Abstract—We propose a decentralized optimization algorithm that preserves the privacy of agents’ cost functions without sacrificing accuracy, termed EFPSN. The algorithm adopts Paillier cryptosystem to construct zero-sum functional perturbations. Then, based on the perturbed cost functions, any existing decentralized optimization algorithm can be utilized to obtain the accurate solution. We theoretically prove that EFPSN is $(\epsilon, \delta)$-differentially private and can achieve infinitesimally small $\epsilon, \delta$ under deliberate parameter settings. Numerical experiments further confirm the effectiveness of the algorithm.

Index Terms—Optimization methods, distributed algorithms, differential privacy.

I. INTRODUCTION

THE PROBLEM of optimizing a global objective function through the cooperation of multiple agents has gained increased attention in recent years [1], [2]. This is driven by the wide applicability of the problem to many engineering and scientific domains, ranging from cooperative control, distributed sensing, multi-agent systems, and sensor networks to large-scale machine learning; see, e.g., [3], [4], [5], [6].

In this letter, we consider a peer-to-peer network of $n$ agents that cooperatively solves:

$$\min_{x \in \mathbb{X}} F(x) = \frac{1}{n} \sum_{i \in \mathcal{N}} f_i(x),$$

where $x$ is the common decision variable, $\mathbb{X}$ and $\mathcal{N}$ denote the domain and the collection of all the agents.

Current gradient-based distributed optimization algorithms require agents to explicitly share optimization variables and/or gradient estimates in every iteration. This is problematic in applications involving sensitive data. Zhu et al. [7] showed that obtaining private data from shared gradients is possible, and the recovery is pixelwise accurate for images and token-wise matching for texts. Consequently, we wish to solve (1) in a private way.

For machine learning problems, one common form of the individual function is $f_i(x) \triangleq \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[l_i(x, \xi_i)]$, where $\mathcal{D}_i$ is the local data distribution of agent $i$, and $\xi_i$ is a data sample or a batch of data samples. In such a case, each function $f_i$ contains information on the data distribution $\mathcal{D}_i$, which is usually sensitive. It is thus crucial to keep the objective functions private. More specifically, by privacy, we refer to keeping all $f_i$ from being inferred by adversaries. In this letter, we consider two types of adversaries:

- The eavesdropper: an external adversary having access to all information transmitted through the communication channels within the network.
- The curious-but-honest adversary: an external adversary that corrupts a subset of the agents. The adversary knows all the information of every corrupted agent $i$, including all the information within agent $i$, e.g., the individual function $f_i$ and all the information passed from its neighbor to agent $i$. However, each corrupted agent obeys the optimization protocol precisely.

Plenty of efforts have been reported to counteract privacy breaches in distributed optimization. Privacy preserving algorithms either process the transmitted messages between agents or change the cost functions. Differential Privacy (DP) [8], [9] was introduced to the context of distributed optimization in [10], [11], [12], [13]. The DP-based methods can be categorized into message-based [10], [11], [12] and function-based ones [13]. The message-based methods inject noises to the messages each agent sends, while the latter adds functional noises into each agent’s cost function $f_i$.

However, the direct combination of DP and distributed optimization suffers from the accuracy-privacy trade-off. Huang et al. [12] observed that the accuracy level of the...
TABLE I
CATEGORIZATION OF EXISTING PRIVACY-PRESERVING DISTRIBUTED OPTIMIZATION ALGORITHMS

| Paper Index | [10], [11], [12] | [13] | [14], [15] | [16] | Ours |
|-------------|-----------------|------|-----------|------|------|
| Message-based | ✔              |      | ✔         |      |      |
| Function-based | ✔            |      | ✔         |      |      |
| Structured-noise | ✔        |      | ✔         |      |      |
| Encryption | ✔              |      | ✔         |      |      |
| DP-based | ✔              |      | ✔         |      |      |

obtained solution is in the order of $O\left(\frac{1}{\epsilon^2}\right)$, where $\epsilon$ is the privacy budget that is inversely proportional to privacy. The papers [14], [15] adopted encryption techniques to preserve privacy in distributed optimization. Nevertheless, since the methods must encrypt in each iteration and it is common to have thousands of iterations in practice, they fall victim to the heavy communication and computation overhead.

Adding structured noise [16] is a workaround for the accuracy-privacy trade-off and the heavy overhead mentioned above. This letter [16] constructed a set of zero-sum Gaussian noises to perturb the affine terms of the objective functions that are assumed to be quadratic. Nonetheless, the method fails under an eavesdropping attack due to the naive communication method. Besides, the privacy analysis in [16] is carried out under a self-defined privacy framework. We provide a categorization of the current privacy-preserving distributed optimization algorithms in Table I.

