Encrypted Multisensor Information Filtering

Mikhail Aristov[*], **Benjamin Noack**[*], **Uwe D. Hanebeck**[*], and **Jörn Müller-Quade**[‡]
[*] Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
[‡] Cryptography and IT Security Group
Institute of Theoretical Informatics
[*,‡] Karlsruhe Institute of Technology (KIT), Germany
mikhail.aristov@student.kit.edu, benjamin.noack@ieee.org, uwe.hanebeck@ieee.org, joern.mueller-quade@kit.edu

***Abstract*—With the advent of cheap sensor technology, multisensor data fusion algorithms have been becoming a key enabler for efficient in-network processing of sensor data. The information filter, in particular, has proven useful due to its simple additive structure of the measurement update equations. In order to exploit this structure for an efficient in-network processing, each node in the network is supposed to locally process and combine data from its neighboring nodes. The aspired in-network processing, at first glance, prohibits efficient privacy-preserving communication protocols, and encryption schemes that allow for algebraic manipulations are often computationally too expensive. Partially homomorphic encryption schemes constitute far more practical solutions but are restricted to a single algebraic operation on the corresponding ciphertexts. In this paper, an additive-homomorphic encryption scheme is used to derive a privacy-preserving implementation of the information filter where additive operations are sufficient to distribute the workload among the sensor nodes. However, the encryption scheme requires the floating-point data to be quantized, which impairs the estimation quality. The proposed filter and the implications of the necessary quantization are analyzed in a simulated multisensor tracking scenario.**

I. Introduction

The acquisition and processing of sensor data is increasingly being implemented in networked systems [1], [2]. To leverage the potential of local processing power and storage capacity, distributed state estimation algorithms are the method of choice. With the advent of Kalman filter theory [3], fusion of estimates provided by independently operating agents has become a central topic of research in multisensor state estimation [4]–[9]. The in-network processing and fusion of sensor data relies on a thorough treatment of the estimation error statistics [10], [11], which are given by error covariance matrices. A key challenge in fusion pertains to the reconstruction of cross-covariance information that characterizes dependencies among the agents' local estimates. Optimal fusion [12], [13] exploits the cross-covariance structure to minimize the estimation error but requires precise knowledge about the very same. Although possibilities to keep track of cross-covariance information have been identified [14] or specific distributed implementations [15], [16] can be used to circumvent the need for reconstruction, these approaches typically increase complexity of local processing and limit flexibility and robustness. Alternatively, suboptimal strategies can be pursued that employ conservative bounds on the cross-covariance matrices. For this purpose, a wide variety of conservative data fusion methods [17]–[24] can be named.

A key concept for the processing of multisensor data is the information form [25] of the Kalman filter, which essentially represents an algebraic reformulation of the Kalman filtering equations—the approaches listed above are, in most cases, derived in information form or can be reformulated accordingly. The *information filter* offers the advantage that updating estimates with sensor observations can be expressed in terms of simple summations. In sensor networks, the information form is particularly useful to preprocess and compress sensor data along multi-hop communication paths. Hierarchical networks benefit from the additive structure, which allows for removing common information between estimates to be fused—the channel filter keeps track of common information and simply subtracts it from the fusion result [8], [26]. Also, fully distributed implementations [27] of the Kalman filter are related to the information form.

The information filter as a key tool to leverage efficient multisensor state estimation heavily relies on the possibility to algebraically manipulate sensor data when nodes exchange data with each other. An important implication is that the transmitted data and the intermediate processing results are disclosed to the receiving nodes in the network. Also, eavesdropping on the nodes' transmissions reveals possibly sensitive data to an adversary. So far, encryption has hardly been considered in the context of multisensor data fusion. A main reason can be seen in the structure of most popular encryption schemes, like AES [28], that prohibit algebraic operations on ciphertexts. With these schemes, encryption of sensor data deprives the information filter of its advantages. However, homomorphic encryption schemes [29]–[32] preserve the possibility of algebraic manipulations. These schemes are considered to be computationally infeasible unless we restrict ourselves to partially homomorphic systems that allow for either the addition or the multiplication of encrypted data without prior decryption. In this paper, we exploit the additive Paillier encryption scheme [30] to implement an encrypted information filter that enables us to protect sensor data from adversaries and to process the data within the network efficiently. We study the implications of the quantization of floating-point data required for the encryption and evaluate the proposed *privacy-preserving information filter* (PPIF) in a simulation.

II. Related Work

In recent years, encrypted signal processing has experienced a surge of interest. An overview of various applications of homomorphic encryption in signal processing is provided in [33], [34]. Also, a privacy-preserving implementation [35] of a Kalman filter has been studied that requires an interactive protocol between the sensor and the system operating the Kalman filter. A difficulty is, in particular, the inversion of the innovation matrix. An encrypted implementation of a model-predictive controller for constrained linear systems has been discussed in [36]. For statistical analyses of data, the additive Paillier scheme is utilized in [37]. First studies on using fully homomorphic schemes for least-squares regression have been conducted in [38], [39]. Yet, these scheme are perceived as rather impractical, and tailor-made solutions are required. Applications of such encryption schemes are particularly appealing in the context of cloud computing [40] and sensor networks [41]. In particular for the latter application scenario, the proposed privacy-preserving information filter proves useful as discussed in the following.

