Encrypted Fast Covariance Intersection Without Leaking Fusion Weights

Marko Ristic[1] and Benjamin Noack[1]

***Abstract*— State estimate fusion is a common requirement in distributed sensor networks and can be complicated by untrusted participants or network eavesdroppers. We present a method for computing the common Fast Covariance Intersection fusion algorithm on an untrusted cloud without disclosing individual estimates or the fused result. In an existing solution to this problem, fusion weights corresponding to estimate errors are leaked to the cloud to perform the fusion. In this work, we present a method that guarantees no data identifying estimators or their estimated values is leaked to the cloud by requiring an additional computation step by the party querying the cloud for the fused result. The Paillier encryption scheme is used to homomorphically compute separate parts of the computation that can be combined after decryption. This encrypted Fast Covariance Intersection algorithm can be used in scenarios where the fusing cloud is not trusted and any information on estimator performances must remain confidential.**

I. Introduction

Data fusion and distributed state estimation have long been active fields of research and continue to find many applications in modern systems today [1], [2]. Methods relying on the Kalman filter and derivatives [3] have become particularly prevalent in this area due to their recursive structure and suitability to modelling estimate cross-correlations typically required for data fusion [4], [5]. The handling of these cross-correlations is especially important when consistent or optimal fusion is desired [6], [7] and presents a key challenge in data fusion when they are not known in advance. To overcome this, some methods propagate cross-correlations through time at the cost of repeated reconstruction [8], [9], [10] and typically add local computational complexity. Alternative methods approximate error cross-correlations with conservative suboptimal estimates to provide consistent fusion instead [11], [12], [13]. One such popular method is Covariance Intersection [11] and its computationally inexpensive approximation, Fast Covariance Intersection [13]. These methods minimise fusion estimate error given possible conservative estimates and are popular due to their compatibility with the information form of the Kalman filter [4], [14].

While effective and often efficient, these data fusion solutions have traditionally been performed on closed networks with trusted participants and inherently imply trust between estimators and fusers. In recent years, the ubiquity of distributed public networks has seen the additional challenges of security and privacy become increasingly relevant in distributed sensing environments and has led to an active field of research in providing security during data processing and fusion tasks [15], [16]. While one can ensure transmitted information is kept secret from eavesdroppers on untrusted networks by using common private and public key encryption schemes [17], tasks involving untrusted participants during computations present a greater challenge. Here, partial computations on encrypted data or the intended leakage of some results are often required or desired to arrive at the final result [18], [19], [20]. One common tool for achieving these requirements is homomorphic encryption [21], [22] which allows operations to be computed on encrypted numbers without decryption. The Pailler encryption scheme [22] allows homomorphic addition and is a frequent choice in fusion applications due to its applicability, simplicity and provable security. In [23], the Paillier scheme is used to perform non-Bayesian measurement fusion with range-sensors without disclosing individual sensor locations or measurements, while in [24], it is used to achieve similar security with a Kalman filter when measurements are linear and sensors form a hierarchical network. When paired with additional schemes that introduce some leakage, a wider variety of tasks can often be solved as well. The work in [25] uses private weighted sum aggregation together with the Paillier scheme to compute decentralised local control inputs

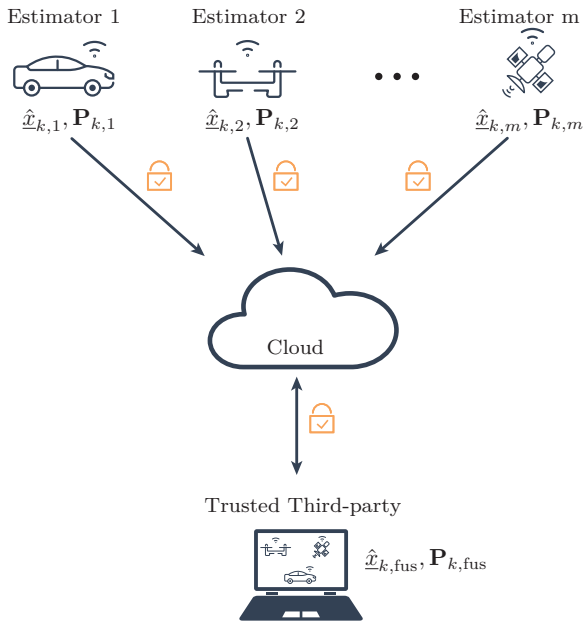

Fig. 1. The participants, inputs and output of the considered fusion scenario.

