On the Privacy-Robustness Trade-off in Distributed Average Consensus

Zarè Palanciyan
Delft University of Technology
the Netherlands
z.palanciyan@student.tudelft.nl

Qiongxiu Li
Aalborg University
Denmark
qili@es.aau.dk

Richard Heusdens
Netherlands Defence Academy
Delft University of Technology
the Netherlands
r.heusdens@tudelft.nl

***Abstract*—Distributed consensus algorithms face a dual challenge in modern networked systems: safeguarding sensitive data through privacy-preserving mechanisms while maintaining robustness against adversarial nodes (e.g., Byzantine faults). While prior work addresses these goals separately, their interplay remains poorly understood, particularly in scenarios where output accuracy must be preserved. In this work, we reconcile these objectives by integrating a subspace perturbation framework, which guarantees privacy by confining noise to redundant network subspaces, with a median absolute deviation (MAD)-based thresholding mechanism to detect active adversarial nodes transmitting corrupted data. Through in-depth analysis, we demonstrate that enhancing privacy via subspace perturbation inherently limits the discriminative power of MAD-based detection, as adversarial updates become statistically indistinguishable from privacy-preserving perturbations. Numerical simulations quantify this tension, demonstrating that as privacy guarantees strengthen, the ability to detect active adversaries diminishes. These findings highlight a core challenge in distributed consensus—achieving both strong privacy and Byzantine robustness simultaneously is inherently difficult.**

***Index Terms*—Primal-dual method of multipliers (PDMM), privacy, subspace perturbation, adversary, detection, median absolute deviation (MAD), privacy-robustness trade-off**

I. Introduction

In recent years, a rising trend is gaining traction which aims to develop a wide range of techniques to perform distributed computations [1]. These techniques enable collaborative data processing across decentralised nodes without a centralised coordinator. For example, this can be seen in applications ranging from wireless sensor networks [2], optimisation [3], and federated learning [4]. Distributed optimisation algorithms such as the alternating direction method of multipliers (ADMM) [5] and the primal-dual method of multipliers (PDMM) [6]–[8] have gained popularity in distributed frameworks. As these algorithms are applied across more fields, they increasingly handle sensitive data, making privacy protection vital [9]. Traditional approaches, such as differential privacy (DP) [10], [11] and secure multiparty computation (SMPC) [12], aim to address these concerns by protecting sensitive data but often do so at the cost of accuracy or increased computational complexity [13], [14]. To overcome these limitations, new methods have been developed that maintain both accuracy and computational efficiency. One such approach is the subspace perturbation framework introduced in [15]–[17] and its variants [18]–[20].

Privacy is not the only concern in decentralised networks. In addition to data being extracted from the network, corrupt data can also be injected into a distributed system [21]. Various types of attacks can introduce corrupt data, such as backdoor attacks in a federated learning environment [22], [23]. Additionally, attacks can be designed to prevent the network from converging to the optimal value. This can be achieved, for example, through random Gaussian attacks [21] or by transmitting malicious data to poison the network [24]. To counter these attacks, various robust detection algorithms have been developed, such as the Krum algorithm [25]. Here, the node calculates the proximity of its neighbours and the similarity of their transmitted data. It then selects the node with the smallest distance to the others as the true update. Another method, called Kardam [26], computes the Lipschitz coefficients of its neighbours and accepts data from those neighbours whose values fall within an acceptable range around the median of the Lipschitz coefficients. Another approach, proposed in [27], detects corrupt nodes by calculating at every time instant the normalised difference of transmitted data among neighbours and determining the maximum deviation from the median. Neighbours of which the averaged distance to the median exceeds a certain threshold are identified as being malicious. When implementing a decentralised network, both privacy and adversarial robustness must be considered. In a distributed network, the optimal solution can be achieved when the appropriate algorithm is used and all nodes share the same goal. However, in the presence of an attack, the network may diverge from its optimal output if robust detection algorithms are not in place. Detection algorithms evaluate and distinguish nodes based on the data they transmit. By comparing individual data updates to the collective behaviour of neighbouring nodes, adversarial nodes can be identified and flagged. However, if the network achieves perfect secrecy, nodes may become indistinguishable from one another, undermining the fundamental principle on which detection algorithms assess corruption.

