Distributed Range-Only Localisation that Preserves Sensor and Navigator Privacies

Marko Ristic [a], Benjamin Noack [a], Uwe D. Hanebeck [b]

[a] *Autonomous Multisensor Systems Group (AMS), Institute for Intelligent Cooperating Systems (ICS), Otto von Guericke University Magdeburg (OVGU), Germany.*

[b] *Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology(KIT), Germany.*

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 typically involve the leakage of sensor or navigator information by communicating measurements or estimates and thus do not preserve participants' privacy. The existing approaches that do provide such guarantees, fail to address sensor and navigator privacy in the common application of model-based range-only estimation, consequently forfeiting broad applicability. In this work, we define a notion of privacy-preserving linear combination aggregation and use it to derive a modified Extended Kalman Filter using range measurements such that navigator location, sensors' locations, and sensors' measurements are kept private during navigation. Additionally, a formal cryptographic backing is presented to guarantee our method's privacy as well as an implementation to evaluate its performance. The novel, provably secure, range-based localisation method has applications in a variety of environments where sensors may not be trusted or estimates are considered sensitive, such as autonomous vehicle localisation or air traffic navigation.

Key words: State estimation; Data privacy; Sensor fusion; Extended Kalman filters.

1 Introduction

Localisation methods in distributed sensor environments have long been an active topic of research [1–3] and have characterised many advancements of Kalman and Bayesian estimation theory [4]. In particular, range-based localisation methods, including signal strength measurements [5,6], acoustic ranges [7] and ultra-wideband ranges [8], have found large application due to the prevalence of suitable sensors. In most cases, these localisation methods require the gathering of measurements centrally, where an estimate of location can be computed. With recent developments in distributed and cloud computing, uses of wireless and public communication channels for data transfer have become widespread, and the additional requirements of data privacy and state secrecy have become particularly relevant [9,10].

Email addresses: marko.ristic@ovgu.de (Marko Ristic), benjamin.noack@ovgu.de (Benjamin Noack), uwe.hanebeck@kit.edu (Uwe D. Hanebeck).

Typical cryptographic secrecy involves hiding all transferred data such that external parties in the communication network learn no new information from acquired encryptions. This can be achieved with common symmetric and public-key encryption scheme such as AES [11] and RSA [12], respectively. In some cases however, partial data leakage or encrypted data processing is required for achieving a desired goal which has led to several homomorphic and functional encryption schemes [13–16] finding uses in signal processing or localisation tasks. In [17], homomorphic encryption is used to make time-independent model-free location estimates where an estimator does not learn sensor measurements or locations. In [18], similar secrecy is achieved with a linear Kalman filter when a hierarchical sensor network is present. In [14,15], privacy-preserving aggregation schemes are presented as a means to compute total powergrid usage without disclosing individual contributions, while in [19,20], centralised and hidden weighted sum aggregations, pWSAc and pWSAh, are introduced as a means for computing local control inputs in a distributed environment without learning individual inputs.

Our contribution in this work presents a range-only localisation method meeting formal cryptographic requirements that ensure sensors keep their measurements and locations private while the navigator keeps its estimates private. We first define a novel notion for private linear combination aggregation and present an implementation that satisfies these requirements, before using it to derive a filter based on the Extended Kalman Filter with no hierarchical sensor layout assumptions. The linear combination aggregation scheme we put forward is in principle similar to the pWSAh scheme in [20], however, a formal definition with different communication assumptions and leakages is given, crucial for its cryptographic security. To the best of the authors' knowledge, no existing method for Bayesian state estimation using range-only sensors and meeting the desired privacy requirements exists.

We motivate this scenario with the example of vehicle localisation in the presence of privately owned measurement stations. While the intention of measurement stations is the accurate navigation of passing vehicles, it may be reasonable to desire identifying location or hardware details to remain unknown to the other present stations and the navigator. Similarly, a navigator may not wish to disclose their most accurate location estimates to untrusted third-party measurers.

In section 2, we introduce both the cryptographic and estimation problems considered in this work, before giving some relevant preliminaries in section 3. Section 4 proposes a cryptographic scheme meeting our desired security properties and section 5 gives a solution to the estimation problem making use of this scheme. Simulation results are discussed in section 6 and concluding remarks are made in section 7.

1.1 Notation

Throughout this work, we make use of the following notation. Underlined characters, $\underline{a}$, denote vectors, uppercase bold characters, $\mathbf{A}$, denote matrices. $\mathbf{A}^{-1}$ is the matrix inverse while $\mathbf{A}^\top$ is its transpose. The expected value of a random variable is denoted $\mathsf{E}[\cdot]$, while the variance of a random scalar and covariance of a random vector by $\mathsf{Var}[\cdot]$ and $\mathsf{Cov}[\cdot]$, respectively. $|a|$ denotes absolute value, $\|\underline{a}\|$ the vector norm, $\{\dots\}$ is used for sets and $\langle\dots\rangle$ for ordered sequences. When estimating a state, notation $\underline{\hat{x}}_{k|l}$ will denote an estimate of $\underline{x}$ at timestep k given measurements from timesteps up to and including l. When discussing cryptography, $\mathcal{E}_k(\cdot)$ and $\mathcal{D}_k(\cdot)$ will denote encryption and decryption with key k, respectively, with k ommited when inferrable from context. $\mathbb{Z}_n$ is used for the set of integers modulo n and $\mathbb{Z}_n^*$ for its multiplicative group. $\mathsf{lcm}(\cdot,\cdot)$ is the lowest common denominator, $\|$ the binary concatination operator and the term *negligible function* refers to its cryptographic definition in [21].

2 Problem Statement

In this work, we consider the context of privacy-preserving range sensor navigation, where we want 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-only 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.

2.1 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 5, we will show how this protocol is sufficient to compute the location estimate at a navigator while meeting our desired privacy goals. The communication protocol is as follows.

At every *instance* t (used to distinguish from an estimation *timestep*), the navigator first broadcasts m weights $\omega_j^{(t)}, j \in \{1,\dots,m\}$ to all sensors $i \in \{1,\dots,n\}$, who individually compute linear combinations $l_i^{(t)} = \sum_{j=1}^m a_{j,i}^{(t)}\omega_i^{(t)}$ 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 note that 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 $a_{i,j}^{(t)}, j \in \{1,\dots,m\}$ 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 5.

