Privacy-Preserving Distributed Graph Filtering

Li, Qiongxiu; Coutino, Mario; Leus, Geert; Christensen, Mads Græsbøll

Published in:
28th European Signal Processing Conference (EUSIPCO)

DOI (link to publication from Publisher):
10.23919/Eusipco47968.2020.9287429

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Privacy-Preserving Distributed Graph Filtering

Qiongxiu Li[1], Mario Coutino[2], Geert Leus[2], Mads Græsbøll Christensen[1]
[1]Audio Analysis Lab, CREATE, Aalborg University, Denmark,
[2]Circuits and Systems Group, Delft University of Technology, The Netherlands,

***Abstract*—With an increasingly interconnected and digitized world, distributed signal processing and graph signal processing have been proposed to process its big amount of data. However, privacy has become one of the biggest challenges holding back the widespread adoption of these tools for processing sensitive data. As a step towards a solution, we demonstrate the privacy-preserving capabilities of variants of the so-called distributed graph filters. Such implementations allow each node to compute a desired linear transformation of the networked data while protecting its own private data. In particular, the proposed approach eliminates the risk of possible privacy abuse by ensuring that the private data is only available to its owner. Moreover, it preserves the distributed implementation and keeps the same communication and computational cost as its non-secure counterparts. Furthermore, we show that this computational model is secure under both passive and eavesdropping adversary models. Finally, its performance is demonstrated by numerical tests and it is shown to be a valid and competitive privacy-preserving alternative to traditional distributed optimization techniques.**

***Index Terms*—distributed computation, distributed graph filters, encryption, graph signal processing, privacy-preserving**

I. Introduction

Modern systems routinely gather large-scale data from different individuals and then draw inferences from these data. To this end, graph signal processing (GSP) has been put forth and proven effective for processing large amounts of networked data by exploiting their inherent structural information [1]. However, to process these networked data in a distributed manner, data exchange among different nodes is required. As the underlying data usually contains sensitive information about each individual node/agent in the network, the data exchanges may raise privacy concerns for example: 1) insecure communication channels which expose the data to eavesdroppers; 2) the trust issues that appear when individuals agree to participate in a distributed computation but are reluctant to reveal their own data to others. In fact, as shown in [2], the identities of individuals and their data are inseparable. For example, with only an anonymous 10-ride bus ticket, the identity of a specific bus passenger can be revealed [3]. Therefore, developing efficient techniques for processing large volumes of data in a privacy-friendly manner is nowadays required, posing new challenges that need to be overcome.

The insecure communication channel concern is usually resolved by assuming securely encrypted channels [4]. The trust issue, on the other hand, is particularly challenging as inferences on the exchanged data are required and classical channel encryption is insufficient. Existing privacy-preserving algorithms for solving such trust issues can be broadly classified into two classes based on established security models: computational and information-theoretical. The first type comprises computationally secure algorithms which ensure privacy through the assumption of computational hardness; that is, the malicious adversary is assumed to be computationally limited thus the secrets cannot be reconstructed efficiently. These algorithms usually adopt popular techniques like homomorphic encryption (HE) [5] and garbled circuit (GC) [6] from secure multiparty computation (SMPC) [7] to achieve privacy-preserving data aggregation [8], [9]. The privacy of the nodes/agents is protected as all the data are first encrypted and then the computations are conducted in the encrypted domain. However, these techniques are usually computationally demanding, and are thus hard to apply in practice.

On the other hand, the information-theoretical security model addresses the privacy issues through an information theory point of view. Distinctly from the computational hardness assumption, it assumes a computationally unlimited adversary and that privacy is achieved only if the information obtained by the adversary just leads to a slightly better (or the same) posterior guess of the private data compared to the prior. Information-theoretical security is usually achieved by obfuscating the private data through noise insertion, it is thus computationally lightweight and has been used in various fields through different noise insertion methods, e.g., zero-sum noise insertion for distributed average consensus [10]–[12]; differentially private Kalman filtering [13]; secret sharing based recursive least squares [14]; subspace noise insertion using distributed optimization [15], [16]. However, these algorithms suffer from a heavy communication overhead as a large number of iterations is required for convergence.