In this paper, we investigate a fundamental trade-off between privacy preservation and adversarial detection in distributed average consensus algorithms. Specifically, we demonstrate that integrating both privacy-protecting mechanisms and attack detection capabilities creates an inherent conflict: enhancing privacy preservation (e.g., through noise injection) can inadvertently diminish the system's ability to identify malicious or compromised nodes. Our analysis reveals that as the level of privacy preservation increases, the detection accuracy for adversarial behaviour declines accordingly, highlighting a critical design challenge for secure and privacy-aware consensus frameworks. We further consolidate these findings through numerical simulations, which quantify the trade-off and validate our theoretical claims.

The paper is organised as follows. In Section II, we define the privacy preservation methods used and describe the adversarial models. Section III presents the problem setup, the network attack model, and the metrics used to demonstrate the privacy-robustness trade-off. In Section IV, we introduce the proposed detection algorithm for active adversarial nodes. Section V provides numerical validation of our findings, and finally, Section VI presents our conclusions.

II. Preliminaries

We present a simple undirected connected graph G as $G = (\mathcal{V}, \mathcal{E})$, where the set of nodes in the network is represented by $\mathcal{V} = \{1, 2, \ldots, n\}$ and the set of edges is represented by $\mathcal{E} = \{e_1, \ldots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$. The neighbourhood of node i is denoted as the set $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$. The degree of node i is then given by $d_i = |\mathcal{N}_i|$. The distributed average consensus algorithm aims to calculate the average of the local data each node holds

$$\mathbf{s}_{\text{ave}} = \frac{1}{n} \sum_{i \in \mathcal{V}} \mathbf{s}_i \tag{1}$$

where $\mathbf{s}_i \in \mathbb{R}^q$ is the local data each node holds and q the dimension of the local data.

A. A/PDMM approach of solving average consensus

The solution to (1) can be achieved in a distributed network by implementing PDMM and reformulating the overall setup problem as follows:

$$\begin{aligned} \min_{\{\mathbf{x}_i : i \in \mathcal{V}\}} \quad & \sum_{i \in \mathcal{V}} f_i(\mathbf{x}_i) \\ \text{subject to} \quad & \forall (i,j) \in \mathcal{E} : \mathbf{B}_{i|j}\mathbf{x}_i + \mathbf{B}_{j|i}\mathbf{x}_j = \mathbf{0}, \end{aligned} \tag{2}$$

where $f_i(\mathbf{x}_i) = \frac{1}{2}||\mathbf{x}_i - \mathbf{s}_i||_2^2$ and $\mathbf{B}_{i|j} \in \mathbb{R}^{q \times q}$ is defined as $\mathbf{B}_{i|j} = \mathbf{I}_q$ if $i < j$, and $\mathbf{B}_{i|j} = -\mathbf{I}_q$ otherwise, where $\mathbf{I}_q$ denotes the $q \times q$ identity matrix. As shown in [28], problem (2) can be solved using A/PDMM. This leads to the following set of update equations for each node:

$$\mathbf{x}_i^{(t+1)} = \arg\min_{\mathbf{x}_i} \left(f_i(\mathbf{x}_i) + \sum_{j \in \mathcal{N}_i} \mathbf{z}_{i|j}^{(t)\top} \mathbf{B}_{i|j}\mathbf{x}_i + \frac{\rho d_i}{2} \|\mathbf{x}_i\|^2 \right),$$

$$(\forall j \in \mathcal{N}_i) \quad \mathbf{y}_{i|j}^{(t+1)} = \mathbf{z}_{i|j}^{(t)} + 2\rho \mathbf{B}_{i|j}\mathbf{x}_i^{(t+1)}, \tag{3}$$

$$(\forall j \in \mathcal{N}_i) \quad \mathbf{z}_{j|i}^{(t+1)} = (1-\theta)\mathbf{z}_{j|i}^{(t)} + \theta \mathbf{y}_{i|j}^{(t+1)}, \tag{4}$$

where θ is the averaging parameter and ρ controls the convergence rate. For $\theta = \frac{1}{2}$ (Douglas-Rachford splitting), $\frac{1}{2}$-averaged PDMM is achieved, which is equivalent to the classical ADMM algorithm [28].