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 participants.

Global Navigator Broadcast We assume that

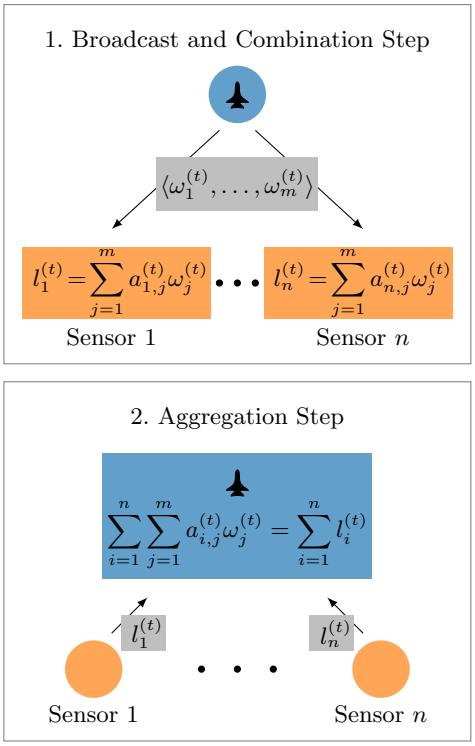

Fig. 1. Required linear combination aggregation steps at instance t.

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 that they follow the localisation 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 [22,23], often requiring additional complicated detection mechanisms, and will not be considered in this work.

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 $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{CombEnc}, \mathsf{AggDec})$. These will be used by a trusted setup party, the navigator and sensors $i \in \{1, \ldots, n\}$. They are defined as follows.

$\mathsf{Setup}(\kappa)$ On input of security parameter κ, generate public parameters 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, $i \in \{1, \ldots, n\}$.

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

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

$\mathsf{AggDec}(t, pk_0, sk_0, l_1^{(t)}, \ldots, l_n^{(t)})$ 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 these 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 new knowledge about the navigator weights $\omega_j^{(t)}$, $j \in \{1, \ldots, m\}$ when receiving only their encryptions from the current and previous instances and having 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, $a_{i,j}^{(t)}$, $j \in \{1, \ldots, m\}$, 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 Plaintext Attack (IND-CPA) [21], 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 1 *A leakage function including weights from the navigator requires extra care to be taken when giving its definition. If an attacker compromises the naviga-*

tor, 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 $(1,0,\ldots,0)$, but that individual sensor values $a_{i,j}^{(t)}$ remain private due to the assumption of a consistent broadcast.

2.2 Estimation problem

The estimation problem we consider, for which we will reformulate communication to the protocol above, 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 must capture the navigator position, x and y, and may contain any other components relevant to the system. It is of the form

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

This state evolves following some known system model, which at timestep k can be written as

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

with noise term $\underline{w}_k$. Measurements of $\underline{x}_k$ follow a measurement model dependent on sensor $i \in \{1,\ldots,n\}$, given by

$$z_{k,i} = h_i(\underline{x}_k) + v_{k,i}\,, \tag{3}$$

with Gaussian measurement noises $v_{k,i} \sim \mathcal{N}(0, r_{k,i})$ and measurement function

$$\begin{aligned} h_i(\underline{x}) &= \left\| \begin{bmatrix} x & y \end{bmatrix}^\top - \underline{s}_i \right\| \\ &= \sqrt{(x - s_{x,i})^2 + (y - s_{y,i})^2}\,, \end{aligned} \tag{4}$$

where

$$\underline{s}_i = \begin{bmatrix} s_{x,i} & s_{y,i} \end{bmatrix}^\top \tag{5}$$

is the location of sensor i.

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 $r_{k,i}$ 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 5.4, 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.

3 Preliminaries

When proposing an encryption scheme meeting the LCAO notion, we will base our method on the additively homomorphic Paillier encryption scheme [13] and the Joye-Libert privacy-preserving aggregation scheme [15]. 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.

3.1 Paillier Encryption Scheme

The Paillier encryption scheme [13] 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 [21]. 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

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

for a randomly chosen $\rho \in \mathbb{Z}_N$. Here, 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. $\forall a_1, a_2 \in \mathbb{Z}_N$,

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

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

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

3.2 Joye-Libert Privacy-Preserving Aggregation Scheme

The Joye-Libert privacy-preserving aggregation scheme [15] is a scheme defined on time-series data and meets the security notion of Aggregator Obliviousness (AO) [14]. 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 $H : \mathbb{Z} \to \mathbb{Z}^*_{N^2}$ is defined and the public key is set to (N, H). n private keys are generated by choosing sk_i, $i \in \{1, \ldots, n\}$, 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

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

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

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

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 $c_i^{(t)}$, $i \in \{1, \ldots, n\}$ are sent to the aggregator, summation and decryption of the aggregated sum are computed by the functions

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

and

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

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

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

removing all noise terms.

3.3 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 [24] 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 [24–26], 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

$$\mathbb{Q}_{M,\phi} = \left\{ q \;\middle|\; \phi q \in \mathbb{N} \wedge -\left\lfloor \frac{M}{2} \right\rfloor \leq \phi q < \left\lfloor \frac{M}{2} \right\rfloor \right\}, \tag{15}$$

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 ϕ to be removed after each encoded multiplication. This is captured by a third parameter d; the number of precision factors present in encodings.

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

$$\mathsf{E}_{M,\phi,d}(a) = \left\lfloor \phi^{d+1} a \right\rceil \pmod{M}. \tag{16}$$

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

$$\mathsf{E}^{-1}_{M,\phi,d}(u) = \begin{cases} \dfrac{u \pmod{M}}{\phi^{d+1}}, & u \pmod{M} \leq \left\lfloor \dfrac{M}{2} \right\rfloor \\ -\dfrac{M - u \pmod{M}}{\phi^{d+1}}, & \text{otherwise} \end{cases}. \tag{17}$$

This encoding scheme provides the following homomorphic operations,

$$\mathsf{E}_{M,\phi,d}(a_1) + \mathsf{E}_{M,\phi,d}(a_2) \pmod{M} = \mathsf{E}_{M,\phi,d}(a_1 + a_2) \tag{18}$$

and

$$\mathsf{E}_{M,\phi,d}(a_1)\mathsf{E}_{M,\phi,d}(a_2) \pmod{M} = \mathsf{E}_{M,\phi,d+1}(a_1 a_2), \tag{19}$$

noting that when $M = N$, the operations and modulus correspond with those in the Paillier homomorphic operations (8), (9) and (10), and the Joye-Libert sum (14).

