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 means they do not remain locally private. Current secure fusion algorithms rely on encryption schemes that do not provide sufficient flexibility and as a result require, often undesired, excess communication between estimate producers. 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 approximate solutions to the fast covariance intersection coefficients 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.**

I. Introduction

Sensor data processing and state estimation have been increasingly prevalent in networked systems [1], [2]. Bayesian state estimation has become a particularly common application since the beginning of Kalman estimation theory [3] and has led to a large interest in the field of state estimation fusion [4]–[8]. Challenges of estimation fusion are closely tied to the handling and merging of estimation error statistics [9]. Cross-correlations between estimation errors characterize dependencies between local estimates and must be considered when performing consistent or optimal fusion [10], [11]. Methods that keep track of the cross-correlation of errors may require repeated reconstruction [12] and typically add local computational complexity and limit usability. An alternative strategy sees the approximation of estimate error cross-correlation based on conservative suboptimal strategies, and has been implemented in a variety of methods [13]–[18]. Covariance Intersection (CI) [14] provides one such popular conservative strategy, from which a less computationally expensive method, the Fast Covariance Intersection (FCI) [17] has been derived. CI is particularly well paired with the information form of the Kalman filter [19]. 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. It has been used to subtract common information between estimates when cross-correlations are known [8] and within fully distributed filter implementations [20].

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

As advancements in distributed algorithms and cloud computing develop, the requirements for privacy and security in such systems has become more apparent [21], [22]. In particular for sensor networks, the desire for sensitive hardware information or estimation methodology to remain private may require the privacy of local measurements or state estimates as well, which can be difficult to achieve in a network containing eavesdroppers or malicious participants. Encryption has until recently been primarily used to secure information transfer between communicating parties. Common symmetric-key encryption schemes such as AES [23] are used to encrypt sent information to its destination, and public-key encryption schemes such as RSA [24] to distribute symmetric keys. However, recent developments in public-key Homomorphic Encryption (HE) schemes [25]–[29], which allow algebraic operations to be performed on encryptions, are leading to many secure applications for signal processing in distributed and cloud computing environments [30]–[35]. Fully Homomorphic Encryption (FHE) schemes [25]–[27] provide all algebraic operations over encryptions, and are often theoretically suitable to processing tasks in a distributed environment securely. However, current implementations are still computationally infeasible for large-scale or real-time processing [36], [37]. Partially Homomorphic Encryption (PHE) schemes [28], [29], providing only a subset of these operations, have instead been a focus for such tasks due to their reduced computational requirements. [31] use PHE to compute a private distributed Information Filter, [32], [33] to compute private distributed control aggregation, [34] for private matrix multiplications, and [35] for private set intersection. These works are, however, due to the limited operations provided by PHE, relatively restricted in complexity and application. Recent developments in new encryption schemes, such as Order Revealing Encryption (ORE) [38]–[40], are providing new light on the possible complexity of securely computable signal processing algorithms. In this paper, we develop a method for secure FCI fusion (SecFCI), such that local and state measurements are kept private to sensors and only fusion weights are made known to the otherwise untrusted fusion center. This is achieved using a combination of ORE and PHE schemes only, and has to the best of our knowledge, not been achieved without the reliance on computationally expensive FHE schemes before.

A. Problem Formulation

B. Notation

Throughout this paper we will use the following notation. Lowercase characters represent scalars, lowercase underlined characters, $\underline{x}$, represent vectors. Uppercase bold characters, $\mathbf{M}$, are reserved for matrices, where $\mathbf{M}^\top$ denotes the matrix transpose, $\mathbf{M}^{-1}$ 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 $\mathcal{D}_{pk}(\cdot)$ and $\mathcal{D}_{ORE,k}(\cdot)$, 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 $\{\cdot\}$ while ordered lists with $[\cdot]$.