As a first step towards providing both computational and communication efficient privacy-preserving networked data processing, in this paper, we focus on addressing the privacy issues in the context of distributed graph filtering, the building block of GSP. As the conventional distributed graph filtering contains mainly two parts: offline learning conducted by a trusted third party (TTP) and online distributed computation by network nodes, we, therefore, propose a complete framework to avoid possible privacy abuse in both the offline and online steps. In the offline step, the TTP receives only the encryption seeds from the nodes and not their private data; while in the online step, an information-theoretical security model is achieved through noise insertion, which protects the private data of each node from being revealed to others in the network.

II. Distributed Processing Over Networks

Consider an N-dimensional signal $\boldsymbol{x}$ residing on a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of N nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of M edges. Further, let $\mathcal{N}_i = \{j | (i,j) \in \mathcal{E}\}$ denote the neighbourhood of the i-th node. The goal of distributed processing over networks is to compute a transformation, $\mathcal{H}$, of the networked data, $\boldsymbol{x}$, i.e.,

$$\boldsymbol{y} = \mathcal{H}(\boldsymbol{x}), \tag{1}$$

in a distributed manner; that is, only employing local data exchanges, i.e., data exchanges among neighboring nodes.

Although $\mathcal{H}$ can take many forms, e.g., optimization problems [17], linear or non-linear transforms [18], here, we focus on transformations that are linear, or that can be properly approximated by a linear transformation, i.e.,

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{2}$$

such as the consensus operation, i.e., $\boldsymbol{H} = \mathbf{1}\mathbf{1}^\top$, which are pervasive in typical network processing tasks, e.g., denoising [19], interpolation [20]. A popular way to implement (or approximate) $\boldsymbol{H}$ is to express it as a K-th order polynomial of a matrix representation of $\mathcal{G}$, i.e.,

$$\boldsymbol{H}_{\rm c} \triangleq \sum_{k=0}^{K} \phi_k \boldsymbol{S}^k, \tag{3}$$

where $\{\phi_k \in \mathbb{R}\}_{k=0}^K$ are the filter coefficients; and $\boldsymbol{S}$ is the so-called graph shift operator in GSP [1]. Common choices of $\boldsymbol{S}$ are the weighted graph adjacency matrix $\boldsymbol{W}$ and the graph Laplacian matrix $\boldsymbol{L}$. By construction, $\boldsymbol{S}$ is an $N \times N$-symmetric matrix for undirected graphs and carries the notion of frequency in the graph setting through its eigen decomposition [1], [21]. The matrix polynomial in (3) is typically referred to as a classical (node-invariant) FIR graph filters. Note that by making use of the local structure of $\boldsymbol{S}$, the computation of (3) can be implemented in a distributed way [22], [23], where each node can locally compute the k-th shift of $\boldsymbol{x}$ by exchanging its previous shifted versions within its neighbourhood, i.e., $\boldsymbol{S}^k\boldsymbol{x} = \boldsymbol{S}(\boldsymbol{S}^{k-1}\boldsymbol{x})$. This distributed implementation is particularly beneficial in saving the communication overhead and computational complexity because a FIR graph filter of order K incurs in a cost of $\mathscr{O}(MK)$.

Though the computational and communication cost scales linearly with K, a large K is usually required to obtain a high approximation accuracy. To alleviate this issue, the constrained edge-variant (CEV) FIR graph filters [24] were proposed to approximate the desired graph filter with a lower order K by endowing it with more degrees of freedom, i.e.,

$$\boldsymbol{H}_{\rm cev} \triangleq \sum_{k=0}^{K} \boldsymbol{\Phi}_k \boldsymbol{S}^k, \tag{4}$$

where the coefficient matrices $\{\boldsymbol{\Phi}_k \in \mathbb{R}^{N\times N}\}_{k=0}^K$ denote the weights that each node assigns to its edges at shift k, and they share the support with $\boldsymbol{S} + \boldsymbol{I}_N$, where $\boldsymbol{I}_N$ is the $N \times N$-identity matrix. It is easy to see that (4) reduces to the classical FIR graph filter if $\boldsymbol{\Phi}_k = \phi_k \boldsymbol{I}_N$. And it also reduces to the node-variant FIR graph filter [23] if $\{\boldsymbol{\Phi}_k\}_{k=0}^K$ are diagonal matrices. Note that while the CEV FIR filters (4) still enjoy a distributed implementation and the communication and computational cost remains $\mathscr{O}(MK)$, they allow for a richer family of linear operators that can be approximated as they are not restricted to linear operators that commute with the graph shift operator.