In general, the choice of a large precision parameter ϕ may reduce quantisation 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

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

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

4 Private Linear Combination Aggregation Scheme

In this section, we introduce an encryption scheme meeting the desired security properties in section 2.1. The scheme is a combination of the Paillier and Joye-Libert schemes and provides encrypted weights meeting IND-CPA and encrypted aggregation meeting the notion of LCAO defined in section 2.1. 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.

As aggregation is typically performed on scalar inputs, we extend our notation to the context of multidimensional estimation data by letting an instance $t_{k,\tau}$ uniquely capture the scalar aggregation during an estimation timestep k for a single element with position index τ. To achieve this in practice, any injective function can be used, such as the concatenation $t_{k,\tau} = k \,\|\, \tau$. The four algorithms defining our scheme are given as follows.

$\mathsf{Setup}(\kappa)$ 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 the number of weights to combine, $m > 1$, and set public parameter $\mathsf{pub} = H$, navigator public key $pk_0 = N$ and navigator private key $sk_0 = (p, q)$. Sensor secret keys are generated by choosing sk_i, $i \in \{1, \dots, n-1\}$ uniformly from $\mathbb{Z}_{N^2}$ and setting the last key as $sk_n = -\sum_{i=1}^{n-1} sk_i \pmod{N^2}$.

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

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

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

$\mathsf{CombEnc}(t_{k,\tau}, pk_0, sk_i, \mathcal{E}(\omega_1^{(k,\tau)}) \dots, a_{i,1}^{(k,\tau)} \dots)$ At instance $t_{k,\tau}$, encrypted linear combination is computed as

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

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

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

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

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

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

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

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

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

Additionally, we note that in the above construction, all weights $\omega_j^{(k,\tau)}$ and values $a_{i,j}^{(k,\tau)}$ are integers and the

resulting linear combinations and aggregation are computed modulo N.

The security proof of this scheme must both show that encrypted weights meet IND-CPA and that 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.

Remark 2 *Given the construction of the scheme above, it can be seen that any weights $\omega_j^{(k,\tau)}$, whose values are known at each sensor, do not need to be broadcast by the navigator. In this case, sensors can replace*

$$\mathcal{E}_{pk_0}(\omega_j^{(k,\tau)})^{a_{i,j}^{(k,\tau)}} = (N+1)^{\omega_j^{(k,\tau)} a_{i,j}^{(k,\tau)}} \rho_j^N \pmod{N^2} \tag{24}$$

in (22)*, by*

$$(N+1)^{\omega_j^{(k,\tau)} a_{i,j}^{(k,\tau)}} \pmod{N^2}. \tag{25}$$

This is due to the removal of ρ_j^N terms during decryption and can be used to reduce the navigator's broadcast communication cost by the number of weights $\omega_j^{(k,\tau)}$ that do not hold any information private to the navigator and are known by the sensors in advance.

5 Privacy-Preserving Localisation

With a concrete scheme meeting the LCAO notion, we can now put forward a localisation filter with communication that can be reformulated to the required protocol. To produce an estimate of the state $\underline{x}_k$, we make use of an algebraic reformulation of the Extended Kalman Filter (EKF), the Extended Information Filter (EIF) [27], which reduces the filter update step to a single summation. The EIF update step requires the predicted state estimate $\hat{\underline{x}}_{k|k-1}$ and estimate covariance $\mathbf{P}_{k|k-1}$ in the information vector and matrix forms

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

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

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

and

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

with Jacobians

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

for sensors $i \in \{1, \ldots, n\}$. After converting the updated information vector and matrix back to state estimate $\hat{\underline{x}}_{k|k}$ and estimate covariance $\mathbf{P}_{k|k}$, the filter's prediction step can be computed by the navigator locally using any suitable filter for the known system model (2).

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

$$\underline{i}_{k,i} = \mathbf{H}_{k,i}^\top r_i^{-1} (z_{k,i} - h_i(\hat{\underline{x}}_{k|k-1}) + \mathbf{H}_{k,i} \hat{\underline{x}}_{k|k-1}) \tag{30}$$

and the measurement matrix

$$\mathbf{I}_{k,i} = \mathbf{H}_{k,i}^\top r_i^{-1} \mathbf{H}_{k,i}, \tag{31}$$

namely, their measurements $z_{k,i}$, measurement variances $r_{k,i}$ and locations $\underline{s}_i$; captured in measurement functions h_i and Jacobians $\mathbf{H}_{k,i}$. However, $\underline{i}_{k,i}$ and $\mathbf{I}_{k,i}$ also require the current predicted state estimate $\hat{\underline{x}}_{k|k-1}$ to be computed (in h_i and $\mathbf{H}_{k,i}$). For this reason, our goal is to reformulate (30) and (31) to be computable at each sensor i as a linear combination of functions of the navigator state estimate (the navigator weights), to be 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 the navigator learns only the aggregation required for updating its state estimate in (27) and (28).

5.1 Range Measurement Modification

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

$$\mathbf{H}_{k,i} = \begin{bmatrix} \frac{\hat{x}_{k|k-1} - s_{x,i}}{\sqrt{(\hat{x}_{k|k-1} - s_{x,i})^2 + (\hat{y}_{k|k-1} - s_{y,i})^2}} \\ \frac{\hat{y}_{k|k-1} - s_{y,i}}{\sqrt{(\hat{x}_{k|k-1} - s_{x,i})^2 + (\hat{y}_{k|k-1} - s_{y,i})^2}} \\ 0 \\ \vdots \end{bmatrix}, \tag{32}$$

