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 to approximate solutions to the fast covariance intersection coefficients can be computed and combined with partially homomorphic encryptions of estimates, to compute 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 []. Bayesian state estimation has become a particularly common application since the beginning of Kalman estimation theory [] and has led to a large interest in the field of state estimation fusion [].

Challenges of estimation fusion are closely tied to the handling and merging of estimation error statistics []. Cross-correlation between estimation errors characterise dependencies between local estimates and must be considered when performing consistent or optimal fusion [].

Methods that keep track of the cross-correlation of errors omit the need for repeated reconstruction [] 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 [].

Covariance Intersection (CI) [] provides one such conservative strategy, from which less computationally expensive methods such as the Fast Covariance Intersection (FCI) [] have been derived. CI is particularly well paired with the information form of the Kalman filter []. This algebraically equivalent form of the standard Kalman filter requires the persistent storing of the information matrix and 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 [] and within fully distributed filter implementations [].

A key step in distributed sensor fusion, and our topic of interest in this paper, is the requirement of transmitting sensor state estimate 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 errors.

Encryption has until recently been primarily used to secure information transfer between communicating parties, with common symmetric-key encryption schemes such as AES [] being used to encrypt sent information to its destination, and public-key encryption schemes such as RSA [] to distribute symmetric keys. However, recent developments in public-key homomorphic encryption (HE) schemes [](FHE, PHE), which allow algebraic operations to be performed on encryptions, are leading to more secure cloud or network applications for signal processing [](lag, Alex, zeros, aris).

Although implementations of Fully Homomorphic Encryption (FHE) schemes exist [](FHE), and provide all algebraic operations on encryptions, current implementations are still computationally infeasible for large-scale signal processing [](ML, survey). Instead, Partially Homomorphic Encryption (PHE) schemes [](PHE), providing typically only one algebraic operation, have been a focus for such processing tasks [](Alex, zeros, aris).

However, due to the limited operations provided by PHE (most commonly addition provided by the Paillier encryption scheme due to its speed and simplicity, []), 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) [], 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.

- Describe what will be in each following section and how the paper structure makes sense

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

A. Notation

- Brief overview of ISAS vector and random vector notation
- Encryption notation, keys omitted where it's obvious from context
- Encryption of matrices and vectors is element-wise
- Real number encoding assumed in all encryption notation, all numbers are real

II. Covariance Intersection and Approximations

- Covariance intersection (CI) is a method for fusing state estimations for different sources when models and cross-correlations are not known
- The fused estimate and estimate covariance are commuted by (1) and (2)
- Note that the estimate and estimate covariance depicted in (1) and (2) are in the Information filter form of the popular Kalman Filter as this simplifies computation.

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

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

- Where values ω_i satisfy (3) and (4) and are chosen in a way to minimise a property of the resulting fused estimate.
- For example minimising the resulting trace would require solving (5)

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

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

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

- Minimising a given non linear cost function such as (5) can be very costly computationally and has led to the development of non-iterative approximation techniques [1], [2], [3]

A. Fast Covariance intersection

- The fast covariance intersection (FCI) algorithm described in [1] is a common method used for approximating the solution to (5) by defining a new constraint on ω_i and solving that instead.
- In the 2 sensor case, (3) now becomes (6), and the additional requirement of (7) is also defined. Solutions are defined analytically and shown in (8)

$$\omega_0 + \omega_1 = 1 \tag{6}$$

$$\omega_0 \operatorname{tr}(\mathbf{P}_0) - \omega_1 \operatorname{tr}(\mathbf{P}_1) = 0 \tag{7}$$

$$\omega_0 = \frac{\operatorname{tr}(\mathbf{P}_1)}{\operatorname{tr}(\mathbf{P}_0) + \operatorname{tr}(\mathbf{P}_1)},\ \omega_1 = \frac{\operatorname{tr}(\mathbf{P}_0)}{\operatorname{tr}(\mathbf{P}_0) + \operatorname{tr}(\mathbf{P}_1)} \tag{8}$$

- When extending to any number of sensors, restriction (7) is generalised to (9)

$$\omega_i \operatorname{tr}(\mathbf{P}_i) - \omega_j \operatorname{tr}(\mathbf{P}_j) = 0,\ (i, j = 1, 2, \ldots, n) \tag{9}$$

- Equation (9) is highly redundant and its largest linearly independent subset can be represented with (10).