[1]Marko Ristic and Benjamin Noack are with the Autonomous Multisensor Systems Group (AMS), Institute for Intelligent Cooperating Systems (ICS), Otto von Guericke University (OVGU), Magdeburg, Germany {marko.ristic, benjamin.noack}@ovgu.de

while leaking only the sum of weighted neighbouring states to the fuser, and in [18], an untrusted cloud can use order-revealing encryptions to compute and leak relative sensor or estimator accuracies allowing the homomorphic computation of the Fast Covariance Intersection algorithm.

In this work, we consider stochastic state estimate fusion on an untrusted cloud similar to [18] but aim to fuse estimates without leaking estimator information at the cloud. Our contribution is a novel method for computing the Fast Covariance Intersection algorithm using only the Paillier encryption scheme and leaking no intermediate estimator information to the untrusted cloud. While in some use cases, leaked relative estimate accuracies can be useful for prioritising communications, this leakage inherently informs the cloud about estimator performances over time, disclosing the times when estimators are out of range or non-functioning and implicitly leaking information about the fused estimate. Therefore, the presented method finds application in a variety of cases where the Fast Covariance Intersection is applicable (measurement fusion, estimate fusion, track-to-track fusion etc.) and when all estimate-dependent information is considered sensitive to an untrusted fusing cloud.

Section II introduces the formal data fusion and security goals and section III gives relevant preliminaries. Section IV introduces our method and analyses its complexity and security, before section V presents simulation results. Concluding remarks and future work are discussed in section VI.

A. Notation

Throughout this work, the following notation is used. Plain characters, a, denote scalars, underlined lowercase characters, $\underline{b}$, denote vectors and uppercase bold characters, $\mathbf{C}$, denote matrices. $\underline{0}$ denotes the zero vector, $\mathbf{C}^{-1}$ the matrix inverse and $\mathrm{tr}(\mathbf{C})$ the matrix trace. Encryption with a public key pk is denoted $\mathcal{E}_{\mathsf{pk}}(\cdot)$ and decryption with a private key sk is denoted $\mathcal{D}_{\mathsf{sk}}(\cdot)$, while integer encoding and decoding with parameters M and ϕ are denoted $\mathsf{E}_{M,\phi}(\cdot)$ and $\mathsf{E}^{-1}_{M,\phi}(\cdot)$, respectively. All encryption and encoding operations on vectors or matrices are performed element-wise and the expression $\otimes_{i=1}^{m}\mathbf{C}_i$ is used to denote element-wise multiplication for multidimensional data. The function $\mathsf{lcm}(a_1, a_2)$ gives the lowest common denominator of integers a_1 and a_2, $\lfloor\cdot\rfloor$ denotes rounding towards zero, $\lfloor\cdot\rceil$ denotes rounding to the nearest integer and the $\log\cdot$ function is assumed to be base-2. The set $\mathbb{Z}_N$ is the set of integers modulo N and $(\mathrm{mod}\ N)$ the associated modulo-N operation.

II. Problem Statement

In this work, we consider an arbitrary time-varying process defined by its state $\underline{x}_k \in \mathbb{R}^n$ for all timesteps $k \in \mathbb{N}$. This process is estimated by m individual estimators i, $1 \leq i \leq m$, each producing a state estimate and an estimate error covariance,

$$\hat{\underline{x}}_{k,i} \in \mathbb{R}^n \text{ and } \mathbf{P}_{k,i} \in \mathbb{R}^{n\times n}\,, \tag{1}$$

respectively, at every timestep k. Our aim at every timestep is for all estimates $\hat{\underline{x}}_{k,i}$ and errors $\mathbf{P}_{k,i}$, $1 \leq i \leq m$, to be sent to a cloud server where a fused estimate and error covariance,

$$\hat{\underline{x}}_{k,\mathsf{fus}} \in \mathbb{R}^n \text{ and } \mathbf{P}_{k,\mathsf{fus}} \in \mathbb{R}^{n\times n}\,, \tag{2}$$

respectively, are computed and are consistent with the estimates in (1).

