) are proposed. The schemes capture a weighted aggregation operation suitable for providing privacy in some control environments and are similar in principle to the aggregation decomposition proposed in this work. The requirements of our aggregation operation differ, however, in that weights are generalized to arbitrary length linear combinations, which must be consistent across all participants, and can vary over time. We also require a formal security proof capturing these weight requirements as well as the explicit capabilities of participants. We define our alternative security notion in Section .

Although similar to the pWSAc communication protocol in [35], we formalise the assumptions

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. The additional safety to passengers provided by accurate aircraft localization, however, may be a goal of all those involved, justifying their cooperation.

Along with encryption schemes and their application to localization-relevant algorithms, encryption schemes with ho-

homomorphic properties also require suitable encoding when used with real numbers. As is the case with range sensors, real-valued 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 [37], a simpler alternative is proposed, which leaks no information about encrypted numbers, but allows only a single unencrypted multiplication between encodings. We use a modified version of [37] due to its simplicity and applicability, and introduce the scheme details in Section .

While some aspects of privacy-preserving localization are solved in the above methods and encryption schemes, they fail to provide a localization solution in our context. In the next section, we will formalize our problem and its accompanying security requirements.

#### A. Notation

Throughout this work, we will use the following notation. Lowercase underlined characters are vectors,  $\underline{a}$ , and random variables are bold,  $\mathbf{a}$ . 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 norm. Subscripts of the form  $\mathbf{x}_{k|j}$  will denote an estimate of  $\mathbf{x}_k$  using measurements from times up to and including  $j$ .  $\mathbb{E}[\cdot]$  and  $\text{Var}[\cdot]$  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 omitted when inferable from context.  $\lfloor \cdot \rfloor$  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. PROBLEM STATEMENT

In this work, we consider the context of privacy-preserving range-sensor navigation, in which we wish for no sensor to learn information about the navigator or other sensors beyond their local measurements, and the navigator to learn no information about individual sensors beyond its location estimate. The problem is two-fold, in that we require explicit cryptographic requirements with a suitable encryption scheme meeting them, as well as an estimation scheme that can use the scheme in the context of range-sensor navigation.

To give a formal cryptographic requirement in a distributed setting, we must first consider the communication requirements of our context, and define the attacker capabilities and the desired security of a suitable encryption scheme. In this section, we will define a communication protocol and the relevant formal definition of security we aim to achieve, followed by the estimation problem to which we will apply it.

#### A. Formal Cryptographic Problem

The communication between the navigator and sensors in our estimation problem will be decomposed into a simple two-step bi-directional protocol that will simplify defining formal

security. In section V, we will show how this protocol is sufficient to compute the location estimate at a navigator while meeting our desired privacy goals.

At every *instance*  $t$  (used to distinguish from an estimation timestep  $k$ ), the navigator first broadcasts a number of weights  $\omega_j^{(t)}, 1 \leq j \leq m$  to all sensors  $1 \leq i \leq n$ , who compute individual linear-combinations  $l_i^{(t)} = \sum_{j=1}^m a_{j,i}^{(t)} \omega_j^{(t)}$  based on their measurement data  $a_{j,i}$ . Linear-combinations are then sent back to the navigator, who computes their sum  $\sum_{i=1}^n l_i^{(t)}$ . This two-step linear-combination aggregation protocol has been visually displayed in figure 1. In addition, we remark that

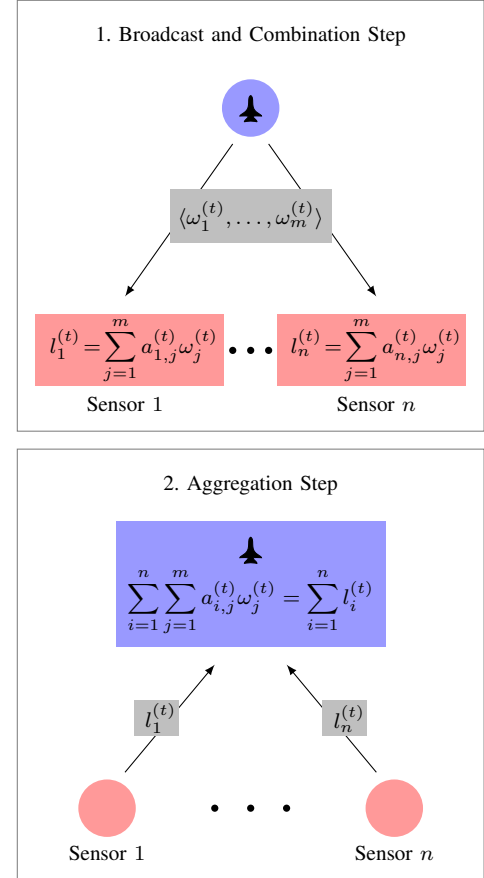


Fig. 1. Required linear-combination aggregation steps at instance  $t$ .

an alternative approach to the two-step protocol is computing  $\sum_{j=1}^m (\omega_j^{(t)} \sum_{i=1}^n a_{i,j}^{(t)})$  at the navigator, requiring only values  $a_{i,j}^{(t)}, 1 \leq j \leq m$  to be sent 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 V.

Before giving a formal definition for the construction and security of our desired encryption scheme, we make the following assumptions on the capabilities of the navigator and sensors.

**Global Navigator Broadcast** We assume that broadcast information from the navigator is received by *all* sensors involved in the protocol.

**Consistent Navigator Broadcast** We assume that broadcast information from the navigator is received equally by all sensors. This means the navigator may not send different weights to individual sensors during a single instance  $t$ .

**Honest-but-Curious Sensors** We adopt the honest-but-curious attacker model for all involved sensors, meaning they are assumed to follow the localization procedure correctly but may store or use any gained sensitive information.

We justify the global broadcast assumption by noting that any subset of sensors within the range of the navigator can be considered a complete group and treated as the global set for estimation purposes, generalising the method, while the wide-spread use of cheap non-directional antennas supports the assumption of consistent broadcasts. The final assumption refers to the known problem of misbehaving sensors [32], [38], often requiring additional complicated detection mechanisms, and will not be considered in this work for the sake of simplicity.

We are now ready to define the type of encryption scheme we want for the specified communication protocol and the security guarantees it should provide. We let a linear-combination aggregation scheme be defined as a tuple of the four algorithms (Setup, Enc, CombEnc, AggDec). These will be used by a trusted setup party, the navigator and sensors  $i$ ,  $1 \leq i \leq n$ . They are defined as follows

**Setup( $\kappa$ )** On input of security parameter  $\kappa$ , generate public parameters  $\text{pub}$ , the number of weights  $m$ , the navigator's public and private keys  $pk_0$  and  $sk_0$  and the sensor private keys  $sk_i$ ,  $1 \leq i \leq n$ .