III. Privacy-Preserving Processing

In this section, we first highlight the privacy concerns of distributed data processing using current FIR graph filters and introduce the adversary models. The latter is an important issue when designing privacy-preserving protocols. We then formally state the problem addressed in this work.

A. Privacy concern

Generally, privacy is associated with individuals and their personal data. Thus, we identify the privacy concern in graph filtering is to protect the input graph signal x_i of each node from being revealed. This is motivated by the fact that this data usually contains sensitive information about the individual like health condition or political views. Unfortunately, this privacy concern is not addressed in current distributed FIR graph filters, since the private data x_i is propagated through the network and exposed to neighbors and eavesdroppers.

B. Adversary model

To evaluate the robustness of a system under different security attacks, we consider the so-call adversary model. An adversary can be both external or internal to the network, and aims to conduct certain malicious activities such as inferring the private data by controlling a number of nodes. These controlled nodes are referred to as corrupted nodes and the others are called non-corrupted or honest nodes. In this paper we will address two widely-used adversary models: eavesdropping and passive. The eavesdropping adversary attempts to infer the private data by eavesdropping the communication channels in the network. The passive adversary, on the other hand, assumes that all the nodes follow the protocol instructions and aims also to infer the private data of honest nodes through the information collected by all the corrupted nodes.

C. Problem formulation

Putting things together, we conclude that a privacy-preserving distributed graph filter should be able to protect private data while computing the filter output. To this end, we formulate the problem as follows; *given a linear transform $\boldsymbol{H}$, design an encryption function $E(\boldsymbol{x})$ and related operator $\boldsymbol{H}^e$ that satisfy the following two requirements simultaneously:*

- *Output correctness*: all nodes should be able to achieve the correct filtering output, i.e.,
$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} = \boldsymbol{H}^e\boldsymbol{x}^e, \tag{5}$$
where $\boldsymbol{x}^e = E(\boldsymbol{x})$ denotes the encrypted input data.
- *Individual privacy*: after encryption, each node uses the encrypted data x_i^e as its input. In addition, the information theoretic security criterion
$$I(X_i; X_i^e) < \epsilon_i, \forall i \in \mathcal{N}, \tag{6}$$

must be met. Here, $I(\cdot)$ denotes mutual information [25]; x_i^e and x_i are assumed to be a realization of random variables X_i^e and X_i, respectively, and ϵ_i is the desired privacy level of node i. Hence, (6) guarantees that the encrypted x_i^e reveals asymptotically no information about the private data x_i,

IV. Proposed approach

Before moving to our proposed method, let us first consider the following straightforward design for both $E(\boldsymbol{x})$ and $\boldsymbol{H}^e$. The privacy of each node can be preserved by considering the following rendition of the edge-variant graph filter [c.f. (4)], i.e.,

$$\boldsymbol{H}_{\text{rev}} = \sum_{k=0}^{K} \boldsymbol{S}^k \boldsymbol{\Phi}_k. \tag{7}$$

Here, each node exploits the graph filter coefficients as encryption seeds to mask its private data and sends the masked data to its neighbours. Note that (4) and (7) are equivalent when $\boldsymbol{\Phi}_k \boldsymbol{S} = \boldsymbol{S}\boldsymbol{\Phi}_k \ \forall k$ and that the graph filter (7) can also be implemented distributively. However, there is a price to pay: instead of having a communication cost $\mathcal{O}(KM)$, the implementation now has an overall communication cost of $\mathcal{O}(K^2M)$. Unfortunately, for this case, although the above-mentioned requirements can be achieved, there seems to be no free lunch and communication complexity has to be sacrificed.

To alleviate the communication overhead, we now proceed to introduce another choice for both $E(\boldsymbol{x})$ and $\boldsymbol{H}^e$ meeting the above-mentioned requirements. We further analyze their performance in two widely-used adversary models.