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

$$h_i'(\underline{x}) = h_i(\underline{x})^2. \tag{33}$$

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 $\hat{\underline{x}}_{k|k-1}$, as

$$\begin{aligned} h'_i(\underline{x}) &= \left\| \begin{bmatrix} x & y \end{bmatrix}^\top - \underline{s}_i \right\|^2 \\ &= (x - s_{x,i})^2 + (y - s_{y,i})^2 \\ &= x^2 + y^2 - 2s_{x,i}x - 2s_{y,i}y + s_{x,i}^2 + s_{y,i}^2 \end{aligned} , \quad (34)$$

and

$$\mathbf{H}'_{k,i} = \begin{bmatrix} 2\hat{x}_{k|k-1} - 2s_{x,i} \\ 2\hat{y}_{k|k-1} - 2s_{y,i} \\ 0 \\ \vdots \end{bmatrix} . \quad (35)$$

Here, h'_i and $\mathbf{H}'_{k,i}$ are linear combinations of $\hat{x}^2_{k|k-1}$, $\hat{y}^2_{k|k-1}$, $\hat{x}_{k|k-1}$ and $\hat{y}_{k|k-1}$. To show how the corresponding modified measurement vectors $\underline{i}'_{k,i}$ and matrices $\mathbf{I}'_{k,i}$ can be similarly rearranged and used for localisation, we also require the existence of measurements following a modified measurement model of the form

$$z'_{k,i} = h'_i(\underline{x}_k) + v'_{k,i} , \quad (36)$$

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

Computing $z'_{k,i}$ and its variance $r'_{k,i}$ from the original measurements $z_{k,i}$ are complicated by the noise term $v_{k,i} \sim \mathcal{N}(0, r_{k,i})$, and simply squaring the original range measurements produces

$$\begin{aligned} z^2_{k,i} &= (h_i(\underline{x}_k) + v_{k,i})^2 \\ &= h'_i(\underline{x}_k) + 2h_i(\underline{x}_k)v_{k,i} + v^2_{k,i} , \end{aligned} \quad (37)$$

with a new noise term $2h_i(\underline{x}_k)v_{k,i} + v^2_{k,i}$, now dependent on the measurement function h_i, and no longer zero-mean. We can compute the mean of this new noise term (a function of the Gaussian term $v_{k,i}$) as $\mathsf{E}[2h_i(\underline{x}_k)v_{k,i} + v^2_{k,i}] = r_{k,i}$ and mean-adjust modified measurements as

$$\begin{aligned} z'_{k,i} &= z^2_{k,i} - r_{k,i} \\ &= h_i(\underline{x}_k)^2 + 2h_i(\underline{x}_k)v_{k,i} + v^2_{k,i} - r_{k,i} \\ &= h'_i(\underline{x}_k) + v'_{k,i} , \end{aligned} \quad (38)$$

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

$$\mathsf{Var}[v'_{k,i}] = 4h_i(\underline{x}_k)^2 r_{k,i} + 2r^2_{k,i} \quad (39)$$

and is also dependent on h_i. To use the modified measurement (38) with the EIF, we require an estimate for $\mathsf{Var}[v'_{k,i}]$ 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. While the naive approach, replacing $h_i(\underline{x}_k)$ with $z_{k,i}$ in (39), may not provide a conservative estimate when $z^2_{k,i} < h_i(\underline{x}_k)^2$, the Gaussianity of $v_{k,i}$ can be exploited to provide a conservative estimate with 95% confidence by shifting the replacement term $z_{k,i}$ by two of its standard deviations $\sqrt{r_{k,i}}$. The modified measurement's variance at timestep k can therefore be conservatively approximated by

$$\begin{aligned} r'_{k,i} &= 4(z_{k,i} + 2\sqrt{r_{k,i}})^2 r_{k,i} + 2r^2_{k,i} \\ &\gtrapprox \mathsf{Var}[v'_{k,i}] , \end{aligned} \quad (40)$$

at each sensor i.

The modified measurement model (36) can now be used for localisation, when measurements are modified by (38) and their new variance estimated with (40).

5.2 Localisation

To complete the EIF update as a linear combination aggregation, modified vectors $\underline{i}'_{k,i}$ and matrices $\mathbf{I}'_{k,i}$, using the modified measurement model (36), can be rearragned as follows.

$$\begin{aligned} \underline{i}'_{k,i} &= \mathbf{H}'^\top_{k,i} r'^{-1}_{k,i} (z'_{k,i} - h'_i(\hat{\underline{x}}_{k|k-1}) + \mathbf{H}'_{k,i}\hat{\underline{x}}_{k|k-1}) \\ &= \begin{bmatrix} \alpha_i^{(k,1)} & \alpha_i^{(k,2)} & 0 & \cdots \end{bmatrix}^\top , \end{aligned} \quad (41)$$

with

$$\begin{aligned} \alpha_i^{(k,1)} &= (2r'^{-1}_{k,i})\hat{x}^3_{k|k-1} + (2r'^{-1}_{k,i})\hat{x}_{k|k-1}\hat{y}^2_{k|k-1} \\ &\quad + (-r'^{-1}_{k,i}s_{x,i})\hat{x}^2_{k|k-1} + (-2r'^{-1}_{k,i}s_{x,i})\hat{y}^2_{k|k-1} \\ &\quad + (2r'^{-1}_{k,i}z'_{k,i})\hat{x}_{k|k-1} + (-2r'^{-1}_{k,i}s^2_{x,i})\hat{x}_{k|k-1} \\ &\quad + (-2r'^{-1}_{k,i}s^2_{y,i})\hat{x}_{k|k-1} + (2r'^{-1}_{k,i}s^3_{x,i}) \\ &\quad + (2r'^{-1}_{k,i}s_{x,i}s^2_{y,i}) + (-2r'^{-1}_{k,i}s_{x,i}z'_{k,i}) \text{ and} \\ \alpha_i^{(k,2)} &= (2r'^{-1}_{k,i})\hat{y}^3_{k|k-1} + (2r'^{-1}_{k,i})\hat{x}^2_{k|k-1}\hat{y}_{k|k-1} \\ &\quad + (-2r'^{-1}_{k,i}s_{y,i})\hat{x}^2_{k|k-1} + (-2r'^{-1}_{k,i}s_{y,i})\hat{y}^2_{k|k-1} \\ &\quad + (2r'^{-1}_{k,i}z'_{k,i})\hat{y}_{k|k-1} + (-2r'^{-1}_{k,i}s^2_{x,i})\hat{y}_{k|k-1} \\ &\quad + (-2r'^{-1}_{k,i}s^2_{y,i})\hat{y}_{k|k-1} + (2r'^{-1}_{k,i}s_{y,i}s^2_{x,i}) \\ &\quad + (2r'^{-1}_{k,i}s^3_{y,i}) + (-2r'^{-1}_{k,i}s_{y,i}z'_{k,i}) , \end{aligned}$$

