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

Response to Reviewers' Comments - Submission L-CSS 20-0282

Marko Ristic Benjamin Noack Uwe D. Hanebeck

May 14, 2020

Dear Dr. Giovanni Cherubini,
Dear Reviewers,

We would like to thank you all for your thorough and encouraging reviews. In this letter, we explain how the reviewer comments, questions, and suggestions have been addressed. Throughout this response, reviewers' comments are typed in blue.

Sincerely,
Marko Ristic, Benjamin Noack, and Uwe D. Hanebeck

Response to the Editor's Report

E.1 Three reports have been collected for this submission. The reviewers agree that the topic is interesting. However, they also raise a number of concern that should be carefully addressed in a revised version.

I also have some specific comments:

1. The problem motivation given in the Introduction is a bit vague. Specifically, while it is known that encryption plays an important role in certain network (control) systems, the need for securing confidentiality of data in sensor networks is less evident. The authors could discuss some concrete applications related to sensor networks where data confidentiality is important.

We are sorry that the role of encryption in sensor networks was not made more clearer in our introduction, and agree with the reviewer that a stronger motivation for our work was lacking. We have changed the introduction to include a discussion on when encryption may be desired in sensor networks specifically, and have added Section I.A describing the scenario motivating our work.

E.2 2. On a similar vein, the paper lacks a clear problem statement defining what is the sensitive information (and why). This has been pointed out also by Reviewer 1.

We agree that a problem statement was missing and in the added new Section I.A, we give the scenario motivating our work, as well as a stronger definition of our problem. In this section, we specify which information is treated as sensitive and which collusions we consider.

E.3 3. It is known that encryption introduces delays, thus affecting performance. The authors could add a discussion on the computation complexity of their method and its trade-off with accuracy. All the reviewers have similar comments (Reviewer 3 is especially critical on this point).

You're right; the manuscript was lacking any run times or complexity analysis of the fusion algorithm, as well as any comparison between methods. We have added the complexity run times of the new algorithm and contrasted them with the unencrypted FCI algorithm, as well as noting the comparative differences to homomorphic operation complexities of some current FHE schemes. These changes are included in the new Section V.A. Additionally, in Section VI we have added a discussion and derivation of the accuracy of the method in terms of algorithms parameters.

E.4 4. The authors could summarize the main result in a theorem. This would help to highlight the method and its properties. As it stands, the analysis of Section V is rather fuzzy. This has been pointed out also by Reviewer 2.

We regret that Section V and our results were not made clearer, and have added Section V.A and updated Section VI to include an explicit complexity analysis of the novel method, and give an upper bound on the accuracy reduction of SecFCI from the FCI algorithm it approximates. We have not stated any explicit theorems as their proofs are beyond the scope of this short manuscript, but have aimed to summarize our results with the added security comments, complexity analysis, and accuracy bound in Sections V and VI.

Response to the Comments of Reviewer 1 (23899)

R1.1 - Equations (9), (11) and (12) are incorrect. Paillier is a probabilistic cryptosystem which means when you encrypt the same message two times it will yield different ciphertexts due to the usage of randomness. Thus, you have to equate the decryption of both sides of all equations.

Thank you for pointing this out. We have corrected the manuscript accordingly.

R1.2 - The paper lacks a clear problem statement and threat model. What are the sensitive information estimates, measurement or both?. A portion of the threat model is mentioned very late with the proposed solution in section IV. The author should mention that collusion between any sensor and the aggregator is not considered as the sensor holds the ORE key.

We thank the reviewer for their comment, and agree that a problem statement was lacking in detail, and missing earlier in the manuscript. We have added Section I.A that states our problem explicitly, as well as the information we treat sensitive and the collusions we consider.

R1.3 - According to (18) and (19), the weights would be public information. A comment should be made on this privacy leakage.

We have now stated that weights are leaked in the problem statement, and when explaining SecFCI. Thank you for pointing this out.

R1.4 - Section IV: The author mentioned, "Since analytical solutions (6) require division, they cannot be computed exactly with the given PHE encryptions of sensor information vectors and information matrices."I disagree with this statement. The Author used an inappropriate encoding mechanism. Check the one used in [1]. The author should mention why they avoid using such an encoding mechanism to solve the problem without going into the proposed direction.

We are sorry for having overlooked the existing literature, and appreciate your comment and reference. We agree with your statement and have updated Section III.C and our references to motivate and justify our choice of encoding.