**Enc( $pk_0, x$ )** The navigator and sensors can encrypt any value  $x$  with the navigator's public key  $pk_0$  and obtain the encryption  $\mathcal{E}_{pk_0}(x)$ .

**CombEnc( $t, pk_0, sk_i, \mathcal{E}_{pk_0}(\omega_1^{(t)}), \dots, \mathcal{E}_{pk_0}(\omega_m^{(t)}), a_{i,1}^{(t)}, \dots, a_{i,m}^{(t)}$ )** At instance  $t$ , sensor  $i$  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  $sk_i$ .

**AggDec( $t, pk_0, sk_0, l_1^{(t)}, \dots, l_n^{(t)}$ )** At instance  $t$ , the navigator 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  $pk_0, sk_0$ .

The security notions we want the above algorithms to meet reflect the previously stated estimation privacy goals. The navigator should learn no information from individual sensors while sensors should learn no information from the navigator or any other sensors. In the context of the introduced communication protocol, this can be summarised as the following notions.

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

**Linear-Combination Aggregator Obliviousness** No colluding subset *excluding* the navigator 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 navigator learns only the sum of all linear combinations weighted by weights of their choice,  $\sum_{i=1}^n l_i^{(t)} = \sum_{i=1}^n \sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}$ .

While indistinguishable weights can be achieved by encrypting weights with an encryption scheme meeting the notion of Indistinguishability under the Chosen Ciphertext Attack (IND-CPA) [39], the novel notion of Linear-Combination Aggregator Obliviousness (LCAO) has been formalised as a typical cryptographic game between attacker and challenger in appendix A. Lastly, we conclude the cryptographic problem definition with the following important remark.

*Remark.* A leakage function including weights from the navigator requires extra care to be taken when giving its definition. If an attacker compromises the navigator (which need not follow the estimation protocol), they have control over the weights, and therefore the leakage function. We note that in the leakage function above,  $\sum_{i=1}^n \sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}$ , an individual sum weighted by the same weight may be learned by an attacker, e.g.,  $\sum_{i=1}^n a_{i,1}^{(t)}$  given weights  $(1, 0, \dots, 0)$ , but that individual sensor values  $a_{i,j}^{(t)}$  remain private due to the assumption of a consistent broadcast.

## B. Estimation problem

The estimation problem we consider, for which we will reduce communication to the protocol required for the introduced cryptographic problem, is localisation with range-only sensors. In this work, we will focus on the two-dimensional case for simplicity but will derive methods suitable for extension to a three-dimensional equivalent. The state that we wish to estimate captures the navigator position and the change in position ( $\Delta$ ), and is given by

$$\underline{x} = [x \quad \Delta x \quad y \quad \Delta y]^\top. \quad (1)$$

The state evolves following a known system model, which at timestep  $k$  is given by

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

with system noise covariance  $\text{Cov}[\underline{w}_k] = \mathbf{Q}_k$ . The measurement model is dependent on sensor  $i$ 's location,

$$\underline{s}_i = [s_{x,i} \quad s_{y,i}]^\top, \quad (3)$$

and is given by

$$z_{k,i} = h_i(\underline{x}_k) + v_{k,i}, \quad (4)$$

with Gaussian measurement noise  $v_{k,i} \sim \mathcal{N}(0, r_i)$  and measurement function  $h_i$ , for sensor  $i$ , as

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

We aim to provide a filter that estimates the navigator's state  $\underline{x}_k$ , at every timestep  $k$ , without learning sensor positions  $\underline{s}_i$ , measurements  $z_{k,i}$  and measurement variances  $r_i$  beyond the information in the corresponding aggregation leakage function.

Similarly, sensors should not learn any information about current state estimates or any other sensor information. Leakage will be further discussed in section V-D, but we note that from any sequential state estimates, following known models, some sensor information leakage can be computed by the navigator. In the context of our leakage function, we will show that this corresponds to the global sums of private sensor information, while individual, or subsets of sensors', information remain private. Similarly, corrupted sensors with access to one or more measurements can produce state estimates of their own, leaking information about navigator state estimates, however, the most accurate estimates, requiring all measurements, will always remain private to the navigator.

### III. PRELIMINARIES

When proposing an encryption scheme meeting the LCAO notion, we will base our method on the additively homomorphic Paillier encryption scheme [20] and the Joye-Libert privacy-preserving aggregation scheme [14]. These schemes have been summarised below. Additionally, the estimation problem we consider uses real-valued inputs and functions, and will require encoding real numbers for use with the aforementioned encryption schemes. The method used for encoding has been summarised afterwards.

#### A. Paillier Encryption Scheme

The Paillier encryption scheme [20] is an additively homomorphic encryption scheme that 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 [39]. The public key is defined by  $(N, g)$  and the 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

$$c = g^a \rho^N \pmod{N^2} \quad (6)$$

for a randomly chosen  $\rho \in \mathbb{Z}_N$ . Here,  $\rho^N$  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

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

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

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

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

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

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

#### B. Joye-Libert Privacy-preserving Aggregation Scheme

The Joye-Libert privacy-preserving aggregation scheme [14] is a scheme defined on time-series data and meets the security notion of Aggregator Obliviousness (AO) [12]. Similarly to the Paillier scheme, it bases its security on the DCRA. A notable difference to a public-key encryption scheme is its need for a trusted party to perform the initial key generation and distribution.

Key generation is computed by first choosing two equal length and sufficiently large primes  $p$  and  $q$ , and computing  $N = pq$ . A hash function  $H : \mathbb{Z} \rightarrow \mathbb{Z}_{N^2}^*$  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  $n$  participants (whose values are to be aggregated), while the last key is set as

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

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

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

Here, we can consider  $H(t)^{sk_i}$  the noise term which hides the easily computable discrete logarithm  $g^{a_i^{(t)}} \pmod{N^2}$ , where  $g = N+1$  (as with the Paillier scheme above).

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

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

and

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

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.

#### C. Integer Encoding for Real Numbers

In both the Paillier and Joye-Libert schemes, as well as the one we introduce, meaningful inputs  $a$  are bounded to  $a \in \mathbb{Z}_N$ . For this reason, real-valued estimation variables require quantisation and integer mapping for encryption and aggregation. We will rely on a generalised Q number encoding [40] due to implementation simplicity and applicability.

