Secure Fast Covariance Intersection Using Partially Homomorphic and Order Revealing Encryption Schemes

Marko Ristic, Benjamin Noack, and Uwe D. Hanebeck

***Abstract*—Fast covariance intersection is a widespread technique for state estimate fusion in sensor networks when cross-correlations are not known and fast computations are desired. The common requirement of sending estimates from one party to another during fusion forfeits local privacy. Current secure fusion algorithms rely on encryption schemes that do not provide sufficient flexibility and as a result require excess communication between estimate producers which is often undesirable. We propose a novel method of homomorphically computing the fast covariance intersection algorithm on estimates encrypted with a combination of encryption schemes. Using order revealing encryption, we show how an approximate solution to the fast covariance intersection weights can be computed and combined with partially homomorphic encryptions of estimates, to calculate an encryption of the fused result. The described approach allows secure fusion of any number of private estimates, making third-party cloud processing a viable option when working with sensitive state estimates or when performing estimation over untrusted networks.**

***Index Terms*—Sensor fusion, Estimation, Kalman filtering**

I. Introduction

SENSOR data processing and state estimation have been increasingly prevalent in networked systems [1]. Bayesian state estimation has become a particularly common application since the beginning of Kalman estimation theory and has led to a large interest in the field of state estimation fusion [2]–[4]. Challenges of estimation fusion are closely tied to the handling and merging of estimation error statistics [5]. Cross-correlations between estimation errors characterize dependencies between local estimates and must be considered when performing consistent or optimal fusion [6], [7]. Methods that keep track of these cross-correlations may require repeated reconstruction [8] and typically add local computational complexity. An alternative strategy sees the approximation of error cross-correlation based on conservative suboptimal strategies, and has been implemented in a variety of methods [9]–[11]. Covariance Intersection (CI) [9] provides one such popular strategy, from which a less computationally expensive method, the Fast Covariance Intersection (FCI) [11] has been derived. CI is particularly well paired with the information form of the Kalman filter [12], [13]. This algebraically equivalent form of the standard Kalman filter requires the persistent storing of the information vector and information matrix instead of the usual state estimate and estimate covariance, and reduces fusion operations to simple summations.

As advancements in distributed algorithms and cloud computing develop, the requirements for privacy and security in such systems have become more apparent [14], [15]. This is particularly pertinent for sensor networks, where the desire for sensitive hardware information or estimation methodology to remain private may require the privacy of local measurements and estimates as well, and is a non-trivial problem in networks containing eavesdroppers or untrusted parties. Encryption has until recently been primarily used to secure information transfer between communicating parties, relying on symmetric-key encryption schemes such as AES [16] to encrypt sent information, and public-key schemes such as RSA [17] to distribute symmetric keys. However, recent developments in public-key Homomorphic Encryption (HE) schemes [18]–[21], which allow algebraic operations to be performed on encryptions, are leading to novel secure applications for signal processing in distributed and cloud computing environments [22]–[27]. Fully Homomorphic Encryption (FHE) schemes [18], [19] allow algebraic operations to be performed on encryptions, and are often theoretically suitable for secure processing in distributed environments, but current implementations are still computationally infeasible for large-scale or real-time processing [28]. Partially Homomorphic Encryption (PHE) schemes [20], [21], providing only a subset of these operations, have been a focus for such tasks due to their reduced computational requirements. [23] use PHE to run a private distributed Information Filter, [24], [25] to compute private distributed control aggregation, [26] for private matrix multiplication, and [27] for private set intersection, however these works are relatively restricted in application due to the limited operations provided by PHE. Recent developments in new encryption schemes, such as Order Revealing Encryption (ORE) [29]–[31], are now providing new light on the possible complexity of securely computable algorithms. In this paper, we develop a method for secure FCI fusion, such that local sensor information is kept private, using a combination of ORE and PHE schemes only, which has to the best of our knowledge not been achieved without the reliance on computationally expensive FHE schemes.

A. Problem Formulation

Our paper is motivated by a key step in multi-sensor fusion, the requirement of transmitting local sensor state estimates and

Marko Ristic, Benjamin Noack, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), Germany.
{marko.ristic,noack,uwe.hanebeck}@kit.edu

covariance information over a network for the computation of their fused result. In particular, we consider centralized FCI fusion, where a party responsible for many networked sensors capable of computing their local state estimates, wishes to have their fused state estimate and covariance computed securely on an untrusted cloud. The same party may query the cloud fusion center for the fused result at any time. To preserve the privacy of local sensor measurements and state estimates, we aim to provide a secure FCI algorithm such that the fusion center does not learn individual sensor measurements, state estimates, or covariances. This will be achieved by encrypted homomorphic fusion, whereby the untrusted cloud learns only the FCI aggregation weights, which will be shown in section IV.