A. Encryption function design

One typical way to maintain privacy is to obfuscate the private data by inserting noise since it is computationally lightweight. We, therefore, propose to design the encryption function using multiplicative noises as

$$E(\boldsymbol{x}) = \text{diag}(\boldsymbol{e})\boldsymbol{x}, \tag{8}$$

where $\boldsymbol{e} = [e_1, \ldots, e_N]^T \in \mathbb{R}^N$ is the encryption vector. Thus the encrypted data of node i is given by $x_i^e = e_i x_i$. Let e_i denote a realization of random variable E_i having differential entropy $h(E_i)$, assuming it exists[1]. The concerned individual privacy for node i is thus $I(X_i; X_i E_i) = h(X_i) - h(X_i | X_i E_i)$. To guarantee privacy, we need the following result.

Proposition 1. *Consider X and Y as independent continuous random variables with mean $\mu(X), \mu(Y) < \infty$ and variance $0 < \text{var}(X) < \text{var}(Y) < \infty$, and let $Z = XY$. Then*

$$\lim_{\text{var(Y)}\to\infty} I(X; Z) = 0,$$

assuming $I(X; Z)$ exists.

Proof. As X is independent with Y, we have $\text{var}(Z) = \text{var}(X)\text{var}(Y) + \text{var}(X)\mu^2(Y) + \text{var}(Y)\mu^2(X)$. Let $\gamma = 1/(\text{var}(Z))^{\frac{1}{2}}$ and define $Z' = \gamma Z$. Hence, Z' has unit variance. We have $I(X; Z) = I(\gamma X; Z')$ as mutual information is scale-invariant. Thus,

$$\lim_{\text{var(Y)}\to\infty} I(X; Z) = \lim_{\gamma\to 0} I(\gamma X; Z') = I(0; Z') = 0.$$

□

So, applying Proposition 1 to the problem at hand, we have

$$\lim_{\text{var}(E_i)\to\infty} I(X_i; X_i E_i) = 0. \tag{9}$$

We conclude that each node is able to achieve arbitrary small information loss by increasing the variance of its encryption seed. Hence, the privacy requirement in (6) is satisfied.

B. Encrypted operator design

To fullfil the output correctness requirement, we should design the encrypted operator $\boldsymbol{H}^e$ satisfying $\boldsymbol{H}^e = \boldsymbol{H}\,\text{diag}(\boldsymbol{e})^{-1}$. To this end, we then further approximate the desired encrypted operator $\boldsymbol{H}^e$ using standard distributed graph filters. Here, we use the constrained edge-variant graph filter as an example, which in turn leads to the following convex optimization problem

$$\begin{aligned} \min_{\{\boldsymbol{\Phi}_k\}} \quad & \left\| \boldsymbol{H}^e - \left(\sum_{k=0}^{K} \boldsymbol{\Phi}_k \boldsymbol{S}^k \right) \right\|_F^2 \\ \text{s.t.} \quad & \text{supp}\{\boldsymbol{\Phi}_k\} = \text{supp}\{\boldsymbol{S} + \boldsymbol{I}_N\} \ \forall k \in [K], \end{aligned} \tag{10}$$

where $\|\cdot\|_F$ is Frobenius norm and $\text{supp}\{\cdot\}$ denotes the support of its argument. To solve the above-mentioned problem, we assume a TTP (cloud/server) to learn the filter coefficients offline. In order to avoid possible data abuse, we have the following assumptions which minimize the amount of information available to each node while still guaranteeing the distributed implementation, i.e.,

Assumption 1. *(Knowledge of the TTP) The TTP has the knowledge of the desired data transformation $\boldsymbol{H}$, the encryption seeds $\boldsymbol{e}$, the graph shift operator $\boldsymbol{S}$ and the filter coefficients $\{\boldsymbol{\Phi}_k\}_{k=0}^K$.*

Assumption 2. *(Knowledge of the nodes) Each node i only has knowledge of its private data x_i, and its corresponding entries of both $\boldsymbol{S}$ and filter coefficients, i.e., $[\boldsymbol{S}]_{i,j\in\mathcal{N}_i\cup\{i\}}$, $\{[\boldsymbol{\Phi}_k]_{i,j\in\mathcal{N}_i\cup\{i\}}\}_{k=0}^K$.*

Note that Assumptions 2 is motivated similarly as the distributed signal processing [17], [26] that each node only has local knowledge of its neighbourhood. That is, each node does not have the knowledge of the desired data transformation $\boldsymbol{H}$. For example, for the distributed average consensus application, i.e., $\boldsymbol{H} = \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$, all the nodes would like to reach an agreement over the network but they do not know the size of the network, i.e., N. In the distributed recursive least squares application [14], each node only has its local observations but not the full knowledge of the whole system. After the offline learning, the filter output can be computed distributedly. It is worth to note that the communication and computational cost remains $\mathcal{O}(KM)$. The proposed approach is summarized in Algorithm 1.