We will consider 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 in terms of total and fractional bits [40], [41], [37], but allows for a direct mapping to integer ranges which are not a power of two. A rational subset  $\mathbb{Q}_{M,\phi}$  is then given by

$$\mathbb{Q}_{M,\phi} = \left\{ q \left| \phi q \in \mathbb{N} \wedge -\left\lfloor \frac{M}{2} \right\rfloor \leq \phi q < \left\lfloor \frac{M}{2} \right\rfloor \right. \right\}, \quad (15)$$

and we can quantize 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 an encryption range  $\mathbb{Z}_N$  is achieved by choosing  $M = N$  and handling negatives by modulo arithmetic. Additionally, we note that the Q number format requires a precision factor  $\phi$  to be removed after each encoded multiplication. This is captured by a third parameter  $d$ ; the number of multiplication factors to add or remove from encodings.

The function for *combined* quantization and encoding,  $E_{M,\phi,d}(a)$ , of a given number  $a \in \mathbb{R}$ , for an integer range  $\mathbb{Z}_M$ , precision  $\phi$  and scaling for  $d$  prior encoded multiplications, is given by

$$E_{M,\phi,d}(a) = \lfloor \phi^{d+1} a \rfloor \pmod{M}. \quad (16)$$

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

$$E_{M,\phi,d}^{-1}(u) = \begin{cases} \frac{u \pmod{M}}{\phi^{d+1}}, & u \pmod{M} \leq \left\lfloor \frac{M}{2} \right\rfloor \\ -\frac{M - u \pmod{M}}{\phi^{d+1}}, & \text{otherwise} \end{cases}. \quad (17)$$

This encoding scheme provides the following homomorphic operations,

$$E_{M,\phi,d}(a_1) + E_{M,\phi,d}(a_2) \pmod{M} = E_{M,\phi,d}(a_1 + a_2) \quad (18)$$

and

$$E_{M,\phi,d}(a_1) E_{M,\phi,d}(a_2) \pmod{M} = E_{M,\phi,d+1}(a_1 a_2), \quad (19)$$

noting that when  $M = N$ , the modulus corresponds with those in the Paillier homomorphic operations (9), (10) and the Joye-Libert sum (14).

In general, the choice of a large precision parameter  $\phi$  may reduce quantization errors introduced in (16), but risks overflow after too many multiplications. Given the largest number of encoded multiplications,  $d_{max}$ , and the largest value to be encoded  $a_{max}$ , the parameter should be chosen such that

$$|\phi^{d_{max}+1} a_{max}| < \left\lfloor \frac{M}{2} \right\rfloor. \quad (20)$$

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

#### IV. PRIVATE LINEAR-COMBINATION AGGREGATION SCHEME

In this section, we introduce an encryption scheme meeting the desired security properties in section II-A. The scheme is a combination of the Paillier and Joye-Libert schemes and provides encrypted weights meeting IND-CPA and encrypted aggregation meeting LCAO. Similarly to its constituents, the scheme bases its security on the DCRA and, as with the Joye-Libert scheme, requires a trusted party for initial key generation and distribution. The four algorithms defining our scheme are given as follows.

**Setup( $\kappa$ )** On input parameter  $\kappa$ , generate two equal length, sufficiently large, primes  $p$  and  $q$ , and compute  $N = pq$ . Define a hash function  $H : \mathbb{Z} \rightarrow \mathbb{Z}_{N^2}^*$ , choose a the number of weights to combine  $m > 1$  and set public parameter  $\text{pub} = H$ , navigator public key  $pk_0 = N$  and navigator private key  $sk_0 = (p, q)$ . Sensor 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  $sk_n = -\sum_{i=1}^{n-1} sk_i \pmod{N^2}$ .

**Enc( $pk_0, x$ )** Public-key encryption is computed by the Paillier encryption scheme with implicit generator  $g = N + 1$ . This is given by

$$\mathcal{E}_{pk_0}(x) = (N + 1)^x \rho^N \pmod{N^2}, \quad (21)$$

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

**CombEnc( $t, pk_0, sk_i, \mathcal{E}_{pk_0}(\omega_1^{(t)}), \dots, \mathcal{E}_{pk_0}(\omega_m^{(t)}), a_{i,1}^{(t)}, \dots, a_{i,m}^{(t)}$ )** Encrypted linear combination at instance  $t$  is computed as

$$l_i^{(t)} = H(t)^{sk_i} \prod_{j=1}^m \mathcal{E}_{pk_0}(\omega_j^{(t)})^{a_{i,j}^{(t)}} \pmod{N^2}, \quad (22)$$

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,  $\rho_i^N$  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.

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

$$\begin{aligned} &\sum_{i=1}^n \sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)} = \\ &\frac{L((l^{(t)})^\lambda \pmod{N^2})}{L((N + 1)^\lambda \pmod{N^2})} \pmod{N}, \end{aligned} \quad (23)$$

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

$$\begin{aligned} l^{(t)} &= \prod_{i=1}^n H(t)^{sk_i} (N+1)^{\sum_{j=1}^m a_{i,j} \omega_j} \rho_i^N \pmod{N^2} \\ &= H(t)^{\sum_{i=1}^n sk_i} \prod_{i=1}^n (N+1)^{\sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}} \rho_i^N \pmod{N^2} \\ &= (N+1)^{\sum_{i=1}^n \sum_{j=1}^m a_{i,j}^{(t)} \omega_j^{(t)}} \rho^N \pmod{N^2}, \end{aligned}$$

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 the resulting linear combinations and aggregation are computed  $\pmod{N}$ .

The security proof of this scheme must show that both encrypted weights meet IND-CPA and encrypted aggregation meets LCAO. As weights are encrypted with the Paillier encryption scheme, the first requirement is already met. To show that aggregation meets LCAO, a reduction proof is given in appendix B.

Lastly, we make the following remark about a possible implicit weight.

*Remark.* The construction of the above scheme supports linearly combining the implicit weight  $\omega_j^{(t)} = 1$ , by replacing

$$\mathcal{E}_{pk_0}(1)^{a_{i,j}^{(t)}} = (N+1)^1 \rho_j^N \pmod{N^2} \quad (24)$$

in (22), by

$$(N+1)^{a_{i,j}^{(t)}} \pmod{N^2}. \quad (25)$$

This is due to the removal of  $\rho_j^N$  terms during decryption and can be used to reduce the navigator's broadcast communication cost by a single weight when it is known that  $\omega_j^{(t)} = 1$ .