R1.5 - I expected to see complexity analysis for n sensor case or show the required number of comparisons and the execution time of each one.

We apologies for having not included any run times or a complexity analysis in the manuscript. We have added the new Section V.A which gives both homomorphic operation complexities and the complexity of the complete fusion algorithm. We have also now noted the number of required comparisons in Section V.

R1.6 - Section I.A: The author mentioned "Epk(a) and EORE;k(a) denote the public-key pk and ORE key k encryptions of a"I suggest to change it to Epk(a) and EORE;k(a) denote the encryption of a using the public-key pk and ORE key k, respectively

— [1] "CryptoImg: Privacy-preserving processing over encrypted images" M Tarek Ibn Ziad, Amr

Alanwar, Moustafa Alzantot, Mani Srivastava

Thank you for the suggestion, we have made the change accordingly.

Response to the Comments of Reviewer 2 (23901)

R2.1 This paper applis the combination of two Encryption strategies, named the Partially Homomorphic Encryption (PHE) and Order Revealing Encryption (ORE), to the Fast Covariance Intersection (FCI) so as to perform secure state estimation. In the proposed method, the algebraic operations of PHE and ORE are utilized to find the fusion weight for FCI. Though the topic is for surely interesting, I think the authors should address the following problems before the subject paper being considered to be published:

1, Though the topic is certainly interesting, the main idea of this paper seems to simply combine two existing strategies together. I think such idea should be motivated more, e.g., why choosing the mixed PHE and ORE strategies? What I can find is that it can reduce computational load, but I think this is not enough. Maybe you can say something concerning the accuracy preservation?

We thank the reviewer for their comment, and regret that our reasoning was not made clearer. We have updated the introduction to more accurately motivate our work. FHE is a very computationally expensive method of homomorphic computation, and is the primary reason for the development of a private fusion method which does not rely on it. We have made this clearer with a complexity discussion in the added Section V.A as well. PHE and ORE are used to achieve this as their combined properties were suited to defining the new SecFCI fusion method.

R2.2 2, In the Introduction, the authors should add a short summary of the technical novelties of this paper. Currently I cannot find anything.

We apologies for not stating the novelty clearly, and have updated the introduction accordingly.

R2.3 3, In simulation the authors only considered FCI and SecFCI. Concerning the fact that authors declared that another encryption scheme, named Fully Homomorphic Encryption (FHE), will lead to infeasible computational load. Furthermore, does the proposed method outperforms the encryption scheme in which only PHE or ORE is adopted? I think it would be nice to involve all the aforementioned strategies in Simulation and report also the computational load of them.

We agree that using different encryption schemes can greatly affect the computational load of the algorithm and apologize for having left out the comparisons of different methods. We also regret that the intention of the simulation was not made clearer, in that it was primarily intended to discuss accuracy. We have now added Section V.A where we give the computational complexity of SecFCI, and compare it to the unencrypted FCI algorithm, as well as making comparisons to the complexity of some current FHE scheme operations. We note that although FHE could be used to implement an encrypted homomorphic FCI, PHE or ORE alone has to the best of our knowledge, no known method to do so. We have for this reason not considered methods adopting only one of these schemes in the complexity analysis or simulation. We have also updated Section VI to make the simulation's purpose clearer and to give a more in-depth discussion on the accuracy of the SecFCI method. Run-times have not been given as an efficient implementation of the required encryption schemes is beyond the scope of this paper, but has been addressed as intended future work in Section VII.

Response to the Comments of Reviewer 3 (24111)

R3.1 The paper present a secure algorithm for covariance intersection implementation. Order revealing encryption is used to find the optimal parameters in the algorithm. Subsequently, the Paillier's encryption, a semi-homomorphic encryption method, to implement the algorithm.

There is a large literature on control, estimation, and optimization with homomorphic encryption.

The authors have not cited any of those studies. This is particularly important as they have dealt with some of the issues raised in this paper. For instance, in Section III.C, the authors discuss encoding real numbers when dealing with encryptions that are implemented on integer rings. This has been investigated and discussed at length in those papers, e.g., see secure and private control using semi-homomorphic encryption in Control Engineering Practice.

We apologies for having missed references in homomorphic signal processing, and thank the reviewer for providing a reference. In particular, we agree that our encoding discussion in Section III.C was heavily lacking in references on encoding for homomorphic encryption. The introduction has been updated to include more references regarding homomorphic and partially homomorphic encryption in signal processing, and Section III.C has been updated to include relevant references and justifications for the encoding scheme used.