In this letter, we integrate the encryption-based scheme and the structured noise method under the functional DP framework. The proposed new method, termed the Encrypted Functional Perturbation with Structured Noise algorithm (EFPSN), enjoys the benefits of several previous methods. In particular, EFPSN adopts the Paillier encryption scheme [17], a homomorphic public-key cryptosystem, to construct zero-sum noises among the agents secretly. The noises are subsequently used to generate functional perturbations for each agent’s cost function. Such a procedure differs from those in [14], [15] and bypasses the heavy communication and computation overhead caused by encryption at every iteration. With the perturbed cost functions, any decentralized optimization algorithms in Table I can be utilized to obtain the accurate solution to problem (1) thanks to the structured noises. We further prove that EFPSN is $(\epsilon, \delta)$-differentially private and can achieve infinitesimally small $\epsilon, \delta$ under deliberate parameter settings. In other words, as long as the curious-but-honest attacker does not corrupt all the neighbors of an agent, EFPSN can provide an arbitrarily strong privacy guarantee without sacrificing accuracy. Finally, we use EFPSN to optimize non-convex functions to illustrate its efficacy.

II. PRELIMINARIES

This section introduces the notation, graph-related concepts and some background knowledge on Hilbert space since generating functional perturbations relies on the orthonormal systems in some Hilbert space.

A. Notation

We use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{R}^d$ the Euclidean space of dimension $d$. The space of scalar-valued infinite sequences is denoted by $\mathbb{R}^N$. Let $\mathbb{Z}, \mathbb{Z}_{\geq 0}$ be the set of integers and positive integers, respectively. Given $w \in \mathbb{Z}_{\geq 0}$ and $\mathbb{Z}_w \triangleq \{0, 1, \ldots, w\}$, $\mathbb{Z}_w$ denotes the set of positive integers which are smaller than $w$ and coprime with $w$. Let $l_2 \subset \mathbb{R}^N$ be the space of infinite square summable sequences. For $D \subseteq \mathbb{R}^d$, $L_2(D)$ denotes the set of square-integrable measurable functions over $D$. $\mathbf{1}$ denotes a column vector with all entries equal to 1. A vector is viewed as a column vector unless otherwise stated. We use $\|\cdot\|$ to denote the Euclidean norm for a vector.

We use $\mathbb{P}[A]$ to denote the probability of an event $A$, $Pr(x)$ the probability distribution function of random variable $X$. For an encryption scheme, $\text{Enc}()$, $\text{Dec}()$ represent the encoder and decoder respectively. $N(\mu, \sigma^2)$ is the (multivariate) Gaussian distribution with (vector) mean $\mu$ and variance (covariance matrix) $\sigma^2$. $N^\dagger$ is the degenerate Gaussian distribution.

B. Graph Related Concepts

We assume that the agents interact on an undirected graph, described by a matrix $W \in \mathbb{R}^{n \times n}$. If agents $i$ and $j$ can communicate with each other, then $w_{ij}$, the $(i,j)$-th entry of $W$, is positive. Otherwise, $w_{ij}$ equals zero. The neighbor set $N_i$ of agent $i$ is defined as the set of agents $\{j|w_{ij} > 0\}$. Note that $i \in N_i$ always holds. Denote $\mathcal{L}$ as the graph Laplacian of the weight matrix $W$. Let $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ be the eigenvalues of $\mathcal{L}$ and $M$ be the unitary matrix that satisfies $\mathcal{L} = M\text{Diag}(\mu_1, \ldots, \mu_n)M^T$. Denote $\mu(\mathcal{L})$ and $\bar{\mu}(\mathcal{L})$ are the second smallest and the largest eigenvalue of $\mathcal{L}$, respectively.

C. Hilbert Spaces

A Hilbert space $\mathcal{H}$ is a complete inner-product space [18]. A set $\{e_k\}_{k \in \mathbb{N}} \subset \mathcal{H}$ is an orthonormal system if $\langle e_k, e_l \rangle = 0$ for $k \neq j$ and $\langle e_k, e_k \rangle = \|e_k\|^2 = 1$ for all $k \in \mathbb{N}$. If, in addition, the set of linear combinations of $\{e_k\}_{k \in \mathbb{N}}$ is dense in $\mathcal{H}$, then $\{e_k\}_{k \in \mathbb{N}}$ is an orthonormal basis. If $\mathcal{H}$ is separable, then any orthonormal basis is countable, and we have $h = \sum_{k=0}^\infty (h,e_k)e_k$, for any $h \in \mathcal{H}$. Define the coefficient sequence $\theta \in \mathbb{R}^\infty$ by $\theta_k = \langle h, e_k \rangle$ for $k \in \mathbb{N}$. Then $\theta \in l_2$ and, by Parseval’s identity, $\|h\| = \|\theta\|$. Let $\Phi : l_2 \to \mathcal{H}$ be the linear bijection that maps the coefficient sequence $\theta$ to $h$. For an arbitrary $D \subseteq \mathbb{R}^d$, $L_2(D)$ is a Hilbert space, and the inner product is the integral of the product of functions. Moreover, $L_2(D)$ is separable. In this letter, we denote $\{e_k\}_{k=0}^\infty$ as an orthonormal basis for $L_2(D)$ and $\Phi : l_2 \to L_2(D)$ the corresponding linear bijection between coefficient sequences and functions.

