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 private to their producing party. Current secure fusion algorithms have a reliance 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 the 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 the 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 insecure 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-correlation between estimation errors characterise 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 omit the need for repeated reconstruction [12] however 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 matrix and information vector instead of the usual state estimate and 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 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. Encryption has until recently been primarily used to secure information transfer between communicating parties, with common symmetric-key encryption schemes such as AES [21] being used to encrypt sent information to its destination, and public-key encryption schemes such as RSA [22] to distribute symmetric keys. However, recent developments in public-key homomorphic encryption (HE) schemes [23]–[25], which allow algebraic operations to be performed on encryptions, are leading to more secure cloud or network applications for signal processing [26]–[28]. Although implementations of Fully Homomorphic Encryption (FHE) schemes exist [29], and provide all algebraic operations on encryptions, current implementations are still computationally infeasible for large-scale signal processing [30], [31]. Instead, Partially Homomorphic Encryption (PHE) schemes [24], [25], providing typically only one algebraic operation, have been a focus for such processing tasks [27], [28]. However, due to the limited operations provided by PHE (most commonly addition provided by the Paillier encryption scheme due to its speed and simplicity, [25]), securely computable processing algorithms have thus far been relatively restricted in complexity and application. The recent development of new encryption schemes, such as Order Revealing Encryption (ORE) [32]–[34], have provided 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 algorithms or in combination with HE schemes. In this paper, we make use of a combination of ORE and PHE schemes to develop a Secure FCI (SecFCI) fusion approximation method that enables us to protect sensor estimates from both eavesdroppers and other algorithm participants.

In section II we will introduce CI and FCI methods relevant to our proposed fusion algorithm, and in section III, the relevant encryption schemes. Sections IV and V will introduce the secure FCI algorithm for the 2 sensor and multi-sensor cases respectively, and VI discusses simulation results and comparisons to the ordinary FCI fusion algorithm.

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

We conclude our findings and plans for future work in VII.

A. 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 public-key pk and ORE key k encryptions of a, 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 estimation fusion algorithm when model cross-correlations are not known. The resulting fused estimate $\hat{\underline{x}}$ and estimate covariance $\mathbf{P}$ can be easily derived from its equations

$$\mathbf{P}^{-1} = \sum_{i=0}^{n} \omega_i \mathbf{P}_i^{-1} \tag{1}$$

and

$$\mathbf{P}^{-1}\hat{\underline{x}} = \sum_{i=0}^{n} \omega_i \mathbf{P}_i^{-1}\hat{\underline{x}}_i \,. \tag{2}$$

Note that (1) and (2) compute the fusion of the information matrix and information vectors defined in [17] and reduce the fusion to a simple weighted sum. Values for ω_i must satisfy

$$\omega_0 + \omega_1 + \cdots + \omega_n = 1 \tag{3}$$

and

$$0 \leq \omega_i \leq 1 \,. \tag{4}$$

and guarantee consistency of the fused estimates. They are chosen in a way to speed up convergence, by minimising a certain specified property of the resulting fused estimate covariance. One such property which may be minimised to guarantee faster convergence is the fused estimate covariance trace, requiring the solution to

$$\underset{\omega_0,\ldots,\omega_n}{\arg\min}\{\mathrm{tr}(\mathbf{P})\} = \underset{\omega_0,\ldots,\omega_n}{\arg\min}\{\mathrm{tr}((\sum_{i=0}^{n}\omega_i\mathbf{P}_i)^{-1})\} \,. \tag{5}$$

However, minimising 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 (5) without the loss of guaranteed consistency. It is computed by defining a new constraint

$$\omega_i \,\mathrm{tr}(\mathbf{P}_i) - \omega_j \,\mathrm{tr}(\mathbf{P}_j) = 0, \;\; (i,j = 1,2,\ldots,n) \tag{6}$$