III. Kalman and Information Filtering

The *Kalman filter* [3], [42] is a well-established inference scheme to estimate the state $\underline{\mathbf{x}}_k \in \mathbb{R}^n$ of a dynamic linear system

$$\underline{\mathbf{x}}_{k+1} = \mathbf{A}_k\,\underline{\mathbf{x}}_k + \mathbf{B}_k\,\underline{\hat{u}}_k + \underline{\mathbf{w}}_k \ , \tag{1}$$

where $k \in \mathbb{N}$ denotes the discrete time. The process dynamics are characterized by the system matrix $\mathbf{A}_k \in \mathbb{R}^{n\times n}$, and a possible input $\underline{\hat{u}}_k \in \mathbb{R}^l$ may affect the state through the control-input matrix $\mathbf{B}_k \in \mathbb{R}^{n\times l}$. Uncertainties affecting the state transition are modeled by the zero-mean white noise $\underline{\mathbf{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$ with covariance matrix $\mathbf{C}_k^w \in \mathbb{R}^{n\times n}$. Observations of the state are provided by a network of $N \in \mathbb{N}$ sensor nodes. An estimate $\underline{\hat{x}}_k^{\mathrm{e}}$ is computed with the aid of the measurements $\underline{\hat{z}}_k^j \in \mathbb{R}^m$, where $j \in \{1,\ldots,N\}$ is the index of the sensor node providing the measurement. The corresponding model

$$\underline{\hat{z}}_k = \mathbf{H}_k^j\,\underline{\mathbf{x}}_k + \underline{\mathbf{v}}_k^j$$

is assumed to be linear with observation matrix $\mathbf{H}_k^j \in \mathbb{R}^{m\times n}$ and noise $\underline{\mathbf{v}}_k^j \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^{z,j})$ with covariance $\mathbf{C}_k^{z,j} \in \mathbb{R}^{m\times m}$. The measurement noise terms are assumed to be mutually uncorrelated and to be uncorrelated with the process noise.

The Kalman filter is initialized with a prior estimate $\underline{\hat{x}}_0^{\mathrm{e}}$ and prior covariance matrix $\mathbf{C}_0^{\mathrm{e}}$ and consists of prediction and filtering steps. In the former step, the process model (1) is used to compute a predicted estimate

$$\underline{\hat{x}}_{k+1}^{\mathrm{p}} = \mathbf{A}_k\,\underline{\hat{x}}_k^{\mathrm{e}} + \mathbf{B}_k\,\underline{\hat{u}}_k \tag{2}$$

with error covariance matrix

$$\mathbf{C}_{k+1}^{\mathrm{p}} = \mathbf{A}_k\,\mathbf{C}_k^{\mathrm{e}}\,\mathbf{A}_k^{\mathrm{T}} + \mathbf{C}_k^w \ . \tag{3}$$

The filtering step is used to incorporate a measurement $\underline{\hat{z}}_k^j$ at time step k. The update of the prior or predicted estimate $\underline{\hat{x}}_k^{\mathrm{p}}$ is given by

$$\underline{\hat{x}}_k^{\mathrm{e}} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k^j)\,\underline{\hat{x}}_k^{\mathrm{p}} + \mathbf{K}_k\,\underline{\hat{z}}_k^j \tag{4}$$

with covariance matrix

$$\mathbf{C}_k^{\mathrm{e}} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k^j)\mathbf{C}_k^{\mathrm{p}}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k^j)^{\mathrm{T}} + \mathbf{K}_k\mathbf{C}_k^{z,j}\mathbf{K}_k^{\mathrm{T}} \tag{5}$$

and Kalman gain

$$\mathbf{K}_k = \mathbf{C}_k^{\mathrm{p}}(\mathbf{H}_k^j)^{\mathrm{T}}\big(\mathbf{H}_k^j\mathbf{C}_k^{\mathrm{p}}(\mathbf{H}_k^j)^{\mathrm{T}} + \mathbf{C}_k^{z,j}\big)^{-1} \ . \tag{6}$$

In order to process multiple estimates, as it is required in the considered multisensor setup, the Kalman filter formulas (4), (5), and (6) have to be applied successively to each measurement. Alternatively, the measurements can be stacked into a vector to perform a blockwise update. However, both sequential and blockwise filtering do not serve well a distributed processing of measurements.