## V. PRIVACY-PRESERVING LOCALIZATION

To produce an estimate of the state  $\underline{x}_k$  such that communication between  $n$  range sensors and the navigator is reducible to the linear-combination aggregation protocol, we make use of an algebraic reformulation of the Extended Kalman Filter (EKF), the Extended Information Filter (EIF) [42], which reduces the filter update step to a single summation. The EIF update step requires the predicted state estimate  $\underline{x}_{k|k-1}$  and estimate covariance  $\mathbf{P}_{k|k-1}$  in the information vector and matrix forms

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

respectively. In this form, the update equations for  $n$  sensor measurements at time  $k$ , given process and measurement models (2) and (4), are given by

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

and

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

with Jacobians

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

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

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

before using the normal EKF prediction equations

$$\underline{x}_{k+1|k} = \underline{f}(\underline{x}_{k|k}) \quad \text{and} \quad \mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^\top + \mathbf{Q}_k, \quad (31)$$

with Jacobian

$$\mathbf{F}_k = \left. \frac{\partial \underline{f}}{\partial \underline{x}} \right|_{\underline{x}_{k|k}}. \quad (32)$$

In the form above, at every timestep  $k$ , all sensitive sensor information required for state estimation is captured in the measurement vector

$$\underline{i}_{k,i} = \mathbf{H}_{k,i}^\top r_i^{-1} (z_{k,i} - h_i(\underline{x}_{k|k-1}) + \mathbf{H}_{k,i} \underline{x}_{k|k-1}) \quad (33)$$

and the measurement matrix

$$\mathbf{I}_{k,i} = \mathbf{H}_{k,i}^\top r_i^{-1} \mathbf{H}_{k,i}, \quad (34)$$

namely, their measurements  $z_{k,i}$ , measurement variances  $r_i$  and locations  $\underline{s}_i$ ; captured in measurement functions  $h_i$  and Jacobians  $\mathbf{H}_{k,i}$ .  $\underline{i}_{k,i}$  and  $\mathbf{I}_{k,i}$ , however, also require the current predicted state estimate  $\underline{x}_{k|k-1}$  to be computed (in  $h_i$  and  $\mathbf{H}_{k,i}$ ). For this reason, our goal is to reformulate (33) and (34) to be computable at each sensor  $i$  as a linear combination of functions of the navigator state estimate (weights), and subsequently aggregated at the navigator. Application of the linear-combination aggregation scheme proposed would in turn guarantee that sensors do not learn the navigator state, and that the navigator learns only the aggregation required for updating its state estimate in (27) and (28).

### A. Range Measurement Modification

The first thing we note when wanting to decompose  $\underline{i}_{k,i}$  and  $\mathbf{I}_{k,i}$  to a linear combination of functions of  $\underline{x}_{k|k-1}$ , is that  $h_i$  cannot be rearranged in this way due to the present square-root. Similarly, the Jacobian of  $h_i$  at  $\underline{x}_{k|k-1}$ ,

$$\mathbf{H}_{k,i} = \left[ \frac{\underline{x}_{k|k-1} - \underline{s}_{x,i}}{\sqrt{(\underline{x}_{k|k-1} - \underline{s}_{x,i})^2 + (\underline{y}_{k|k-1} - \underline{s}_{y,i})^2}} \right], \quad (35)$$

cannot be either. We, therefore, consider the modified measurement functions

$$h'_i(\underline{x}) = h_i(\underline{x})^2. \quad (36)$$

A measurement function in this form allows rearrangement of  $h'_i$  and the corresponding Jacobian  $\mathbf{H}'_{k,i}$  to a linear combination of powers of location elements in  $\underline{x}_{k|k-1}$ , as

$$\begin{aligned} h'_i(\underline{x}) &= \left\| \begin{bmatrix} x & y \end{bmatrix}^\top - \underline{s}_i \right\|^2 \\ &= (x - s_{x,i})^2 + (y - s_{y,i})^2 \\ &= x^2 + y^2 - 2s_{x,i}x - 2s_{y,i}y + s_{x,i}^2 + s_{y,i}^2 \end{aligned} \quad (37)$$

and

$$\mathbf{H}'_{k,i} = \begin{bmatrix} 2x_{k|k-1} - 2s_{x,i} \\ 2y_{k|k-1} - 2s_{y,i} \end{bmatrix}. \quad (38)$$

Here,  $h'_i$  and  $\mathbf{H}'_{k,i}$  are linear combinations of  $x_{k|k-1}^2$ ,  $y_{k|k-1}^2$ ,  $x_{k|k-1}$  and  $y_{k|k-1}$ . To show how  $\underline{z}'_{k,i}$  and  $\mathbf{I}'_{k,i}$  can be similarly rearranged, we also require the existence of measurements following a modified measurement model of the form

$$z'_{k,i} = h'_i(\underline{x}_k) + v'_{k,i}, \quad (39)$$

where  $z'_{k,i}$  is the modified measurement, and noise term  $v'_{k,i}$  is zero-mean and has a known variance  $r_{k,i}$ .

The approximation of  $z'_{k,i}$  and its variance  $\text{Var}[v'_{k,i}]$  from the original measurements  $z_{k,i}$  are complicated by the noise term  $v_{k,i} \sim \mathcal{N}(0, r_i)$  in (4). Squaring the original range measurements produces

$$\begin{aligned} z_{k,i}^2 &= (h_i(\underline{x}_k) + v_{k,i})^2 \\ &= h_i^2(\underline{x}_k) + 2h_i(\underline{x}_k)v_{k,i} + v_{k,i}^2, \end{aligned} \quad (40)$$

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

$$\mathbb{E}[2h_i(\underline{x}_k)v_{k,i} + v_{k,i}^2] = r_i \quad (41)$$

and mean-adjust modified measurements as

$$\begin{aligned} z'_{k,i} &= z_{k,i}^2 - r_i \\ &= h_i^2(\underline{x}_k) + 2h_i(\underline{x}_k)v_{k,i} + v_{k,i}^2 - r_i \\ &= h'_i(\underline{x}_k) + v'_{k,i}, \end{aligned} \quad (42)$$

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

$$\text{Var}[v'_{k,i}] = 4h_i^2(\underline{x}_k)r_i + 2r_i^2 \quad (43)$$

and is also dependent on  $h_i(\underline{x}_k)$ . To use the modified measurement (42) with the EIF, we require an estimate for  $\text{Var}[v'_{k,i}]$  at the sensor as well. Additionally, a conservative estimate (*i.e.*, a larger variance resulting in less confidence in measurements) is desirable to reduce filter divergence. The naive approach, replacing  $h_i(\underline{x}_k)$  with  $z_{k,i}$ , only provides a conservative estimate when

$$\begin{aligned} z_{k,i} &> h_i(\underline{x}_k) \\ \implies v_{k,i} &> 0, \end{aligned} \quad (44)$$

which cannot be guaranteed. Instead, we provide a conservative estimate with 95% confidence due to the Gaussianity of  $v_{k,i}$ , by adjusting the replacement term  $z_{k,i}$  by two of its standard deviations  $\sqrt{r_i}$ . The modified measurement's variance at time  $k$  is therefore conservatively approximated by

$$\begin{aligned} r_{k,i} &= 4(z_{k,i} + 2\sqrt{r_i})^2 r_i + 2r_i^2 \\ &\approx \text{Var}\{v'_{k,i}\} \end{aligned} \quad (45)$$

for each sensor  $i$ .