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

$$\omega_0 + \omega_1 = 1 \tag{7}$$

and

$$\omega_0 \,\mathrm{tr}(\mathbf{P}_0) - \omega_1 \,\mathrm{tr}(\mathbf{P}_1) = 0 \,, \tag{8}$$

with analytical solutions given by

$$\omega_0 = \frac{\mathrm{tr}(\mathbf{P}_1)}{\mathrm{tr}(\mathbf{P}_0) + \mathrm{tr}(\mathbf{P}_1)}, \;\; \omega_1 = \frac{\mathrm{tr}(\mathbf{P}_0)}{\mathrm{tr}(\mathbf{P}_0) + \mathrm{tr}(\mathbf{P}_1)} \,. \tag{9}$$

When computed for the n sensor case, the highly redundant (6) can have its largest linearly independent subset represented by

$$\omega_i \,\mathrm{tr}(\mathbf{P}_i) - \omega_{i+1} \,\mathrm{tr}(\mathbf{P}_{i+1}) = 0, \;\; (i = 1,2,\ldots,n) \,, \tag{10}$$

and requires the solution to the linear problem

$$\begin{bmatrix} \mathcal{P}_0 & -\mathcal{P}_1 & 0 & \cdots & 0 \\ 0 & \mathcal{P}_1 & -\mathcal{P}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathcal{P}_{n-1} & -\mathcal{P}_n \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_0 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

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

Our proposed filter aims to solve FCI fusion, namly (1), (2) and (11), homomorphically, such that using only encryptions from each sensor i we are able to 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 have made use of two types of function-providing encryption schemes. Public-key additive Partially Homomorphic Encryption schemes [25], [35] provide a single homomorphic addition operation on cyphertexts such that

$$\mathcal{E}(a) \oplus \mathcal{E}(b) = \mathcal{E}(a+b) \tag{12}$$

holds. While symettric-key Order Revealing Encryption schemes [32], [33] provide a secure comparison function, allowing the comparison of encrypted values via

$$f(\mathcal{E}_{ORE}(a), \mathcal{E}_{ORE}(b)) = cmp(a,b) \,. \tag{13}$$

The formal security of encryption schemes consists of a security goal and a formal threat model [36]. Indistinguishability of ciphertexts under the adaptive chosen ciphertext attack model (IND-CCA2) is the commonly considered strongest security guarantee [37], however, cannot be satisfied by any homomorphic encryption scheme 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) [38]. Similarly, ORE schemes aim to protect against the ordered chosen-plaintext attack (IND-OCPA) or weaker simulation-based security defined in [32] which allows for some leakage.

A. Additive Partially Homomorphic Encryption

The additive PHE scheme we have used is the Paillier encryption scheme [25] due to its implementation simplicity, and computational speed. The Paillier scheme provides two homomorphic operations on encrypted data, namely

$$\mathcal{E}_{pk}(a)\mathcal{E}_{pk}(b) \pmod{N^2} = \mathcal{E}_{pk}(a+b \pmod{N}) \quad (14)$$

and

$$\mathcal{E}_{pk}(a)^c \pmod{N^2} = \mathcal{E}_{pk}(ca \pmod{N}),\ c \in \mathbb{Z}_N^*\,, \quad (15)$$

where the modulus N is computed as the product of 2 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 protects against the IND-CPA security goal and attacker model.

B. Left-Right Order Revealing Encryption

The ORE scheme we have used is Lewi's symmetric-key Left-Right encryption scheme [33] which 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

$$\begin{aligned} enc^L_{ORE}(k,x) &= \mathcal{E}^L_{ORE,k}(x) \\ enc^R_{ORE}(k,y) &= \mathcal{E}^R_{ORE,k}(y)\,, \end{aligned} \quad (16)$$

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

$$cmp_{ORE}(\mathcal{E}^L_{ORE}(x),\ \mathcal{E}^R_{ORE}(y)) = cmp(x,y)\,. \quad (17)$$

Note that no decryption function is provided, as encryptions are only required to provide a means of secure comparison. The Lewi ORE encryption scheme does not satisfy IND-OCPA security due to its apparent difficulty, and instead satisfies the simulation-based notion of ORE security [32], [33].