As we assume the querying party is the owner of all individual sensors, the threat model to be considered is that of network eavesdroppers and a malicious fusion center, with no possible collusion between sensors and the fusion center.

B. Notation

Throughout this paper we will use the following notation. Lowercase characters represent scalars, and underlined characters, $\underline{x}$, represent vectors. Uppercase bold characters, $\mathbf{M}$, are for matrices, where $\mathbf{M}^{-1}$ denotes the matrix inverse, and $\mathrm{tr}(\cdot)$ the trace function. Covariance matrices will be represented by $\mathbf{P}$. $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{ORE,k}(a)$ denote the encryption of a using the public-key pk and ORE key k, respectively, and similarly with the decryption functions $\mathcal{D}_{sk}(\cdot)$ and $\mathcal{D}_{ORE,k}(\cdot)$ with secret key sk, where any required real-number encodings of the number a are assumed to be performed. $\mathcal{E}(a)$ and $\mathcal{E}_{ORE}(a)$ may be used for brevity when the encryption keys can be inferred from context. All encryption of vectors and matrices are defined element-wise, with elements given by $\mathcal{E}(\mathbf{P}_{i,j}) = \mathcal{E}(\mathbf{P})_{i,j}$. Sets are represented as $\{\cdot\}$ and ordered lists with $[\cdot]$.

II. Covariance Intersection and Approximations

Covariance Intersection (CI), introduced in [9], provides a consistent state estimate fusion algorithm when cross-correlations are not known. The resulting fused estimate $\hat{\underline{x}}$ and covariance $\mathbf{P}$ can be easily derived from its equations

$$\mathbf{P}^{-1} = \sum_{i=1}^{n} \omega_i \mathbf{P}_i^{-1},\ \mathbf{P}^{-1}\hat{\underline{x}} = \sum_{i=1}^{n} \omega_i \mathbf{P}_i^{-1}\hat{\underline{x}}_i\,. \tag{1}$$

Note that (1) computes the fusion of the information vectors and information matrices defined in [11] and reduces the fusion to a weighted sum. Values for weights ω_i must satisfy

$$\omega_1 + \omega_2 + \cdots + \omega_n = 1,\ 0 \le \omega_i \le 1\,, \tag{2}$$

which guarantees consistency of the fused estimates. They are chosen in a way to speed up convergence and minimize error by minimizing a certain specified property of the resulting fused estimate covariance. One such property, the covariance trace, requires the solution to

$$\underset{\omega_1,\ldots,\omega_n}{\arg\min}\{\mathrm{tr}(\mathbf{P})\} = \underset{\omega_1,\ldots,\omega_n}{\arg\min}\left\{\mathrm{tr}\left(\left(\sum_{i=1}^{n}\omega_i\mathbf{P}_i^{-1}\right)^{-1}\right)\right\} \tag{3}$$

for computing weights ω_i. However, minimizing this non-linear cost function can be very computationally costly and has led to the development of faster approximation techniques.

A. Fast Covariance intersection

The Fast Covariance Intersection (FCI) algorithm from [11] is a non-iterative method for approximating the solution to (3) without the loss of guaranteed consistency. It is computed by defining a new constraint

$$\omega_i\,\mathrm{tr}(\mathbf{P}_i) - \omega_j\,\mathrm{tr}(\mathbf{P}_j) = 0,\ i,j = 1,2,\ldots,n \tag{4}$$

on ω_i and solving the resulting equations instead. In the two sensor case, this results in the solving of

$$\omega_1\,\mathrm{tr}(\mathbf{P}_1) - \omega_2\,\mathrm{tr}(\mathbf{P}_2) = 0,\ \omega_1 + \omega_2 = 1\,. \tag{5}$$

When computed for n sensors, the highly redundant (4) can have its largest linearly independent subset represented by

$$\omega_i\,\mathrm{tr}(\mathbf{P}_i) - \omega_{i+1}\,\mathrm{tr}(\mathbf{P}_{i+1}) = 0,\ i = 1,2,\ldots,n-1\,, \tag{6}$$

and requires the solution to the linear problem

$$\begin{bmatrix} \mathcal{P}_1 & -\mathcal{P}_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \mathcal{P}_{n-1} & -\mathcal{P}_n \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{7}$$

where we let $\mathcal{P}_i = \mathrm{tr}(\mathbf{P}_i)$.