II. Covariance Intersection and Approximations

Covariance Intersection (CI), introduced in [14], provides a consistent state estimate fusion algorithm when cross-correlations are not known. The resulting fused estimate $\hat{\underline{x}}$ and estimate 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 [17] and reduces the fusion to a simple weighted sum. Values for 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 fused estimate covariance trace, requires the solution to

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

However, minimizing this non-linear cost function can be very costly computationally and has led to the development of the non-iterative approximation technique in [17].

A. Fast Covariance intersection

The Fast Covariance Intersection (FCI) algorithm from [17] is a common method used 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

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

with analytical solutions given by

$$\omega_1 = \frac{\mathrm{tr}(\mathbf{P}_2)}{\mathrm{tr}(\mathbf{P}_1)+\mathrm{tr}(\mathbf{P}_2)},\ \omega_2 = \frac{\mathrm{tr}(\mathbf{P}_1)}{\mathrm{tr}(\mathbf{P}_1)+\mathrm{tr}(\mathbf{P}_2)}\,. \tag{6}$$

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

$$\omega_i\,\mathrm{tr}(\mathbf{P}_i) - \omega_{i+1}\,\mathrm{tr}(\mathbf{P}_{i+1}) = 0,\ i = 1,2,\ldots,n-1\,, \tag{7}$$

and requires the solution to the linear problem

$$\begin{bmatrix} \mathcal{P}_1 & -\mathcal{P}_2 & 0 & \cdots & 0 \\ 0 & \mathcal{P}_2 & -\mathcal{P}_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \mathcal{P}_{n-1} & -\mathcal{P}_n \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{8}$$

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

Our proposed filter aims to solve FCI fusion, namely (1) and (8), homomorphically, such that using only encrypted values from each sensor i, we can produce valid encryptions of fused estimates without the need for decryption.

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. Public-key additive Partially Homomorphic Encryption (PHE) schemes [29], [41] provide a single homomorphic addition operation $\oplus$ on cyphertexts such that

$$\mathcal{D}(\mathcal{E}(a)\oplus\mathcal{E}(b)) = a + b \tag{9}$$

holds. Symmetric-key Order Revealing Encryption (ORE) schemes [38], [39] provide a secure comparison function, allowing the comparison of encrypted values via

$$f(\mathcal{E}_{ORE}(a),\mathcal{E}_{ORE}(b)) = \mathrm{cmp(a,b)}\,. \tag{10}$$

The formal security of encryption schemes consists of a security goal and a formal threat model [42]. Indistinguishability of ciphertexts under the adaptive chosen ciphertext attack model (IND-CCA2) is commonly considered the strongest security guarantee [43]. However, no homomorphic encryption scheme provides security against chosen ciphertext attack models 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) [44]. Similarly, ORE schemes aim to protect against simulation-based security defined in [38] or the harder to achieve ordered chosen-plaintext attack model (IND-OCPA).

A. Additive Partially Homomorphic Encryption

The additive PHE scheme we use is the Paillier encryption scheme [29] due to its implementation simplicity, and computational speed. The Paillier scheme provides two homomorphic operations on encrypted data,

$$\mathcal{D}_{pk}(\mathcal{E}_{pk}(a)\mathcal{E}_{pk}(b) \pmod{N^2}) = a + b \pmod{N} \tag{11}$$

and

$$\mathcal{D}_{pk}(\mathcal{E}_{pk}(a)^c \pmod{N^2}) = c \cdot a \pmod{N},\ c \in \mathbb{Z}_N\,, \tag{12}$$

where the modulus N is computed as the product of two large primes chosen randomly during 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. Left-Right Order Revealing Encryption