III. ALGORITHM DESIGN

In this section, we propose the Encrypted Functional Perturbation with Structured Noise algorithm (EFPSN in short) that privately solves the problem (1). Unlike the majority of privacy-preserving algorithms, EFPSN does not sacrifice accuracy for privacy.

To achieve privacy, EFPSN adds structured, zero-sum in specific, functional perturbations on cost functions $\{f_i\}_{i \in \mathbb{N}}$. EFPSN consists of two phases, as shown in Algorithm 1.

In phase I, the agents in the network cooperate to generate functional perturbations in a way that is immune to
Algorithm 1 Encrypted Functional Perturbation With Structured Noise

**Input:** Cost function \( \{f_i\}_{i \in \mathcal{N}} \), noise precision order \( P \), perturbation order \( K \), \( E_n \), \( \text{De} \), \( \Phi \), and \( \{\sigma_k\}_{k \in \mathbb{Z}_K} \).

**Output:** \( x^* \)

**Phase 1 – Masking cost functions**

1. for \( i \in \mathcal{N} \) do
2. Generate key pair \((\tilde{K}_i, \hat{K}_i)\), and randomly choose \( r_i \) following the Paillier encryption scheme
3. Share public key \( \hat{K}_i \) with agent \( j \in \mathcal{N}_i \)
4. end for
5. for \( (j, k) \in \mathcal{N}_i \times \mathbb{Z}_K \) do
6. Generate random noise \( \eta_{ijk} \sim N(0, \sigma_k^2) \)
7. Calculate \( \zeta_{ijk} = \text{En}([10^P \eta_{ijk}], \hat{K}_j, r_j) \)
8. Send \( \zeta_{ijk} \) to agent \( j \)
9. end for
10. for \( k \in \mathbb{Z}_K \) do
11. Calculate \( \tilde{\eta}_{ik} = \sum_{j \in \mathcal{N}_i} \eta_{ijk} - 10^{-P} \text{De}(\prod_{j \in \mathcal{N}_i} \zeta_{ijk}, \tilde{K}_i, \hat{K}_i) \)
12. end for
13. end for
14. \( \hat{f}_i = \Phi(\Phi^{-1}(f_i) + \tilde{\eta}_i) = f_i + \Phi(\tilde{\eta}_i) \), \( \tilde{\eta}_i = [\tilde{\eta}_{i0}, ..., \tilde{\eta}_{iK}, 0, ...] \in \mathbb{R}^N \)
15. end for

**Phase 2 – Distributed optimization**

16. Execute any distributed optimization algorithm on the masked functions \( \{\hat{f}_i\}_{i \in \mathcal{N}} \)

Eavesdropping attacks and curious-but-honest attacks to some extent. First, they generate keys and random numbers required by the Paillier encryption scheme. Then, they encrypt and send random noise \([10^P \eta_{ijk}] \) (Paillier encryption only works for integers) to their neighbors, and the noises are further decrypted to construct the zero-sum perturbation.

In Line 12, each agent \( i \) calculates \( \tilde{\eta}_{ik} \) by subtracting the sum of noises it receives from the sum of noises it sends. Due to Paillier encryption’s homomorphic property, \( \text{De}(\prod_{j \in \mathcal{N}_i} \zeta_{ijk}) = \sum_{j \in \mathcal{N}_i} \text{De}(\zeta_{ijk}) \), and each agent only decodes once for each \( k \in \mathbb{Z}_K \). This saves computation, especially in a large graph. Since \( \lim_{P \to \infty} 10^{-P} \text{De}(\zeta_{ijk}) = \eta_{ijk} \), we have \( \lim_{P \to \infty} \sum_{j \in \mathcal{N}_i} \tilde{\eta}_{ik} = \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathbb{Z}_K} (\eta_{ijk} - \eta_{ikk}) = 0, \forall k \). Therefore, we have generated a set of zero-sum signals \( \{\tilde{\eta}_{ik}\}_{i \in \mathcal{N}} \) given large \( P \).

Paillier encryption scheme guarantees privacy under eavesdropping attacks. Under the curious-but-honest attacks, agent \( i \) privacy is maintained given the attacker does not corrupt all \( i \)'s neighbors, as the noise coefficient sequence \( \tilde{\eta}_i \) of agent \( i \in \mathcal{N} \) remains unknown to the attacker.