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

- The solution to (10) and (3) can be represented as the simultaneous equations problem shown in (11) where $\mathcal{P}_i = \operatorname{tr}(\mathbf{P}_i)$

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

- Our goal for secure fusion is to solve (1), (2) and (11) using only encryptions from each sensor i.

III. Homomorphic and Order Revealing Encryption

- To achieve a secure solution to the FCI fusion problem, we have focused on two types of function providing encryption schemes.
- Additive partially homomorphic encryption (PHE) schemes such ones defined in [4] and [5] provide a homomorphic addition operation as shown in (12).
- Order revealing encryption (ORE) schemes such as those in [6] and [7] provide a function which allows the comparison of encrypted values as shown in (13)

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

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

A. Additive Partially Homomorphic Encryption

- We have used the Paillier encryption scheme described in [4] for the computation of the FCI due to its speed and implementation simplicity.
- The Paillier encryption scheme is a public key scheme, where encryptions are made with a public key but only the secret key holder can decrypt them.
- It provides two homomorphic operations on encrypted data, shown in (14) and (15). The modulus N is computed as a product of 2 large primes which are part of the secret key.

$$\mathcal{E}(a)\mathcal{E}(b) \pmod{N} = \mathcal{E}(a + b \pmod{N}) \tag{14}$$

$$\mathcal{E}(a)^c \pmod{N} = \mathcal{E}(ca \pmod{N}),\ c \in \mathbb{Z}_N \tag{15}$$

- Encrypted numbers must be less than N, and negative numbers can be handled by storing integers in "two's complement" binary form, that is taking $[0, \frac{N}{2})$ as all possible positive numbers, and $[\frac{N}{2}, N)$ as the decreasing negative integers.

B. Real Number Encoding for Homomorphic Encryption

- The Paillier encryption scheme can only encrypt, add, and multiply with integers. Due to the prevalence of real number values in sensor outputs and estimation processes, some form of encoding is required for these numbers to be encrypted.
- Real numbers, typically stored as floating-point numbers in sensor hardware, are converted to integers using the "Q" number format. A real number a can be encoded to an integer e using (16), where the largest encodable real number has an integer part of i bits. Integer bits i and fractional bits f are chosen such that the largest encoded value can still be encrypted.

$$e = \begin{cases} \lfloor 2^f a \rfloor & a < 2^i \\ \lfloor 2^f (2^i - a) \rfloor & a \geq 2^i \end{cases} \tag{16}$$

- While the encoded real numbers are consistent under addition, multiplication by constants requires that a factor of $\frac{1}{2^f}$ be removed.
- Since division is not supported under the encryption scheme, the number of multiplications performed on an encrypted value must be bounded and handled when decoding. This also decreases the size of the largest encodable real number that can be decoded correctly.
- In our case, only a single multiplication is required, and decoding of an integer e to a real number f is performed by (17).

$$a = \begin{cases} \frac{e}{2^f} & e < 2^{(i+f)} \\ \frac{e}{2^{2f}} & e \geq 2^{(i+f)} \end{cases} \tag{17}$$

C. Left-Right Order Revealing Encryption

- For the ORE scheme we have considered in particular the Left-Right encryption scheme described in [7] which will help in preventing information leakage as described in section IV.
- The key difference between this scheme and others is how numbers are compared. Left-Right encryption allows any number to be encrypted as either a "Left" or "Right" encryption, but only a "Left" encryption can be compared with a "Right" encryption. The provided function of the encryption scheme are shown in (18)
- The ORE scheme described requires a single symmetric key for the encryption of either "Left" or "Right" encryptions, which can be compared without any key.

$$\begin{aligned} encrypt^L_{ORE}(sk, x) &= \mathcal{E}^L_{ORE}(x) \\ encrypt^R_{ORE}(sk, y) &= \mathcal{E}^R_{ORE}(y) \\ compare_{ORE}(\mathcal{E}^L_{ORE}(x),\ \mathcal{E}^R_{ORE}(y)) &= cmp(x, y) \end{aligned} \tag{18}$$

IV. Secure Fast Covariance Intersection with 2 Sensors

- First we will consider the 2 sensor case of secure FCI (SecFCI), before extending it to any number of sensors in section V. We consider sensor 0 which produces the estimate $\underline{\hat{x}}_0$ and covariance $\mathbf{P}_0$, and similarly sensor 1 producing $\underline{\hat{x}}_1$ and $\mathbf{P}_1$.
- From (1) and (2) we can see that CI is particularly suited to PHE schemes. Sensor encrypted estimates and covariances can be combined additively using (14) and (15). The equations using the Paillier encrtyption systems to compute CI are given in (19) and (20).