[1]If E_i is a discrete random variable, the conditions are given in terms of the Shannon entropy $H(E_i)$.

Algorithm 1 Privacy-preserving distributed FIR graph filters

Offline learning (secure channels)
Each node $i \in \mathcal{N}$ chooses its encryption seed $\boldsymbol{e}_i$ based on its desired privacy level ϵ_i, and sends it to the TTP.
The TTP first computes $\boldsymbol{H}^e = \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{e})^{-1}$ and solves the problem (10), and then sends the corresponding information $[\boldsymbol{S}]_{i,j\in\mathcal{N}_i\cup\{i\}}, \{[\boldsymbol{\Phi}_k]_{i,j\in\mathcal{N}_i\cup\{i\}}\}_{k=0}^K$ to node i.
Online distributed computation (non-secure channels)
Each node i initializes $x_i^{(0)} = x_i^e$.
while $k = 0, \ldots, K$ **do**
  Collect $x_j^{(k)}$ from all neighbours $j \in \mathcal{N}_i$
  Store locally $z_i^{(k)} = \sum_{j\in\mathcal{N}_i\cup\{i\}} [\boldsymbol{\Phi}_k]_{ij} x_j^{(k)}$
  Compute $x_i^{(k+1)} = \sum_{j\in\mathcal{N}_i\cup\{i\}} [\boldsymbol{S}]_{ij} x_j^{(k)}$
  Send $x_i^{(k+1)}$ to all neighbours $j \in \mathcal{N}_i$
end while
Set $y_i = \sum_{k=0}^K z_i^{(k)}$

C. Privacy analysis under adversary models

We now analyze the individual privacy concern under both an eavesdropping and a passive adversary. For an eavesdropping adversary, we assume that the communication required in offline learning is conducted through secure channels. That is, the channels should be securely encrypted [4] when transmitting the encryption seeds to the TTP and receiving the associated filter coefficients and graph shift operator. As a consequence, the secure channel encryption cost is $2N$. The online distributed computation step, as it is an iterative process, we remark that $\forall k > 0 : X_i \to X^{(0)} \to X^{(k)}$ forms a Markov chain where vector $X^{(k)} = [X_1^{(k)}, \ldots, X_N^{(k)}]^\top$ denotes all random variables over the network at iteration k. By the data processing inequality [25] we have

$$I(X_i; X^{(0)}) \geq I(X_i; X^{(k)}),\ \forall k > 0, \tag{11}$$

which implies that all the information transmitted in the online step will not reveal additional information. As a consequence, the online step is secure against eavesdropping adversaries without requiring secure channel encryption at all. For the case of a passive adversary, recall Assumptions 1-2, we can see that the corrupted nodes cannot reconstruct the encryption seeds $\boldsymbol{e}$ as $\boldsymbol{H}$ is only known to the TTP. We then conclude that the private data of an honest node is guaranteed even if all other nodes are corrupted (assuming the TTP is not corrupted).

V. Numerical results

We now proceed to present the performance of the proposed approach for approximating user-provided frequency responses and compare it with the distributed optimization technique. We randomly generate a community graph using the GSP toolbox [27] with $N = 40$ nodes and choose the normalized Laplacian as the graph shift operator $\boldsymbol{S}$. Here, we consider as maximum filter order $K = 15$. To show the performance of approximating different frequency responses, we consider two cases commonly used in the GSP community:

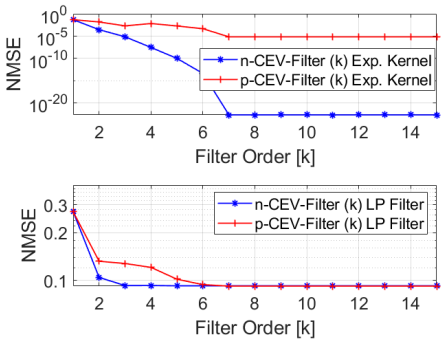

Fig. 1. Approximation error comparison between the p-CEV and the n-CEV graph filter for different orders. (Top) Results for the exponential kernel. (Bottom) Results for an ideal low-pass filter

1) the exponential kernel
$$h(\lambda) := e^{-\gamma(\lambda-\mu)^2},$$
where γ denotes the spectrum decaying factor and μ is the central parameter;
2) the ideal low pass filter
$$h(\lambda) = \begin{cases} 1 & 0 \leq \lambda \leq \lambda_c \\ 0 & \text{otherwise} \end{cases},$$
where λ_c denotes the cutoff frequency.

