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 the sensor estimate errors are leaked to the cloud in order to perform the fusion. In this work, we present a method that guarantees no leakage at 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 seperate 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 relative sensor 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] and typically add local computational complexity. Alternative methods approximate error cross-correlations with conservative suboptimal estimates to provide consistent fusion instead [9], [10], [11]. One such popular method is Covariance Intersection [9] and it computationally inexpensive approximation, Fast Covariance Intersection [11]. 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], [12].

While effective and often efficient, these data fusion solutions have traditionally been performed on closed networks with trusted participants and inherently imply a trust between sensors and estimators. 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 reserach in providing security during data processing and fusion tasks [13], [14]. While ensuring transimitted information is kept secret from eavesdroppers on untrusted networks is achievable with common private and public key encryption schemes [15], tasks involving untrusted participants during computations present a greater challenge. Here, partial computations on encrypted data or the leakage of intermediate results are often required for arriving at the final result [16], [17]. One common tool for achieving these requirements is homomorphic encryption [18], [19] which allows operations to be computed on encrypted numbers without decryption. The Pailler encryption scheme [19] allows homomorphic addition and is a frequent choice in fusion application due to its applicability, simplicity and provable security. In [20], the Paillier scheme is used to perform non-Bayesian measurement fusion with range-sensors without disclosing individual sensor locations or measurements, while in [21], it is used in to achieve similar security with a Kalman filter when measurements are linear and sensors form a heirarchical network. When paired with additional schemes that introduce some leakage, a wider variety of tasks can often be solved as well. [22] uses private weighted sum aggregation together with the Paillier scheme to compute decentralised local control inputs while leaking only the sum of weighted neightbouring states, while in [16], an untrusted cloud can use order-revealing encryptions to compute and leak relative sensor 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 [16], but aim to fuse estimates with no leakage 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 information to the untrsuted cloud. While in some use-cases, leaked relative sensor accuracies can be useful for prioritising sensor communications, this leakage inherently informs the cloud about sensor performances over time, disclosing the times when sensors are out of range or non-functioning and implicitly leaking information about the fused estimate. Therefore, the presented method finds application when all sensor information is considered sensitive and when the fusing cloud is not trusted.

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

[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

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. $\mathbf{C}^{-1}$ denotes 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 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 elementwise and the expression $\otimes_{i=1}^{m}\mathbf{C}_i$ is used to denote elementwise multiplication for multidimensional data.

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).

Simulataneously, 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) [15] is desired for all transimitted and processed information. This is inline with common security goals in the field and suitable for homomorphic encryption.

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 [11] 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 [19], [15] 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 equal bit length, p and q, and computing $N = pq$. The public key is defined by $\mathsf{pk} \triangleq N$ and the private key by $\mathsf{sk} \triangleq (p, q)$.

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 $r \in \mathbb{Z}_N$. 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 elementwise encoding and encryption, and rely on a generalised Q number encoding [23]. 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

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

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 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 evlaluated 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

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

and

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

In this form, innermost summations

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

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

Algorithm 2 Cloud Fusion

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

Algorithm 3 Fusion Query

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

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 querying third party operations.

V. Security Analysis

The provable security of the method in section IV is relatively straight forward. Our aim for IND-CPA security of all information receieved 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 eavedroppers do not hold the secret key sk, IND-CPA is met at all parties.

We do, however, note an implicit assumption that is made when encrypting multidimensional data elementwise. While individual elements are indistinguishable, elementwise encryption does not encrypt the estimate dimension n, which remains implicitly public for all computations. While some methods that allow the complete homomorphic encryption of vectors do exist [24], they are left for future work and considered beyond the scope of this work. Instead, we acknowledge the implicit leakage of state dimension n and note that, while this may leak information about the fusion's use-case, state estimate themselves remain hidden.

VI. Simulation and Results

Fusion estimates and error covariances from the encrypted FCI method in section IV differ from unencrypted FCI only when quantisation errors are large or summation overflow occurs. 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 the similarilty in performance between the encrypted and unencrypted FCI fusion algorithms with a simulation written in the Python programming language, using the phe Paillier encryption scheme library [25] 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 for each simulation. Here, sampling of covariances was performed by uniformely sampling scaling and correlation components, $s_x\,,\ s_y \sim \mathcal{U}(0, 6)$ and $c \sim \mathcal{U}(0, \pi/2)$, respectively, and setting

$$\mathbf{R}_i = \begin{bmatrix} \cos c & -\sin c \\ \sin c & \cos c \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} \cos c & \sin c \\ -\sin c & \cos c \end{bmatrix} . \tag{22}$$

The fusion results of 1000 simulation runs are shown in figure . From the figure, we

VII. Conclusion

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. 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.

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

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] C. Data61, “Python paillier library,” https://github.com/data61/python-paillier, 2013.