The *information filter* [25] is an algebraic reformulation of the Kalman filter. In the information form, the filtering step, i.e. equations (4) and (5), can be written as

$$(\mathbf{C}_k^{\mathrm{e}})^{-1}\,\underline{\hat{x}}_k^{\mathrm{e}} = (\mathbf{C}_k^{\mathrm{p}})^{-1}\,\underline{\hat{x}}_k^{\mathrm{p}} + \sum_{j=1}^{N}\underline{i}_k^j \ , \tag{7}$$

$$(\mathbf{C}_k^{\mathrm{e}})^{-1} = (\mathbf{C}_k^{\mathrm{p}})^{-1} + \sum_{j=1}^{N}\mathbf{I}_k^j \ , \tag{8}$$

where $\underline{i}_k^j$ and $\mathbf{I}_k^j$ are the information vectors and matrices, respectively, provided by the sensor nodes. They are locally computed by

$$\underline{i}_k^j = (\mathbf{H}_k^j)^{\mathrm{T}}\big(\mathbf{C}_k^{z,j}\big)^{-1}\,\underline{\hat{z}}_k^j \ , \tag{9}$$

$$\mathbf{I}_k^j = (\mathbf{H}_k^j)^{\mathrm{T}}\big(\mathbf{C}_k^{z,j}\big)^{-1}\mathbf{H}_k^j \ . \tag{10}$$

The update equations (7) and (8) offer the advantage that the sum of information parameters can be computed in a distributed fashion—each node can compute parts of the sum by combining the received information parameters. The combination with $\underline{\hat{x}}_k^{\mathrm{p}}$ and $\mathbf{C}_k^{\mathrm{p}}$ and the prediction step (2) and (3) is then performed in the data sink. In the following sections, a privacy-preserving formulation of the information filter update (7) and (8) is derived and studied.

IV. Cryptography Fundamentals

A. Provable Security

Modern cryptography is the "study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks" [43]. While cryptography has traditionally been associated with encryption, it covers a much broader spectrum of topics, including hash functions, message authentication, digital signatures, digital commitment schemes, zero-knowledge proofs, oblivious transfer, secure multiparty computation, etc. As a science, modern cryptography aims to provide rigorous *proofs* that certain cryptographic primitives and protocols fulfill specific *security definitions* under precisely formulated *assumptions*.

A specific security definition consists of a quantifiable *security goal* and a formal threat model. In case of encryption, a common security goal is the indistinguishably (IND) of ciphertexts, meaning that an adversary cannot distinguish

encryptions of equal-length plaintext messages from each other. The intuition behind this is that if all ciphertexts appear indistinguishable, no attacker can learn anything about their contents. The *threat model*, in turn, specifies how powerful the attacker is, in particular whether its running time is bounded by a probabilistic polynomial time (PPT) or is computationally unbounded. The former assumption lets us prove "computational security", while the latter leads to "information-theoretic security" or "perfect secrecy" (although, in practice, the latter is often prohibitively expensive).

The threat model also specifies the attacker's capabilities, e.g. whether it can only eavesdrop on encrypted communication (EAV), encrypt arbitrary plaintexts ("chosen-plaintext attack", CPA), or decrypt arbitrary ciphertexts in preparation for or even during the attack ("chosen-ciphertext attack", or CCA1, and "adaptive CCA", or CCA2, respectively) [44]. A common encryption security definition is IND-CPA: no PPT-bounded attacker may be able to distinguish encrypted messages, even if given the option to encrypt arbitrary messages under the same key. This definition is also known as "semantic security". More generally in secure multiparty computations, adversaries are classified as passive (also referred as "semi-honest") or active ("malicious"), based on whether corrupted parties still adhere to the protocol or deviate from it arbitrarily [45].

The actual proof of security of a given encryption scheme usually entails reducing it to the assumed (if not proven) hardness of certain problems. For example, the popular RSA (Rivest-Shamir-Adleman) encryption relies on the assumption that prime factorization is NP-hard, and proving its security requires showing that any efficient attacker who breaks RSA would also be able to factorize large composite numbers in PPT. RSA is therefore considered secure until the aforementioned assumption is proven incorrect.

B. Private-key and Public-key Encryption

Two paradigms are currently dominant in the field of cryptographic encryption. A private-key (also known as secret-key or "symmetric") encryption scheme uses the same encryption key (usually a random bit string of length λ) for both encrypting and decrypting messages, and consists of three algorithms:

$$\begin{aligned} sk &\leftarrow \mathrm{KeyGen}(1^\lambda) \\ c &\leftarrow \mathrm{Enc}_{sk}(m) \\ m &\leftarrow \mathrm{Dec}_{sk}(c) \end{aligned}$$

Meanwhile, public-key ("asymmetric") encryption uses two keys: the public key pk is freely distributed and can be used only to encrypt messages, while the secret key sk is kept private and used to decrypt ciphertexts encrypted under its corresponding public key:

$$\begin{aligned} pk, sk &\leftarrow \mathrm{KeyGen}(1^\lambda) \\ c &\leftarrow \mathrm{Enc}_{pk}(m) \\ m &\leftarrow \mathrm{Dec}_{sk}(c) \end{aligned}$$

Private-key encryption algorithms generally have the advantage of being faster and having smaller key and ciphertext sizes. Public-key encryption, on the other hand, simplifies key management, as public keys can be distributed over insecure channels (although some infrastructure is normally needed to ensure their authenticity), and n-way communication only requires $O(n)$ key pairs instead of $O(n^2)$ symmetric keys.

Popular private-key encryption schemes include the Advanced Encryption Standard (AES) [28], Serpent [46], and Twofish [47]. Some popular public-key encryption schemes include RSAEP-OAEP [48], ElGamal [29], Cramer-Shoup [49], and Paillier [30] cryptosystems.