B. Privacy

There are various privacy-preserving distributed average consensus algorithms proposed, such as DP-based [29]–[31], SMPC based [32]–[34] and subspace-perturbation based approaches [15]–[17]. In this paper, we deploy subspace perturbation to achieve privacy preservation, motivated by its efficiency and flexibility; it has been shown to achieve similar privacy guarantees to SMPC and DP-based approaches under specific parameter configurations [20]. The implementation of subspace perturbation is straightforward: it involves sampling the initialised auxiliary variable $\mathbf{z}^{(0)} \in \mathbb{R}^{mq}$ from a high-variance noise distribution to protect local node data through statistical obfuscation. This is achieved by splitting the space into two subspaces, the convergent subspace and the non-convergent subspace. Privacy is imposed by perturbing the non-convergent subspace with noise, while perturbations in the non-convergent subspace do not affect the output accuracy. As shown in [17], perfect secrecy can asymptotically be achieved this way, by introducing finite variance in the noise in the non-convergent subspace of the auxiliary variable $\mathbf{z}^{(0)}$.

C. Adversarial models

In this work, we consider three types of nodes that can exist within the network. The first type is the honest node, which follows the averaging process as intended and does not attempt to infer the local data of other nodes. The other two types are adversarial nodes: passive adversarial nodes (also known as honest-but-curious) and active adversarial nodes. Passive adversarial nodes follow the protocol's instructions similar to honest nodes but attempt to infer as much information as possible, in collaboration with other passive adversarial nodes, about the local data s_i of the honest nodes. In contrast, active adversarial nodes seek to disrupt the network by transmitting arbitrary updates based on their malicious intentions, which could cause the network to diverge or converge to a malicious point.

Let $\mathcal{V}_h$ denote the set of honest nodes and $\mathcal{V}_c$ the set of adversarial nodes such that $\mathcal{V} = \mathcal{V}_h \cup \mathcal{V}_c$ and $\mathcal{V}_h \cap \mathcal{V}_c = \emptyset$. In addition, let $\mathcal{V}_{c,p}$ denote the set of passive adversarial nodes and $\mathcal{V}_{c,a}$ the set of active adversarial nodes so that $\mathcal{V}_c = \mathcal{V}_{c,p} \cup \mathcal{V}_{c,a}$ and $\mathcal{V}_{c,p} \cap \mathcal{V}_{c,a} = \emptyset$.

III. Problem Definition

The local data utilised in private distributed networks for which PDMM can be applied to, should not be revealed to outsiders or adversarial nodes. To solve this issue, one must implement privacy-preserving frameworks in their network. The works in [17] [20] [15] have shown that it is possible to implement such a framework in PDMM and to achieve perfect secrecy, without sacrificing the output correctness of the network. However, our findings in this work show there is another trade-off which occurs when perfect secrecy is achieved. We found that higher levels of privacy in a network for the private local data of its nodes come at the expense of detecting adversarial nodes that corrupt their local data with malicious intent. This trade-off highlights the difficulty in balancing between privacy and adversarial node detection.

A. Active adversarial attack

An adversarial model can utilise multiple different attacks to reach its goal [35]. However, in this work, we will only focus on a single attack, in which the adversarial node corrupts its local data to make the network converge to a non-optimal point.