The ORE scheme we have used is Lewi's symmetric-key Left-Right encryption scheme [39] that has the added property of only allowing certain comparisons between cyphertexts. This property can be used to decide which values may not be compared as will be shown in section IV and is described as follows. Two encryption functions allow integers to be encrypted as either a "Left" (L) or "Right" (R) encryption by

$$\begin{aligned} \mathrm{enc}^{\mathrm{L}}_{\mathrm{ORE}}(\mathrm{k}, \mathrm{x}) &= \mathcal{E}^{\mathrm{L}}_{\mathrm{ORE,k}}(\mathrm{x}) \\ \mathrm{enc}^{\mathrm{R}}_{\mathrm{ORE}}(\mathrm{k}, \mathrm{y}) &= \mathcal{E}^{\mathrm{R}}_{\mathrm{ORE,k}}(\mathrm{y})\,, \end{aligned} \tag{13}$$

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

$$\mathrm{cmp}_{\mathrm{ORE}}(\mathcal{E}^{\mathrm{L}}_{\mathrm{ORE}}(\mathrm{x}),\ \mathcal{E}^{\mathrm{R}}_{\mathrm{ORE}}(\mathrm{y})) = \mathrm{cmp}(\mathrm{x}, \mathrm{y})\,. \tag{14}$$

Note that no decryption function is provided, as only encryptions are required to provide a means of secure comparison. The Lewi ORE encryption scheme provides security against the simulation-based security model [38] 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 acommpanies encrypted processing [32], [45], [46], and a requirement for our estimate fusion algorithm.

While some encoding schemes for additive homomorphic encryption provide additional operations such as homomorphic right bit shifting [45], they typically complicate the homomorphic operations, and in the case of [45] leak the exponent information of encrypted real numbers. We have instead relied on a simplified version of the encoding scheme defined in [32].

We consider encoding real numbers representable as rational fixed-point numbers consising of a single sign bit, i integer bits and f fractional bits. Each encodable rational number defined by its $b = 1 + i + f$ bits, is encoded to the positive integer range $[0, 2^b)$. This is computed as the conversion to the signed Q number format [47] and is equivalent to the encoding function from [32]. The conversion of any real number a to an encoded fixed-point rational is given by

$$e = \lfloor 2^f a \rfloor \pmod{2^b}\,. \tag{15}$$

While encoded Q numbers are consistent under addition, each multiplication requires a factor of $1/2^f$ to be removed. As shown in [32], particular cases of encoded multiplication can be computed exactly when using the Paillier encryption scheme, however, we simplify multiplication by taking advantage of the FCI algorithm, which guarantees only a single homomorphic multiplication as will be shown in section IV, and handle this when decoding.

In addition to not removing the multiplication factor homomorphically, we relax the requirement from [32] for multiplied encoded numbers not to overflow. We allow any amount of multiplicative overflow (from the single multiplication operation) by decoding positive integers from the range $[0, 2^{2b})$ to a real number representable as a rational fixed-point number consisting of a single sign bit, $2i + 1$ integer bits, and $2f$ fractional bits. Decoded real numbers are representable by $2b$ bits rather than the encodable ones of b bits. Decoding is defined as the conversion from the signed Q number format [47] and defined by

$$a = \begin{cases} 2^{-2f}\left(e \pmod{2^{2b}}\right) & e < 2^{2b-1} \\ 2^{-2f}\left(2^{2b} - (e \pmod{2^{2b}})\right) & e \geq 2^{2b-1} \end{cases}. \tag{16}$$

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

$$(2^{(1+i+f)} - 1)^2 \leq N - 1 \tag{17}$$

holds.