C. Homomorphic Encryption

The core idea of homomorphic encryption (HE) is to combine two messages encrypted under the same key in such a way that produces a new valid ciphertext containing a meaningful combination of the old ones' contents, without exposing any information about them. For example, an encryption scheme with the additive homomorphic property allows us, without the need for a decryption key, to combine the encryptions of two numbers into a valid encryption of the sum of those numbers:

$$\mathrm{Enc}_{pk}(m_1) \odot \mathrm{Enc}_{pk}(m_1) = \mathrm{Enc}_{pk}(m_1 + m_2)$$

Both symmetric and asymmetric encryption schemes can have homomorphic properties, although most well-studied homomorphic encryptions are public-key. Two popular asymmetric schemes, by Taher ElGamal [29] and by Pascal Paillier [30], allow multiplicative and additive homomorphic operations on ciphertexts, respectively. These schemes are referred to as "partially homomorphic encryption" (PHE), as only one homomorphic algebraic operation is possible under them.

In contrast, "fully homomorphic encryption" (FHE) schemes allow for both addition and multiplication. The first such scheme to be proven secure was proposed by Craig Gentry in 2009 [50], but its computational complexity made it highly impractical for real applications. Although a number of more efficient fully-homomorphic schemes have been proposed since (see the survey in [51]), even the newest ones are still infeasible for large-scale data processing [38].

Having homomorphic properties has no adverse effect on the semantic security of a given scheme [31]. However, the inherent malleability of ciphertexts means that no homomorphic encryption can be secure against adaptive chosen-ciphertext attacks (IND-CCA2), which is the strongest commonly-considered encryption security guarantee [44].

D. Paillier Cryptosystem

The Paillier encryption scheme was proposed by Pascal Paillier in 1999 [30]. It is proven semantically secure under the decisional composite residuosity assumption, and has several useful homomorphic properties that make it attractive for secure signal processing [35]. Specifically, it allows for homomorphic addition of encrypted values and, consequently, multiplication of a ciphertext by a plaintext integer value:

$$\begin{aligned} \forall m \in \mathbb{Z}_n : \mathrm{Enc}_{pk}(m) &\in \mathbb{Z}_{n^2} \\ \mathrm{Enc}_{pk}(m_1) \cdot \mathrm{Enc}_{pk}(m_2) &= \mathrm{Enc}_{pk}(m_1 + m_2) \\ (\mathrm{Enc}_{pk}(m_1))^{m_2} &= \mathrm{Enc}_{pk}(m_1 \cdot m_2) \end{aligned}$$

where $\mathbb{Z}_n$ represents integers modulo n, a product of two large primes that is part of the public key. All homomorphic computations on plaintexts under Paillier scheme are thus carried out in the finite ring $\mathbb{Z}_n$.

V. Privacy-Preserving Information Filter

In this section we describe our proposal for a protocol we refer to as "privacy-preserving information filter" (or "PPIF" for short). We attempt to identify and to leverage potential synergies between the various tools and methods employed in the fields of sensor fusion and state estimation and of modern cryptography. The starting point of and the intuition behind PPIF is the question: How can we use the additive homomophic properties of our chosen encryption scheme (specifically, the Paillier cryptosystem) to implement secure multisensor fusion in a somewhat realistic scenario?

A. Scenario

Consider the following setup: A mobile agent traverses unknown terrain and must rely on measurements by multiple externally-operated sensors for its localization. For the sake of simplicity and without loss of generality, we assume that the agent has no internal sensors of its own. The agent and the individual sensors can communicate with each other over insecure channels, but neither the agent, nor the sensor grid operator can fully trust that the other has not been corrupted by a third party attacker. Here, like in most research on secure signal processing [33], we assume a static PPT-bounded semi-honest (passive) adversary.

The agent now defines the following (informal) security goals:

A1: No adversary eavesdropping on the agent's communication with the sensors may gain any meaningful information about its location.

A2: No adversary corrupting one or more sensors may gain any more meaningful information than it would just from these corrupted sensors. In particular, a corrupted sensor should be unable to learn anything from its honest neighbors.

The sensor grid operator defines its security goal as:

S1: No adversary eavesdropping on the communication or compromising the agent's system may gain any meaningful information about the locations and the measurement models of any individual sensor.

Clearly, an encryption of some kind is required to securely transmit the sensors' measurement data to the agent over insecure communication channels. In the following, we assume that the agent possesses a public/private key pair, and that its public key has been distributed to every sensor ahead of time.

B. Prediction Step

While the agent has no internal sensors, it is reasonable to assume that it at least knows its own system model. Because all information needed for this step is available to the agent locally in plaintext, this computation can occur unencrypted. The prediction step is conducted by means of the Kalman filter equations (2) and (3) for the estimate and the covariance matrix, respectively.