B. Trade-off metrics

To discuss the trade-off between the effectiveness of privacy-preserving techniques and adversarial detection methods, four metrics will be compared.

a) Information-theoretical privacy metric: measures the mutual information between the private data and all information available to the adversary. Let $\mathcal{O}$ denote the set of information obtained by the adversary and $\mathcal{X}_i$ the private data of node i. The mutual information $I(X_i; \mathcal{O})$ is given by

$$I(X_i; \mathcal{O}) = H(X_i) - H(X_i|\mathcal{O}) \leq H(X_i), \tag{5}$$

where $H(\cdot)$ denotes the (Shannon) entropy. If $I(X_i; \mathcal{O}) = 0$, the adversary cannot gain any information about the private data by observing $\mathcal{O}$. If $I(X_i; \mathcal{O}) = H(X_i)$, the adversary has complete knowledge about the private data. Thus, higher mutual information indicates greater potential privacy leakage.

b) Output correctness: measures the distance of the output of the network to the optimal solution. This is assessed by taking the mean square error (MSE) between the $\mathbf{x}_i$-values and the optimal solution $\mathbf{x}^*$, given by $\frac{1}{n}||\mathbf{x}_i^{(t)} - \mathbf{x}^*||^2$.

c) False alarm rate (FAR): measures whether nodes implementing the detection algorithm are misclassifying honest nodes as active adversarial nodes. Let $D(i,j)$ denote the number of times a node has been identified of being malicious within a time frame of L samples. The FAR is defined as $\mathit{FAR}(k) = \frac{1}{|\mathcal{E}_H|}\sum_{(i,j)\in\mathcal{E}_H} \mathbb{I}(D(i,j)(kL) > \frac{L}{2})$, where $\mathcal{E}_H \subseteq (\mathcal{V} \setminus V_{c,a}) \times (\mathcal{V} \setminus V_{c,a})$, $k \in \mathbb{N}$, and $\mathbb{I}$ is the indicator function.

d) Rate of misdetection (MDR): measures whether nodes implementing the detection algorithm are misclassifying active adversarial nodes as honest nodes. This is quantified as follows: $\mathit{MDR}(k) = \frac{1}{|\mathcal{E}_A|}\sum_{(i,j)\in\mathcal{E}_A} \mathbb{I}(D(i,j)(kL) < \frac{L}{2})$, where $\mathcal{E}_A \subseteq (\mathcal{V} \setminus V_{c,a}) \times V_{c,a}$.

IV. Method of Detection

The method of detection of active adversarial nodes in this paper is based on the work of [27]. Here, the method of determining whether a node is an active adversary or not is defined with the transmitted variable $\mathbf{y}_{\cdot|i}$ node i receives from its neighbours. The following assumptions are made to implement the detection algorithm.

Assumption 1: The amount of active adversarial neighbours a node has is less than half of the total amount of its neighbours. Hence, $\frac{d_i}{2} > |\{j \in \mathcal{N}_i \cap \mathcal{V}_{c,a} \mid (i,j) \in \mathcal{E}\}|$.

Assumption 2: The graph G remains connected even when the node-set $\mathcal{V}_{c,a}$ is removed.

Algorithm 1 Detection and Mitigation

```
Input: Threshold scaling $\alpha$, Segment length $L$.
Set $D(i,j) = 0$ for each $i,j \in V$.
for $t = 1, 2, \ldots$ do
    for all $i \in \mathcal{V}$ do
        for each agent $j \in N_i$ do
            Compute $\Delta Y_{i,j}(t), \delta_i$ according to (6), (7)
        end for
        if $\Delta Y_{i,j}(t) > \delta_i$ then
            Increase $D(i,j)$.
        end if
        if $t \equiv 0 \pmod{L}$ then
            if $D(i,j) > \frac{L}{2}$ then
                Node $i$ ignores[1] the updates of $j$ for the next L
                iterations, and stops sending updates to $j$.
            else
                Node $i$ continues using the updates of node $j$.
            end if
            Set $D(i,j)$ = 0.
        end if
    end for
end for
```

If an adversarial node were to steer away from the objective of the honest nodes by corrupting its local data $\mathbf{s}_i$, its transmitted variable $\mathbf{y}_{i|\cdot}$ would have a larger distance to the transmitted variables of other honest nodes. In the case of PDMM, because of the minus-sign difference between the $\mathbf{y}$ variables, the absolute value of $\mathbf{y}_{i|j}$ is taken. Leveraging Assumption 1, the median of the data of neighbouring nodes will be the data value of one of the honest neighbours.