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 will consider is one where all sensors are capable of running local estimators, as well as the PHE and ORE encryption schemes described in section III. Each sensor i computes its state estimate $\underline{\hat{x}}_i$ and covariance matrix $\mathbf{P}_i$ and sends the relevant encrypted information to a single fusion center that computes the fused state estimate and covariance matrix homomorphically. A third, querying party, can request and use the current encrypted fused information from the fusion center at any time. The querying party is the key holding party in this network 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 [24]. 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 values for each ω_i. In the two sensor case, homomorphic fusion is computed by

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_1^{-1})^{\omega_1}\mathcal{E}(\mathbf{P}_2^{-1})^{(1-\omega_1)} \tag{18}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\underline{\hat{x}}) = \mathcal{E}(\mathbf{P}_1^{-1}\underline{\hat{x}}_1)^{\omega_1}\mathcal{E}(\mathbf{P}_2^{-1}\underline{\hat{x}}_2)^{(1-\omega_1)}\,, \tag{19}$$

where we note that $\omega_2 = 1-\omega_1$ due to the CI requirement (2). We also note that in (18) and (19), each resulting value will have exactly one Q encoding multiplication factor to remove, and can be decoded exactly by using (16).

In the two sensor case, all that remains for computing CI homomorphically is the calculation of parameter ω_1. For this, we approximate the solution to the FCI fusion algorithm. Since analytical solutions (6) require division, they cannot be computed exactly with the given PHE encryptions of sensor information vectors and information matrices. Instead, we discretize ω_i by step-size s, such that $s < 1$ and $p = 1/s \in \mathbb{Z}$, and approximate (5) with ORE. An ordered discretization of values $\omega^{(x)}$ is defined by

$$[\omega^{(1)}, \ldots, \omega^{(p)}] = [0, s, \ldots, 1-s, 1]\,, \tag{20}$$

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

$$[\mathcal{E}^L_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_1)), \ldots, \mathcal{E}^L_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_1))]\,, \tag{21}$$

and similarly sensor 2's by

$$[\mathcal{E}^R_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_2)), \ldots, \mathcal{E}^R_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_2))]\,. \tag{22}$$

Note that Sensor 1 uses only L ORE while sensor 2 uses only R ORE and that both lists are ordered. Lists (21) and (22) 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

$$\omega_1\,\mathrm{tr}(\mathbf{P}_1) = (1-\omega_1)\,\mathrm{tr}(\mathbf{P}_2)\,. \tag{23}$$

If we reverse (22), we obtain the equivalent list

$$[\mathcal{E}^R_{ORE}((1-\omega^{(1)})\,\mathrm{tr}(\mathbf{P}_2)), \ldots, \mathcal{E}^R_{ORE}((1-\omega^{(p)})\,\mathrm{tr}(\mathbf{P}_2))] \tag{24}$$

which when decrypted and plotted over (21) shows that the intersecting point gives the solution to (23) and therefore, (5). However, (22) and (24) consist of L and R ORE encryptions respectively, and the intersection must be approximated by locating the consecutive $\omega^{(x)}$ discretisations 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

$$\omega_1 \approx \frac{1}{2}(\omega^{(x)} + \omega^{(x+1)})\,. \tag{25}$$

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

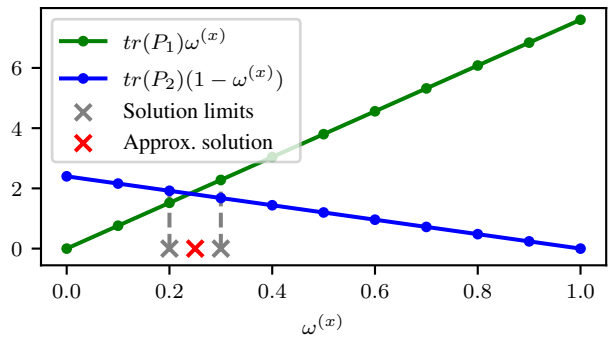


Fig. 1: Approximation of ω_1 with discretisation step-size $s = 0.1$. Only comparisons between line points are used.

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 (18) and (19).