Simultaneously, we consider the desired security of the fusion process. The fusing cloud and estimators are treated as *honest-but-curious*, that is, they follow protocols correctly but may use learned information for malicious gain, and a trusted third party exists that can query the cloud at any timestep k to obtain the fusion results in (2). We aim for the cloud, estimators and eavesdroppers to learn no additional information from observed estimates and fusions, (1) and (2), respectively, beyond their local estimates. To guarantee this, cryptographic *Indistinguishability under the Chosen Plaintext Attack* (IND-CPA) [17] is desired for all transmitted and processed information. This is in line with common security goals in the field and is suitable for homomorphic encryption. The communications of this scenario are summarised graphically in figure 1.

III. Preliminaries

When proposing our solution in section IV, we make use of the Fast Covariance Intersection algorithm, the Paillier homomorphic encryption scheme and a common integer encoding for floating-point numbers. These preliminaries are summarised below.

A. Fast Covariance Intersection

The Fast Covariance Intersection (FCI) algorithm [13] provides a consistent suboptimal fusion of estimates in the form (1) when cross-correlations between the measurements are not known, providing a fast, non-iterative and general solution. Fusion is given by

$$\mathbf{P}_{k,\mathsf{fus}} = \left(\sum_{i=1}^{m}\omega_{k,i}\mathbf{P}_{k,i}^{-1}\right)^{-1} \tag{3}$$

and

$$\hat{\underline{x}}_{k,\mathsf{fus}} = \mathbf{P}_{k,\mathsf{fus}}\sum_{i=1}^{m}\omega_{k,i}\mathbf{P}_{k,i}^{-1}\hat{\underline{x}}_{k,i}\,, \tag{4}$$

with positive fusion weights $\omega_{k,i} \in \mathbb{R}^+$, $1 \leq i \leq m$, such that

$$\sum_{i=1}^{m}\omega_{k,i} = 1 \tag{5}$$

at each timestep k. In FCI, these weights are computed non-iteratively, as

$$\omega_{k,i} = \frac{1/\,\mathrm{tr}(\mathbf{P}_{k,i})}{\sum_{i=1}^{m} 1/\,\mathrm{tr}(\mathbf{P}_{k,i})}\,. \tag{6}$$

B. Paillier Encryption Scheme

The Paillier encryption scheme [22], [17] is an additively homomorphic encryption scheme meeting the IND-CPA security notion. Key generation of the scheme is performed by choosing two sufficiently large primes of an equal bit length,

p and q, and computing $N = pq$. The public key is defined by $\mathsf{pk} \coloneqq N$ and the private key by $\mathsf{sk} \coloneqq (p, q)$.

Encryption of a plaintext message $a \in \mathbb{Z}_N$, resulting in a ciphertext $\mathcal{E}_{\mathsf{pk}}(a) \in \mathbb{Z}^*_{N^2}$, is computed by

$$\mathcal{E}_{\mathsf{pk}}(a) = (N+1)^a r^N \pmod{N^2} \tag{7}$$

with a randomly chosen $r \in \mathbb{Z}_N$. The decryption of the ciphertext is computed by

$$\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(a)) = \frac{L\left(\mathcal{E}_{\mathsf{pk}}(a)^\lambda \pmod{N^2}\right)}{L\left((N+1)^\lambda \pmod{N^2}\right)} \pmod{N} \tag{8}$$

where $\lambda = \mathsf{lcm}(p-1, q-1)$ and $L(u) = \frac{u-1}{N}$. In addition to encryption and decryption, the following homomorphic properties are provided by the Paillier encryption scheme. $\forall a_1, a_2 \in \mathbb{Z}_N$,

$$\begin{aligned}\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(a_1)\mathcal{E}_{\mathsf{pk}}(a_2) &\pmod{N^2}) \\ &= a_1 + a_2 \pmod{N},\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(a_1)(N+1)^{a_2} &\pmod{N^2}) \\ &= a_1 + a_2 \pmod{N},\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(a_1)^{a_2} &\pmod{N^2}) \\ &= a_1 a_2 \pmod{N}.\end{aligned} \tag{11}$$