C. Real Number Encoding for Homomorphic Encryption

Both encryption schemes in sections III-A and III-B are defined over positive integers, while the Paillier scheme further gives an upper bound N to the size of an encryptable integer. 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 the handling of both fractional, and negative numbers.

Negative numbers can be handled using the common two's complement method of representing negative integers [39]. 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 a more complicated matter, due to the homomorphic multiplication property of the Paillier encryption scheme (15). Fractional numbers are represented as integers using the quantising Q number format [40]. The encoding of a 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 is given by $2^{(i+f)}-1$. Encoding is performed by

$$e = \lfloor 2^f a \rfloor\,. \quad (18)$$

While encoded Q numbers are consistent under addition, multiplication requires a factor of $1/2^f$ to be removed. As homomorphic division is not supported by the Paillier encryption scheme, 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

$$a = \frac{e}{2^{2f}}\,. \quad (19)$$

While the largest encryptable integer is given by $N/2-1$, the largest encodable real number must account for the additional multiplication factor and must have i and f chosen such that

$$(2^{(i+f)}-1)^2 \leq \frac{N}{2} - 1 \quad (20)$$

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 estimation filters, as well as the PHE and ORE encryption schemes described in section III. Each sensor i computes its state estimate and covariance matrix, $\underline{\hat{x}}_i$, $\mathbf{P}_i$ and sends the relevant encrypted information to a single fusion centre which aims to compute the fused state estimate and covariance matrix homomorphically. A third, querying party, can request and use the current encrypted fused information from the fusion centre 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 [22]. 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 the CI equations (1) and (2) we can see that both fusion equations can be computed on PHE encryptions of sensor information matrices and information vectors, given valid values for each ω_i. In the two-sensor case, homomorphic fusion is computed by

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_0^{-1})^{\omega_0}\mathcal{E}(\mathbf{P}_1^{-1})^{(1-\omega_0)} \tag{21}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\hat{\underline{x}}) = \mathcal{E}(\mathbf{P}_0^{-1}\hat{\underline{x}}_0)^{\omega_0}\mathcal{E}(\mathbf{P}_1^{-1}\hat{\underline{x}}_1)^{(1-\omega_0)}\,, \tag{22}$$

where we note that $\omega_1 = 1-\omega_0$ due to the CI requirements (3) and (4). We also note that in (21) and (22), each resulting value will have exactly one Q encoding multiplication factor to remove, and can be decoded exactly by using (19).

In the two-sensor case, all that remains for computing CI homomorphically is the calculation of parameter ω_0. For this, we approximate the solution to the FCI fusion algorithm. Since analytical solutions (9) require division, they cannot be computed exactly with the given PHE encryptions of sensor information matrices and information vectors. Instead, we discretise ω_0 by step-size s, such that $s < 1$ and $s|1$, and turn to ORE to approximate (8). Each sensor computes $\mathrm{tr}(\mathbf{P})\omega$ for each ordered discretisation $\omega \in [0, 0+s, \ldots, 1-s, 1]$, and encrypts it with its ORE key k. Each sensor i uses L ORE if i is even, and R ORE otherwise. The resulting encryptions are kept ordered and sent alongside the PHE encryptions of local information matrix and information vector estimates. Sensor 0's list is defined by

$$[\mathcal{E}^L_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_0)),\ \omega \in [0, 0+s, \ldots, 1-s, 1]] \tag{23}$$

and similarly sensor 1's by

$$[\mathcal{E}^R_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_1)),\ \omega \in [0, 0+s, \ldots, 1-s, 1]]\,. \tag{24}$$

Lists (23) and (24) are then used to estimate the FCI values of ω_0 and ω_1 before computing (21) and (22).