Our proposed filter aims to solve FCI fusion, namely (1) and (7), using only encrypted values from each sensor i, and leaking only the weight values $\omega_1,\ldots,\omega_n$.

III. Homomorphic and Order Revealing Encryption

To achieve a secure solution to the FCI fusion problem, we make use of two types of function-providing encryption schemes. The Paillier additive PHE scheme [21] which provides a single homomorphic addition operation, and the Lewi ORE scheme [30] which provides a secure comparison function.

The formal security of an encryption scheme consists of a security goal and a formal threat model [32]. Indistinguishability of ciphertexts under the adaptive chosen ciphertext attack model (IND-CCA2) is commonly considered the strongest security guarantee, however no homomorphic encryption scheme provides security against IND-CCA2 due to their apparent ability to create valid cyphertexts via homomorphic operations. Instead, PHE schemes aim to protect against the weaker assumption of the chosen plaintext attack model (IND-CPA) [33]. Similarly, ORE schemes aim to protect against simulation-based security defined in [29] or the harder to achieve ordered chosen-plaintext attack model (IND-OCPA).

A. Paillier Partially Homomorphic Encryption Scheme

We use the Paillier additive PHE scheme due to its implementation simplicity, and computational speed. The Paillier scheme provides two homomorphic operations on encrypted data, namely

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(a)\mathcal{E}_{pk}(b) \pmod{N^2}) = a + b \pmod{N} \tag{8}$$

and

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(a)^c \pmod{N^2}) = c \cdot a \pmod{N},\ c \in \mathbb{Z}_N\,, \tag{9}$$

where the modulus N is computed as the product of two large random primes chosen at key-generation. The public and secret keys are shown as pk and sk respectively, and plaintext messages $a, b \in \mathbb{Z}_N$. The Paillier encryption scheme successfully provides security against the IND-CPA model.

B. Lewi Left-Right Order Revealing Encryption

For ORE, we use the Lewi symmetric-key Left-Right ORE scheme as it has the added property of only allowing certain comparisons between cyphertexts. This property can be used to decide which values may not be compared, which will be shown in section IV. It is described as follows: two encryption functions allow integers to be encrypted as either a "Left" (L) or "Right" (R) encryption by

$$\begin{aligned} \mathrm{enc}^{\mathrm{L}}_{\mathrm{ORE}}(\mathrm{k},\mathrm{x}) &= \mathcal{E}^{\mathrm{L}}_{\mathrm{ORE,k}}(\mathrm{x}) \\ \mathrm{enc}^{\mathrm{R}}_{\mathrm{ORE}}(\mathrm{k},\mathrm{y}) &= \mathcal{E}^{\mathrm{R}}_{\mathrm{ORE,k}}(\mathrm{y}) \end{aligned}, \tag{10}$$

and only comparisons between an L and an R encryption are possible, by

$$\mathrm{cmp}_{\mathrm{ORE}}(\mathcal{E}^{\mathrm{L}}_{\mathrm{ORE}}(\mathrm{x}),\ \mathcal{E}^{\mathrm{R}}_{\mathrm{ORE}}(\mathrm{y})) = \mathrm{cmp}(\mathrm{x},\mathrm{y})\,. \tag{11}$$

Note that no decryption function is provided as only encryptions are required to provide a secure comparison. The Lewi ORE encryption scheme provides security against the simulation-based security model [29] but is not secure against the IND-OCPA model.

C. Real Number Encoding for Homomorphic Encryption

Both encryption schemes in sections III-A and III-B are defined over positive integers, and the Paillier scheme bounds the largest encryptable integer by $N-1$. Due to the prevalence of real numbers in estimation theory, integer encoding of real numbers is an active field of research that accompanies encrypted processing [24], [34], [35], and a requirement for our estimate fusion algorithm. While some encoding schemes for additive homomorphic encryption provide additional operations such as homomorphic division [34], they typically complicate the homomorphic operations, and in [34] leak exponent information of the encrypted real number. We have instead relied on the simpler encoding in [24].