V. Multi-sensor Secure Fast Covariance Intersection

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

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_1^{-1})^{\omega_1}\cdots\mathcal{E}(\mathbf{P}_n^{-1})^{\omega_n} \tag{26}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\underline{\hat{x}}) = \mathcal{E}(\mathbf{P}_1^{-1}\underline{\hat{x}}_1)^{\omega_1}\cdots\mathcal{E}(\mathbf{P}_n^{-1}\underline{\hat{x}}_n)^{\omega_n}\,. \tag{27}$$

As with the two sensor case, encoded results from (26) and (27) contain exactly one multiplication factor to remove and can be decoded exactly with (16). Again we are just left with computing the weights $\omega_1, \ldots, \omega_n$.

Our approach to the n sensor case is to solve each $n-1$ conditions in (7) 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).

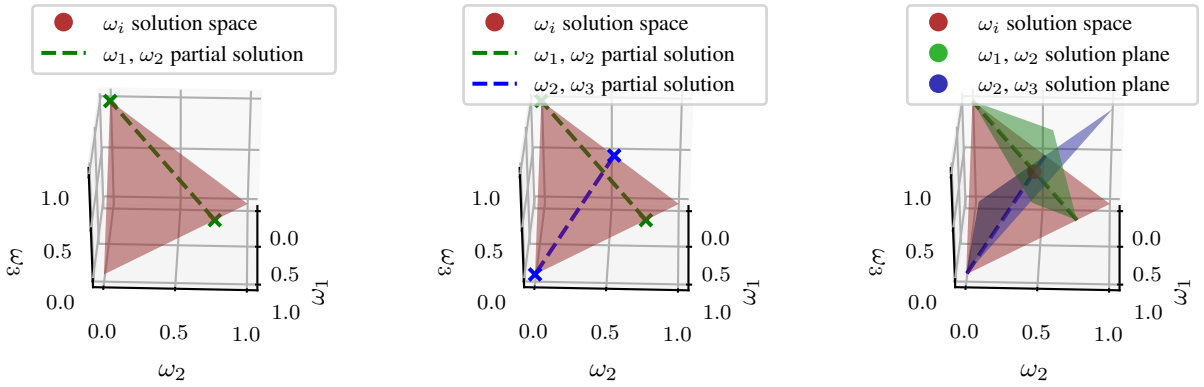
This can be visualized in the three sensor case, which requires solving partial solutions

$$\omega_1\,\mathrm{tr}(\mathbf{P}_1) - \omega_2\,\mathrm{tr}(\mathbf{P}_2) = 0,\ \omega_1 + \omega_2 = 1 - \omega_3 \tag{28}$$

and

$$\omega_2\,\mathrm{tr}(\mathbf{P}_2) - \omega_3\,\mathrm{tr}(\mathbf{P}_3) = 0,\ \omega_2 + \omega_3 = 1 - \omega_1\,. \tag{29}$$

We can use the two sensor method from section IV to solve (28) 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 (28) which can be seen plotted over the possible solution space in Fig. 2(a). Fig. 2(b) shows both partial solutions (28) and (29) 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.



(a) Partial solution to (28).

(b) Partial solutions to (28) and (29).

(c) Partial solutions as planes perpendicular to the solution space.

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

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

$$a_1\omega_1 + a_2\omega_2 + a_3\omega_3 + a_4 = 0\,, \tag{30}$$

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

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \\ a_1^{(2)} & a_2^{(2)} & a_3^{(2)} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} a_3^{(1)} \\ a_4^{(2)} \\ 1 \end{bmatrix}\,, \tag{31}$$

where $a_i^{(j)}$ denotes parameter i of partial solution j.

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

$$a_1\omega_1 + \cdots + a_n\omega_n + a_{n+1} = 0\,. \tag{32}$$

Due to their inherent orthogonality, $n-1$ partial solution hyperplanes are guaranteed to intersect at exactly one point when at most 1 sensor has $\mathrm{tr}(\mathbf{P}_i) = 0$. The hyperplane intersection results in the linear system

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & \cdots & a_n^{(1)} \\ a_1^{(2)} & a_2^{(2)} & \cdots & a_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{(n-1)} & a_2^{(n-1)} & \cdots & a_n^{(n-1)} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} a_{n+1}^{(1)} \\ a_{n+1}^{(2)} \\ \vdots \\ a_{n+1}^{(n-1)} \\ 1 \end{bmatrix}\,, \tag{33}$$

and gives the solution to the SecFCI ω_i values.