From (8) and (7) we know that ω_0 must satisfy

$$\omega_0\,\mathrm{tr}(\mathbf{P}_0) = (1-\omega_1)\,\mathrm{tr}(\mathbf{P}_1)\,. \tag{25}$$

If we reverse (24), we obtain the equivalent list

$$[\mathcal{E}^R_{ORE}((1-\omega)\,\mathrm{tr}(\mathbf{P}_1)),\ \omega \in [0, 0+s, \ldots, 1-s, 1]] \tag{26}$$

which when decrypted and plotted over (23) shows that the intersecting point gives the solution to (8). However, (24) and (26) consist of L and R ORE encryptions respectively, and the intersection must be approximated by locating the consecutive ω discretisations where the sign of comparisons changes. This can be seen in Fig. 1, and can be performed in $O(log(1/s))$ by a binary search. Consecutive ω_l and ω_u for which list comparisons differ can be used to estimate the true intersection, and ω_0, by

$$\omega_0' = \frac{1}{2}(\omega_l + \omega_u)\,. \tag{27}$$

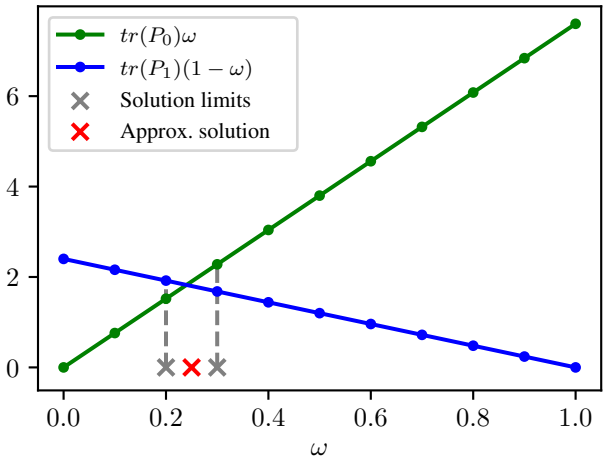

Fig. 1. Approximation of ω_0 with discretisation step-size $s = 0.1$. Only comparisons between line points are used.

In the case a comparison returns equality, the exact value of ω can be taken to be ω_0.

V. Multi-sensor Secure Fast Covariance Intersection

When computing the SecFCI fusion for n sensors, we solve (1) and (2) homomorphically by computing

$$\mathcal{E}(\mathbf{P}^{-1}) = \mathcal{E}(\mathbf{P}_0^{-1})^{\omega_0}\cdots\mathcal{E}(\mathbf{P}_n^{-1})^{\omega_n} \tag{28}$$

and

$$\mathcal{E}(\mathbf{P}^{-1}\hat{\underline{x}}) = \mathcal{E}(\mathbf{P}_0^{-1}\hat{\underline{x}}_0)^{\omega_0}\cdots\mathcal{E}(\mathbf{P}_n^{-1}\hat{\underline{x}}_n)^{\omega_n}\,. \tag{29}$$

As with the two-sensor case, encoded results from (28) and (29) contain exactly one multiplication factor to remove and can be decoded exactly with (19). Again we are just left with computing the weights $\omega_0, \ldots, \omega_n$.

Our approach to the n-sensor case, is to solve each $n-1$ conditions in (10) using the two-sensor method, and combining partial solutions to compute the final result. When we consider each ω_i a dimensional axis, partial solutions can be considered geometrically, as hyperplanes of $n-2$ dimension, over the $n-1$ dimensional solution space given by (3) and (4). This can be visualised in the three-sensor case, which requires solving partial solutions

$$\omega_0\,\mathrm{tr}(\mathbf{P}_0) - \omega_1\,\mathrm{tr}(\mathbf{P}_1) = 0 \tag{30}$$

and

$$\omega_1\,\mathrm{tr}(\mathbf{P}_1) - \omega_2\,\mathrm{tr}(\mathbf{P}_2) = 0\,. \tag{31}$$