The existence of modified measurements  $z'_{k,i}$  and their estimated variances  $r_{k,i}$  are sufficient for sensors to compute measurement vector  $\underline{z}'_{k,i}$  and matrix  $\mathbf{I}'_{k,i}$  in the required form for linear-combination aggregation.

## B. Localization

Now that range measurements can be modified by (42), and the modified model (39) can be used, corresponding modified measurement vectors  $\underline{z}'_{k,i}$  and matrices  $\mathbf{I}'_{k,i}$  can be computed as linear combinations of powers of location elements. This is given by

$$\begin{aligned} \underline{z}'_{k,i} &= \mathbf{H}_{k,i}'^\top r_{k,i}^{-1} (z'_{k,i} - h'_i(\underline{x}_{k|k-1})) + \mathbf{H}_{k,i}' \underline{x}_{k|k-1} \\ &= \begin{bmatrix} (2r_{k,i}^{-1})x_{k|k-1}^3 + (2r_{k,i}^{-1})x_{k|k-1}y_{k|k-1}^2 \\ + (-r_{k,i}^{-1}s_{x,i})x_{k|k-1}^2 + (-2r_{k,i}^{-1}s_{x,i})y_{k|k-1}^2 \\ + (2r_{k,i}^{-1}z'_{k,i})x_{k|k-1} + (-2r_{k,i}^{-1}s_{x,i}^2)x_{k|k-1} \\ + (-2r_{k,i}^{-1}s_{y,i}^2)x_{k|k-1} + (2r_{k,i}^{-1}s_{x,i}^3) \\ + (2r_{k,i}^{-1}s_{x,i}s_{y,i}^2) + (-2r_{k,i}^{-1}s_{x,i}z'_{k,i}) \\ (2r_{k,i}^{-1})y_{k|k-1}^3 + (2r_{k,i}^{-1})x_{k|k-1}y_{k|k-1}^2 \\ + (-2r_{k,i}^{-1}s_{y,i})x_{k|k-1}^2 + (-2r_{k,i}^{-1}s_{y,i})y_{k|k-1}^2 \\ + (2r_{k,i}^{-1}z'_{k,i})y_{k|k-1} + (-2r_{k,i}^{-1}s_{x,i}^2)y_{k|k-1} \\ + (-2r_{k,i}^{-1}s_{y,i}^2)y_{k|k-1} + (2r_{k,i}^{-1}s_{y,i}s_{x,i}^2) \\ + (2r_{k,i}^{-1}s_{y,i}^3) + (-2r_{k,i}^{-1}s_{y,i}z'_{k,i}) \end{bmatrix} \end{aligned} \quad (46)$$

and

$$\begin{aligned} \mathbf{I}'_{k,i} &= \mathbf{H}_{k,i}'^\top r_{k,i}^{-1} \mathbf{H}_{k,i}' \\ &= \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}, \end{aligned} \quad (47)$$

with

$$\begin{aligned} \beta_{11} &= (4r_{k,i}^{-1})x_{k|k-1}^2 + (-8r_{k,i}^{-1}s_{x,i})x_{k|k-1} + (4r_{k,i}^{-1}s_{x,i}^2), \\ \beta_{12} &= (4r_{k,i}^{-1})x_{k|k-1}y_{k|k-1} + (-4r_{k,i}^{-1}s_{y,i})x_{k|k-1} \\ &\quad + (-4r_{k,i}^{-1}s_{x,i})y_{k|k-1} + (4r_{k,i}^{-1}s_{x,i}s_{y,i}), \\ \beta_{21} &= \beta_{12}, \text{ and} \\ \beta_{22} &= (4r_{k,i}^{-1})y_{k|k-1}^2 + (-8r_{k,i}^{-1}s_{y,i})y_{k|k-1} + (4r_{k,i}^{-1}s_{y,i}^2). \end{aligned}$$

The above rearrangements give  $\underline{z}'_{k,i}$  and  $\mathbf{I}'_{k,i}$  as linear combinations of

$$\begin{aligned} \{x_{k|k-1}^3, y_{k|k-1}^3, x_{k|k-1}^2y_{k|k-1}, x_{k|k-1}y_{k|k-1}^2, \\ x_{k|k-1}^2y_{k|k-1}, y_{k|k-1}^2x_{k|k-1}, x_{k|k-1}y_{k|k-1}, y_{k|k-1}\}, \end{aligned} \quad (48)$$

which now capture all the private state information in  $\underline{x}_{k|k-1}$  required at the sensors. The corresponding EIF update steps (27) and (28) then become

$$\underline{y}_{k|k} = \underline{y}_{k|k-1} + \sum_{i=1}^n \underline{z}'_{k,i} \quad (49)$$

and

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{i=1}^n \mathbf{I}'_{k,i}, \quad (50)$$

respectively. Additionally, we note the possible extension to the three-dimensional case.

*Remark.* The above has been derived for two-dimensional localisation but can be similarly derived for three-dimensional case. However, we note that the number of weights increases combinatorially with the number of dimensions, thus affecting the cost of communication as well.



The final step required for computing (46) and (47) at the sensors and (49) and (50) at the navigator using the proposed LCAO scheme, is handling the unique instance variables  $t$  in the context of estimation timesteps  $k$ . This is done by elementwise linear combination of (46) and (47) and setting  $t$  to the concatenation

$$t = k \parallel v \parallel w \parallel \tau \quad (51)$$

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

### C. Pseudocode

Measurement modification, real number encoding and linear combination aggregation are all required to compute the modified EIF from the previous section in a privacy-preserving manner. In this section, we summarise this entire localisation process and give the pseudocode for its execution. For brevity, we will assume  $\phi$  and  $M = N$  from section III-C to be public information, and simplify the encoding notation  $E_{N,\phi,d}(\cdot)$  to  $E_d(\cdot)$ . The privacy-preserving localization filter consists of the following steps.