C. Integer Encoding for Homomorphic Encryption

As the Paillier scheme bounds inputs to $a \in \mathbb{Z}_N$, multidimensional real-valued state estimates and error covariances require a suitable integer mapping to allow for their encryption and to preserve homomorphic operations. We achieve this with element-wise encoding and encryption, and rely on a generalised Q number encoding [26]. Parametrised by a range $M \in \mathbb{N}$ and a fractional precision $\phi \in \mathbb{N}$, the encoding of a number $x \in \mathbb{R}$ is given by

$$\mathsf{E}_{M,\phi}(x) = \lfloor \phi x \rceil \pmod{M}, \tag{12}$$

while decoding of an encoded number $u \in \mathbb{Z}_M$ is given by

$$\mathsf{E}^{-1}_{M,\phi}(u) = \begin{cases} \dfrac{u \pmod{M}}{\phi}, & u \pmod{M} \leq \left\lfloor \dfrac{M}{2} \right\rfloor \\ -\dfrac{M - u \pmod{M}}{\phi}, & \text{otherwise} \end{cases}. \tag{13}$$

This encoding provides the homomorphic property

$$\begin{aligned}\mathsf{E}_{M,\phi}(a_1) + \mathsf{E}_{M,\phi}(a_2) &\pmod{M} \\ &= \mathsf{E}_{M,\phi}(a_1 + a_2)\end{aligned} \tag{14}$$

for $a_1, a_2 \in \mathbb{R}$ when

$$|\phi(a_1 + a_2)| < \left\lfloor \frac{M}{2} \right\rfloor. \tag{15}$$

Here, we note that when $M = N$, (14) coincides with the Paillier homomorphic operations (9) and (10), while when N is very large ($N > 2^{1024}$), (15) can generally be ignored.

IV. Encrypted Fast Covariance Intersection

With the problem and preliminaries introduced, we can now present our encrypted FCI method that leaks no estimator information to the fusing cloud. The core idea behind the method is to postpone the evaluation of operations that cannot be performed homomorphically until partial results are queried and decrypted by the key-holding third party. The remaining operations can then be evaluated on unencrypted inputs to produce the correct results.

First, we note that the FCI fusion equations (3) and (4) can be rearranged and substituted with weights (6) to obtain the equations

$$\mathbf{P}_{k,\mathsf{fus}} = \left(\left(\sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \right)^{-1} \sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \mathbf{P}_{k,i}^{-1} \right)^{-1} \tag{16}$$

and

$$\underline{\hat{x}}_{k,\mathsf{fus}} = \mathbf{P}_{k,\mathsf{fus}} \left(\sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \right)^{-1} \sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \mathbf{P}_{k,i}^{-1} \underline{\hat{x}}_{k,i}. \tag{17}$$

In this form, innermost summations

$$\begin{aligned}&\sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})}, \ \sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \mathbf{P}_{k,i}^{-1} \text{ and} \\ &\qquad \sum_{i=1}^{m} \frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})} \mathbf{P}_{k,i}^{-1} \underline{\hat{x}}_{k,i}\end{aligned} \tag{18}$$

combine information from individual estimators i and are computable homomorphically given suitable encryptions. Encryptions of these sums can then be decrypted by the key-holding third party, before remaining inversions and multiplications in (16) and (17) can be computed to obtain the final results. To depict this process, pseudocode for the encryption at estimators, fusion at the cloud and decryption by the third party are provided in algorithms 1, 2 and 3, respectively.

Algorithm 1 Estimator Encryption

procedure ESTIMATE(i, k, pk, ϕ)
  Estimate $\underline{\hat{x}}_{k,i}$ locally
  Estimate $\mathbf{P}_{k,i}$ locally
  ▷ Public key is encoding and encryption modulus
  $N \leftarrow \mathsf{pk}$
  ▷ Encode scaling, covariance and estimate components
  $\tilde{s}_{k,i} \leftarrow \mathsf{E}_{N,\phi}\left(\frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})}\right)$
  $\tilde{\mathbf{C}}_{k,i} \leftarrow \mathsf{E}_{N,\phi}\left(\frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})}\mathbf{P}_{k,i}^{-1}\right)$
  $\underline{\tilde{e}}_{k,i} \leftarrow \mathsf{E}_{N,\phi}\left(\frac{1}{\mathrm{tr}(\mathbf{P}_{k,i})}\mathbf{P}_{k,i}^{-1}\underline{\hat{x}}_{k,i}\right)$
  ▷ Encrypt scaling, covariance and estimate components
  $s_{k,i} \leftarrow \mathcal{E}_{\mathsf{pk}}\left(\tilde{s}_{k,i}\right)$
  $\mathbf{C}_{k,i} \leftarrow \mathcal{E}_{\mathsf{pk}}\left(\tilde{\mathbf{C}}_{k,i}\right)$
  $\underline{e}_{k,i} \leftarrow \mathcal{E}_{\mathsf{pk}}\left(\underline{\tilde{e}}_{k,i}\right)$
  Send $s_{k,i}$, $\mathbf{C}_{k,i}$ and $\underline{e}_{k,i}$ to fusing cloud
end procedure