Fig. 2 shows partial solution (30) plotted over the possible solution space. Partial solution points are given by solving the two-sensor case for ω_j and ω_k when letting all $\omega_i = 0,\ i \neq j, k$, plus an additional point for each $\omega_i = 1,\ i \neq j, k$ with $\omega_j = \omega_k = 0$. Both partial solutions to (30) and (31) can be seen plotted over the solution space in Fig. 3. Finding the final solution from all partial solutions is computed by

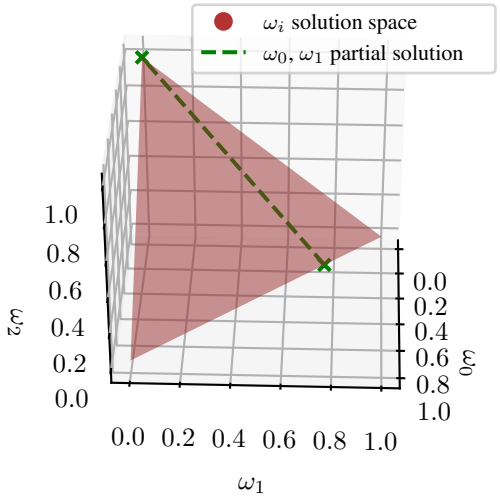

Fig. 2. Partial solution to (30) over plane of ω_0, ω_1, and ω_2 solution space.

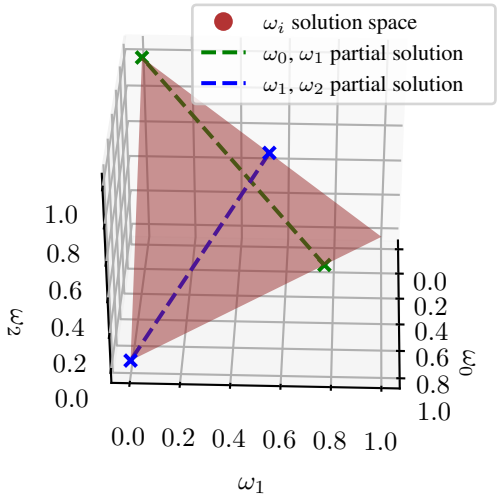

Fig. 3. Partial solutions to (30) and (31) over plane of ω_0, ω_1, and ω_2 solution space.

finding their intersection. This can be seen in Fig. 3 as the intersection of the ω_0, ω_1 and ω_1, ω_2 partial solution lines. Due to their inherent orthogonality, $n-1$ partial solution hyperplanes are guaranteed to intersect at exactly one point when at most 1 sensor has $\mathrm{tr}(\mathbf{P}_i) = 0$.

To simplify computing the partial solution intersection, we define equivalent hyperplanes of dimension $n-1$, perpendicular to the solution space, and solve the resulting n simultaneous linear equations.

In the three-sensor case, each partial solution line can be defined by a plane perpendicular to the solution space plane

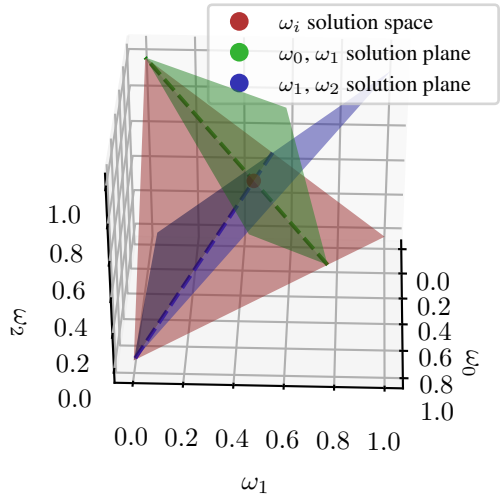

Fig. 4. Partial solutions from Fig. 3 as planes perpendicular to the solution space plane. Intersection point gives SecFCI solutions for each ω_i.