We consider encoding real numbers representable as rational fixed-point numbers of b bits, consisting of a single sign bit, i integer bits, and f fractional bits. Thus, each encodable rational number is defined by its $b = 1+i+f$ bits. As in [24], encoding is performed to allow for multiplication, which requires an operation modulus of $b+2f$ to avoid the requirement for comparisons. Conversion of any real number a to an encoded rational fixed-point number is given by

$$e = \lfloor 2^f a \rfloor \pmod{2^{b+2f}}\,. \tag{12}$$

Multiplication of such encoded numbers requires a factor of $1/2^f$ to be removed. As shown in [24], cases of encoded multiplication can be computed exactly when using Paillier encryption, however, FCI guarantees only one homomorphic multiplication which we handle when decoding for simplicity. Decoding is defined by

$$a = \begin{cases} 2^{-2f}\left(e \pmod{2^{b+2f}}\right) & e < 2^{b+2f-1} \\ 2^{-2f}\left((e \pmod{2^{b+2f}}) - 2^{b+2f}\right) & e \geq 2^{b+2f-1} \end{cases}. \tag{13}$$

and will be correct given only a single encoded multiplication has occurred.

Since the largest encryptable integer is given by $N-1$, the largest encodable real number must account for this. Thus, the integer bits i and fractional bits f must be chosen such that

$$\begin{aligned} N &\geq 2^{b+2f} \\ &\geq 2^{1+i+3f} \end{aligned}. \tag{14}$$

IV. Two-sensor Secure Fast Covariance Intersection

In this section, we will introduce the Secure FCI (SecFCI) fusion algorithm for the two sensor case, before extending it to the n sensor case in section V. The network model we consider is described in section I-A, where sensors are capable of running local estimators, as well as the PHE and ORE encryption schemes from section III. Each sensor i computes its state estimate $\hat{\underline{x}}_i$ and covariance matrix $\mathbf{P}_i$ and sends relevant encrypted information to an untrusted cloud fusion center. The querying party is the key holding party and generates the PHE public key pk, secret key sk, and ORE symmetric key k. pk is made available to all parties in the network, and k is made available to the sensors only, via any standard public-key scheme such as RSA [17]. When encrypting with ORE key k, individual sensors are limited to using only L or R ORE encryption to reduce local information leakage. Thus, consecutive ORE encryptions from any sensor cannot be used to infer local information directly, and can only be compared to encryptions from sensors using the alternate ORE encryption.

From (1), we can see that both CI fusion equations can be computed on PHE encryptions of sensor information vectors and information matrices, given valid unencrypted values for each ω_i. For this reason, we allow the leakage of all weights ω_i. Thus, in the two sensor case, homomorphic fusion is computed by

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_1^{-1})^{\omega_1}\mathcal{E}(\mathbf{P}_2^{-1})^{(1-\omega_1)} \tag{15}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\hat{\underline{x}}) = \mathcal{E}(\mathbf{P}_1^{-1}\hat{\underline{x}}_1)^{\omega_1}\mathcal{E}(\mathbf{P}_2^{-1}\hat{\underline{x}}_2)^{(1-\omega_1)}\,, \tag{16}$$

where we note that $\omega_2 = 1-\omega_1$ due to the CI requirement (2). We also note that in (15) and (16), each resulting value will have exactly one encoding multiplication factor to remove, and can be decoded exactly by using (13).

All that remains for computing CI homomorphically, in the two sensor case, is the calculation of parameter ω_1. For this, we approximate the solution to FCI. Since our encoding scheme in section III-C does not allow division, the exact result of (5) is approximated. This is accomplished by discretizing ω_i by step-size s, such that $s < 1$ and $p = 1/s \in \mathbb{Z}$, and approximating (5) with ORE. An ordered discretization of values $\omega^{(x)}$ is defined by

$$[\omega^{(1)}, \ldots, \omega^{(p)}] = [0, s, \ldots, 1-s, 1]\,, \tag{17}$$

and computed by each sensor i. Each $\omega^{(x)}$ is multiplied by $\mathrm{tr}(\mathbf{P}_i)$ and encrypted with ORE key k. Sensor 1's list is defined by

$$[\mathcal{E}^L_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_1)), \ldots, \mathcal{E}^L_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_1))]\,, \tag{18}$$

and similarly sensor 2's by

$$[\mathcal{E}^R_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_2)), \ldots, \mathcal{E}^R_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_2))]\,. \tag{19}$$

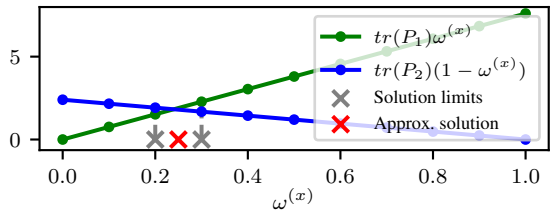


Fig. 1: Approximation of ω_1 with discretization step-size $s = 0.1$. Only comparisons between line points are used.