Algorithm 2 Cloud Fusion

procedure FUSE(k, pk)
  Receive $s_{k,i}$, $\mathbf{C}_{k,i}$ and $\underline{e}_{k,i}$ for all $1 \leq i \leq m$
  $N \leftarrow \mathsf{pk}$
  $s_k \leftarrow \prod_{i=1}^{m} s_{k,i} \pmod{N^2}$
  $\mathbf{C}_k \leftarrow \otimes_{i=1}^{m} \mathbf{C}_{k,i} \pmod{N^2}$
  $\underline{e}_k \leftarrow \otimes_{i=1}^{m} \underline{e}_{k,i} \pmod{N^2}$
  Store s_k, $\mathbf{C}_k$ and $\underline{e}_k$ in case of query
end procedure

Algorithm 3 Fusion Query

procedure GETRESULT(k, pk, sk, ϕ)
  Query and receive s_k, $\mathbf{C}_k$ and $\underline{e}_k$ from fusing cloud
  $N \leftarrow \mathsf{pk}$
  ▷ Decrypt
  $\tilde{s}_k \leftarrow \mathcal{D}_{\mathsf{sk}}(s_k)$
  $\tilde{\mathbf{C}}_k \leftarrow \mathcal{D}_{\mathsf{sk}}(\mathbf{C}_k)$
  $\underline{\tilde{e}}_k \leftarrow \mathcal{D}_{\mathsf{sk}}(\underline{e}_k)$
  ▷ Decode
  $\bar{s}_k \leftarrow \mathsf{E}_{N,\phi}^{-1}(\tilde{s}_k)$
  $\bar{\mathbf{C}}_k \leftarrow \mathsf{E}_{N,\phi}^{-1}\left(\tilde{\mathbf{C}}_k\right)$
  $\underline{\bar{e}}_k \leftarrow \mathsf{E}_{N,\phi}^{-1}(\underline{\tilde{e}}_k)$
  ▷ Compute Fusion
  $\mathbf{P}_{k,\mathsf{fus}} \leftarrow \left(\bar{s}_k^{-1} \cdot \bar{\mathbf{C}}_k\right)^{-1}$
  $\underline{\hat{x}}_{k,\mathsf{fus}} \leftarrow \mathbf{P}_{k,\mathsf{fus}} \cdot \bar{s}_k^{-1} \cdot \underline{\bar{e}}_k$
  return $\underline{\hat{x}}_{k,\mathsf{fus}}$, $\mathbf{P}_{k,\mathsf{fus}}$
end procedure

Remark 1: Along with allowing the summations to be performed homomorphically on the cloud, we note that this form of the FCI also allows the cloud's partial fusion operations to be evaluated sequentially. This can be seen in algorithm 2, where individual components $s_{k,i}$, $\mathbf{C}_{k,i}$ and $\underline{e}_{k,i}$ from each estimator can continue to be aggregated as additional estimators send their estimate information. This, in turn, supports the dynamic joining and leaving of estimators in the network without affecting the cloud or the operations of a trusted third party. The security implications of such an extension are discussed further in section IV-B.

A. Complexity

The method described provides a level of security when relying on an untrusted cloud for fusing estimates. However, the added reliance on an encryption scheme and the additional computations at the third party intuitively increase the computational complexity of the algorithm and the required capabilities of participating parties. Here, we present the complexity of operations during fusion, required by each party at every timestep k. We assume encoding and decoding operations have complexity $O(1)$ (due to their comparative insignificance when compared to associated encryption and decryption operations) and use [22] to obtain the encryption complexities of the Paillier encryption scheme in table I, with security parameter $\lambda = \log N$. In table II, we compare the complexities of the unencrypted FCI algorithm and the method presented in this work. It can be seen that the burden of computation is greatly increased at the estimators and the third party, in particular when dimension n is large and when a long encryption key N is used. Naturally, an application of the proposed method would need to consider these requirements in terms of computation time and required hardware.