and

$$\mathbf{I}'_{k,i} = \mathbf{H}'^{\top}_{k,i} r'^{-1}_{k,i} \mathbf{H}'_{k,i} = \begin{bmatrix} \alpha_i^{(k,3)} & \alpha_i^{(k,4)} & 0 & \cdots \\ \alpha_i^{(k,5)} & \alpha_i^{(k,6)} & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{42}$$

with

$$\begin{aligned} \alpha_i^{(k,3)} &= (4r'^{-1}_{k,i})\hat{x}^2_{k|k-1} + (-8r'^{-1}_{k,i} s_{x,i})\hat{x}_{k|k-1} \\ &\quad + (4r'^{-1}_{k,i} s^2_{x,i}), \\ \alpha_i^{(k,4)} &= (4r'^{-1}_{k,i})\hat{x}_{k|k-1}\hat{y}_{k|k-1} + (-4r'^{-1}_{k,i} s_{y,i})\hat{x}_{k|k-1} \\ &\quad + (-4r'^{-1}_{k,i} s_{x,i})\hat{y}_{k|k-1} + (4r'^{-1}_{k,i} s_{x,i} s_{y,i}), \\ \alpha_i^{(k,5)} &= \alpha_i^{(k,4)} \text{ and} \\ \alpha_i^{(k,6)} &= (4r'^{-1}_{k,i})\hat{y}^2_{k|k-1} + (-8r'^{-1}_{k,i} s_{y,i})\hat{y}_{k|k-1} \\ &\quad + (4r'^{-1}_{k,i} s^2_{y,i}). \end{aligned}$$

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

$$\begin{aligned} \{&\hat{x}^3_{k|k-1},\ \hat{y}^3_{k|k-1},\ \hat{x}^2_{k|k-1}\hat{y}_{k|k-1},\ \hat{x}_{k|k-1}\hat{y}^2_{k|k-1}, \\ &\hat{x}^2_{k|k-1},\ \hat{y}^2_{k|k-1},\ \hat{x}_{k|k-1}\hat{y}_{k|k-1},\ \hat{x}_{k|k-1},\ \hat{y}_{k|k-1}\}, \end{aligned} \tag{43}$$

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

$$\underline{\hat{y}}_{k|k} = \underline{\hat{y}}_{k|k-1} + \sum_{i=1}^{n} \underline{i}'_{k,i} \tag{44}$$

and

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{i=1}^{n} \mathbf{I}'_{k,i}, \tag{45}$$

respectively.

Remark 3 *The above has been derived for two-dimensional localisation but can be similarly derived for the three-dimensional case. However, the number of weights increases combinatorially with the number of dimensions, thus affecting the cost of communication as well.*

5.3 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 ϕ and $M = N$ from section 3.3 to be public information and thus simplify the encoding notation $\mathsf{E}_{N,\phi,d}(\cdot)$ to $\mathsf{E}_d(\cdot)$. The privacy-preserving localisation filter consists of the following steps.

Setup The Setup algorithm from section 4 is run only once by a trusted party, N and H are made public, and the navigator and sensor secret keys, $sk_0 = \lambda = \mathsf{lcm}(p-1, q-1)$ and $sk_i, i \in \{1, \ldots, n\}$, are distributed accordingly.

Prediction At each timestep k, the navigator computes the prediction of the current state and its covariance with a local filter before encrypting weights (43) 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 (38) and (40) before computing encryptions of $\underline{i}'_{k,i}$ and $\mathbf{I}'_{k,i}$ using algorithm $\mathsf{CombEnc}$ for each element 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 EIF update equations (44) and (45). This is given by algorithm 3.

Algorithm 1 Navigator Prediction

procedure PREDICTION($\underline{\hat{x}}_{k-1|k-1}$, $\mathbf{P}_{k-1|k-1}$, N)
  Compute $\underline{\hat{x}}_{k|k-1}$ with a local filter
  Compute $\mathbf{P}_{k|k-1}$ with a local filter
  Compute $\mathsf{E}_0(\hat{x}^3_{k|k-1})$ by (16)
  Compute $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}^3_{k|k-1}))$ by (21)
  Broadcast $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}^3_{k|k-1}))$ to sensors
  for Remaining weights in (43) **do**
    Broadcast weight in the form above
  end for
  return $\underline{\hat{x}}_{k|k-1}, \mathbf{P}_{k|k-1}$
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 $\mathsf{E}_d(\cdot)$ denote elementwise operations with the same parameters.

5.4 Leakage

With the privacy-preserving EIF defined in the previous section, we can now interpret the aggregation leakage of an LCAO scheme in the context of range sensor localisation. The leakage function from the AggDec algorithm corresponds to the information vector and matrix sums, $\sum_{i=1}^{n} \underline{i}'_{k,i}$ and $\sum_{i=1}^{n} \mathbf{I}'_{k,i}$, respectively. However, recalling that a compromised navigator can learn the individual sums weighted by the same weight,