$$\mathcal{E}(\mathbf{P}) = \mathcal{E}(\mathbf{P}_0)^{\omega_0} \mathcal{E}(\mathbf{P}_1)^{(1-\omega_0)} \tag{19}$$

$$\mathcal{E}(\mathbf{P}\underline{\hat{x}}) = \mathcal{E}(\mathbf{P}_0\underline{\hat{x}}_0)^{\omega_0} \mathcal{E}(\mathbf{P}_1\underline{\hat{x}}_1)^{(1-\omega_0)} \tag{20}$$

$$\tag{21}$$

- What remains is the computation of the parameter ω. The FCI solutions for the 2 sensor case given by (8), cannot be computed with PHE encryptions due to the required division.
- Instead we discretise ω to steps of some size $s < 1$ such that s divides 1, and use the ORE scheme to compute (7). Each sensor discretises ω and multiplies each discretisation with $\mathrm{tr}(\mathbf{P}^{-1})$ of its current coveriance $\mathbf{P}$. Each result is then encrypted with the Left-Right ORE scheme.
- Sensor 0 encrypts using the Left scheme as shown in (22) and similarly for sensor 1 using Right encryption in (23).

$$[\mathcal{E}^L_{ORE}(\omega\, \mathrm{tr}(\mathbf{P}_0^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]] \tag{22}$$

$$[\mathcal{E}^R_{ORE}(\omega\, \mathrm{tr}(\mathbf{P}_1^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]] \tag{23}$$

- The two ordered encrypted lists (22) and (23) are recieved at the fusion center, and used to estimate ω.
- To compute the FCI value for ω we want the intersection between the two lines described by $\mathrm{tr}(\mathbf{P}_0)\omega$ and $\mathrm{tr}(\mathbf{P}_1)(1-\omega)$. Note that in the 2 sensor case, the discretised list of $\mathrm{tr}(\mathbf{P}_i)(1-\omega)$ can be obtained by simply reversing the list for $\mathrm{tr}(\mathbf{P}_i)\omega$.
- Fig. 1 shows the list from sensor 0 and the reversed list of sensor 1 plotted over ω with a step size $s = 0.1$. Since values from one list can be compared with those from the other using the ORE operations described in (18), the exact intersection can be approximated in $O(log(\frac{1}{s}))$ steps by performing a binary search.

- Once the two consecutive differing comparisons from vertically aligned points in 1 are found, the FCI ω can be approximated by taking the middle value between the two bounds. This is computed simply with (24).
- In the case where the comparison function from the ORE scheme returns exact equality, the exact value of ω is known and can be taken as the approximation.

$$\omega_0' = \frac{1}{2}(a + b),\ \omega_1' = (1 - \omega_0) \tag{24}$$

V. Multi-sensor Secure Fast Covariance Intersection

- In the multi-sensor case, the same method of using PHE encryptions of each sensor's estimate and covariance is used to compute (1) and (2). Again we are left with the task of computing the weights ω_i.

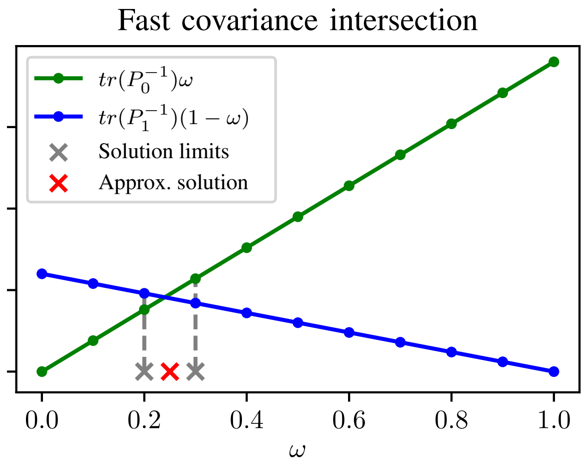

Fig. 1. Approximation of ω_0 with discretisation step size $s = 0.1$. Only comparisons of the ordered values sent from either estimator are used.