TABLE I
COMPUTATION COMPLEXITY OF ENCRYPTION OPERATIONS.

Operation	Complexity
Encryption	$O(\log^3 N)$
Decryption	$O(\log^3 N)$
Addition	$O(\log^2 N)$
Scalar mult.	$O(\log^3 N)$

TABLE II
COMPUTATION COMPLEXITY AT PARTIES DURING FUSION.

	FCI	Our Method
Estimator	$O(1)$	$O\left(n^2 \log^3 N\right)$
Fusion	$O(mn^3)$	$O\left(mn^2 \log^2 N\right)$
Third party	$O(1)$	$O\left(n^2 \log^3 N + n^3\right)$

B. Security Analysis

The provable security of the presented method is relatively straightforward. Our aim for IND-CPA security of all information received by the cloud, sent by the estimators or observable by eavesdroppers is achieved by the homomorphic Paillier encryption scheme. Since all transmitted information is encrypted and the cloud, estimators and eavesdroppers do not hold the secret key sk, IND-CPA is met at all parties.

We note, however, an implicit assumption made when encrypting multidimensional data element-wise. While individual elements are indistinguishable, element-wise encryption does not encrypt the estimate's dimension n, which remains implicitly public. While existing methods allow the complete homomorphic encryption of vectors [27], they are left for future work and considered beyond the scope of this work. Instead, we acknowledge the implicit leakage of n and note that, while this may leak information about the fusion's use case, state estimates remain hidden. Additionally, intuitive extensions to the scheme, such as the dynamic joining and leaving of estimators in remark 1, may introduce further implicit leakages that must be considered if security is analysed. In this example, the periodic estimation may leak to the cloud when estimators are within an estimation range or context, and a solution may be sending dummy measurements with $s_{k,i} = \mathcal{E}_{\mathsf{pk}}(\mathsf{E}_{N,\phi}(0))$ when estimator i is out of range. This

extension is presented only as an example of when care needs to be taken to maintain desired security goals, but in general, extensions and solutions are task-dependent and not a focus of this work.

V. Simulation

Fusion estimates and error covariances from the proposed encrypted FCI method differ from unencrypted FCI only when quantisation errors are large or summation overflows occur. As stated in section III-C, when the Paillier modulus N is large, these errors can often be considered negligible. In this section, we demonstrate this similarity in performance between the encrypted and unencrypted FCI fusion algorithms with a simulation. Code was written in the Python programming language, using the phe Paillier encryption scheme library [28] and a 512 bit length key (bit length of N). The simulation implements a linear constant velocity model,

$$\underline{x}_k = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \underline{x}_{k-1} + \underline{w}_k \,, \tag{19}$$

with noise term $\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q})$ and

$$\mathbf{Q} = 10^{-3} \cdot \begin{bmatrix} 0.42 & 1.25 & 0 & 0 \\ 1.25 & 5 & 0 & 0 \\ 0 & 0 & 0.42 & 1.25 \\ 0 & 0 & 1.25 & 5 \end{bmatrix} . \tag{20}$$

At each timestep k, the system state $\underline{x}_k$ is estimated by $m = 4$ estimators, $1 \leq i \leq 4$, using a standard linear Kalman filter (KF) [3] and producing estimates and error covariances $\hat{\underline{x}}_{k,i}$ and $\mathbf{P}_{k,i}$, respectively. The measurements used by the KF, $\underline{z}_{k,i}$, follow the measurement models

$$\underline{z}_{k,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \underline{x}_k + \underline{v}_{k,i} \,, \tag{21}$$

with noise terms $\underline{v}_{k,i} \sim \mathcal{N}(\underline{0}, \mathbf{R}_i)$ and covariances sampled indepedently, resulting in

$$\mathbf{R}_1 = \begin{bmatrix} 4.77 & -0.15 \\ -0.15 & 4.94 \end{bmatrix} , \; \mathbf{R}_2 = \begin{bmatrix} 2.99 & -0.55 \\ -0.55 & 4.44 \end{bmatrix} , \\ \mathbf{R}_3 = \begin{bmatrix} 2.06 & 0.68 \\ 0.68 & 1.96 \end{bmatrix} \text{ and } \mathbf{R}_4 = \begin{bmatrix} 1.17 & 0.80 \\ 0.80 & 0.64 \end{bmatrix} . \tag{22}$$

The fusion results of 1000 simulation runs are shown in figure 2. From the figure, we can see the expected similarity in performance between the encrypted and unencrypted FCI methods. Additionally, we note that the current recommended key length for the Paillier encryption scheme is 2048 bits [29], easily supporting a modulus N and fractional precision ϕ that guarantee similar performance.