In the simplest case, the system vector $\hat{\underline{x}}_k$ consists only of the agent's current global coordinates, so for all k, $\mathbf{A}_k$ is the identity, the control matrix $\mathbf{B}_k$ is null, and its movement speed is mapped in the process noise $\mathbf{C}_k^w$. With this, the entire prediction step is reduced to a single matrix addition:

$$\mathbf{C}_{k+1}^{\mathrm{p}} = \mathbf{C}_k^{\mathrm{e}} + \mathbf{C}_k^w.$$

C. Filter Step

To attain the security goal S1, the sensors in PPIF do not communicate with the agent individually, but instead, aggregate their (encrypted) data locally and transmit the total sum of their observations to the agent at once. This aggregation is made possible by the homomorphic properties of the encryption. We propose that the sensors be arranged in a tree-like hierarchy, with each one encrypting its measurement of the agent's location, and sending it "upstream" to its respective hub, until finally the central hub of the grid sends the aggregated measurements to the agent. What is being encrypted and transmitted here, however, are not the raw measurements $\hat{\underline{z}}_k^j$ (where j is the sensor's identifier in the grid), but their information form $\underline{i}_k^j$, as well as their respective covariances $\mathbf{I}_k^j$, also in information form. They are given by (9) and (10).

Recall that in the information filter, the filtering step consists simply of summing up all information vectors, then multiplying them by the inverse of the sum of all information matrices to obtain the estimate. Because it does not matter where exactly this summation occurs, the additive homomorphic property of the encryption scheme allows each sensor hub to pre-aggregate the encrypted information vectors and matrices it receives before sending them "upstream". Crucially, the hub gains no knowledge of the actual measurement data, and the message it sends is computationally indistinguishable from any it received.

The agent ultimately receives a single message from the central hub, containing an encrypted aggregate vector $\underline{i}_k = \sum_j \underline{i}_k^j$ and matrix $\mathbf{I}_k = \sum_j \mathbf{I}_k^j$, which it decrypts with its secret key and merges into its (unencrypted) prediction according to

$$\begin{aligned}\mathbf{C}_k^{\mathrm{e}} &= \left((\mathbf{C}_k^{\mathrm{p}})^{-1} + \mathbf{I}_k\right)^{-1}, \\ \hat{\underline{x}}_k^{\mathrm{e}} &= \mathbf{C}_k^{\mathrm{e}} \left((\mathbf{C}_k^{\mathrm{p}})^{-1} \hat{\underline{x}}_k^{\mathrm{p}} + \underline{i}_k\right).\end{aligned}$$

Note that because the agent only receives aggregate values, rather than individual measurement results, it would be very hard for any adversary controlling it to reconstruct the layout of the sensor grid or to obtain details of the individual sensors from this data, as doing so would require solving an underdetermined system of equations with an unknown number of unknowns. This is because the total number of aggregated measurements is never transmitted and may vary from aggregate to aggregate, e.g. if some sensors are unable to obtain a measurement because of terrain. This finally allows us to satisfy the security goal S1.

D. Encoding the Measurement Data

A major hurdle when applying cryptographic methods to signal processing is the fundamental difference in their computation domains: whereas signal processing operates primarily

with real floating-point numbers, the Paillier cryptosystem and many other homomorphic encryptions operate on finite groups, specifically integers modulo a large semiprime. Before encrypting their measurements, the sensors in PPIF must therefore quantize the raw floating-point data into fixed-point values. We define this quantization $q : \mathbb{R} \to \mathbb{Z}$ as $q(r) = \lfloor 2^f r \rceil$, where f is the desired fractional precision in bits. The reverse conversion $q^{-1} : \mathbb{Z} \to \mathbb{R}$ is then given as $q^{-1}(d) = 2^{-f} d$. The precision loss of this conversion is bounded by $2^{-(f+1)}$.

Because the Paillier cryptosystem only accepts plaintexts from the finite ring $\mathbb{Z}_n$ (integers modulo semiprime n) rather than $\mathbb{Z}$, we only have a limited range of accepted values. In practice, however, this is not particularly restrictive, even with a large fractional precision, since n needs to be very large to guarantee sufficient security. The German Federal Office for Information Security, for instance, currently recommends at least 2000 bit length for factoring moduli in RSA and similar cryptosystems [52].

An additional challenge arises when we want to encrypt negative numbers, since the range of $\mathbb{Z}_n$ is normally defined as $[0, n)$. This, however, is elegantly solved through a quirk of modular arithmetic, namely that $-a \equiv (n - a) \pmod{n}$. With this, we can map negative measurements to the upper half of the plaintext domain and still reap full benefits of the homomorphic operations. While this mapping effectively halves the largest allowed plaintext value, it is hardly significant, given the enormous size of the plaintext domain.

Finally, we must define the encoding of arrays (vectors and matrices). Since Paillier cannot natively encrypt such structures, we must preserve the array layout, simply replacing plaintext values with their encryptions. Fortunately, every transmitted message in the PPIF protocol contains the same number of values, since the sizes of $\underline{i}_k^j$, $\mathbf{I}_k^j$, and their respective aggregates are fixed and dependent only on the size of the agent's system state. Because each message has the exact same length, the semantic security offered by the Paillier schema holds for them just as well as for the individual ciphertexts therein (see [43, Theorem 11.6]).