Algorithm 2 Measurement at Sensor i

procedure MEASUREMENT(i, $s_{x,i}$, $s_{y,i}$, $r_{k,i}$, N, H)
  Measure $z_{k,i}$
  Compute $z'_{k,i}$ by (38)
  Compute $r'_{k,i}$ by (40)
  Recieve $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}^3_{k|k-1}))$
  for Remaining weights in (43) **do**
    Recieve weight in the form above
  end for
  Let $\boldsymbol{\alpha}_i^{(k,\tau)}$ represent the encryption of $\alpha_i^{(k,\tau)}$ in (41) and (42)
  $\boldsymbol{\alpha}_i^{(k,1)} \leftarrow \mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}^3_{k|k-1}))^{\mathsf{E}_0(2r'^{-1}_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}_{k|k-1}\hat{y}^2_{k|k-1}))^{\mathsf{E}_0(2r'^{-1}_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}^2_{k|k-1}))^{\mathsf{E}_0(-r'^{-1}_{k,i}s_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{y}^2_{k|k-1}))^{\mathsf{E}_0(-2r'^{-1}_{k,i}s_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}_{k|k-1}))^{\mathsf{E}_0(2r'^{-1}_{k,i}z'_{k,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}_{k|k-1}))^{\mathsf{E}_0(-2r'^{-1}_{k,i}s^2_{x,i})}\cdot$
  $\mathcal{E}_{pk_0}(\mathsf{E}_0(\hat{x}_{k|k-1}))^{\mathsf{E}_0(-2r'^{-1}_{k,i}s^2_{y,i})}\cdot$
  $(N+1)^{\mathsf{E}_1(2r'^{-1}_{k,i}s^3_{x,i})}(N+1)^{\mathsf{E}_1(2r'^{-1}_{k,i}s_{x,i}s^2_{y,i})}\cdot$
  $(N+1)^{\mathsf{E}_1(-2r'^{-1}_{k,i}s_{x,i}z'_{k,i})}H(k\,\|\,1) \pmod{N^2}$
  Compute remaining $\boldsymbol{\alpha}_i^{(k,\tau)}$ using (41), (42), (22) and the remark from section 4 in the form above
  for $\tau \leftarrow 1$ to 6 **do**
    Send $\boldsymbol{\alpha}_i^{(k,\tau)}$ to the navigator
  end for
end procedure

Algorithm 3 Navigator Update

procedure UPDATE($\hat{\underline{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, N, λ)
  for $\tau \leftarrow 1$ to 6 **do**
    Receive $\boldsymbol{\alpha}_i^{(k,\tau)}$ from each sensor $i \in \{1,\dots,n\}$
  end for
  Let $\boldsymbol{\alpha}^{(k,\tau)}$ represent an encryption of $\sum_{i=1}^n \alpha_i^{(k,\tau)}$
  for $\tau \leftarrow 1$ to 6 **do**
    $\boldsymbol{\alpha}^{(k,\tau)} \leftarrow \prod_{i=1}^n \boldsymbol{\alpha}_i^{(k,\tau)}$
    Compute $\mathcal{D}_{sk_0}(\boldsymbol{\alpha}^{(k,\tau)})$ with λ by (23)
    Compute $\mathsf{E}_1^{-1}(\mathcal{D}_{sk_0}(\boldsymbol{\alpha}^{(k,\tau)}))$ by (17)
  end for
  Construct $\sum_{i=1}^n \underline{i}'_{k,i}$ and $\sum_{i=1}^n \mathbf{I}'_{k,i}$ from decoded decryptions above
  $\hat{\underline{y}}_{k|k} \leftarrow \mathbf{P}^{-1}_{k|k-1}\hat{\underline{x}}_{k|k-1} + \sum_{i=1}^n \underline{i}'_{k,i}$
  $\mathbf{Y}_{k|k} \leftarrow \mathbf{P}^{-1}_{k|k-1} + \sum_{i=1}^n \mathbf{I}'_{k,i}$
  $\hat{\underline{x}}_{k|k} \leftarrow \mathbf{Y}^{-1}_{k|k}\hat{\underline{y}}_{k|k}$
  $\mathbf{P}_{k|k} \leftarrow \mathbf{Y}^{-1}_{k|k}$
  return $\hat{\underline{x}}_{k|k}, \mathbf{P}_{k|k}$
end procedure

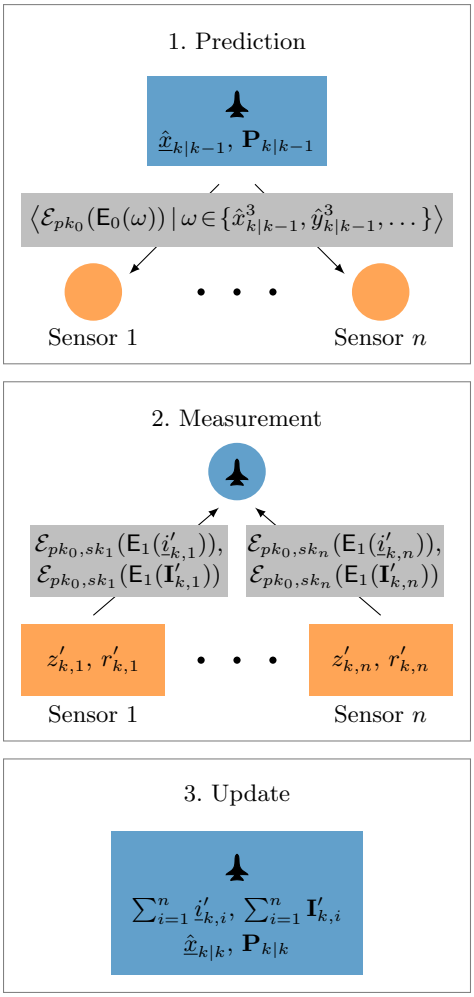

Fig. 2. Procedure at timestep k for the proposed privacy-preserving EIF.

$\{\sum_{i=1}^n 2r_{k,i}^{-1}, \sum_{i=1}^n -r_{k,i}^{-1}s_{x,i}, \sum_{i=1}^n -2r_{k,i}^{-1}s_{x,i}, \dots\}$ can be leaked as well. From this leakage, we can see that private sensor information, $z'_{k,i}$, $r'_{k,i}$ and $\underline{s}_i$, is present only in their complete sums

$$\sum_{i=1}^n z'_{k,i}\,,\ \sum_{i=1}^n r'_{k,i}\,,\ \sum_{i=1}^n s_{x,i} \text{ and } \sum_{i=1}^n s_{y,i}\,, \tag{46}$$

which in practice correspond to their averages. Therefore, in the context of our proposed localisation method, LCAO leakage corresponds to the averages of sensor private information, while individual sensor information remains private.