in the form

$$a_0\omega_0 + a_1\omega_1 + a_2\omega_2 + a_3 = 0\,, \tag{32}$$

as can be seen in Fig. 4. The resulting linear system and solution is then given by

$$\begin{bmatrix} a_{0,0} & a_{1,0} & a_{2,0} \\ a_{0,1} & a_{1,1} & a_{2,1} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} a_{3,0} \\ a_{3,1} \\ 1 \end{bmatrix}, \tag{33}$$

where $a_{i,j}$ denotes parameter i of partial solution j.

In the case with n sensors, the hyperplane equations are given by

$$a_0\omega_0 + a_1\omega_1 + \cdots + a_n\omega_n + a_{n+1} = 0\,, \tag{34}$$

and simultaneous equations and solution by

$$\begin{bmatrix} a_{0,0} & a_{1,0} & \cdots & a_{n,0} \\ a_{0,1} & a_{1,1} & \cdots & a_{n,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0,n-1} & a_{1,n-1} & \cdots & a_{n,n-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} a_{n+1,0} \\ a_{n+1,1} \\ \vdots \\ a_{n+1,n-1} \\ 1 \end{bmatrix}. \tag{35}$$

As all ORE comparisons between sensor values are done between sequential sensors, 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 is given by

$$\begin{aligned} &[\mathcal{E}^L_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]],\ i \text{ even} \\ &[\mathcal{E}^R_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]],\ i \text{ odd}\,. \end{aligned} \tag{36}$$

VI. Simulation Results

We have implemented a simulation to demonstrate and compare the SecFCI algorithm. Three sensors independently measured a two-dimensional constant-speed linear process,

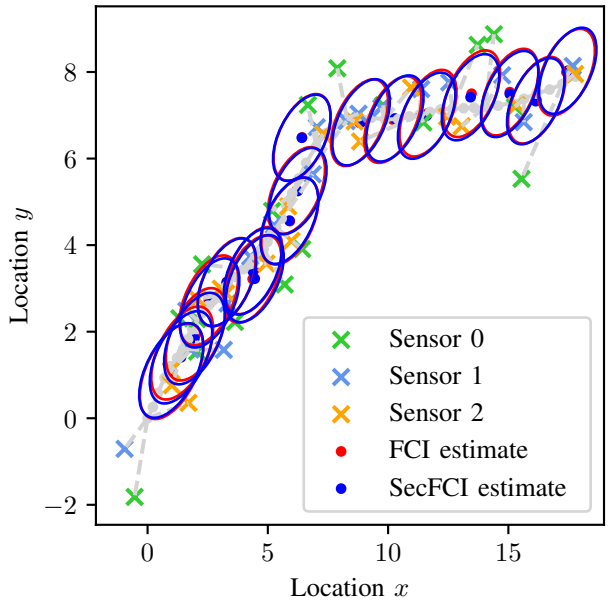

Fig. 5. Tracking simulation comparing FCI and SecFCI methods.

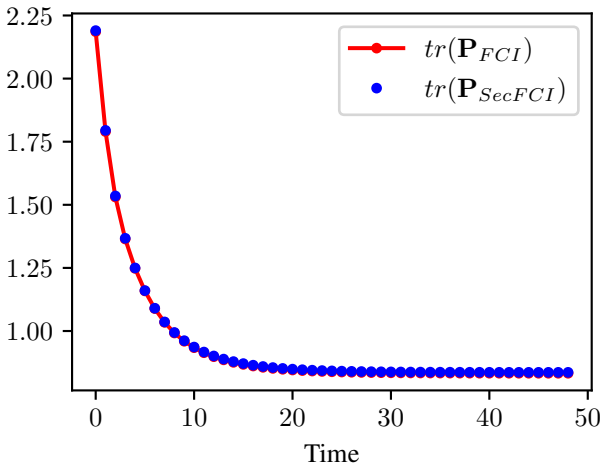

Fig. 6. Traces of fused covariance matrices for FCI and SecFCI.