R3.2 One drawback of encrypted methods is the added overhead to the computational complexity. I think the paper benefits more if the authors discuss the added complexity. If this algorithm is to be implemented in real-time, would we be able to run all the necessary computations? Would the increased complexity results in slower implementation?

We agree that encrypted methods typically have the drawback of added complexity and apologize for having left this out of the manuscript. We have added a new Section V.A which discusses the added complexity of the fusion algorithm when compared to the unencrypted FCI algorithm. While we have implemented a simulation of the algorithm, its primary purpose was to demonstrate the accuracy of the method, which we have now made clearer in Section VI. Run-times have not been given as an efficient implementation of the required encryption schemes is beyond the scope of this paper, but has been addressed as intended future work in Section VII.

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~~ an approximate solution to the fast covariance intersection ~~coefficients~~ 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.**

I. Introduction

Sensor data processing and state estimation have been increasingly prevalent in networked systems ~~[1], [2]~~ [1]. 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]~~ [2], [4], [7]. 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~~ these cross-correlations 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]~~ [14], [16], [17]. 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]~~ [19], [20]. 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].~~

~~A key step in distributed sensor fusion, and our topic of interest in this paper, is the requirement of transmitting sensor state estimate and covariance information between network nodes for~~ As advancements in distributed algorithms and cloud computing develop, the requirements for privacy and security in such systems have become more apparent [21], [22]. In particular for sensor networks, the ~~computation of a final fused result. Network eavesdroppers or curious fusion nodes are not prevented from learning possibly sensitive local state estimates and covariances~~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. Common symmetric-key encryption schemes such as AES ~~[23]~~ [24] are used to encrypt sent information~~to its destination~~, and public-key ~~encryption~~ schemes such as RSA [25] to distribute symmetric keys. However, recent developments in public-key Homomorphic Encryption (HE) schemes ~~[26]–[28]~~ [26]–[29], which allow algebraic operations to be performed on encryptions, are leading to ~~more secure cloud or network~~ novel secure applications for signal processing ~~[?], [?], [30]. Although implementations of~~ in distributed and cloud computing environments [30]–[35]. Fully Homomorphic Encryption (FHE) schemes ~~exist [36], and~~ [26], [29] provide all algebraic operations ~~on encryptions,~~ over encryptions, and are often theoretically suitable for secure processing in distributed environments. However, current implementations are still computationally infeasible for large-scale ~~signal processing [37], [38]. Instead,~~ or real-time processing [38]. Partially Homomorphic Encryption (PHE) schemes [27], [28], providing ~~typically only one algebraic operation~~only a subset of these operations, have been a focus for such ~~processing tasks [?], [30]. Most commonly, addition is provided by the Paillier encryption scheme [28] due to its speed and simplicity. However,~~ tasks due to their reduced computational requirements. [30] use PHE to run a private distributed Information Filter, [32], [33] to compute private distributed control aggregation, [34] for private matrix multiplication, and [35] for private set intersection. These works are, however, due to the limited operations provided ~~, securely~~

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

~~computable processing algorithms have thus far been~~ by PHE, relatively restricted in ~~complexity and application. The recent development of~~ application. Recent developments in new encryption schemes, such as Order Revealing Encryption (ORE) ~~[?], [39], [40], have provided~~ [39]–[41], are now providing new light on the possible complexity of ~~signal processing algorithms that can be computed securely . Thus far, ORE has found little application in the context of signal processing algorithmsor in combination with HE schemes~~securely computable algorithms. In this paper, we ~~make use of a~~ develop a method for secure FCI fusion, such that local sensor information is kept private, using a combination of ORE and PHE schemes ~~to develop a Secure FCI (SecFCI) fusionmethod that enables us to protect sensor estimates from both eavesdroppers and other algorithm participants~~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 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, as will be shown in section IV.