Finally in Line 14, the agents perturb the cost functions by adding \( \Phi(\tilde{\eta}_i) \). \( \Phi(\cdot) \) is a function that maps a sequence in \( \ell_2 \) to a function in \( \ell_2(D) \). Such a construction depends on the orthonormal system \( \{e_k\}_{k \in \ell} \) in \( \mathcal{H} \) one chooses. For instance, given the orthonormal system \( \{e_k\}_{k \in \ell} \) and a sequence \( \tilde{\eta}_i \) and a sequence \( \tilde{\eta}_i = \{\tilde{\eta}_{ik}\}_{k \in \ell} \). The orthonormal system we use will be specified later.

Since \( \{\tilde{\eta}_{ik}\}_{i \in \mathcal{N}} \) is zero-sum when \( P \to \infty \), we have \( \lim_{P \to \infty} \sum_i \Phi(\tilde{\eta}_i) = \lim_{P \to \infty} \sum_i \sum_k \tilde{\eta}_{ik} e_k = 0 \). Namely, the perturbing functions are zero-sum when \( P \to \infty \). Additionally, the decrypted text in Line 12 is of precision \( 10^{-P} \). Consequently, the error brought by \( P \) will be dominated by the floating-point error once \( P \) is set to a moderately large value. Though \( \{\Phi(\tilde{\eta}_i)\}_{i \in \mathcal{N}} \) is zero-sum, each agent \( i \) gains privacy from the non-zero functional perturbation \( \Phi(\tilde{\eta}_i) \).

Then, the agents may conduct any distributed optimization algorithm on \( \{\hat{f}_i\}_{i \in \mathcal{N}} \). Since \( \sum_i \hat{f}_i(x) = \sum_i f_i(x) \), the solution solves problem (1) when \( P \to \infty \). Namely, EFPSN solves problem (1) precisely given proper \( P \).

**Remark 1:** EFPSN combines encryption, functional perturbation, and structured noise and is superior to considering any one of the techniques. In previous message-based methods, encryption at every iteration results in heavy communication and computation overhead, whereas the function-level encryption in EFPSN alleviates such a pain: only insignificant communication and computation overhead occur in phase I. Specifically, only \( K+1 \) real numbers are transmitted from each agent to its neighbor. Regarding the existing function-based methods, the obtained solution after functional perturbation suffers from a privacy-related deviation from the solution to problem (1), and thus leads to privacy-accuracy tradeoff. For EFPSN, however, the optimization accuracy is independent of the privacy budget, which will be elaborated more in the following sections. Moreover, using structured noise alone fails with the presence of eavesdropping, limiting its applicability.

**Remark 2:** Most homomorphic encryption methods, including Paillier’s, only work with integers as they rely on modular operations. Thus, we convert a real number to an integer by multiplying \( 10^P \), where \( P \) is a tunable parameter to trade off error and communication overhead.

**IV. PRIVACY ANALYSIS**

Here, we analyse the privacy-related property of EFPSN under the framework of differential privacy. First, the definition of \( \mathcal{V} \)-adjacency, originally proposed in [13], and the \((\epsilon, \delta)\)-DP definition under the functional setting are introduced.

**Definition 1 (\( \mathcal{V} \)-adjacency):** Given any normed vector space \((\mathcal{V}, \|\cdot\|_\mathcal{V})\) with \( \mathcal{V} \subseteq \ell_2(D) \), two sets of functions \( F = \{f_i\}_{i \in \mathcal{N}} \), \( F' = \{f'_i\}_{i \in \mathcal{N}} \subseteq \ell_2(D) \) are \( \mathcal{V} \)-adjacent if there exists \( i \in \mathcal{N} \) such that \( f_i = f'_i \), \( i \neq I \), and \( f_i - f'_i \in \mathcal{V} \).

**Definition 2 ((\( \epsilon, \delta \))-Differential Privacy):** Consider a random map \( M : \ell_2(D)^n \to \mathcal{X} \) from the function space \( \ell_2(D)^n \) to an arbitrary set \( \mathcal{X} \). Given \( \epsilon, \delta \in \mathbb{R}_{\geq 0} \), the map \( M \) is (\( \epsilon, \delta \))-differentially private if, for any two \( \mathcal{V} \)-adjacent sets of functions \( F = \{f_i\}_{i \in \mathcal{N}} \) and \( F' = \{f'_i\}_{i \in \mathcal{N}} \), and for any \( O \subseteq \mathcal{X} \) one has \( \mathbb{P}(M(F) \in O) \leq e^\epsilon \cdot \mathbb{P}(M(F') \in O) + \delta \).

Obviously, when \( \epsilon, \delta \to 0 \), the probabilities of \( M \) generating the same output from adjacent inputs are very close. Therefore, inferring information about the input from the output is barely possible. Privacy is thus preserved.
The adjacency space $V_q$ is chosen as follows. Given $q > 1$, use the weight sequence $\{k^q\}_{k=1}^{\infty}$ and let the adjacency space be the image of the resulting weighted $l_2$ space under $\Phi$, i.e., $V_q = \Phi(\delta \in \mathbb{R}^{|\mathcal{V}|}; \sum_{k=1}^{\infty} k^q \delta_k < \infty)$, where $\delta_k$ is the $k$th element of $\delta$. The rationale of considering such a space will be made clear from the analysis. Moreover, $\|f\|_{V_q} = \left( \sum_{k=1}^{\infty} (k^{2q} \delta_k^2) \right)^{1/2}$, where $\delta = \Phi^{-1}(f)$ is a norm on $V_q$.