**Setup** Run only once, the Setup algorithm from section IV is run by a trusted third party,  $N$  and  $H$  are made public, and the navigator and sensor secret keys,  $sk_0 = \lambda = \text{lcm}(p-1, q-1)$  and  $sk_i$ ,  $1 \leq i \leq n$ , are distributed accordingly.

**Prediction** At each timestep  $k$ , the navigator computes the typical EKF prediction equations (V) before encrypting weights (48) with algorithm Enc and broadcasting them to the sensors. This is given by algorithm 1.

**Measurement** At each timestep  $k$ , sensors modify their measurements with (42) and (45) before computing  $\underline{i}'_{k,i}$  and  $\mathbf{I}'_{k,i}$  using algorithm CombEnc elementwise and sending them back to the navigator. This is given by algorithm 2.

**Update** At each timestep  $k$ , the navigator aggregates and decrypts recieved measurement vectors and matrices with algorithm AggDec, before computing the modified EIF update equations (49) and (50). This is given by algorithm 3.

---

#### Algorithm 1 Navigator Prediction

---

```

1: procedure PREDICTION( $\underline{x}_{k-1|k-1}$ ,  $\mathbf{P}_{k-1|k-1}$ ,  $\underline{f}$ ,  $\mathbf{Q}$ ,  $N$ )
2:   Compute  $\mathbf{F}_k$  by (32)
3:    $\underline{x}_{k|k-1} \leftarrow \underline{f}(\underline{x}_{k-1|k-1})$ 
4:    $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k$ 
5:   Compute  $E_0(x_{k|k-1}^3)$  by (16)
6:   Compute  $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}^3))$  by (21)
7:   Broadcast  $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}^3))$  to sensors
8:   for Remaining weights in (48) do
9:     Broadcast weight in the form above
10:  end for
11:  return  $\underline{x}_{k|k-1}$ ,  $\mathbf{P}_{k|k-1}$ 
12: end procedure

```

---

Algorithms 1, 2 and 3 have also been summarised graphically in figure 2. Here, for brevity,  $\mathcal{E}_{pk_0, sk_i}(\cdot)$  and  $E_d(\cdot)$  denote elementwise operations with the same parameters.

---

#### Algorithm 2 Measurement at Sensor $i$

---

```

1: procedure MEASUREMENT( $i$ ,  $s_{x,i}$ ,  $s_{y,i}$ ,  $r_i$ ,  $N$ ,  $H$ )
2:   Measure  $z_{k,i}$ 
3:   Compute  $z'_{k,i}$  by (42)
4:   Compute  $r_{k,i}$  by (45)
5:   Recieve  $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}^3))$ 
6:   for Remaining weights in (48) do
7:     Recieve weight in the form above
8:   end for
9:   Let  $\alpha_v^{(i)}$  represent an encryption of element  $v$  in  $\underline{i}'_{k,i}$  from (46)
10:   $\alpha_1^{(i)} \leftarrow \mathcal{E}_{pk_0}(E_0(x_{k|k-1}^3))^{E_0(2r_{k,i}^{-1})}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}y_{k|k-1}^2))^{E_0(2r_{k,i}^{-1})}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}^2))^{E_0(-r_{k,i}^{-1}s_{x,i})}$ .
     $\mathcal{E}_{pk_0}(E_0(y_{k|k-1}^2))^{E_0(-2r_{k,i}^{-1}s_{x,i})}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}))^{E_0(2r_{k,i}^{-1}z'_{k,i})}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}))^{E_0(-2r_{k,i}^{-1}s_{x,i}^2)}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}))^{E_0(-2r_{k,i}^{-1}s_{y,i}^2)}$ .
     $(N+1)^{E_1(2r_{k,i}^{-1}s_{x,i}^3)}(N+1)^{E_1(2r_{k,i}^{-1}s_{x,i}s_{y,i}^2)}$ .
     $(N+1)^{E_1(-2r_{k,i}^{-1}s_{x,i}z'_{k,i})}H(k \parallel 1 \parallel 1 \parallel 0) \pmod{N^2}$ 
11:  Compute  $\alpha_2^{(i)}$  using (46), (22) and remark IV in the form above
12:  for  $v \leftarrow 1$  to 2 do
13:    Send  $\alpha_v^{(i)}$  to the navigator
14:  end for
15:  Let  $\beta_{vw}^{(i)}$  represent an encryption of element  $(v, w)$  in  $\mathbf{I}'_{k,i}$  from (47)
16:   $\beta_{11}^{(i)} \leftarrow \mathcal{E}_{pk_0}(E_0(x_{k|k-1}^2))^{E_0(4r_{k,i}^{-1})}$ .
     $\mathcal{E}_{pk_0}(E_0(x_{k|k-1}))^{E_0(-8r_{k,i}^{-1}s_{x,i})}$ .
     $(N+1)^{E_1(4r_{k,i}^{-1}s_{x,i}^2)}$ .
     $H(k \parallel 1 \parallel 1 \parallel 1) \pmod{N^2}$ 
17:  Compute remaining  $\beta_{vw}^{(i)}$  using (47), (22) and remark IV in the form above
18:  for  $v \leftarrow 1$  to 2 do
19:    for  $w \leftarrow 1$  to 2 do
20:      Send  $\beta_{vw}^{(i)}$  to the navigator
21:    end for
22:  end for
23: end procedure

```

---

### D. Leakage

With the algorithm defined, we can analyze the localization leakage when the strictly defined LCAO leakage is taken into account. We recall the assumptions on participant capabilities made in Section . Global and 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 knowledge about other sensors' measurements when navigator behavior is dependent on its estimate. This may be achievable by detecting changes in measurements after reporting false ones. As misbehaving sensors are also a known problem which complicates homomorphic operations, active