6 Simulation and Results

As well as having shown the theoretical backing for the security of our scheme, we have simulated the proposed localisation method to evaluate its performance. A two-

dimensional, linear, constant velocity process model,

$$\underline{x}_k = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \underline{x}_{k-1} + \underline{w}_k \,,$$

where noise term $\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q})$ and

$$\mathbf{Q} = \frac{1}{10^3} \cdot \begin{bmatrix} 0.4 & 0 & 1.3 & 0 \\ 0 & 0.4 & 0 & 1.3 \\ 1.3 & 0 & 5.0 & 0 \\ 0 & 1.3 & 0 & 5.0 \end{bmatrix} ,$$

was simulated and tracked with the algorithms in section 5.3, using a linear Kalman filter for the navigator's local state prediction. Code was written in the C programming language using the MPI library [28] to support asynchronous computations by the sensors and navigator. The MG1 mask generation function and the SHA256 hash function, from the OpenSSL library [29], were used to implement the required hash function H, and the Libpaillier library [30] was used for the Paillier encryption scheme. Additionally, GNU libraries, GSL [31] and GMP [32], were used for algebraic operations and multiple-precision encoded integers, respectively. All execution was performed on a 3.33GHz Xeon W3680 CPU, running on the Windows Subsystem for Linux (WSL).

We have considered multiple sensor layouts, each with four sensors, to capture the dependence of estimated modified measurement variances $r'_{k,i}$ on the original measurements $z_{k,i}$. These layouts of varying sensor distances are shown next to the simulation initial state and a sample track in figure 3. To demonstrate the accuracy of the method, we have compared the root mean square error (RMSE) of the privacy-preserving filter to the standard EIF using unmodified measurements, which is algebraically equivalent to the EKF typically used in industry for linearising non-linear state estimation. Estimation in each layout from figure 3 consisted of 50 filter iterations and was run 1000 times. Unmodified measurement variances were taken as $r_{k,i} = 5$ for all $k > 0$ and a large fractional precision factor, $\phi = 2^{32}$, was chosen. The results can be seen in figure 4. From these results, we can see a strong similarity in filter performance between the privacy-preserving method and that of the traditional EIF. We can also see that the varying average distances between sensors and the navigator have little impact on the differences in performance. We attribute this similarity in RMSE to the conservativeness of estimated modified measurement variances $r'_{k,i}$, resulting in few additional filter divergences, and to the high fractional precision factor, keeping computations consistent with the floating-point arithmetic of the EIF.

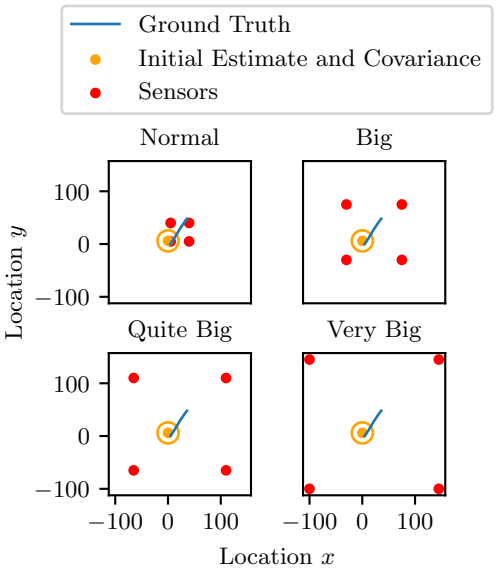

Fig. 3. Different simulation layouts with varying distances between navigator and sensors.

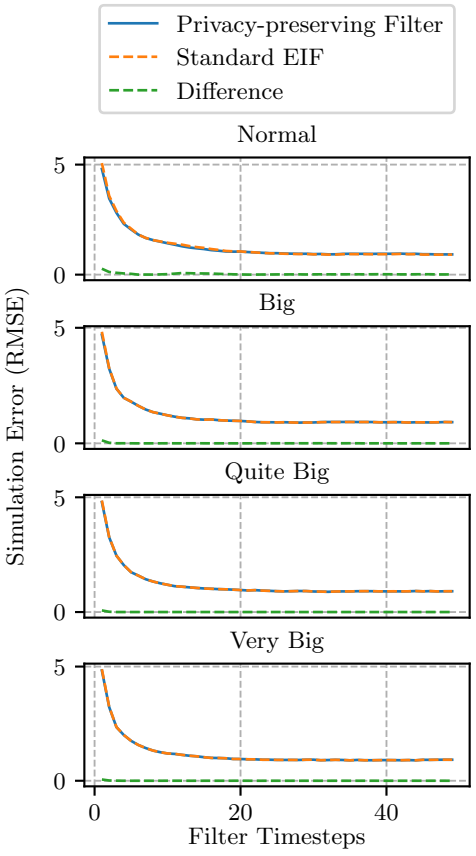

Fig. 4. Average RMSE of our privacy-preserving filter and the standard EIF for different layouts.

In addition to filter error, computational performance is important to consider when relying on cryptographic methods. Figure 5 shows the averages of 10 execution times when varying the numbers of sensors and key sizes (bit lengths of N). Here, increasing the number of sensors primarily affected the number of inter-process communications and aggregation modular multiplications due to the asynchronous implementation. We can see from the

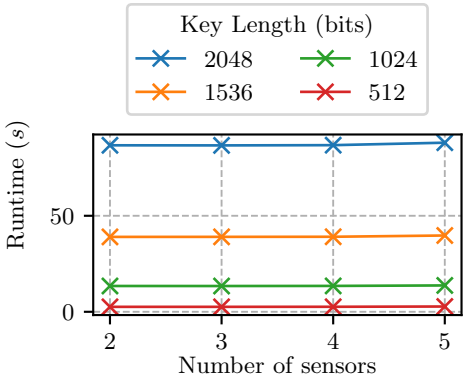

Fig. 5. Runtimes for varying key sizes and numbers of sensors.

figure that the predominant computational costs stem from cryptographic computations and are directly dependent on the chosen key size. The key size should be chosen such that sufficient security is achieved, and the current recommendation, when relying on the DCRA for security (difficulty of factorising N), is the use of 2048 bit length keys [33]. For our implementation of the filter, run on the aforementioned hardware, this results in a filter update computation duration of roughly $1.7s$.

7 Conclusion

We have presented a localisation filter in the presence of range-only sensors, which preserves both navigator and sensor privacies. A suitable cryptographic scheme has been introduced and a filter implementation compared and evaluated. Privacy-preserving range-only localisation is suitable for use in environments where sensor networks are untrusted or location is considered private and we hope to extend the method to broader measurement models in the future. Additional future work includes exploring more computationally efficient encryption schemes and filter reformulations to decrease the filter update duration, exploring the possible security implications of sensors that are not only honest-but-curious and expanding the LCAO notion to guarantee that consistent broadcasts are always made by the navigator.