Note that Sensor 1 uses only L ORE while sensor 2 uses only R ORE, and that both lists are ordered. Lists (18) and (19) are sent alongside PHE encryptions of local information vector and information matrix estimates to the fusion center which uses them to estimate the FCI values of ω_1 and ω_2.

From (5) we know that ω_1 must satisfy

$$\omega_1 \operatorname{tr}(\mathbf{P}_1) = (1-\omega_1)\operatorname{tr}(\mathbf{P}_2)\,. \tag{20}$$

If we reverse (19), we obtain a list equivalent to one with values $\mathcal{E}^R_{ORE}((1-\omega^{(x)})\operatorname{tr}(\mathbf{P}_2))$ for each discretization step x. When the reversed list is decrypted and plotted over (18) the intersection gives the solution to (20) and therefore, (5). However, (18) and reversed (19) consist of L and R ORE encryptions respectively, and the intersection must be approximated by locating consecutive $\omega^{(x)}$ discretizations where the sign of comparisons changes. This can be seen in Fig. 1, and can be performed in $O(\log p)$ ORE comparisons using a binary search. Consecutive $\omega^{(x)}$ and $\omega^{(x+1)}$ for which list comparisons differ can be used to estimate the true intersection, and ω_1, by

$$\omega_1 \approx 0.5(\omega^{(x)} + \omega^{(x+1)})\,. \tag{21}$$

In the case a comparison returns equality, the exact value of $\omega^{(x)}$ can be taken to be ω_1.

The fusion center can then use its values for ω_1 and $\omega_2 = 1-\omega_1$ and the received PHE encryptions of local information vectors and information matrices to compute (15) and (16).

V. Multi-sensor Secure Fast Covariance Intersection

When computing the SecFCI fusion for n sensors, we solve (1) homomorphically by computing

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_1^{-1})^{\omega_1}\cdots\mathcal{E}(\mathbf{P}_n^{-1})^{\omega_n} \tag{22}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\hat{\underline{x}}) = \mathcal{E}(\mathbf{P}_1^{-1}\hat{\underline{x}}_1)^{\omega_1}\cdots\mathcal{E}(\mathbf{P}_n^{-1}\hat{\underline{x}}_n)^{\omega_n}\,. \tag{23}$$

As with the two sensor case, encoded results from (22) and (23) contain exactly one multiplication factor to remove and can be decoded exactly with (13). Again we are just left with the task of computing the plaintext weights $\omega_1,\ldots,\omega_n$.

Our approach to the n sensor case is to solve each $n-1$ conditions in (6) using the two sensor method, and combining partial solutions to compute the final result. When we consider a Euclidean dimension for each ω_i, partial solutions can be considered geometrically as hyperplanes of $n-2$ dimension, over the $n-1$ dimensional solution space given by (2).

This can be visualized in the three sensor case, which requires solving partial solutions

$$\omega_1\operatorname{tr}(\mathbf{P}_1) - \omega_2\operatorname{tr}(\mathbf{P}_2) = 0,\ \ \omega_1+\omega_2 = 1-\omega_3 \tag{24}$$

and

$$\omega_2\operatorname{tr}(\mathbf{P}_2) - \omega_3\operatorname{tr}(\mathbf{P}_3) = 0,\ \ \omega_2+\omega_3 = 1-\omega_1\,. \tag{25}$$

We can use the two sensor method from section IV to solve (24) exactly when $\omega_3 = 0$, and know that when $\omega_3 = 1$, then $\omega_1 = \omega_2 = 0$. These two points are enough to define the two-dimensional partial solution (24) which can be seen plotted over the possible solution space in Fig. 2(a). Fig. 2(b) shows both partial solutions (24) and (25) plotted over the solution space. The final solution from all partial solutions is computed by finding their intersection. This can be seen in Fig. 2(b) as the intersection of the (ω_1,ω_2) and (ω_2,ω_3) partial solution lines.