For data aggregation purposes, the arrays can be simply summed up element-wise using the homomorphic addition operation of the Paillier schema. To conserve computing power and bandwidth, we may exploit the symmetry of $\mathbf{I}_k^j$ and only encrypt and transmit its main diagonal and values above it.

VI. Evaluation

In this section, we evaluate our proposed PPIF protocol with regard to its accuracy and security.

A. Accuracy

The first thing to note when evaluating our PPIF protocol is that the encryption itself has no effect on the accuracy of its resultant estimates. This follows from the basic correctness property of the encryption scheme, i.e. that $\mathrm{Dec}_{sk}(\mathrm{Enc}_{pk}(m)) = m$, which was proven in Paillier's original work. This property holds regardless of whether homomorphic operations have been applied to the ciphertext, as long as the plaintext within remains bounded by the public modulus n (or by the range $[-\frac{n}{2}, \frac{n}{2})$ if we encode for negative plaintexts), which, as we have discussed in the previous section, should be the case in realistic scenarios.

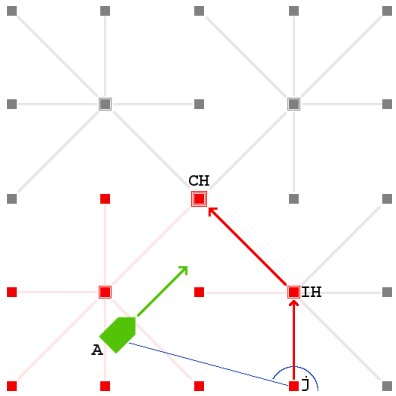

Fig. 1: Each sensor j measures the distance and angle to agent A and sends the information form to its intermediary hub IH for aggregation, which, in turn, sends it on to the central hub CH. Grayed-out sensors are outside the maximum measurement range.

It is clear then, that the only phase of our protocol that can reduce its resultant accuracy (compared to an unencrypted execution on the original floating-point inputs) is our quantization method. To evaluate its effects under different parameters, we have developed a simulation[1] (see Figure 1), wherein a mobile agent is initially placed on a uniformly random position at the edge of the 2D square field. Fo the sake of simplicity, its velocity vector is sampled from $\mathcal{N}(0, \mathbf{C}_k^w)$ once at the start of the run and held constant for its entire duration. The agent collects measurements from the sensors and estimates its own location at every time step until it leaves the field, updating the mean squared error (MSE) of both the encrypted and the unencrypted estimators each time. The experiment is repeated $10\,000$ times to obtain N_{est} estimates.

The field is covered with a grid of 25 equidistant radar-like sensors, arranged in a three-tier hierarchy, with the central sensor aggregating all others' data before sending it to the agent. Each sensor simulates a noisy measurement of the agent's relative location in polar coordinates (with known uncorrelated variances $\sigma_\varphi^2, \sigma_r^2$) and converts it to a linearized model in global Cartesian coordinates. The result and its estimated covariance (both in information form) are then quantized, encrypted, and sent "upstream". Each sensor has a maximum detection range $r_{\max}$ and generates no measurements of its own until the agent comes within said range (hubs do, however, send measurements "upstream" if they receive any).

Table I shows some results produced by our simulation. In all examples, the width of the simulated field is fixed at $100\,\mathrm{m}$, σ_v is $5\,\mathrm{m/s}$; "BP" stands for the fractional precision in bits. We observe that the PPIF and the unencrypted information filter show very similar accuracy for fractional precision of 16 bit or higher, with their mean squared error converging

[1]The Python code is available at https://github.com/KIT-ISAS/PPIF.

TABLE I: Mean squared errors of the simulated PPIF estimates and controls under different parameters.

BP	σ_φ	σ_r	$r_{\max}$	N_{est}	MSE_{PPIF}	MSE_{IF}
8	5°	2 m	50 m	190 991	1.497 207 52	1.398 596 18
8	5°	2 m	200 m	189 437	0.730 031 35	0.689 676 14
8	15°	5 m	50 m	186 230	11.945 691 1	8.567 850 67
16	5°	2 m	50 m	192 918	1.392 861 12	1.392 861 58
16	5°	2 m	200 m	187 828	0.687 671 72	0.687 672 86
16	15°	5 m	50 m	188 072	8.640 616 91	8.640 362 42
24	5°	2 m	50 m	192 897	1.393 730 54	1.393 730 55
24	5°	2 m	200 m	192 211	0.690 740 24	0.690 740 24
24	15°	5 m	50 m	196 549	8.643 080 32	8.643 080 91

down to a few square micrometers. We therefore conclude that even with a relatively small fractional precision, the estimates produced by the PPIF are as accurate for practical purposes as those of an unencrypted information filter.