As we assume the querying party is the owner of all individual sensors, we consider the threat model 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, ~~lowercase~~ and underlined characters, $\underline{x}$, represent vectors. Uppercase bold characters, $\mathbf{M}$, are ~~reserved~~ for matrices, where $\cancel{\mathbf{M}^\top}$ $\mathbf{M}^{-1}$ 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~~encryptions of a,~~, respectively, and similarly with the decryption functions $\cancel{\mathcal{D}_{pk}(\cdot)}$ $\mathcal{D}_{sk}(\cdot)$ 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 $\{\cdot\}$ ~~while~~ and 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 weights ω_i must satisfy

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

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

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

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

A. *Fast Covariance intersection*

The Fast Covariance Intersection (FCI) algorithm from [17] is a ~~common method used~~ non-iterative method for approximating the solution to (3) without the loss of guaranteed consistency. It is computed by defining a new constraint

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

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~~

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

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

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

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), ~~homomorphically, such that~~ using only encrypted values from each sensor i, ~~we can produce valid encryptions of fused estimates without the need for decryption~~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. ~~Public-key additive Partially Homomorphic Encryption (PHE) schemes [28], [42] provide~~ The Paillier additive PHE scheme [28] providing a single homomorphic addition operation~~on cyphertexts such that~~

$$\cancel{\mathcal{E}(a) \oplus \mathcal{E}(b) = \mathcal{E}(a+b)}$$

~~holds. Symmetric-key Order Revealing Encryption (ORE) schemes [39], [40] provide~~ , and the Lewi ORE scheme [40] providing a secure comparison function~~, allowing the comparison of encrypted values via~~

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

.

The formal security of encryption schemes consists of a security goal and a formal threat model [43]. Indistinguishability of ciphertexts under the adaptive chosen ciphertext attack model (IND-CCA2) is commonly considered the strongest security guarantee~~[44]. However,~~ , however, no homomorphic encryption scheme provides security against ~~chosen ciphertext attack models~~ 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) [45]. Similarly, ORE schemes aim to protect against simulation-based security defined in [39] or the harder to achieve ordered chosen-plaintext attack model (IND-OCPA).

A. ~~Additive~~ Paillier Partially Homomorphic Encryption Scheme

~~The additive PHE scheme we use is the Paillier encryption scheme [28]~~ 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

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

and

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

where the modulus N is computed as the product of two large ~~primes chosen randomly during~~ 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

~~The OREscheme we have used is Lewi 's~~ For ORE, we use the Lewi symmetric-key Left-Right ~~encryption scheme [40] that~~ 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 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

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

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 ~~means of~~ secure comparison. The Lewi ORE encryption scheme provides security against the simulation-based security model [39] 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 ~~with the upper bound N to the size of an encryptable integer~~ bounds the largest encryptable integer by $N-1$. Due to the prevalence of real numbers in estimation theory, ~~typically stored as floating-point numbers in modern-day hardware, an~~ integer encoding of real ~~number values is required for their encryption. This requires handling both negative and fractional numbers~~ numbers is an active field of research that acommpanies encrypted processing [32], [46], [47], and a requirement for our estimate fusion algorithm. While some encoding schemes for additive homomorphic encryption provide additional operations such as homomorphic division [46], they typically complicate the homomorphic operations, and in [46] leak exponent information of the encrypted real number. We have instead relied on a simplified version of the encoding in [32].

~~Negative numbers can be handled using the common two's complement method of representing negative integers [48]. This is done by splitting the total range of allowable integers $[0, N)$ in half, and letting the upper half $[\frac{N}{2}, N)$ represent negative integers. From this, we can see that the value of the largest encryptable integer is now given by $N/2 - 1$ and that the addition of two's complement numbers is automatically preserved due to modulo arithmetic.~~

~~The handling of fractional numbers proves to be more complicated, due to the homomorphic multiplication property of the Paillier encryption scheme . Fractional numbers are represented as integers using the quantising~~ 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 by conversion to the signed Q number format [49] ~~. The encoding of a~~ and is equivalent to the encoding from [32]. The conversion of any real number a ~~, with maximum integer bits i and fractional bits f is represented by an $i+f$ bit long integer e, such that the maximum encoding~~ to an encoded fixed-point rational is given by ~~$2^{(i+f)}-1$. Encoding is performed by~~

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

While encoded Q numbers are consistent under addition, each multiplication requires a factor of $1/2^f$ to be removed. As ~~homomorphic division is not supported by the Paillier encryptionscheme,~~ shown in [32], 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. Additionally, knowing the number of multiplications ~~performed on encrypted values must be bounded and handled when decoding. As will be shown in section IV, our fusion method will always require that a single multiplication factor be removed and leads to the decoding of an integer e to a real number a being given by~~ allows us to relax the requirement for multiplied encoded numbers not to overflow [32]. We decode real numbers representable by $2b$ bits, which corresponds to the range of b bit encoded numbers after a single multiplication. Decoding is defined as the conversion from the signed Q number format and defined by