- For the ease of diagrams, we will demonstrate how this can be done in the three sensor case, and provide equations for the n sensor case.
- Computing the ω_i values for FCI in the three sensor case requires the solving of (25) and (26).

$$\omega_0 \operatorname{tr}(\mathbf{P}_0) - \omega_1 \operatorname{tr}(\mathbf{P}_1) = 0 \tag{25}$$
$$\omega_1 \operatorname{tr}(\mathbf{P}_1) - \omega_2 \operatorname{tr}(\mathbf{P}_2) = 0 \tag{26}$$

- Our method solves the multiple equations by approximating the partial solutions of each, similarly to the two sensor case, and solving the newly obtained, unencrypted, multiple equations.
- Partial solutions are treated as hyperplanes on the possible solutions space, and their intersection gives the solution to the multiple equations. We consider each ω_i a dimension, and defined hyperplane points as $(\omega_0, \omega_1, \ldots, \omega_n)$ accordingly.
- Solving (25) as was done in the two sensor case with (22) and (23), gives the partial solution of (25) when $\omega_2 = 0$ in the three sensor case. The other defining points of the partial solution of ω_0 and ω_1 are obtained when each remaining $\omega_i = 1$. In the three sensor case, this gives one aditional point; (0,0,1). In Fig. 2 the partial solution of (25) has been plotted over the solution space defined by (3) and (4) when $n = 2$.
- Next we compute the partial solution for ω_1, and ω_2 in the same way. The additional known point defining this partial solution space is $(1, 0, 0)$. Fig 3 shows both partial solutions plotted over the solution space. Note that the intersection of the partial solutions gives the approximate solution for the FCI ω values.
- The partial solutions provide $n - 1$ hyperplanes, each of dimension $n - 2$. When computing the partial solutions' intersection, we define hyperplanes of dimension $n - 1$ for each partial solution, by adding a dimension perpendicular to the solution space. This can be seen in Fig. 4.

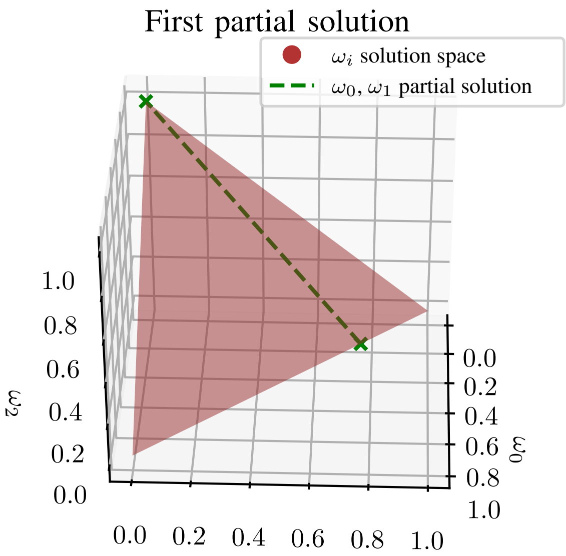

Fig. 2. Partial solution from equation (25) plotted on the plane of all possible values of ω_0, ω_1, and ω_2.

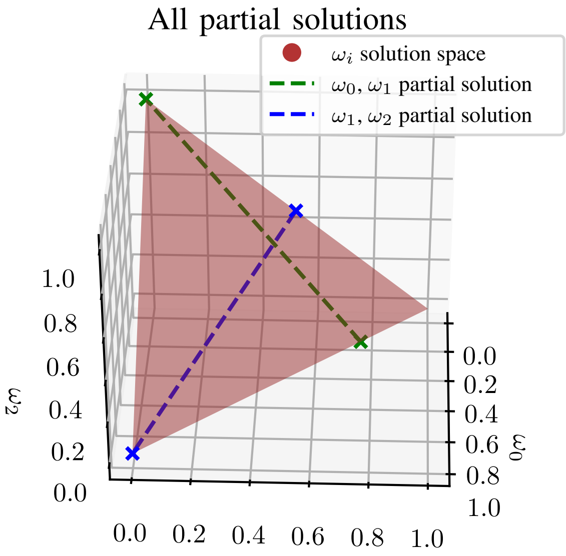

Fig. 3. Partial solutions from equations (25) and (26) plotted on the plane of all possible values of ω_0, ω_1, and ω_2.