To simplify computing the partial solution intersection, we define equivalent planes for each of the partial solutions, perpendicular to the solution space, in the form

$$a_1\omega_1 + a_2\omega_2 + a_3\omega_3 + a_4 = 0\,, \tag{26}$$

and solve the resulting linear system for finding the intersection of all planes and the solution space. This is given by

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \\ a_1^{(2)} & a_2^{(2)} & a_3^{(2)} \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix} a_3^{(1)} \\ a_4^{(2)} \\ 1\end{bmatrix}\,, \tag{27}$$

where $a_i^{(j)}$ denotes parameter i of partial solution j, and has been shown visually in Fig. 2(c).

In the n sensor case, we can similarly solve partial solutions by first using the method from section IV to solve equations with two parameters ω_k and ω_{k+1} when letting all $\omega_i = 0,\ i \neq k, k+1$. For each equation we can then compute remaining partial solution points at $\omega_i = 1,\ i \neq k, k+1$ with $\omega_j = 0,\ j \neq i$. Perpendicular hyperplanes can then be similarly defined in the form

$$a_1\omega_1 + \cdots + a_n\omega_n + a_{n+1} = 0\,. \tag{28}$$

Due to their inherent orthogonality, and that all meaningful covariance traces are strictly positive, the $n-1$ partial solution hyperplanes are guaranteed to intersect at exactly one point. The hyperplane intersection results in the linear system

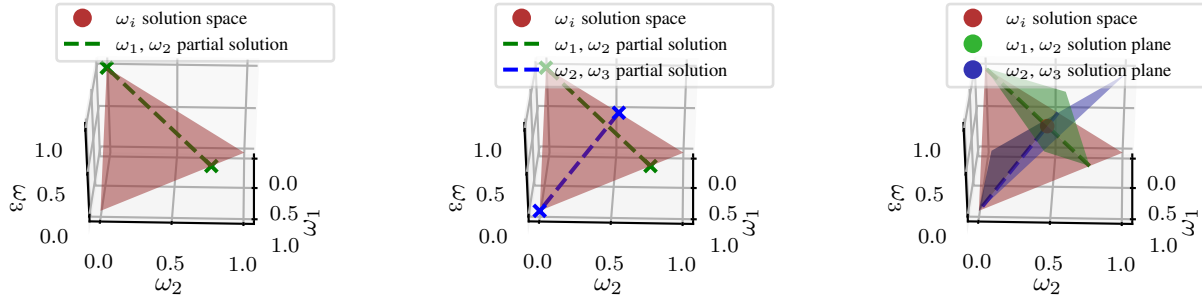
$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & \cdots & a_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{(n-1)} & a_2^{(n-1)} & \cdots & a_n^{(n-1)} \\ 1 & 1 & \cdots & 1 \end{bmatrix}\begin{bmatrix}\omega_1\\ \vdots\\ \omega_{n-1}\\ \omega_n\end{bmatrix} = \begin{bmatrix} a_{n+1}^{(1)} \\ \vdots \\ a_{n+1}^{(n-1)} \\ 1\end{bmatrix}\,, \tag{29}$$

and gives the solution to the SecFCI ω_i weights.

As all $O(n\log p)$ ORE comparisons are done between sequential sensors i and $i+1$, L and R ORE encryptions can be used to the same effect as for the two sensor case. The ORE ordered list sent from each sensor i is given by

$$\begin{aligned} &[\mathcal{E}^L_{ORE}(\omega^{(1)}\operatorname{tr}(\mathbf{P}_i)),\ldots,\mathcal{E}^L_{ORE}(\omega^{(p)}\operatorname{tr}(\mathbf{P}_i))],\ i \text{ odd} \\ &[\mathcal{E}^R_{ORE}(\omega^{(1)}\operatorname{tr}(\mathbf{P}_i)),\ldots,\mathcal{E}^R_{ORE}(\omega^{(p)}\operatorname{tr}(\mathbf{P}_i))],\ i \text{ even.} \end{aligned} \tag{30}$$

When combining (30) with PHE encryptions of local information vectors and information matrices, SecFCI can be computed entirely homomorphically by (22) and (23).



(a) Partial solution to (24). (b) Partial solutions to (24) and (25). (c) Partial solutions as planes.

Fig. 2: Partial solutions over ω_1, ω_2, and ω_3 solution space.

TABLE I: Computation complexity of encryption operations.