Let $\mathbf{m}_i$ denote the median of the neighbouring data of node i, given by

$$\mathbf{m}_i = \text{med}\{|\mathbf{y}_{i|j}| : j \in \mathcal{N}_i\},$$

and let $\Delta Y_{i,j}$ be defined as

$$\Delta Y_{i,j} = \||\mathbf{y}_{i|j}| - \mathbf{m}_i\|_\infty, \quad j \in \mathcal{N}_i. \tag{6}$$

The values of $\Delta Y_{i,j}$ will then be compared to the scaled median absolute deviation (SMAD), given by

$$\text{SMAD}_i = \alpha\, \text{med}\{||\mathbf{y}_{i|j}| - \mathbf{m}_i|_\infty : j \in \mathcal{N}_i\}, \tag{7}$$

where α is a scaling factor. This threshold determines whether a node is corrupt or not. Again, leveraging Assumption 1, the SMAD is a powerful method of determining a threshold because it compares the distance other neighbours have from a neighbour that is guaranteed to be honest.

Utilising Assumption 1 and Assumption 2, the nodes can flag and isolate an adversarial neighbour using Algorithm 1 while also being able to achieve the optimal solution in the network for the objective function with its constraints. The method of detection would be implemented in PDMM between

[1] The utilisation of the updates of node j stops, but node i keeps receiving the updates for the next test, but does not acknowledge it as its neighbour.

(3) and (4), to determine whether or not it should use the update or acknowledge the neighbour.

The argument of the trade-off between privacy preservation and adversarial detection can be made with any detection method. This is a fundamental trade-off, as stronger privacy measures inherently reduce the ability to gather information for adversarial detection. As explained previously, if the mutual information equals zero, then independent of any detection method, no information about the private data of the adversarial nodes can be gained.

V. Simulations

In this section, we will present the simulation results to show the trade-off between achieving higher levels of privacy and the ability to detect active adversarial nodes in the network. Here we simulated a distributed network by generating a random geometric graph (RGG) with $n = 50$ nodes and a communication radius of $r = \sqrt{\frac{2\log(n)}{n}}$, as this ensures that the graph is connected [36]. The data will be scalar-valued ($q = 1$). We will introduce a single corrupt node b in the graph, for which its local data will be $s_b = 10^4$, this will make the average converge to a non-optimal point if not removed. The other nodes will have local data which is generated with a Gaussian distribution around a mean of 25 with a variance of 30 ($s_i \sim \mathcal{N}(25, 30)) : \forall i \in (\mathcal{V} \setminus V_{c,a})$. First, for $\theta = \frac{1}{2}$ (ADMM), we set $\rho = 1, \alpha = 13, L = 5$, as for the case of $\theta = 1$ (PDMM), we set $\rho = 1, \alpha = 10, L = 2$. Figure 1 and 2 show the FAR, MDR and MSE output correctness, with the implementation of the privacy-preserving framework and the detection algorithm for ADMM and PDMM, respectively.

A. False alarm rate

The top subplot of the figures corresponds to the FAR. It is observed that even as the noise variance in $z^{(0)}$ increases, the FAR remains low, indicating how infrequently honest nodes are incorrectly flagged as adversarial nodes. This is the case for both ADMM and PDMM.

B. Rate of misdetection

The middle subplot shows the MDR, which decreases slower with higher noise variance. The high MDR shows that the detection algorithm often misses the adversarial node when the noise variance in $z^{(0)}$ is large. For $\sigma^2 = 10^4$ it is seen that for ADMM the MDR converges to 0, but for PDMM the MDR does not converge.

C. Output correctness

The bottom subplot shows the MSE, demonstrating that when the MDR is nonzero, $MDR \neq 0$, the MSE remains high because the adversarial node b can still skew the network's output for both ADMM and PDMM. Thus, while increased noise in $z^{(0)}$ preserves privacy, it also hinders the detection algorithm's ability to detect the adversarial node, revealing the trade-off and validating our theoretical claims.