In Fig. 1, we demonstrate the normalized approximation error in terms of Frobenius norm between the desired, $\boldsymbol{H}$, and the fitted, $\boldsymbol{H}_{\text{fit}}$, frequency response, i.e., $\|\boldsymbol{H} - \boldsymbol{H}_{\text{fit}}\|_F^2 / \|\boldsymbol{H}\|_F^2$, of the proposed privacy-preserving CEV (p-CEV) filter with the non-private CEV (n-CEV) filter for both scenarios. As we can see, both filters saturate at an order of $K = 7$ but the proposed p-CEV has a higher error floor for the case of the exponential kernel. And the proposed p-CEV requires a few orders more to achieve the same error floor as the n-CEV for approximating the ideal low pass filter. Overall, we conclude that the proposed approach is able to approximate the desired frequency responses and keeps its private data protected.

In Fig. 2, we compare the normalized error as a function of iteration number between desired filter output $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$ and the approximated one $\boldsymbol{y}_{\text{fit}} = \boldsymbol{H}_{\text{fit}}\boldsymbol{x}$ of the proposed p-CEV graph filter and the privacy-preserving alternative using distributed optimization in the average consensus application. In particular, we choose to compare with the privacy-preserving primal-dual method of multipliers (p-PDMM) proposed in [15] as it also achieves information-theoretical security by noise insertion. In both algorithms, we set the noise variance as 100 times the variance of the associated private data, thereby guaranteeing a similar amount of noise perturbation. As we can see, the noise insertion will lead to a high initial error for p-PDMM therefore requiring more iterations to converge. To provide insights into the question on *what is the difference between GSP and distributed optimization* from the privacy-preserving perspective, we compare these two approaches in terms of several important parameters in Table I:

1) Both approaches obtain the information-theoretical security model and consider the same adversary models: passive and eavesdropping.

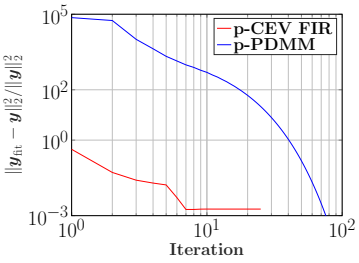

Fig. 2. Normalized output error versus the number of iterations of the proposed p-CEV filter and the p-PDMM for the average consensus problem

2) Perfect approximation is not possible and thus the proposed approach is not as accurate as the p-PDMM, while the price to pay for the accuracy is the communication cost. The iteration number T of the p-PDMM is usually far larger than the filter order K of the proposed approach, i.e., $T \gg K$.
3) Both approaches require secure channels in the initialization step to guarantee the inserted noise cannot be eavesdropped, but the complexity of the proposed approach is usually lower, i.e., $M > N$.
4) The proposed approach assumes that the trusted third party is not corrupted to guarantee the privacy of each node while the p-PDMM requires each honest node has one honest neighbour.

We conclude that the proposed approach is beneficial in solving distributed signal processing tasks in terms of both communication and computational cost. More specifically, the proposed approach is preferred if the required accuracy is not very strict, while the p-PDMM is more attractive if the system requires strictly accurate solution, but with a price of a higher communication cost.