Operation	Complexity
Paillier enc.	$O(\log^3 N)$
Paillier dec.	$O(\log^3 N)$
Paillier add.	$O(\log^2 N)$
Paillier scalar mult.	$O(\log^3 N)$
Lewi L enc.	$O(\log^2 N)$
Lewi R enc.	$O(\log^2 N)$
Lewi comp.	$O(\log^2 N)$

Briefly considering the security of our scheme, we note that any leaked information from ORE lists (30), as described in [29], can be considered a subset of knowing the estimated fusion weights $\omega_1, \ldots, \omega_n$, which specify relative sizes of sensor covariance traces, and we already consider public. Thus only IND-CPA and IND-OCPA (after accounting for leakage through public weights) encryptions are made available to the fusion center.

A. Computational Complexity

Given the state estimates and estimate errors at each sensor, we wish to show the computational complexity of the SecFCI algorithm for the n sensor case. We will assume that both Lewi ORE and Paillier PHE schemes use the same length security parameter (and equivalently key size), such that $\lambda_{Lewi} = \lambda_{Paillier} = \log N$, where λ_s represents encryption scheme s's security parameter, and N the Paillier modulus and encryptable integer limit. We also note the distinction between floating-point or small integer operations, which are typically treated as having $O(1)$ runtime, and large integer operations whose complexities are dependent on bit length. While architectures exist for speeding up encryption operations [16], we consider software implementations and treat large integer operations in terms of bit operations explicitly.

From [21], [30], and the assumptions made above, we have summarized the operation complixites of the two schemes in Table I. In contrast to some current FHE schemes, these operations are of a much lower complexity than [36], which has complexity $O(\lambda^{10})$ for integer operations, and [19], which computes single bit operations in $O(\lambda^{3.5})$ adding significant overhead for integer arithmetic.

Finally, applying the operations from Table I to the SecFCI algorithm, we summarize the total complexity of SecFCI at the sensors and the fusion center in Table II, with the unencrypted complexities of FCI shown for reference.

TABLE II: Computation complexity at sensors and fusion center.

	FCI	SecFCI
Sensors	$O(1)$	$O\left(p \log^2 N + \log^3 N\right)$
Fusion	$O(n^3)$	$O\left(n \log p \log^2 N + n \log^3 N + n^3\right)$

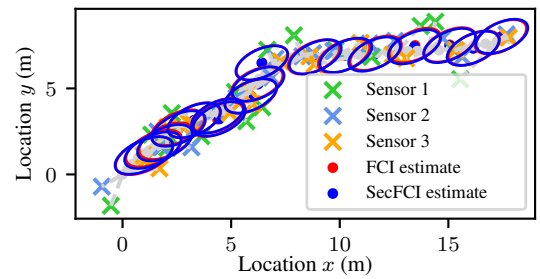


Fig. 3: Tracking simulation comparing SecFCI and FCI.

VI. Simulation Results

We have implemented a simulation to demonstrate the accuracy of SecFCI approximating FCI. Three sensors independently measure a constant-speed linear process and simultaneously run a Kalman filter on their measurements. Estimates are sent both encrypted and unencrypted to a fusion center which computes the SecFCI and FCI fusions on the received data respectively. Encrypted estimates are comprised of PHE encryptions of the information vector and information matrix, $\mathcal{E}(\mathbf{P}_i^{-1}\underline{\hat{x}}_i)$ and $\mathcal{E}(\mathbf{P}_i^{-1})$, in addition to the ORE list given by (30) with discretization step $s = 0.1$. Unencrypted estimates consist of the state estimate $\underline{\hat{x}}_i$ and covariance $\mathbf{P}_i$. The trajectory and fused estimates are shown in Fig. 3.

To derive an upper bound on the accuracy difference between SecFCI and FCI, we note the two factors which introduce inconsistency between the two methods: the encoding method from section V-A, and the difference in fusion weights. Due to the possibility of choosing sufficiently large integer and fractional bit lengths i and f, we will only consider the error caused by the difference in weights. We will treat this error as the distance between respective weight vectors

$$\begin{aligned} \underline{\omega}_{SecFCI} &= \left(\omega_{1,SecFCI}, \ldots, \omega_{n,SecFCI}\right), \\ \underline{\omega}_{FCI} &= \left(\omega_{1,FCI}, \ldots, \omega_{n,FCI}\right) \end{aligned} \tag{31}$$

where $\omega_{i,s}$ denotes weight ω_i from algorithm s. From section IV we see that the largest difference $|\omega_{i,FCI} - \omega_{i,SecFCI}|$ is strictly bounded by $s/2$. As shown in section V, when more sensors are involved, a tighter bound on this difference is dependent on the value of $\underline{\omega}_{i,FCI}$, but will remain strictly