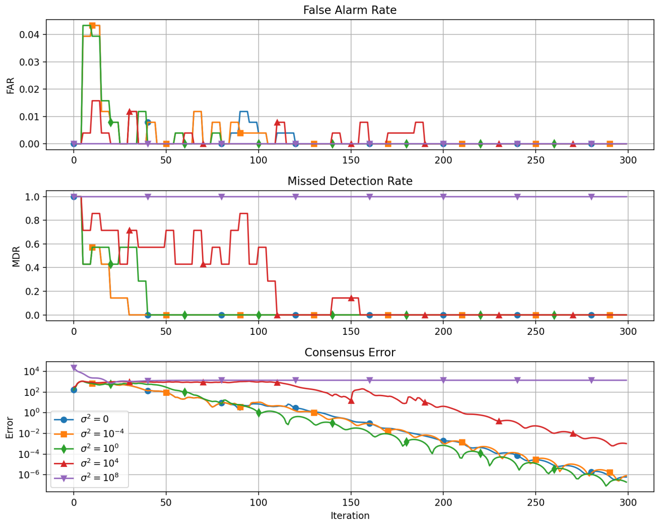

Fig. 1. ADMM simulation with different variance levels for the noise in $z^{(0)}$.

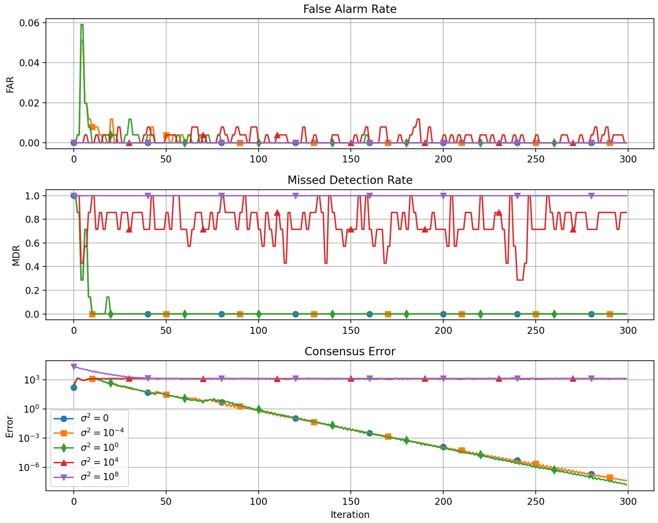

Fig. 2. PDMM simulation with different variance levels for the noise in $z^{(0)}$.

VI. Conclusion

In this paper, we presented a hybrid approach for detecting active adversarial nodes while utilising a privacy-preserving framework for the PDMM algorithm. We have shown that when the data of nodes achieves perfect secrecy, it becomes impossible for any detection algorithm to detect active adversarial nodes in distributed average consensus algorithms. Therefore, when the mutual information converges to zero ($I(X_i; \mathcal{O}) = 0$), no information about the private data can be inferred, making the nodes indistinguishable from one another with respect to one another's private data. Numerical results under various settings further consolidate our claims.

References

[1] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *Automatic Control, IEEE*

Transactions on, vol. 54, 2009.

[2] Z. Chen, M. Dahl, and E. G. Larsson, “Decentralized learning over wireless networks: The effect of broadcast with random access,” in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 316–320.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine learning, vol. 3, no. 1, pp. 1–122*, 2011.

[6] G. Zhang and R. Heusdens, “Distributed optimization using the primal-dual method of multipliers,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, 2017.

[7] T. W. Sherson, R. Heusdens, and W. B. Kleijn, “Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory,” *IEEE transactions on signal and information processing over networks*, vol. 5, no. 2, pp. 334–347, 2018.

[8] R. Heusdens and G. Zhang, “Distributed optimisation with linear equality and inequality constraints using pdmm,” *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[9] M. Mageshwari and R. Naresh, “Decentralized data privacy protection and cloud auditing security management,” in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2022, pp. 103–109.