As all $O(n \log p)$ ORE comparisons are done between sequential sensors i and $i+1$, seen in (7), 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

$$\begin{aligned} &[\mathcal{E}^L_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_i)), \ldots, \mathcal{E}^L_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_i))],\ i \text{ odd} \\ &[\mathcal{E}^R_{ORE}(\omega^{(1)}\,\mathrm{tr}(\mathbf{P}_i)), \ldots, \mathcal{E}^R_{ORE}(\omega^{(p)}\,\mathrm{tr}(\mathbf{P}_i))],\ i \text{ even}\,. \end{aligned} \tag{34}$$

When combining (34) with PHE encryptions of local information vectors and information matrices, SecFCI can be computed entirely homomorphically by (26) and (27).

TABLE I: Computation complexity of encryption operations.

Operation	Complexity
Paillier encryption	$O(\log N \log^2 N^2)$
Paillier decryption	$O(\log N \log^2 N^2)$
Paillier addition	$O(\log^2 N^2)$
Paillier scalar multiplication	$O(\log N \log^2 N^2)$
Lewi L encryption	$O(\log^2 N^2)$
Lewi R encryption	$O(\log^2 N^2)$
Lewi comparison	$O(\log^2 N^2)$

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 make the assumption that both the Lewi ORE and Paillier PHE schemes use the same length security parameter (and equivalently key size), such that

$$\lambda_{Lewi} = \lambda_{Paillier} = \log N\,, \tag{35}$$

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 dependant on bit length. While hardware architectures exist for speeding up encryption bit operations [48], we will consider software implementations and treat large integer operations in terms of bit operations explicitly.

From [29], [39], 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 [26], which

TABLE II: Computation complexity at sensors and fusion center.

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

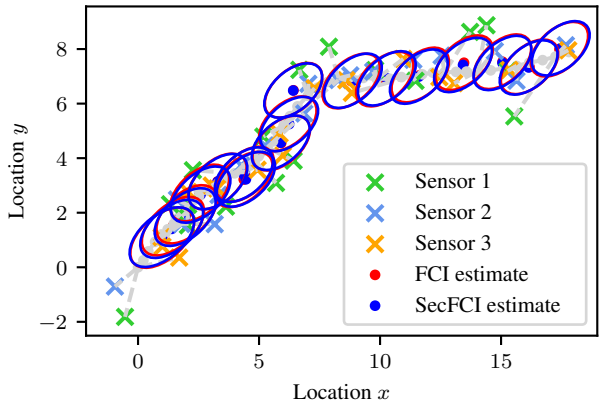


Fig. 3: Tracking simulation comparing SecFCI and FCI.

has complexity $O(\lambda^{10})$ for integer operations, or [27], which computes bit operations in $O(\lambda^{3.5})$.

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

VI. Simulation Results

We have implemented a simulation to demonstrate the accuracy of SecFCI and compare it to FCI. Three sensors independently measure a two-dimensional constant-speed linear process and simultaneously run a linear 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 a PHE encryption 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 (34) (discretization step-size $s = 0.1$). Unencrypted estimates consist of the state estimate $\underline{\hat{x}}_i$ and covariance matrix $\mathbf{P}_i$. The trajectory and fused estimates are shown in Fig. 3.