**Algorithm 3** Navigator Update

---

```

1: procedure UPDATE( $\underline{x}_{k|k-1}, \mathbf{P}_{k|k-1}, N, \lambda$ )
2:   for  $v \leftarrow 1$  to 2 do
3:     Receive  $\alpha_v^{(i)}$  from each sensor  $1 \leq i \leq n$ 
4:   end for
5:   for  $v \leftarrow 1$  to 2 do
6:     for  $w \leftarrow 1$  to 2 do
7:       Receive  $\beta_{vw}^{(i)}$  from each sensor  $1 \leq i \leq n$ 
8:     end for
9:   end for
10:  Let  $\alpha_v$  represent an encryption of element  $v$  in
     $\sum_{i=1}^n \underline{i}'_{k,i}$ 
11:  for  $v \leftarrow 1$  to 2 do
12:     $\alpha_v \leftarrow \prod_{i=1}^n \alpha_v^{(i)}$ 
13:    Compute  $\mathcal{D}_{sk_0}(\alpha_v)$  with  $\lambda$  by (23)
14:    Compute  $E_1^{-1}(\mathcal{D}_{sk_0}(\alpha_v))$  by (17)
15:  end for
16:  Construct  $\sum_{i=1}^n \underline{i}'_{k,i}$  from decoded decryptions above
17:  Let  $\beta_{vw}$  represent an encryption of element  $(v, w)$  in
     $\sum_{i=1}^n \mathbf{I}'_{k,i}$ 
18:  for  $v \leftarrow 1$  to 2 do
19:    for  $w \leftarrow 1$  to 2 do
20:       $\beta_{vw} \leftarrow \prod_{i=1}^n \beta_{vw}^{(i)}$ 
21:      Compute  $\mathcal{D}_{sk_0}(\beta_{vw})$  with  $\lambda$  by (23)
22:      Compute  $E_1^{-1}(\mathcal{D}_{sk_0}(\beta_{vw}))$  by (17)
23:    end for
24:  end for
25:  Construct  $\sum_{i=1}^n \mathbf{I}'_{k,i}$  from decoded decryptions above
26:   $\underline{y}_{k|k} \leftarrow \mathbf{P}_{k|k-1}^{-1} \underline{x}_{k|k-1} + \sum_{i=1}^n \underline{i}'_{k,i}$ 
27:   $\mathbf{Y}_{k|k} \leftarrow \mathbf{P}_{k|k-1} + \sum_{i=1}^n \mathbf{I}'_{k,i}$ 
28:   $\underline{x}_{k|k} \leftarrow \mathbf{Y}_{k|k}^{-1} \underline{y}_{k|k}$ 
29:   $\mathbf{P}_{k|k} \leftarrow \mathbf{Y}_{k|k}^{-1}$ 
30:  return  $\underline{x}_{k|k}, \mathbf{P}_{k|k}$ 
31: end procedure

```

---

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  $\sum_{i=1}^n \underline{i}'_{k,i}$  and  $\sum_{i=1}^n \mathbf{I}'_{k,i}$ , however, as stated in Remark II-A, weights chosen by a corrupted navigator mean that individual sums of coefficients can be leaked as well. That is, sums of coefficients in (46) and (47) (*i.e.*,  $\sum_{i=1}^n 2\mathbf{r}_{k,i}^{-1}$ ,  $\sum_{i=1}^n -\mathbf{r}_{k,i}^{-1}s_{x,i}$ ,  $\sum_{i=1}^n -2\mathbf{r}_{k,i}^{-1}s_{x,i}$ ,  $\sum_{i=1}^n 2\mathbf{r}_{k,i}^{-1}z'_{k,i}$ , ...) are leaked to a corrupted navigator.

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

$$\sum_{i=1}^n z'_{k,i}, \sum_{i=1}^n \mathbf{r}_{k,i}, \sum_{i=1}^n s_{x,i} \text{ and } \sum_{i=1}^n s_{y,i}. \quad (52)$$

From these sums and the definitions (42) and (45), 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.

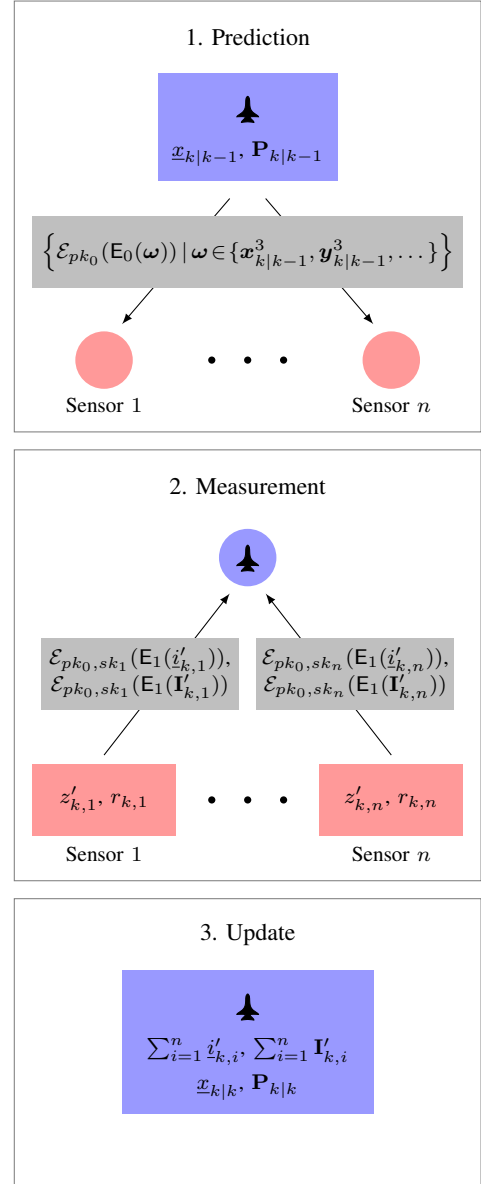


Fig. 2. Procedure at timestep  $k$  for the proposed privacy-preserving EIF.

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

## VI. SIMULATION AND RESULTS

To demonstrate our proposed approach, we have implemented the algorithm in Section 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 (42), 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 3. To evaluate the accuracy of our method, we have run 100

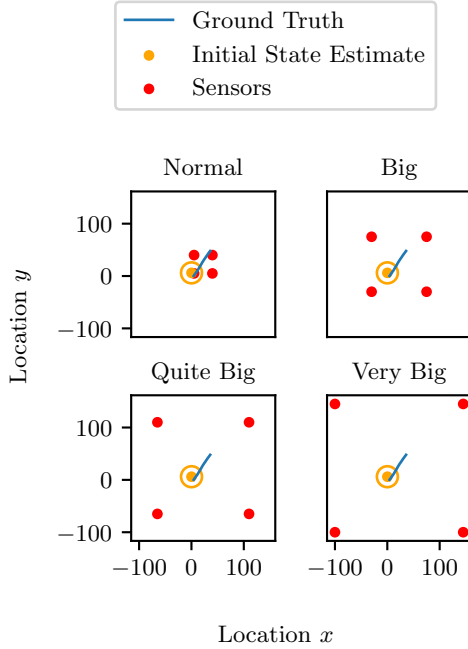


Fig. 3. The considered layouts, with varying average distances between navigator and sensors.

simulations, all with 50 filter iterations, for each of the sensor layouts with sensor measurement variances set to  $r_i = 5$  simulation units. Due to the large encoding modulus  $N$ , no notable advantage is gained when choosing a low fractional precision factor  $\phi$ , 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 4. 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 V-A 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 (45) and the sufficiently large fractional precision parameter  $\phi$ .