References

[1] J.A. Pierce. An Introduction to Loran. *Proceedings of the IRE*, 34(5):216–234, 1946.

[2] M. Liggins, C. Y. Chong, D. Hall, and J. Llinas. *Distributed Data Fusion for Network-Centric Operations*. CRC Press, 2012.

[3] Xinya Li, Zhiqun Daniel Deng, Lynn T. Rauchenstein, and Thomas J. Carlson. Contributed Review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements. *Review of Scientific Instruments*, 87(4):921–960, 2016.

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

[5] Qi Wang, Zhansheng Duan, X. Rong Li, and Uwe D. Hanebeck. Convex Combination for Source Localization Using Received Signal Strength Measurements. In *21st International Conference on Information Fusion (Fusion 2018)*, pages 323–330, Cambridge, UK, 2018. IEEE.

[6] Tian He, Chengdu Huang, Brian M Blum, John A Stankovic, and Tarek Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *9th Annual International Conference on Mobile Computing and Networking*, pages 81–95, 2003.

[7] F. Beutler and U.D. Hanebeck. A New Nonlinear Filtering Technique for Source Localization. In *3rd IEEE Conference on Sensors (Sensors 2004)*, volume 1, pages 413–416, 2004.

[8] S. Gezici, Zhi Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via Ultra-Wideband Radios: A Look at Positioning Aspects for Future Sensor Networks. *IEEE Signal Processing Magazine*, 22(4):70–84, 2005.

[9] Michael Brenner, Jan Wiebelitz, Gabriele von Voigt, and Matthew Smith. Secret Program Execution in the Cloud Applying Homomorphic Encryption. In *5th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pages 114–119, 2011.

[10] Kui Ren, Cong Wang, and Qian Wang. Security Challenges for the Public Cloud. *IEEE Internet Computing*, 16(1):69–73, 2012.

[11] Shay Gueron. Intel Advanced Encryption Standard (AES) New Instructions Set. *Intel Corporation*, 2010.

[12] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM (CACM)*, 21(2):120–126, 1978.

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

[14] Elaine Shi, T-H Hubert Chan, and Eleanor Rieffel. Privacy-Preserving Aggregation of Time-Series Data. *Annual Network & Distributed System Security Symposium (NDSS)*, page 17, 2011.

[15] Marc Joye and Benoît Libert. A Scalable Scheme for Privacy-Preserving Aggregation of Time-Series Data. In *International Conference on Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 111–125. Springer, 2013.

[16] Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized Multi-Client Functional Encryption for Inner Product. In *Advances in Cryptology (ASIACRYPT)*, Lecture Notes in Computer Science, pages 703–732. Springer, 2018.

[17] Amr Alanwar, Yasser Shoukry, Supriyo Chakraborty, Paul Martin, Paulo Tabuada, and Mani Srivastava. PrOLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption. In *16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 41–52, 2017.

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

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

[20] Andreea B. Alexandru and George J. Pappas. Private Weighted Sum Aggregation. *arXiv*, 2020.

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

[22] Loukas Lazos and Radha Poovendran. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *ACM Workshop on Wireless Security (WiSe)*, page 21, Philadelphia, PA, USA, 2004. ACM.

[23] Irad Ben-Gal. Outlier Detection. In *Data Mining and Knowledge Discovery Handbook*, pages 131–146. Springer, Boston, MA, USA, 2005.

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

[25] Moritz Schulze Darup, Adrian Redder, and Daniel E. Quevedo. Encrypted Cooperative Control Based on Structured Feedback. *IEEE Control Systems Letters*, 3(1):37–42, 2019.

[26] Farhad Farokhi, Iman Shames, and Nathan Batterham. Secure and Private Control Using Semi-Homomorphic Encryption. *Control Engineering Practice*, 67:13–20, 2017.

[27] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*. Academic Press, 1982.

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

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

[30] John Bethencourt. Libpaillier. http://acsc.cs.utexas.edu/libpaillier/, 2010.

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

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

[33] Elaine Barker, Lily Chen, Allen Roginsky, Apostol Vassilev, Richard Davis, and Scott Simon. Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography. Technical Report NIST SP 800-56Br2, National Institute of Standards and Technology, Gaithersburg, MD, USA, March 2019.

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 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, $\mathcal{E}_{pk_0}(\omega_j^{(t)})$, $j \in \{1, \ldots, m\}$.

(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:

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

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 $\omega_j^{(t^*)}$, $j \in \{1, \ldots, m\}$ 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 A.1 *An encryption scheme meets LCAO security if no probabilistic adversary, running in polynomial-time with respect to security parameter κ, has more than a negligible advantage in winning the above security game. That is, for all adversaries $\mathcal{A}$, there exists a negligible function η, such that*

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

with probabilities taken over randomness introduced by $\mathcal{A}$, and in Setup, Enc *and* CombEnc.

B LCAO Scheme Proof

The scheme in section 4 will be shown to meet LCAO by contrapositive. We show that for any 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

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

for *any* negligible functions η_1, η_2 and security parameter κ. 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, [15]).

PROOF. Consider adversary $\mathcal{A}$ playing the LCAO game. The following is a construction of an adversary $\mathcal{A}'$ playing the AO game [14] 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 $\omega_j^{(t)}$, $j \in \{1, \ldots, m\}$ and random values $\rho_j \in \mathbb{Z}_N$, $j \in \{1, \ldots, m\}$, and return encryptions

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

to $\mathcal{A}$.

Combine Query When $\mathcal{A}$ submits combine query $(i, t, a_{i,1}^{(t)}, \ldots, a_{i,m}^{(t)})$, choose weights $\omega_j^{(t)}, j \in \{1, \ldots, m\}$ 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 $\omega_j^{(t^*)}, j \in \{1, \ldots, m\}$ 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, $i \in \{1, \ldots, n\}$ 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 exactly, and to $\mathcal{A}$, $\mathcal{A}'$ follows the scheme in section 4 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 η, such that

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

for security parameter κ. 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.