- Each partial solution in the three sensor case is therefore now defined as a plane of the form (27), where x, y, z represent the $\omega_0, \omega_1, \omega_2$ axes.

$$a_0 x + a_1 y + a_2 z + d = 0 \tag{27}$$

- We now have n equations, one of which is the solution space equation, and n values of $'omega_i$ to solve for. In the three sensor case, the solution to the linear problem (28) provides the approximate solution to each ω_i value in FCI.

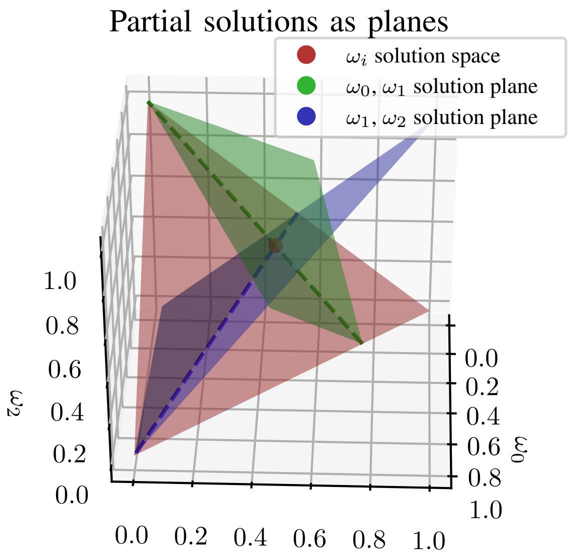

Fig. 4. Partial solutions from fig. 3 plotted as planes perpendicular to the plane of possible solutions. Intersection point gives solution values of ω_i for Fast Covariance Intersection.

$$\begin{bmatrix} a_0^0 & a_1^0 & a_2^0 \\ a_0^1 & a_1^1 & a_2^1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} d^0 \\ d^1 \\ 1 \end{bmatrix} \tag{28}$$

- In the case of n sensors, hyperplanes are defined similarly and produce the linear equation (29).

$$\begin{bmatrix} a_0^0 & a_1^0 & \cdots & a_n^0 \\ a_0^1 & a_1^1 & \cdots & a_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_0^n & a_1^n & \cdots & a_n^{n-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} d^0 \\ d^1 \\ \vdots \\ d^{n-1} \\ 1 \end{bmatrix} \tag{29}$$

- Each of the partial solutions is approximated using consecutive sensor lists encrypted with ORE. This allows “Left-Right” ORE to still be used, by alternating which sensor uses which encryption.
- The required ordered lists sent from each sensor i are described by (30).

$$\begin{aligned} &[\mathcal{E}^L_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]],\ i \text{ is even} \\ &[\mathcal{E}^R_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]],\ i \text{ is odd} \end{aligned} \tag{30}$$

VI. Simulation Results

- A simulation was implemented to demonstrate the accuracy of SecFCI fusion when compared to traditional FCI fusion.
- A constant-speed linear process model was used, with two independent cartesian sensors making white Gaussian noisy measurements of the ground truth. Both sensors ran linear Kalman Filters [] on their measurements producing a local estimate and estimate covariance.

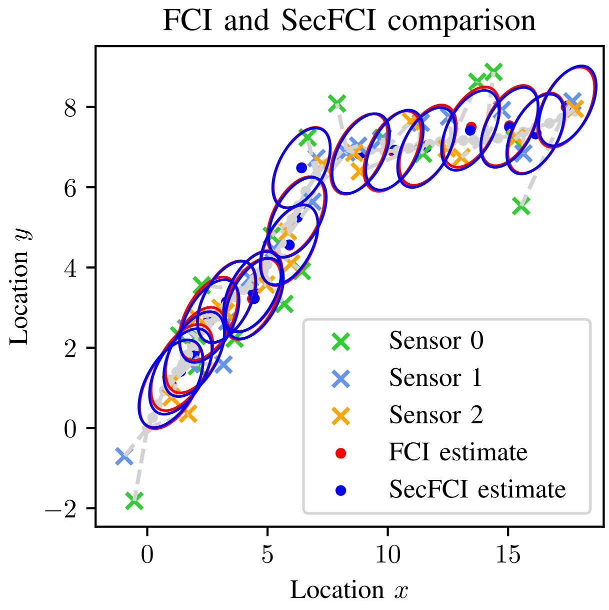

Fig. 5. Tracking simulation comparing Fast Covariance Intersection and our Secure Fast Covariance Intersection fusion methods.