[10] C. Dwork, F. McSherry, K. Nissim, A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proc. Theory of Cryptography Conf. , pp. 265-284*, 2006.

[11] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[12] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[13] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private distributed convex optimization via functional perturbation,” *IEEE Trans. Control Netw. Syst., vol. 5, no. 1, pp 395-408*, 2018.

[14] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, “Privacy-preserving distributed processing: Metrics, bounds, and algorithms,” in *IEEE Trans. Inf. Forensics Secur.*, vol. 16, 2021, pp. 2090–2103.

[15] Q. Li, R. Heusdens, and M. G. Christensen, “Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5895–5899.

[16] Q. Li, R. Heusdens and M. G. Christensen, “Convex optimization-based privacy-preserving distributed least squares via subspace perturbation,” in *Proc. Eur. Signal Process. Conf.*, 2020.

[17] Q. Li, R. Heusdens, and M. G. Christensen, “Privacy-preserving distributed optimization via subspace perturbation,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 5983–5996, 2020.

[18] S. O. Jordan, Q. Li, and R. Heusdens, “Privacy-preserving distributed optimisation using stochastic PDMM,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2024, pp. 8571–8575.

[19] Q. Li, R. Heusdens, and M. G. Christensen, “Communication efficient privacy-preserving distributed optimization using adaptive differential quantization,” *Signal Process.*, vol. 194, p. 108456, 2022.

[20] Q. Li, J. S. Gundersen, M. Lopuhaä-Zwakenberg, and R. Heusdens, “Adaptive differentially quantized subspace perturbation (adqsp): A unified framework for privacy-preserving distributed average consensus,” *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1780–1793, 2024.

[21] X. Dong, Z. Wu, Q. Ling, and Z. Tian, “Byzantine-robust distributed online learning: Taming adversarial participants in an adversarial environment,” *IEEE Transactions on Signal Processing*, vol. 72, pp. 235–248, 2024.

[22] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, 2020, pp. 2938–2948.

[23] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, “Poison frogs! targeted clean-label poisoning attacks on neural networks,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, p. 6106–6116.

[24] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, “Detection of false data injection attacks in smart grid: A secure federated deep learning approach,” *IEEE Transactions on Smart Grid*, vol. PP, 2022.

[25] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017.

[26] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, “Asynchronous Byzantine machine learning (the case of SGD),” in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 1145–1154.

[27] O. Shalom, A. Leshem, and A. Scaglione, “Localization of data injection attacks on distributed m-estimation,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 655–669, 2022.

[28] T. W. Sherson, R. Heusdens, and W. B. Kleijn, “Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 334–347, 2019.

[29] M. Kefayati, M. S. Talebi, H. R. Rabiee and B. H. Khalaj, “Secure consensus averaging in sensor networks using random offsets,” in *Proc. of the IEEE Int. Conf. on Telec., and Malaysia Int. Conf. on Commun., pp. 556–560*. IEEE, 2007.

[30] Z. Huang, S. Mitra, and G. Dullerud, “Differentially private iterative synchronous consensus,” in *ACM workshop Privacy electron. Soc., pp. 81–90*, 2012.

[31] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design,” *Automatica, vol. 81, pp. 221–231*, 2017.

[32] R. C. Hendriks, Z. Erkin, and T. Gerkmann, “Privacy preserving distributed beamforming based on homomorphic encryption,” in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2013.

[33] Q. Li and M. G. Christensen, “A privacy-preserving asynchronous averaging algorithm based on shamir’s secret sharing,” in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[34] Q. Li, I. Cascudo, and M. G. Christensen, “Privacy-preserving distributed average consensus based on additive secret sharing,” in *Proc. Eur. Signal Process. Conf., pp. 1-5*, 2019.

[35] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, “Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets,” *AAAI*, vol. 33, no. 1, pp. 1544–1551, 2019.

[36] J. Dall and M. Christensen, “Random geometric graphs,” *Phys. Rev. E*, vol. 66, p. 016121, 2002.