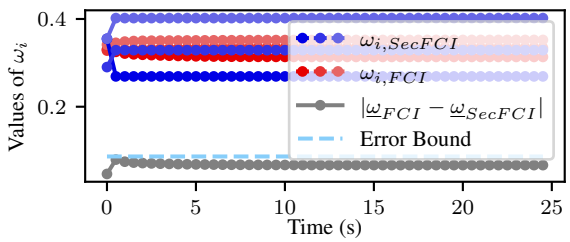


Fig. 4: $\underline{\omega}_{SecFCI}$ and $\underline{\omega}_{FCI}$ components.

bounded by $s/2$. Therefore, we can give a strict upper bound on the distance between weight vectors as

$$|\underline{\omega}_{FCI} - \underline{\omega}_{SecFCI}| < 0.5\sqrt{ns^2}\,. \tag{32}$$

Finally, components of $\omega_{i,SecFCI}$, $\omega_{i,FCI}$ and the errors $|\underline{\omega}_{FCI} - \underline{\omega}_{SecFCI}|$, have been plotted over time in Fig. 4, and show the computed error bound when $n = 3$ and $s = 0.1$.

VII. Conclusion

FCI is a commonly used, and efficiently computable, approximation to the CI optimization problem that requires the sharing of local sensor estimates to compute their fusion. We propose a secure approximation to FCI, SecFCI, to compute the fused estimate homomorphically. The novel encrypted fusion approach may find uses in various security-critical applications or over untrusted networks subject to eavesdroppers and malicious participants. Possible future work includes run-time comparisons with FHE implementations, giving a computational bound for its practicality, and quantification of fusion weight leakages via formal security proofs.

References

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

[2] D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Algorithms for a Multi-sensor System," in *15th IEEE Conference on Decision and Control (CDC 1976)*, 1976, pp. 570–574.

[3] H. 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, 1988.

[4] C. Y. Chong, "Forty Years of Distributed Estimation: A Review of Noteworthy Developments," in *IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2017)*, 2017, pp. 1–10.

[5] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, *Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion*. CRC Press, 2017, pp. 169–192.

[6] Y. Bar-Shalom, "On The Track-to-track Correlation Problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.

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

[8] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck, "Optimal Sample-based Fusion for Distributed State Estimation," in *19th International Conference on Information Fusion (Fusion 2016)*, 2016, pp. 1600–1607.

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

[10] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, 2017.

[11] W. Niehsen, "Information Fusion Based On Fast Covariance Intersection Filtering," in *5th International Conference on Information Fusion (Fusion 2002)*, vol. 2, 2002, pp. 901–904.

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

[13] F. Pfaff, B. Noack, U. D. Hanebeck, F. Govaers, and W. Koch, "Information Form Distributed Kalman Filtering (IDKF) with Explicit Inputs," in *20th International Conference on Information Fusion (Fusion 2017)*, 2017, pp. 1–8.

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

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

[16] S. Gueron, "Intel Advanced Encryption Standard (AES) New Instructions Set," *Intel Corporation*, 2010.

[17] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM (CACM)*, vol. 21, no. 2, pp. 120–126, 1978.

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

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

[20] 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.

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

[22] 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, 2012.

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

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

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

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

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

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

[29] N. Chenette, K. Lewi, S. A. Weis, and D. J. Wu, "Practical Order-Revealing Encryption with Limited Leakage," in *IACR Fast Software Encryption (FSE)*. Springer, 2016, pp. 474–493.

[30] K. Lewi and D. J. Wu, "Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 1167–1178.

[31] D. Bogatov, G. Kollios, and L. Reyzin, "A Comparative Evaluation of Order-Preserving and Order-Revealing Schemes and Protocols," *IACR Cryptology*, vol. 2018, p. 953, 2018.

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

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

[34] M. T. I. Ziad, A. Alanwar, M. Alzantot, and M. Srivastava, "CryptoImg: Privacy Preserving Processing Over Encrypted Images," in *Conference on Communications and Network Security (CNS)*, 2016, pp. 570–575.

[35] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Advances in Cryptology (ASIACRYPT)*, T. Takagi and T. Peyrin, Eds., 2017, vol. 10624, pp. 409–437.

[36] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer, 2010, pp. 24–43.