Now we introduce our main theorem about privacy.

**Theorem 1:** Given $q > 1$, $\gamma > 0$, $p \in (1/2, q - 1/2)$, the chosen Voronoi space, and $\sigma^2 = \frac{\gamma}{p}$, the mechanism in Alg. 1 is $(\epsilon, \delta)$-differential private when the precision order and the perturbation order $P, K \to \infty$, with $\epsilon = \left( \frac{\mu L \sqrt{\mu A}}{\sqrt{2}} \right) (\frac{3}{2} + \frac{R \sqrt{\mu L A}}{\sqrt{2}})$, $\delta = e^{-\frac{\epsilon^2}{2}}$, where $A = \frac{1}{\gamma} \sqrt{\frac{2}{(q - p)}} |f_i - f_j|^2_{V_q}$, and $R$ is an arbitrary positive real number.

**Proof:** When the precision order $P \to \infty$, the encryption process is precise. For convenience, we can ignore the encryption process and the flooring in Alg. 1. Denote $\eta_k \triangleq \sum_{j \in N_k} \eta_{jk} = \sum_{j \in N_k} \eta_{jk}, \eta_k \triangleq [\eta_{k0}, \ldots, \eta_{kn}]$. Under such a case, $\mathcal{M}$ in algorithm 1 is essentially adding functional perturbation constructed from a set of degenerate Gaussian noises. Specifically, given $F = \{f_i\}_{i \in N}$, we have $\mathcal{M}(F) = \{f_i + \Phi(\eta_i)\}_{i \in N}$, and

$$\eta_k = [\eta_{k1}, \ldots, \eta_{kn}]^T \sim \mathcal{N}(0_n, 2\sigma_k^2 L), \forall k \in \mathbb{Z}_K$$

(2)

are the zero-sum coefficients.

From [16], we have

$$\mathcal{P}_{\eta_k}(y) \begin{cases} \frac{1}{\sqrt{\det((4\pi \sigma^2 L)^{(n-1)})}} \exp(-\frac{\|y - \Phi^T \eta_k\|^2}{4\sigma^2 L}), & y^T 1 = 0 \\ 0, & \text{else} \end{cases}$$

where $\Phi = \text{MDiag}(0, 1/\mu_2, \ldots, 1/\mu_n)\text{M}^T$ and $\det^*(4\pi \sigma^2 L) = (4\pi \sigma^2 L)^{n-1} \sum_{i=2}^{n} \mu_i$.

Let $F' = \{f'_i\}_{i \in N}$ be a set of $\mathcal{V}$-adjacent functions of $F$ that differ only in the $i$th element. Let $\Psi^{-1} : L_2(D)^n \to \mathbb{R}^{n \times n}$ be the map such that $\Psi^{-1}(F) = \{\Phi^{-1}(f'_i)\}_{i \in N}$. Define $\Phi^{-1}_k : L_2(D) \to \mathbb{R}^{k+1}, \Phi^{-1}_k : L_2(D) \to \mathbb{R}$ as the map that returns the first $k+1$ coefficients and the $(k+1)$-th coefficient of $\Phi^{-1}(\cdot)$, respectively. Similarly, define $\Psi^{-1}_k : L_2(D)^n \to \mathbb{R}^{n \times (k+1)}, \Psi^{-1}_k : L_2(D)^n \to \mathbb{R}^n$ as the map that returns the first $k+1$ columns and the $(k+1)$-th column of $\Psi^{-1}(\cdot)$, respectively. For any $O \in L_2(D)^n$, $O - F \triangleq \{g_i\}_{i \in N} \in L_2(D)^n\{|g_i + f_i|_{i \in N}\} \in \mathcal{O}$.

We have

$$\mathbb{P}\{\mathcal{M}(F') \in \mathcal{O}\} = \mathbb{P}\{\{\eta_i\}_{i \in N} \in \Psi^{-1}(O - F')\}$$

(3)

$$= \lim_{K \to \infty} \int_{\Psi^{-1}_k(O-F')}^{K} \prod_{k=0}^{K} \mathcal{P}_{\eta_k}(y_k) dy_0 \ldots dy_K$$

Let $\xi_k = [\xi_{k1}, \ldots, \xi_{kn}]^T$, by the linearity of $\Phi$, we have

$$\Psi^{-1}_k(O - F') = \Psi^{-1}_k(O - F) + \xi_k$$

(4)

Combing (3) and (4), we have

$$\mathbb{P}\{\mathcal{M}(F') \in \mathcal{O}\} = \lim_{K \to \infty} \int_{\Psi^{-1}_k(O-F')}^{K} \prod_{k=0}^{K} \mathcal{P}_{\eta_k}(y_k + \xi_k) dy_0 \ldots dy_K$$

To prove that $\mathcal{M}$ is $(\epsilon, \delta)$-DP, it suffices to show the ratio of $\prod_{k=0}^{K} \mathcal{P}_{\eta_k}(\xi_k)$ over $\prod_{k=0}^{K} \mathcal{P}_{\eta_k}(\xi_k + \xi_k)$ is bounded by $e^\epsilon$ with probability at least $(1 - \delta)$.