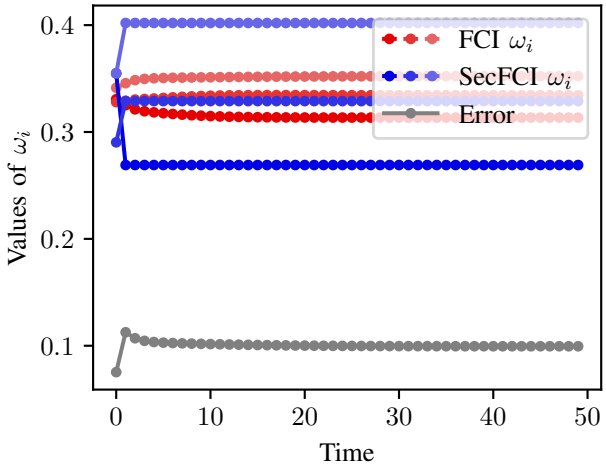

Fig. 7. FCI and SecFCI ω_i values and their difference.

and simultaneously run a linear Kalman filter on their measurements. Estimates were then sent both unencrypted and encrypted to a fusion centre which computed the FCI and SecFCI fusions on the received data respectively. Unencrypted estimates consisted of the state estimate and covariance matrix $\hat{\underline{x}}_i$, $\mathbf{P}_i$, while encrypted estimates were comprised of PHE encryptions of the information matrix and information vector $\mathcal{E}(\mathbf{P}_i^{-1}\hat{\underline{x}}_i)$ and $\mathcal{E}(\mathbf{P}_i^{-1})$, and the ORE list given by (36). The trajectory and fused estimates are shown in Fig. 5.

The traces of the fused covariance matrices from both FCI and SecFCI fusion are shown in Fig. 6. As can be seen, the trace difference is very small, and due to both methods approximating CI's minimisation function (5), neither is necessarily a bound on the other. Values of each ω_i over time have been plotted in Fig. 7. Due to discretisation, as can be expected, SecFCI values of ω_i stay constant, but may jump with sufficient change in the true FCI ω_i values they estimate.

VII. Conclusion

FCI is a commonly used, efficiently computable, approximation to the CI optimisation problem which requires the sharing of local sensor estimates to compute their fusion. We have proposed 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. It is clear that particular computational requirements are necessary for the computation of SecFCI, and that it may only be practical in specific scenarios. Namely, the ability for sensors to compute local estimation and encryption is required. 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 quantize ORE leakage concerning SecFCI fusion and produce formal security proofs and assumptions for the novel algorithm.

References

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

[2] C.-Y. Chong, "Forty years of distributed estimation: A review of noteworthy developments," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct. 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, Mar. 1960.

[4] D. Willner, C.-B. Chang, and K.-P. Dunn, "Kalman filter algorithms for a multi-sensor system," Jan. 1977, pp. 570–574.

[5] C. Y. Chong, "HIERARCHICAL ESTIMATION," p. 17.

[6] C.-Y. Chong, K.-C. Chang, and S. Mori, "Distributed Tracking in Distributed Sensor Networks," in *1986 American Control Conference*, Jun. 1986, pp. 1863–1868.

[7] H. Hashemipour, S. Roy, and A. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988, conference Name: IEEE Transactions on Automatic Control.

[8] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, Oct. 1994.

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

[10] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, Apr. 1981, conference Name: IEEE Transactions on Automatic Control.

[11] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, Jun. 2004.

[12] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck, "Optimal sample-based fusion for distributed state estimation," in *2016 19th International Conference on Information Fusion (FUSION)*, Jul. 2016, pp. 1600–1607.