To derive a strict upper bound on the accuracy difference between SecFCI and FCI, we note that the two factors which introduce inconsistency between the two methods are the encoding of real values from Section V-A, and the difference in weights computed by (33) and (8) respectively. Due to the small error encoding by quantizing introduces to floating-point arithmetic [31], and 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, denoted

$$\underline{\omega}_{SecFCI} = \begin{bmatrix}\omega_{1,SecFCI} & \cdots & \omega_{n,SecFCI}\end{bmatrix}^\top \tag{36}$$

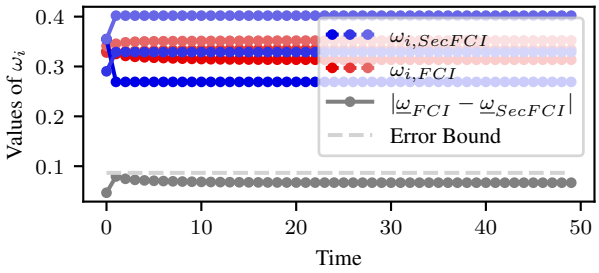


Fig. 4: $\underline{\omega}_{SecFCI}$ and $\underline{\omega}_{FCI}$ values and difference.

and

$$\underline{\omega}_{FCI} = \begin{bmatrix}\omega_{1,FCI} & \cdots & \omega_{n,FCI}\end{bmatrix}^\top \tag{37}$$

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 $s/2$. Section V shows that when more sensors are involved, a tighter bound on this difference is dependant on the value of $\underline{\omega}_{i,FCI}$, but will remain strictly bounded by $s/2$. Therefore, we can give a generous upper bound on the distance between weight vectors as

$$|\underline{\omega}_{FCI}-\underline{\omega}_{SecFCI}| < \sqrt{n\left(\frac{s}{2}\right)^2}. \tag{38}$$

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

VII. Conclusion

FCI is a commonly used, 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 signal processing approach may find uses in various security-critical applications or over untrusted networks. Possible future work includes a run-time comparison between SecFCI and potential FHE implementations, giving a computational bound for its practicality. Also, we hope to further quantify ORE leakage concerning SecFCI fusion and produce formal security proofs and assumptions for the novel algorithm.

References

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

[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)*, 2017, pp. 1–10.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, 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)*, 1976, pp. 570–574.

[5] C. Y. Chong, "Hierarchical Estimation," in *MIT/ONR Workshop on C3 Systems*, 1979, pp. 205–220.

[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)*, 1986, pp. 1863–1868.

[7] 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.

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

[9] H. Fourati, *Multisensor Data Fusion : From Algorithms and Architectural Design to Applications*. CRC Press, 2017.

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

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

[12] 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)*, 2016, pp. 1600–1607.

[13] N. Carlson, "Federated Filter for Fault-tolerant Integrated Navigation Systems," in *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS 1988)*, 1988, pp. 110–119.

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

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

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

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

[18] D. Fränken and A. Hüpper, "Improved Fast Covariance Intersection For Distributed Data Fusion," in *Proceedings of the 7th International Conference on Information Fusion (Fusion 2005)*, vol. 1, 2005, p. 7.

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

[20] 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)*, 2017, pp. 1–8.

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

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

[23] J. Daemon and V. Rijmen, "Announcing the Advanced Encryption Standard (AES)," *Federal Information Processing Standards Publication*, vol. 197, 2001.

[24] 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.

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

[26] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in *Advances in Cryptology (EUROCRYPT)*, 2010, pp. 24–43.

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

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

[28] 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.

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

[30] 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.

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

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

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

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

[36] Y. Du, L. Gustafson, D. Huang, and K. Peterson, "Implementing ML Algorithms with HE," in *MIT Course 6. 857: Computer and Network Security*, 2017.

[37] 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.

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

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

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

[41] S. Goldwasser and S. Micali, "Probabilistic Encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

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

[43] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-key Encryption Schemes," in *Advances in Cryptology (CRYPTO 1998)*. Springer, 1998, pp. 26–45.

[44] 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.

[45] 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.

[46] 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.

[47] E. L. Oberstar, "Fixed-Point Representation & Fractional Math," *Oberstar Consulting*, vol. 9, 2007.

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