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

~~While~~ Since the largest encryptable integer is given by ~~$N/2-1$~~ $N-1$, the largest encodable real number must account for the additional multiplication factor ~~in its encoding~~ when encoded. Thus, the integer ~~and fractional~~ bits i and fractional bits f must be chosen such that

$$(2^{\cancel{(i+f)}\,(1+i+f)} - 1)^2 \leq \cancel{\frac{N}{2}-1}\, N-1\,. \tag{14}$$

~~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~~ consider is described in section I-A, where sensors are capable of running local estimators, as well as the PHE and ORE encryption schemes ~~described in~~ from section III. Each sensor i computes its state estimate $\hat{\underline{x}}_i$ and covariance matrix $\mathbf{P}_i$ and sends ~~the~~ relevant encrypted information to ~~a single fusion centerthat 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.~~ an untrusted cloud fusion center. 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 [25]. 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. ~~In the~~ 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 Q encoding multiplication factor to remove, and can be decoded exactly by using (13).

~~In the two sensor case, all~~ 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 ~~the FCIfusion algorithm. Since analytical solutions require division, they cannot be computed exactly with the given PHE encryptions of sensor information vectors and information matrices. Instead, we discretize~~ FCI. Since our encoding scheme in section III-C does not provide 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 ~~approximate~~ approximating (5) with ORE. An ordered discretization of values $\omega^{(x)}$ is defined by

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

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

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

and similarly sensor 2's by

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

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

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

If we reverse (19), we obtain ~~the equivalent list~~

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

~~which when~~ a list equivalent to one with values $\mathcal{E}^R_{ORE}((1-\omega^{(x)})\,\mathrm{tr}(\mathbf{P}_2))$ for each discretization step x. When the reversed list is decrypted and plotted over ~~shows that the intersecting point~~ (18) the intersection gives the

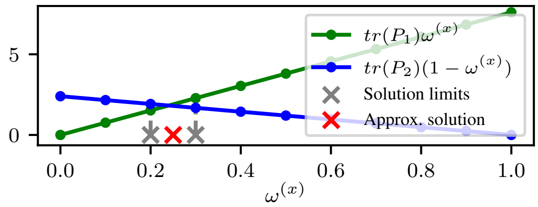


Fig. 1: Approximation of ω_1 with discretisation step-size $s = 0.1$. Only comparisons between line points are used.

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 ~~the~~ consecutive $\omega^{(x)}$ discretisations where the sign of comparisons changes. This can be seen in Fig. 1, and can be performed in ~~$O(log(p))$ by~~ $O(\log p)$ 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

$$\omega_1 \approx \cancel{\frac{1}{2}}0.5(\omega^{(x)} + \omega^{(x+1)})\,. \tag{21}$$

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 computing the plaintext weights $\omega_1, \dots, \omega_n$.

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 ~~partial solution~~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 ~~an equation~~ equations with two parameters ω_k and ω_{k+1} when letting all $\omega_i = 0,\; i \neq k, k+1$~~, and can compute a remaining point for each~~. 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~~when at most 1 sensor has $\operatorname{tr}(\mathbf{P}_i) = 0$~~. The hyperplane intersection results in the linear system

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

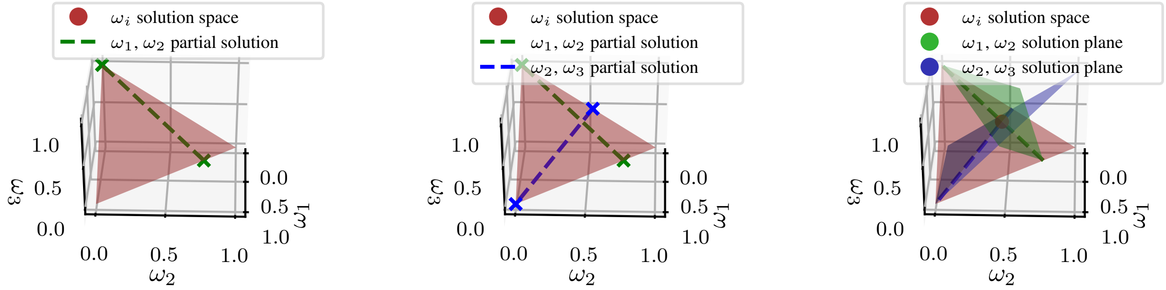
and gives the solution to the SecFCI ω_i ~~values~~weights.