- Sensors provided their estimates to a fusion center both in unencrypted form, and encrypted form. Unencrypted estimates consisted of only the estimates $\hat{\underline{x}}_i$ and $\mathbf{P}_i$, while encrypted estimates were composed of PHE encryption of the estimate in the information filter form, $\mathcal{E}(\mathbf{P}_i^{-1})$ and $\mathcal{E}(\mathbf{P}_i^{-1}\hat{\underline{x}}_i)$, and the ordered ORE encryptions defined in (30).
- The fusion center then performs FCI fusion on the unencrypted estimates and our SecFCI fusion on the encrypted information. A portion of the trajectory and fused estimates are shown in Fig. 5.
- Fig. 6 plots the resulting fused estimate covariances' traces over the course of the simulation. Approximation of exact FCI fusion results in slightly differing covariance traces as expected. In our simulation, the resulting covariance trace of the SecFCI fusion algorithm is consistently higher than that of FCI fusion. However, this is not always the case, due to FCI itself computing an approximation to the trace minimising choice of ω.
- In Fig. 7 FCI and SecFCI values for each ω_i are plotted throughout a portion of the trajectory. SecFCI values for ω_i stay constant over time as can be expected due to the discretisation of partial solution estimates.

VII. Conclusion

- Fast Covariance Intersection commonly used method for the approximation of the non-linear optimisation problem of Covariance Intersection fusion but requires the sharing of local sensor estimates with each other or a centralised fusion server. We have proposed an approximation to the FCI which can be computed

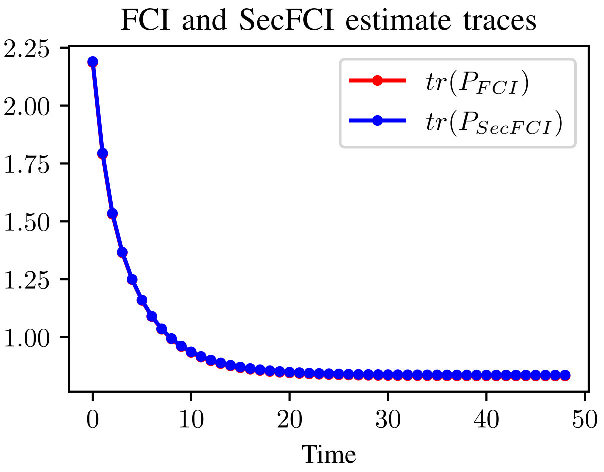

Fig. 6. Plot showing the fused estimate covariance trace throughout a tracking simulation, for both Fast Covariance Intersection and our Secure Fast Covariance Intersection

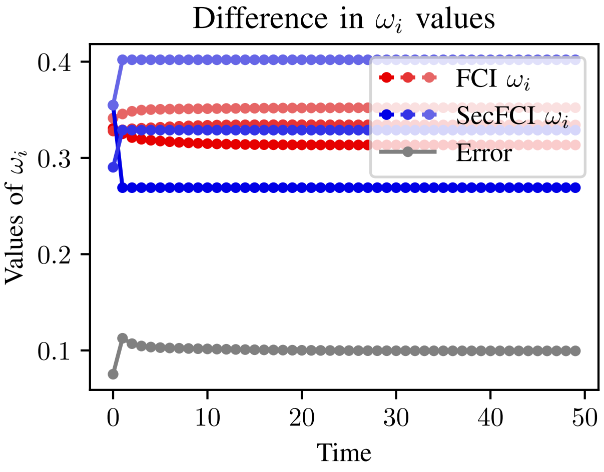

Fig. 7. Plot showing the difference in ω_i values between Fast Covariance Intersection and our Secure Fast Covariance Intersection, throughout a tracking simulation.

given only encrypted estimate and estimate covariance information from each sensor.

- Uses for a secure fusion algorithm can be found in various security-critical applications, or with untrusted networks and fusion centers.
- In future work, we would like to assess the estimate data leakage implications of ORE encrypted covariance information and produce security assumptions and proofs for the signal processing technique.
- We are also interested in how the computational performance of SecFCi may compare with alternative FHE and real number encoding techniques.

References

[1] 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.

[2] 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.–.

[3] J. Cong, Y. Li, G. Qi, and A. Sheng, "An order insensitive sequential fast covariance intersection fusion algorithm," *Information Sciences*, vol. 367-368, pp. 28–40, Nov. 2016.

[4] 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.

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

[6] 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.

[7] 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.