VI. Conclusion

In this work, we have presented a method for computing encrypted Fast Covariance Intersection homomorphically on an untrusted cloud and discussed its security guarantees. The method ensures no estimator information leakage at the cloud, eavesdroppers or other estimators and an accompanying simulation demonstrates its minimal effect on estimation performance when compared to the unencrypted algorithm. Applications include a variety of distributed fusion tasks when external fusing computations are required such as weather forecasting and vehicle localisation. Future work on the topic aims to extend the method to include multivariable encryption, hiding the dimension variable n, and generalising to decentralised environments where individual fusing parties are untrusted.

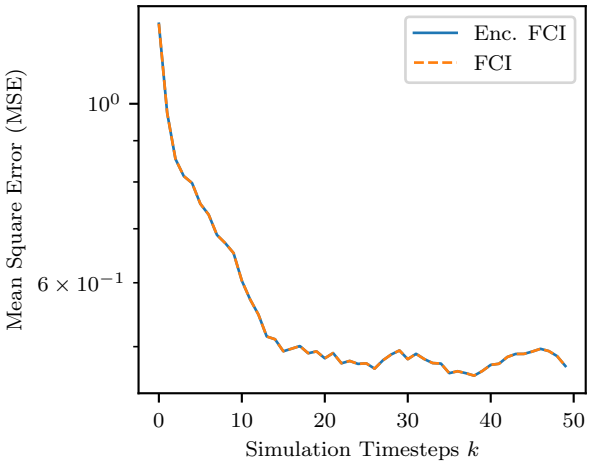

Fig. 2. Average RMSE of encrypted and unencrypted FCI fusion over 1000 simulations.

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Dover Publications, 1979.

[2] D. Simon, *Optimal State Estimation: Kalman, H Infinity and Nonlinear Approaches*. Wiley-Interscience, 2006.

[3] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons, 2012.

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

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

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

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

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

[9] S. Radtke, B. Noack, U. D. Hanebeck, and O. Straka, "Reconstruction of Cross-Correlations with Constant Number of Deterministic Samples," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1638–1645.

[10] S. Radtke, B. Noack, and U. D. Hanebeck, "Fully Decentralized Estimation Using Square-Root Decompositions," *Journal of Advances in Information Fusion*, vol. 16, no. 1, pp. 3–16, 2021.

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

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

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

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

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

[16] 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, pp. 114–119.

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

[18] M. Ristic, B. Noack, and U. D. Hanebeck, "Secure Fast Covariance Intersection Using Partially Homomorphic and Order Revealing Encryption Schemes," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 217–222, 2021.

[19] E. Shi, T.-H. H. Chan, and E. Rieffel, "Privacy-Preserving Aggregation of Time-Series Data," *Annual Network & Distributed System Security Symposium (NDSS)*, p. 17, 2011.

[20] M. Ristic, B. Noack, and U. D. Hanebeck, "Cryptographically Privileged State Estimation With Gaussian Keystreams," *IEEE Control Systems Letters*, vol. 6, pp. 602–607, 2022.

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

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

[23] A. Alanwar, Y. Shoukry, S. Chakraborty, P. Martin, P. Tabuada, and M. Srivastava, "PrOLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption," in *16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 41–52.

[24] 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.

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

[26] E. L. Oberstar, *Fixed-Point Representation and Fractional Math*. Oberstar Consulting, 2007.

[27] A. B. Alexandru and G. J. Pappas, "Private Weighted Sum Aggregation," *IEEE Transactions on Control of Network Systems*, 2021.

[28] CSIRO's Data61, "Python Paillier Library," https://github.com/data61/python-paillier, 2013.

[29] E. Barker, L. Chen, A. Roginsky, A. Vassilev, R. Davis, and S. Simon, "Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-56Br2, Mar. 2019.