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

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

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

Briefly considering the security of our scheme, we note that any leaked information from ORE lists (30), as described in [39], can be considered a subset of knowing the estimated fusion weights $\omega_1, \dots, \omega_n$, 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) Partial solution to (24). (b) Partial solutions to (24) and (25). (c) Partial solutions as planes~~perpendicular to the solution space~~.

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

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	$O(n^3)$	$O\left((\log p + \log N) n \log^2 N^2 + n^3\right)$

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 [24], we consider software implementations and treat large integer operations in terms of bit operations explicitly.

From [28], [40], 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 [50], which has complexity $O(\lambda^{10})$ for integer operations, and [29], 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.

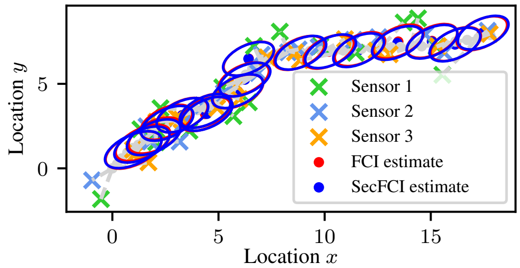


Fig. 3: Tracking simulation comparing ~~FCI and~~ SecFCI and FCI.

VI. Simulation Results

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

~~The traces of the fused covariance matrices from both FCI~~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. Encoding 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 ~~SecFCI fusion are shown in Fig. ?? where a very small difference in trace can be seen. As SecFCI approximates FCI which in turn is an approximation to the CI's trace minimization function , neither method guarantees a resulting trace smaller than the other. Traces of fused covariance matrices for FCI and SecFCI. Values of each~~ f, we will only consider the error caused by the difference in weights. We will treat this error as the distance

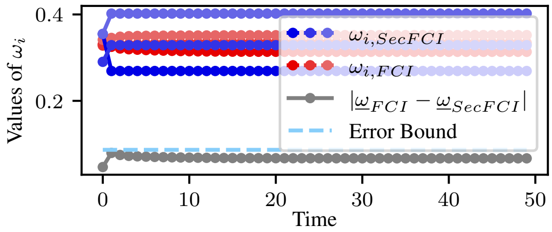


Fig. 4: ~~FCI~~ ω_{SecFCI} and ~~SecFCI ω_i values and their difference~~ω_{FCI} components.

between respective weight vectors

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

where $\omega_{i,s}$ denotes weight ω_i ~~over time~~ 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 dependent on the value of $\omega_{i,FCI}$, but will remain strictly bounded by $s/2$. Therefore, we can give a strict upper bound on the distance between weight vectors as

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

Finally, components of $\omega_{i,SecFCI}$, $\omega_{i,FCI}$ and the errors $|\omega_{FCI} - \omega_{SecFCI}|$, have been plotted over time in Fig. 4~~. Due to discretization, as can be expected, SecFCI values of ω_i stay constant but may jump with sufficient change in true FCI ω_i values, as can be seen near the start of the simulation~~, and show the commputed 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 ~~signal processing~~ fusion 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~~ comparisons with FHE implementations, giving a computational bound for its practicality~~. Also, we hope to further quantify ORE leakage concerning SecFCI fusion and produce~~ , and quantification of fusion weight leakages via 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 *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 *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 *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] 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.

[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 *19th International Conference on Information Fusion (Fusion 2016)*, 2016, pp. 1600–1607.

[13] N.~Carlson, "Federated Filter for Fault-tolerant Integrated Navigation Systems," in *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 *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 *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 *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 *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 IEEE International Conference on Digital Ecosystems and Technologies (DEST 2011)*, 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] S.~Gueron, "Intel Advanced Encryption Standard (AES) New Instructions Set," *Intel Corporation*, 2010.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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 *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 *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] C.~Gentry and S.~Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2011, pp. 129–148.

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

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

[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 1998)*. Springer, 1998, pp. 26–45.

[45] M.~Chase *et al.*, "Security of Homomorphic Encryption," *Technical Report, HomomorphicEncryption.org, Redmond WA, USA*, 2017.

[46] 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.

[47] 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.

[48] D. J.~Lilja and S. S.~Sapatnekar, *Designing Digital Computer Systems with Verilog*. Cambridge University Press, 2004.

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

[50] 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.