In addition to accuracy, computational requirements are an important factor in the adoption of novel cryptographic

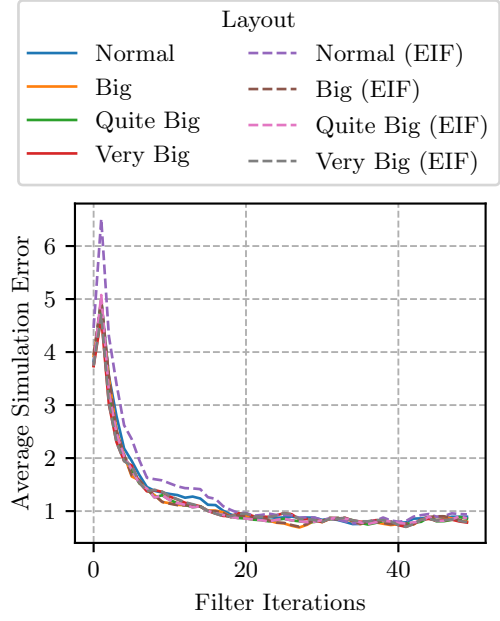


Fig. 4. Filter error for different layouts, where solid lines show the results from our privacy-preserving filter and dashed lines from an unmodified EIF.

methods. Figure 5 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

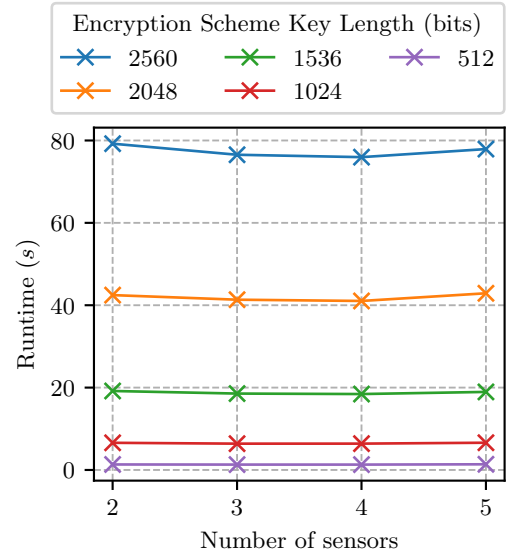


Fig. 5. Runtimes for varying key sizes and numbers of sensors.

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 dependent 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 2048 bit-long keys [48].

## VII. CONCLUSION

To develop a Bayesian localization 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 Paillier and Joye-Libert encryption and aggregation schemes, respectively. 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 localization 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

#### LINEAR-COMBINATION AGGREGATOR OBLIVIOUSNESS (LCAO)

The following game between attacker and challenger defines the security notion of LCAO.

**Setup** The challenger chooses security parameter  $\kappa$ , runs the  $\text{Setup}(\kappa)$  algorithm and gives  $\text{pub}$ ,  $m$  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  $(i, t, a_{i,1}^{(t)}, \dots, a_{i,m}^{(t)})$  such that for any two chosen combine query tuples  $(i, t, a_{i,1}^{(t)}, \dots, a_{i,m}^{(t)})$  and  $(i', t', a_{i',1}^{(t')}, \dots, a_{i',m}^{(t')})$ , the following condition holds:

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

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

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

and

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

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

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

**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

$$\text{Adv}^{\text{LCAO}}(\mathcal{A}) := \left| \mathbb{P}[b' = b] - \frac{1}{2} \right|.$$

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

### APPENDIX B LCAO SCHEME PROOF

The scheme in section IV will be shown to meet LCAO by contrapositive. We show that for an adversary  $\mathcal{A}$  playing against a challenger using the scheme, we can always create an adversary  $\mathcal{A}'$  playing against a challenger  $\mathcal{C}$  using the Joye-Libert scheme, such that

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

for some negligible functions  $\eta_1, \eta_2$  and security parameter  $\kappa$ . That is, if we assume our scheme is not LCAO secure, then the Joye-Libert scheme is not AO secure (which is not the case [14]).

*Proof.* Consider adversary  $\mathcal{A}$  playing the LCAO game. The following is a construction of an adversary  $\mathcal{A}'$  playing the AO game [12] against a challenger  $\mathcal{C}$  using the Joye-Libert aggregation scheme.

**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  $(i, t, a_{i,1}^{(t)}, \dots, a_{i,m}^{(t)})$ , 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  $\rho^N$  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 received secret key  $sk_i$  to  $\mathcal{A}$ .

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

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

and

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

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

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

and

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

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

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

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

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

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 exactly, and to  $\mathcal{A}$ ,  $\mathcal{A}'$  follows the scheme in section IV exactly. Since  $\mathcal{A}'$  runs in polynomial-time to security parameter when  $\mathcal{A}$  does, and no non-negligible advantage adversary to  $\mathcal{C}$  exists, we conclude that no non-negligible advantage adversary  $\mathcal{A}$  exists. That is, there exists a negligible function  $\eta$ , such that

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

for security parameter  $\kappa$ . Lastly, 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.  $\square$

## 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.
- [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 Multi-sensor Systems*. CRC press, 1998.
- [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.
- [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] A. B. Alexandru and G. J. Pappas, "Private Weighted Sum Aggregation," *arXiv*, 2020.
- [36] Google, "Encrypted-bigquery-client," <https://github.com/google/encrypted-bigquery-client>, 2015.
- [37] F. Farokhi, I. Shames, and N. Batterham, "Secure and Private Control Using Semi-Homomorphic Encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, 2017.
- [38] I. Ben-Gal, "Outlier Detection," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2005, pp. 131–146.
- [39] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall, 2008.
- [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. 2000.



**Marko Ristic** received his diploma in software engineering in 2018 at the University of Melbourne, Australia. Since 2019, he has been pursuing a Ph.D. at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests include encrypted and privacy-preserving signal processing, focusing on state estimation, sensor fusion, and distributed filtering.



**Benjamin Noack** received his diploma in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2009. Afterwards, he obtained his Ph.D. in 2013 at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. Since 2013 he is a senior researcher at the Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the areas of multi-sensor data fusion, distributed and decentralized Kalman filtering, combined stochastic and set-membership approaches to state estimation,

and event-based systems.



**Uwe D. Hanebeck** is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS). He obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in Electrical Engineering from the Technical University in Munich, Germany. His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. He is author and coauthor of more than 500 publications in various high-ranking journals and conferences and an IEEE Fellow.