We know that

$$\prod_{k=0}^{K} \mathcal{P}_{\eta_k}(\xi_k + \xi_k) = \exp \left( \sum_{k=0}^{K} \frac{2x_k^2 \xi_k + 2x_k^2 \xi_k}{4\sigma_k^2} \right)$$

$$\leq \exp \left( \frac{1}{\mu(L)} \left( \sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2} \right) \right)$$

Denote

$$\text{Rat} \triangleq \exp \left( \frac{1}{\mu(L)} \left( \sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2} \right) \right)$$

We show that Rat is bounded with certain probability.

Since $\{f'_i\}_{i \in N}$ and $\{f'_i\}_{i \in N}$ only differ in one element, $\xi_k$ has at most one non-zero element. Noting that $\eta_k$ is random and drawn from $N^t(0_n, 2\sigma_k^2 L)$, $\xi_k$ is a univariate Gaussian random variable. From (2), each element of $\eta_k$ is at most of variance $2\sigma_k^2 L$. Thus, $\sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2}$ is a univariate Gaussian random variable with variance less than or equal to $\frac{1}{2} \mu(L) \sum_{k=0}^{K} ||\xi_k||^2$. We further bound its variance following a similar way in the proof of [13, Th. V.2]:

$$\sum_{k=0}^{K} ||\xi_k||^2 \leq \sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2} \leq \left( \sum_{k=0}^{K} \frac{1}{(k+1)!} \right)^{\frac{1}{2}} \left( \sum_{k=0}^{K} \frac{2x_k^2 \xi_k}{4\sigma_k^2} \right)^{\frac{1}{2}}$$

$$= \left( \gamma \sqrt{\frac{2}{(q - p)}} \right) |f_i - f_j|^2_{V_q} \chi \triangleq A.$$

Let $R$ be an arbitrary positive real number. When $\sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2}$ holds, we have

$$\text{Rat} \leq \exp \left( \frac{1}{\mu(L)} \left( \frac{A}{4} + \frac{R \sqrt{\mu(L)A}}{\sqrt{2}} \right) \right)$$

From the Chernoff bound for Gaussian RV, we have

$$\mathbb{P}\left( \sum_{k=0}^{K} \frac{2x_k^2 \xi_k + \sum_{k=0}^{K} ||\xi_k||^2}{4\sigma_k^2} \geq \frac{1}{2} \mu(L) \sum_{k=0}^{K} ||\xi_k||^2 \right) \leq e^{-\frac{\epsilon^2}{2}}.$$
Remark 3: The communication graph affects $\epsilon$ as it determines $\tilde{\mu}(L)$ and $\mu(L)$. Strongly and evenly connected graph benefits privacy. In addition, larger $\gamma$ (more noise) contributes to better privacy. Specifically, arbitrarily large $R$, $\gamma$ s.t. $R \gamma \sim o(1)$ results in arbitrarily small $\epsilon$ and $\delta$. Namely, under specific adversary, we can attain an arbitrarily private mechanism while not sacrificing the accuracy.

Remark 4: Infinite $P, K$ are required for obtaining a DP bound. In particular, infinite $P$ guarantees the decrypted noise to be Gaussian. In practice, however, a moderately large $P$ is sufficient as the floating-point error will dominate. Infinite $K$ is compulsory since private information may exist in all orders of the cost function. However, suppose that every cost function $f_i$ is at most of order $K_{\text{order}}$, where a function $f_i$ is of finite order $K_{\text{order}}$ if $\Phi_{K_{\text{order}}}^{-1}(f_i) \neq 0$ and $\Phi_k^{-1}(f_i) = 0$ for all $k > K_{\text{order}}$, then $K = K_{\text{order}}$ suffices to provide the same privacy bound as in the theorem. Namely, a finite $K$ yields privacy guarantee for cost functions belonging to a specific class, e.g., quadratic functions. In the simulations, we can see that even choosing $K < K_{\text{order}}$ can preserve the privacy quite well.

Remark 5: Given finite ordered $\{f_i\}$, the DP bound exists for any normed space as $\|f_i - f_j\|_{\nabla^2}^2$ is bounded by a finite sum and any norm on a finite-dimensional space is equivalent.