[13] N. Carlson, "Federated filter for fault-tolerant integrated navigation systems," in *IEEE PLANS '88.,Position Location and Navigation Symposium, Record. 'Navigation into the 21st Century'.*, Nov. 1988, pp. 110–119.

[14] S. J. Julier, "A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations," p. 5.

[15] J. Sijs, M. Lazar, and P. Bosch, "State fusion with unknown correlation: Ellipsoidal intersection," in *Proceedings of the 2010 American Control Conference*, Jun. 2010, pp. 3992–3997.

[16] B. Noack, J. Sijs, and U. D. Hanebeck, "Inverse covariance intersection: New insights and properties," in *2017 20th International Conference on Information Fusion (Fusion)*. Xi'an, China: IEEE, Jul. 2017, pp. 1–8.

[17] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, vol. 2, Jul. 2002, pp. 901–904 vol.2.

[18] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *2005 7th International Conference on Information Fusion*, vol. 1, Jul. 2005, pp. 7 pp.–.

[19] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Routledge, May 2019.

[20] F. Pfaff, B. Noack, U. D. Hanebeck, F. Govaers, and W. Koch, "Information form distributed Kalman filtering (IDKF) with explicit inputs," in *2017 20th International Conference on Information Fusion (Fusion)*. Xi'an, China: IEEE, Jul. 2017, pp. 1–8.

[21] "Announcing the Advanced Encryption Standard (AES)," NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD, Tech. Rep., Nov. 2001.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[23] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing - STOC '09*. Bethesda, MD, USA: ACM Press, 2009, p. 169.

[24] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," in *Advances in Cryptology*, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds. Berlin, Heidelberg: Springer, 1985, pp. 10–18.

[25] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology — EUROCRYPT '99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Springer Berlin Heidelberg, 1999, pp. 223–238.

[26] R. L. Lagendijk and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, Jan. 2013.

[27] A. B. Alexandru, M. S. Darup, and G. J. Pappas, "Encrypted Cooperative Control Revisited," p. 7.

[28] M. Aristov, B. Noack, U. D. Hanebeck, and J. Muller-Quade, "Encrypted Multisensor Information Filtering," in *2018 21st International Conference on Information Fusion (FUSION)*. Cambridge, United Kingdom: IEEE, Jul. 2018, pp. 1631–1637.

[29] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Advances in Cryptology – EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed. Springer Berlin Heidelberg, 2011, pp. 129–148.

[30] Y. Du, L. Gustafson, D. Huang, and K. Peterson, "Implementing ML Algorithms with HE," p. 14.

[31] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 79:1–79:35, Jul. 2018.

[32] N. Chenette, K. Lewi, S. A. Weis, and D. J. Wu, "Practical Order-Revealing Encryption with Limited Leakage," in *Fast Software Encryption*, T. Peyrin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, vol. 9783, pp. 474–493.

[33] K. Lewi and D. J. Wu, "Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*. Vienna, Austria: ACM Press, 2016, pp. 1167–1178.

[34] D. Bogatov, G. Kollios, and L. Reyzin, "A Comparative Evaluation of Order-Preserving and Order-Revealing Schemes and Protocols," p. 19.

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

[36] J. Katz, *Introduction to Modern Cryptography: Principles and Protocols*, 1st ed. Chapman and Hall/CRC, Aug. 2007.

[37] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Advances in Cryptology — CRYPTO '98*, G. Goos, J. Hartmanis, J. van Leeuwen, and H. Krawczyk, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, vol. 1462, pp. 26–45, series Title: Lecture Notes in Computer Science.

[38] M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, J. Hoffstein, K. Lauter, S. Lokam, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "SECURITY OF HOMOMORPHIC ENCRYPTION," p. 27.

[39] D. J. Lilja and S. S. Sapatnekar, "Designing Digital Computer Systems with Verilog," p. 176.

[40] E. L. Oberstar and O. Consulting, "Fixed-Point Representation & Fractional Math," p. 19.