B. Security

In a previous section, we have already described the intuition behind the proof of security of the PPIF against semi-honest (passive) adversaries. We have identified that such an adversary can eavesdrop on communication between sensors and agents, or take control of ("corrupt") either the agent, or one or more of the sensors. Note that our security definition did not require the sensor grid operator to be unable to learn anything about the agent's location, as that would deprive the grid of its primary function; for the same reason, our protocol offers no security against an adversary who corrupts the majority of the sensor nodes. With this in mind, we have shown that:

- An eavesdropper cannot gain any information on the state and properties of the agent and individual sensors, since all data packages it can intercept have the same length and are computationally indistinguishable from each other under the decisional composite residuosity assumption.
- A corrupted sensor node cannot learn anything about the agent beyond what it can directly measure, because all messages it receives from its neighbors are computationally indistinguishable. Among other things, it cannot distinguish how many measurements, if any, have been aggregated in the message it had received.
- A corrupted agent cannot obtain any meaningful data about the positions and internal properties of individual sensors, as doing so would require it to solve an underdetermined system of equations with an unknown number of variables.

At this point, however, we have to establish the limitations of our security definition. For one, it offers no protection against trivial side-channel attacks. An eavesdropper monitoring network traffic in a realistic scenario can, for example, gain a pretty good idea of the sensor grid's topology, simply by observing which nodes communicate with each other. While this weakness can be mitigated, e.g. by having sensors randomly send zero vector encryptions to arbitrary nodes to hide the real payload, it would result in substantial additional computational and network load. For simplicity's sake, we have also neglected the necessity of authenticating each message in order to avoid man-in-the-middle injections; in practice, such authentication would be implemented via digital signatures.

Lastly, while the protocol is relatively secure against passive adversaries, it is highly vulnerable against malicious (active) ones. Gaining control over any of the sensors, regardless of its position in the grid, allows an active adversary to completely falsify the estimation results by transmitting arbitrary "measurements" with an unrealistically small covariance as overinflated information vectors. While such attacks are also possible in the unencrypted setup, encrypting the measurements lets the attacker hide its presence, as individual nodes cannot implement local sanity checks on data sent by their neighbors and isolate the malicious sensors. A much more complex protocol is required to defend against active adversaries, which will be the subject of our future research.

VII. Conclusion

We have described a practical solution for privacy-preserving multisensor information filtering. Our PPIF protocol leverages the additive homomorphic property of the Paillier encryption scheme, and we have provided both a security definition for it and a basic intuition for its cryptographic proof. We have also evaluated its accuracy and found it comparable to that of an unencrypted information filter. We therefore conclude that our protocol is efficient and secure enough to be used in suitable real world settings.

It is clear, however, that our protocol solves a very special formulation of the state estimation problem. Specifically, in order to use only homomorphic additions in the encrypted domain, all parties (particularly the sensors) must have sufficient computational power to carry out all other necessary operations, e.g. matrix inversion and multiplication, locally in plaintext. Furthermore, all data must be decrypted for both the prediction and the final filtering step, making it impossible to securely outsource these computations, e.g. into the cloud.

Our work here can serve as the foundation for future solutions that would transcend the above limitations, but further research into combining modern cryptography and signal processing is necessary before these can be found.

References

[1] D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, Eds., *Distributed Data Fusion for Network-Centric Operations*, ser. The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2013.

[2] C.-Y. Chong, "Forty Years of Distributed Estimation: A Review of Noteworthy Developments," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2017)*, Bonn, Germany, Oct. 2017.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[4] D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Algorithms for a Multi-Sensor System," in *Proceedings of the 15th IEEE Conference on Decision and Control (CDC 1976)*, Clearwater, FL, USA, Dec. 1976.

[5] C.-Y. Chong, "Hierarchical estimation," in *MIT/ONR Workshop on C3 Systems*, Monterey, California, USA, 1979.

[6] C.-Y. Chong, K.-C. Chang, and S. Mori, "Distributed Tracking in Distributed Sensor Networks," in *Proceedings of the 1986 American Control Conference (ACC 1986)*, Seattle, Washington, USA, 1986.

[7] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized Structures for Parallel Kalman Filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988.

[8] S. Grime and H. F. Durrant-Whyte, "Data Fusion in Decentralized Sensor Networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, Oct. 1994.

[9] B. Noack, *State Estimation for Distributed Systems with Stochastic and Set-membership Uncertainties*, ser. Karlsruhe Series on Intelligent Sensor-Actuator-Systems 14. Karlsruhe, Germany: KIT Scientific Publishing, 2013.

[10] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. John Wiley & Sons, 2001.

[11] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion," in *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, H. Fourati, Ed. CRC Press, Aug. 2015, pp. 169–192.

[12] Y. Bar-Shalom and L. Campo, "On the Track-to-Track Correlation Problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, Apr. 1981.

[13] S.-L. Sun and Z.-L. Deng, "Multi-sensor Optimal Information Fusion Kalman Filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, Jun. 2004.

[14] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck, "Optimal Sample-Based Fusion for Distributed State Estimation," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016.

[15] F. Govaers and W. Koch, "An Exact Solution to Track-to-track Fusion at Arbitrary Communication Rates," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, Jul. 2012.

[16] M. Reinhardt, B. Noack, and U. D. Hanebeck, "Advances in Hypothesizing Distributed Kalman Filtering," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[17] N. A. Carlson, "Federated Filter for Fault-tolerant Integrated Navigation Systems," in *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS'88)*, Orlando, Florida, USA, 1988, pp. 110–119, record 'Navigation into the 21st Century' (IEEE Cat. No.88CH2675-7).