V. NUMERICAL EXPERIMENTS

Here, we evaluate the performance of the EFPSN algorithm under non-convex settings numerically.\(^1\) And we compare it with the non-zero-sum functional perturbation mechanism in [13]. Before displaying the results, we demonstrate how the orthonormal system $\{e_k\}_{k \in \mathbb{N}}$ based on which we generate the coefficients-function mapping $\Phi$, is constructed.

A. Generating the Orthonormal System

Since the number of elements of an orthonormal basis grows factorially as the number of variables increases, it becomes prohibitively large for real world applications. Consequently, for a function with $M$ variables, we pick $m < M$ variables to perturb. Also, instead of generating an orthonormal basis, we only generate an orthonormal system in $L_2$ with $N$ elements. Since the generated orthonormal system belong to some orthonormal basis, perturbing the orthonormal system is equivalent to perturbing the orthonormal basis with some noise coefficient being 0. Therefore, the noises are still zero-sum and Gaussian.

The orthonormal system is constructed from the Gram-Schmidt orthonormalization of the Taylor functions. Given the tuple $(K, m, N)$, we randomly generate $N$ elements of Taylor functions of $m$ variables, of which the sum of the order is smaller than or equal to $K$. Then we orthonormalizing all terms using Gram-Schmidt method. E.g., when $(K, m, N) = (3, 2, 5)$ under noise level $\gamma = 1$, the orthonormal system is $\{0.5, 0.87x_2, 3.30x_1^2 - 1.98x_2, 0.87x_1, 2.91x_2^2 - 0.97x_2\}$ and the noise sequence is $\{0.18, 0.63, -0.37, 0.82, 2.02\}$. The corresponding perturbing function is $5.85x_1^2x_2 - 2.14x_2^2 + 0.71x_1$.

\(^1\)Experiments under convex settings are available in the extended version of this letter [19].

Fig. 2. Mean Squared Gradient and Accuracy for LeNet.

B. Accuracy Test

In this part, we validate the accuracy of the proposed EFPSN method. We consider an image classification task on MNIST using Convolutional Neural Network. We adopt the classic LeNet [20], which consists of 13426 parameters. We choose to perturb the bias of its last linear layer, and again we set $(K, m, N) = (1, 10, 10)$. The experiments are conducted under different noise level $\gamma \in \{1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}, 1e^{-3}, 1e^{-2}\}$.

Consider 5 agents in the network, connected as shown in Fig. 1(b). Each agent holds the same number of randomly assigned training data points. We adopt decentralized stochastic gradient descent in Phase II in Alg. 1. The batch size is set to 64. The initial learning rate equals 0.2. Each agent conducts 10000 gradient updates. The learning rate remains fixed in the first 2000 steps, and drops to $4e^{-5}$ at the last step.

Fig. 2(a) depicts the relation between noise magnitude $\gamma$, and the squared norm of the average gradient of all the agents, $\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \nabla f_i(x)^2$. When $\gamma = 1e^{-2}$, the results from EFPSN and the non-zero-sum algorithm are nearly identical, both close to the noise-free case. However, as $\gamma$ increases, the solution obtained from the non-zero-sum method starts to deviate from optima, and such deviation spikes at a roughly constant rate. The blue line does not rise until $\gamma = 1e^3$, at which point the orange line is 6 magnitudes higher.

The rise in the blue line stems from the disagreements between local decision variables $x_i$. Note that our perturbed function only guarantees zero-sum when each agent holds the same decision variable. Yet slight differences between $x_i$ exist due to finite step size. With EFPSN, one can always generate a more accurate solution using a finer learning rate.

Fig. 2(b) demonstrates the classification accuracy under different algorithms and noise levels. The pattern matches that of Fig. 2(a). Generally, the non-zero-sum algorithm, the test accuracy starts to drop dramatically when $\gamma$ reaches $1e^2$. In
contrast, EFPSN’s solution remains as accurate as the noise-free case until $\gamma = 1e^4$. Namely, EFPSN provides much more privacy budgets while not degenerating accuracy.

C. Privacy Test

To further validate EFPSN’s efficacy in privacy-preserving, we conduct iDLG [21] attack on agents with zero-sum and non-zero-sum noise, respectively.

With either EFPSN or the non-zero-sum noise method, each agent receives some functional perturbation of roughly the same magnitude. Since iDLG is carried out at the agent level, it makes no difference between the two algorithms. Therefore, we conduct iDLG on agent 1 in Fig. 1(b), with the noise generated from EFPSN at different noise level.

Specifically, we assume the mixing matrix is known to the attacker. And the attacker has access to at least one of the communication channels connected to agent 1 (either eavesdropping or corrupting 1’s neighbor will do). Therefore, the attacker knows agent 1’s perturbed gradient $\nabla f_1(x^1_k)$ and decision parameters $x^1_k$. But it does not know the functional perturbation $\Phi(\tilde{\eta}_i)$. Consequently, the true gradient $\nabla f_1(x^1_k)$ remains unrevealed. Namely, the attacker is recovering the raw data using inexact gradient information. And the larger the noise level $\gamma$, the more inexact the gradient is.