TABLE I
COMPARISON WITH DISTRIBUTED OPTIMIZATION

	Proposed	p-PDMM [15]
Security model	Information-theoretic	Information-theoretic
Adversary model	Passive/Eavesdropping	Passive/Eavesdropping
Accuracy	Approximate	Accurate
Communication complexity	$\mathcal{O}(MK)$	$\mathcal{O}(MT)$
Secure channels	$\mathcal{O}(N)$	$\mathcal{O}(M)$
Honest neighbour	No	Yes
Trusted Third Party	Yes	No

VI. CONCLUSIONS

In this paper, we proposed a communication and computationally efficient solution to achieve privacy-preserving distributed graph filters which allow each node to compute its desired output and protect its own private data simultaneously. To protect the privacy, each node first inserts noise to obfuscate its private data and sends the obfuscated data to its neighborhood. Numerical tests demonstrated that the proposed approach is able to approximate some desired graph frequency responses with a small filter order and that it is a competitive alternative compared to the privacy-preserving distributed optimization approach in the distributed average consensus problem.

REFERENCES

[1] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644-1656,*, 2013.
[2] D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE Signal Process. Magazine, vol. 30, no. 5, pp. 86–94*, 2013.
[3] G. Avoine, L. Calderoni, J. Delvaux, D. Maio, and P. Palmieri, "Passengers information in public transport and privacy: Can anonymous tickets prevent tracking?," *Int. J. Inf. Manage, vol. 34, pp. 682-688*, 2014.
[4] D. Dolev, C. Dwork, O. Waarts, M. Yung, "Perfectly secure message transmission," *J. Assoc. Comput. Mach., vol. 40, no. 1, pp. 17-47,*, 1993.
[5] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology–CRYPTO, pp. 643–662*. Springer, 2012.
[6] A. C. Yao, "Protocols for secure computations," in *FOCS, pp. 160–164*, 1982.
[7] R. Cramer, I. B. Damgrd, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*, Cambridge University Press, 2015.
[8] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy-preserving distributed speech enhancement for wireless sensor networks by processing in the encrypted domain," in *ICASSP, pp. 7005–7009*, 2013.
[9] M. H. Ruan, M. Ahmad, Y. Q. Wang, "Secure and privacy-preserving average consensus," in *Proc. Workshop Cyber-Phys. Syst. Secur. Privacy, pp. 123-129*, 2017.
[10] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Automat Contr., vol. 62, no. 2, pp. 753–765*, 2017.
[11] J. He, L. Cai, C. Zhao, P. Cheng, X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw., vol. 5, no. 1, pp. 127–138*, 2019.
[12] N. Gupta, J. Kat and N. Chopra, "Statistical privacy in distributed average consensus on bounded real inputs," in *ACC, pp 1836-1841*, 2019.
[13] K. H. Degue and J. L. Ny, "On differentially private kalman filtering," in *Proc. IEEE Global Conf. Signal Inf. Process., pp. 487-491*, 2017.
[14] K. Tjell, I. Cascudo and R. Wisniewski, "Privacy preserving recursive least squares solutions," in *ECC, pp.3490–3495*, 2019.
[15] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *ICASSP, pp. 5895-5899*, 2020.
[16] Q. Li, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: a general framework," in *arXiv preprint arXiv: 2004.13999*, 2020.
[17] T. Sherson, W. B. Kleijn, R. Heusdens, "A distributed algorithm for robust lcmv beamforming," in *ICASSP, pp. 101-105*, 2016.
[18] Lopes, Cassio G and Sayed, Ali H, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Trans. Signal Process., vol. 56, no. 7, pp. 3122-3136,*, 2008.
[19] J. Pang, G. Cheung, A. Ortega, O. C. Au, "Optimal graph laplacian regularization for natural image denoising," in *ICASSP, pp 2294-2298*, 2015.
[20] SK Narang, A Gadde, A Ortega, "Signal processing techniques for interpolation in graph structured data," in *ICASSP, pp 5445-5449*, 2013.
[21] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Magazine, vol. 30, no. 3, pp. 83-98,*, vol. 30, no. 3, pp. 83–98, 2013.
[22] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *DCOSS, pp 1–8*, 2011.
[23] S. Santiago, M. Antonio G and R. Alejandro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process., vol. 65, no. 15, pp. 4117-4131*, 2017.
[24] M. Coutino, E. Isufi, G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process., vol. 67, no. 9, pp. 2320-2333,*, 2019.
[25] T. M. Cover and J. A. Tomas, *Elements of information theory*, John Wiley & Sons, 2012.
[26] T. Sherson, R. Heusdens, W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE Trans. Signal Inf. Process. Netw., vol. 5, no. 2, pp 334-347*, 2018.
[27] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, 2014.