[18] S. J. Julier and J. K. Uhlmann, "A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations," in *Proceedings of the IEEE American Control Conference (ACC 1997)*, vol. 4, Albuquerque, New Mexico, USA, Jun. 1997, pp. 2369–2373.

[19] J. Sijs and M. Lazar, "State-fusion with Unknown Correlation: Ellipsoidal Intersection," *Automatica*, vol. 48, no. 8, pp. 1874–1878, Aug. 2012.

[20] B. Noack, J. Sijs, and U. D. Hanebeck, "Algebraic Analysis of Data Fusion with Ellipsoidal Intersection," in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.

[21] B. Noack, J. Sijs, and U. D. Hanebeck, "Inverse Covariance Intersection: New Insights and Properties," in *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*, Xi'an, China, Jul. 2017.

[22] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized Data Fusion with Inverse Covariance Intersection," *Automatica*, vol. 79, pp. 35–41, May 2017.

[23] B. Chen, G. Hu, D. W. C. Ho, and L. Yu, "Distributed Covariance Intersection Fusion Estimation for Cyber-Physical Systems With Communication Constraints," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4020–4026, Dec. 2016.

[24] J. Ajgl, M. Šimandl, M. Reinhardt, B. Noack, and U. D. Hanebeck, "Covariance Intersection in State Estimation of Dynamical Systems," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[25] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, Florida, USA: CRC Press, Inc., 1998.

[26] B. Noack, D. Lyons, M. Nagel, and U. D. Hanebeck, "Nonlinear Information Filtering for Distributed Multisensor Data Fusion," in *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, California, USA, Jun. 2011.

[28] Federal Information Processing Standards Publication, "Announcing the Advanced Encryption Standard (AES)," 2001.

[27] F. Pfaff, B. Noack, U. D. Hanebeck, F. Govaers, and W. Koch, "Information Form Distributed Kalman Filtering (IDKF) with Explicit Inputs," in *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*, Xi'an, China, Jul. 2017.

[29] 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.

[30] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.

[31] C. Gentry, "Computing Arbitrary Functions of Encrypted Data," *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, Mar. 2010.

[32] I. Sharma, "Fully Homomorphic Encryption Scheme with Symmetric Keys," *arXiv:1310.2452 [cs]*, Oct. 2013, arXiv: 1310.2452.

[33] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted Signal Processing for Privacy Protection: Conveying the Utility of Homomorphic Encryption and Multiparty Computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, Jan. 2013.

[34] T. Shortell and A. Shokoufandeh, "Secure Signal Processing Using Fully Homomorphic Encryption," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science, S. Battiato, J. Blanc-Talon, G. Gallo, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham: Springer International Publishing, 2015, vol. 9386.

[35] F. J. González-Serrano, A. Amor-Martín, and J. Casamayón-Antón, "State Estimation Using an Extended Kalman Filter with Privacy-Protected Observed Inputs," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2014, pp. 54–59.

[36] M. S. Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo, "Towards Encrypted MPC for Linear Constrained Systems," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 195–200, Apr. 2018.

[37] T. Y. Youn, N. S. Jho, and K. Y. Chang, "Practical Additive Homomorphic Encryption for Statistical Analysis over Encrypted Data," in *2016 International Conference on Platform Technology and Service (PlatCon)*, Feb. 2016, pp. 1–7.

[38] Y. Du, L. Gustafson, D. Huang, and K. Peterson, "Implementing ML Algorithms with HE," MIT Course 6.857: Computer and Network Security, 2017.

[39] P. M. Esperança, L. J. M. Aslett, and C. C. Holmes, "Encrypted accelerated least squares regression," *arXiv:1703.00839 [cs, stat]*, Mar. 2017, arXiv: 1703.00839.

[40] M. Togan and C. Pleşca, "Comparison-Based Computations Over Fully Homomorphic Encrypted Data," in *2014 10th International Conference on Communications (COMM)*, May 2014, pp. 1–6.

[41] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks," in *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Jul. 2005, pp. 109–117.

[42] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[43] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed. CRC Press, 2015.

[44] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Advances in Cryptology — CRYPTO '98*, H. Krawczyk, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 26–45.

[45] C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols*. Springer Berlin Heidelberg, 2010.

[46] E. Biham, R. Anderson, and L. Knudsen, "Serpent: A new block cipher proposal," in *International Workshop on Fast Software Encryption*. Springer, 1998, pp. 222–238.

[47] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-bit block cipher," *NIST AES Proposal*, vol. 15, 1998.

[48] B. Kaliski and J. Staddon, "PKCS# 1: RSA cryptography specifications version 2.0," Tech. Rep., 1998.

[49] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Advances in Cryptology — CRYPTO '98*, H. Krawczyk, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 13–25.

[50] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC'09*. Stanford University and IBM Watson, May 2009.

[51] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *arXiv preprint arXiv:1704.03578*, 2017.

[52] *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, Bundesamt für Sicherheit in der Informationstechnik Std., Rev. 2018-01, Jan. 2018.