Fig. 3 represent the iDLG attacker’s inference result on LeNet. The top left subfigure is the raw data. And the remaining ones are the adversary’s estimate of the raw data at different iterations (from 0 to 240). As $\gamma$ increases, the retrieved picture becomes blurred. Interestingly, perturbing the original problem functionally equals to directly perturbing the dataset. Generally, after $\gamma \geq 1e^3$, the recovered picture is unrecognizable for humans. When $\gamma \geq 1e^3$, however, the accuracy of the model trained by the non-zero-sum method has dropped below 10% (as shown in Fig. 2). This suggests that the non-zero-sum solution would be too inaccurate to provide enough privacy. In contrast, the EFPSN solution has comparable accuracy to the noise-free case. Thus, EFPSN is capable of preserving privacy without degenerating accuracy.

VI. CONCLUSION

In the letter, we proposed the Encrypted Functional Perturbation with Structured Noise algorithm that solves the decentralized optimization problem 1 privately and accurately. Given exact consensus, EFPSN eliminates the privacy-accuracy trade-off by constructing a zero-sum functional perturbation. Since such construction requires secure communication between agents, we adopt the Paillier encryption scheme to defend eavesdropping attacks. We rigorously proved the privacy property of EFPSN under the differential privacy framework. Simulations confirmed the efficacy of EFPSN in protecting privacy while maintaining accuracy.

REFERENCES

[1] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Trans. Autom. Control, vol. 54, no. 1, pp. 48–61, Jan. 2009.
[2] S. Pu and A. Nedić, “Distributed stochastic gradient tracking methods,” Math. Program., vol. 187, no. 1, pp. 409–457, 2021.
[3] S. Pu, A. Garcia, and Z. Lin, “Noise reduction by swarming in social foraging,” IEEE Trans. Autom. Control, vol. 61, no. 12, pp. 4007–4013, Dec. 2016.
[4] T. Yan, Y. Gu, T. He, and J. A. Stankovic, “Design and optimization of distributed sensing coverage in wireless sensor networks,” ACM Trans. Embedded Comput. Syst., vol. 7, no. 3, pp. 1–40, 2008.
[5] R. Murphy and P. M. Pardalos, Cooperative Control and Optimization, vol. 66. New York, NY, USA: Springer, 2002.
[6] A. G. Roy, S. Siddiqui, S. Pölder, N. Navab, and C. Wachinger, “BrainTorrent: A peer-to-peer environment for decentralized federated learning,” 2019, arXiv:1905.06731.
[7] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” 2019, arXiv:1906.08935.
[8] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” Found. Trends Theor. Comput. Sci., vol. 9, nos. 3–4, pp. 211–407, 2013.
[9] J. Cortés, G. E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G. J. Pappas, “Differential privacy in control and network systems,” in Proc. IEEE 55th Conf. Decis. Control (CDC), Las Vegas, NV, USA, 2016, pp. 4252–4272.
[10] T. Ding, S. Zhu, J. He, C. Chen, and X. Guan, “Differentially private distributed optimization via state and direction perturbation in multiagent systems,” IEEE Trans. Autom. Control, vol. 67, no. 2, pp. 722–737, Feb. 2022.
[11] Y. Lou, L. Yu, and S. Wang, “Privacy preservation in distributed subgradient optimization algorithms,” Dec. 2015, arXiv:1512.08822.
[12] Z. Huang, S. Mitra, and N. Vaidya, “Differentially private distributed optimization,” in Proc. Int. Conf. Distrib. Comput. Netw., Jan. 2015, pp. 1–10.
[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, Mar. 2018.
[14] Y. Lu and M. Zhu, “Privacy preserving distributed optimization using homomorphic encryption,” Automatica, vol. 96, pp. 314–325, Oct. 2018.
[15] C. Zhang and Y. Wang, “Enabling privacy-preservation in decentralized optimization,” IEEE Trans. Control Network Syst., vol. 6, no. 2, pp. 679–689, Jun. 2019.
[16] N. Gupta, S. Gade, N. Chopra, and N. H. Vaidya, “Preserving statistical privacy in distributed optimization,” IEEE Contr. Syst. Lett., vol. 5, pp. 779–784, 2021.
[17] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Proc. Int. Conf. Theor. Appl. Cryptograph. Techn., 1999, pp. 223–238.
[18] W. Rudin, Real and Complex Analysis. New York, NY, USA: McGraw-Hill, 1987.
[19] Y. Zhou and S. Pu, “Private and accurate Decentralized optimization via encrypted and structured functional perturbation,” 2022, arXiv:2209.01756.
[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[21] B. Zhao, K. R. Mopuri, and H. Bilen, “iDLG: Improved deep leakage from gradients,” 2020, arXiv:2001.02610.