

Analog Secret Sharing with Applications to Private Distributed Learning

Mahdi Soleymani, Hessam Mahdavifar, and A. Salman Avestimehr

Abstract— We consider the critical problems of distributed computing and learning over data while keeping it private from the computational servers. The state-of-the-art approaches to this problem rely on quantizing the data into a finite field, so that the cryptographic approaches for secure multiparty computing can then be employed. These approaches, however, can result in substantial accuracy losses due to fixed-point representation of the data and computation overflows. To address these critical issues, we propose a novel algorithm to solve the privacy-preserving distributed computing problem when data is in the analog domain, e.g., the field of real/complex numbers. We characterize the privacy of the data from both information-theoretic and cryptographic perspectives, while establishing a connection between the two notions in the analog domain. More specifically, the well-known connection between the distinguishing security (DS) and the mutual information security (MIS) metrics is extended from the discrete domain to the analog domain. This is then utilized to bound the amount of information about the data leaked to the servers in our protocol, in terms of the DS metric, using well-known results on the capacity of single-input multiple-output (SIMO) channel with correlated noise. It is shown how the proposed framework can be adopted to do computation tasks when data is represented using floating-point numbers. We then show that this leads to a fundamental trade-off between the privacy level of data and accuracy of the result. By extending the setup to distributed learning, we show how to train a machine learning model using the proposed algorithm while keeping the data as well as the trained model private. Then numerical results are shown for experiments on several datasets. Furthermore, experimental advantages are shown comparing to fixed-point implementations over finite fields.

I. INTRODUCTION

Distributed learning systems involve dispersing data among many servers that operate in parallel with the aim of collectively completing a certain computational job, e.g., computing a certain function over a dataset or iteratively updating the model parameters trained on a dataset [1]. Then the results generated by *sufficiently* many local servers are collected in order to recover the desired outcome, e.g., the output of the given function over the dataset or the updated model parameters. One of the major concerns in such distributed systems is to preserve

the privacy of the dataset while dispersing it among the servers. More specifically, the dataset may contain highly sensitive information, e.g., biometric data of patients in a hospital [2] or customers' data of a company [3], necessitating that *almost* no information about the dataset is revealed to the computational servers. Such a privacy constraint is often generalized to ensure that any subset of colluding servers, up to a certain size, can not gain *almost* any information about the dataset.

Since both the transmitter/encoder and the receiver/decoder in a distributed computing setup are the same entity, referred to as a *master* node, the notion of *privacy*, as opposed to *security*, is used. However, the same metrics that are used in conventional secure communication/computation setups, including information-theoretic security [4] as well as well-known notions of semantic security and distinguishing security in the cryptography literature emanating from [5], can be used to measure the privacy level of the data in such emerging distributed computing setups.

The seminal Shamir's secret sharing scheme and its various versions are often used to provide information-theoretic security for data, referred to as a secret, while distributing it among a set of servers/users [6]. Also, Shamir's scheme serves as the backbone of most of the existing schemes on privacy-preserving multiparty computing such as the celebrated BGW scheme [7]. The idea can be illustrated via an example as follows. Consider a given dataset X and two computational servers, referred to as servers 1 and 2. Suppose that the function $f(X) = aX$, where a is a scalar, needs to be computed over the dataset X . The data symbols in X as well as a are considered as elements of a finite field \mathbb{F}_q . Then a random N is generated, with the same size as the dataset X and entries generated independently and uniformly at random from \mathbb{F}_q . Then N and $X + N$, also referred to as secret shares, are given to the servers 1 and 2, respectively. Since N and $X + N$ are both uniformly distributed, the servers do not learn anything about X individually. The servers return aN and $a(X + N)$. Then aX is recovered by subtracting the former from the latter.

In Shamir's scheme, the secret/data symbols are always assumed to be elements of a finite field. Consequently, the state-of-the-art schemes treat the data symbols in the given dataset as finite field elements in order to employ Shamir's secret sharing, see, e.g., [7]. However, quantizing the data into a finite field can result in substantial accuracy losses mainly due to computation overflows. In practice, the dataset X consists of real/complex values often represented as floating-point numbers. One possible method to avoid overflow errors is to bypass quantizing the data into a finite field and carry out the computations over the real/complex numbers. This approach eliminates the need for mapping back the computation results from the underlying finite field to the analog domain and hence, is not prone to accuracy loss due to overflow errors. However, in

This material is based upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0053 and FA8750-19-2-1005, ARO award W911NF1810400, NSF grants CCF-1703575, CCF-1763673, CCF-1763348, CCF-1909771 and CCF-1941633, and MLWINS-2002874, ONR Award No. N00014-16-1-2189, and a gift from Cisco, a gift from Intel/Avast/Borsetta via the PrivateAI institute, a Cisco research gift and contract, and gifts from Qualcomm and Intel. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

M. Soleymani and H. Mahdavifar are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48104 (email: mahdy@umich.edu and hessam@umich.edu).

A. Salman Avestimehr is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: avestimehr@ee.usc.edu).

this approach X can not be perfectly secured in an information-theoretic sense, i.e., the mutual information between X and $X + N$, denoted by $I(X; X + N)$, being exactly zero. These are the main challenges that need to be properly addressed when designing privacy-preserving distributed learning algorithms in the infinite fields of \mathbb{R}/CC , also referred to as the *analog domain*.

A. Our contributions

In this paper, we provide a framework to construct the counterpart of Shamir’s secret sharing scheme in the analog domain. This framework is then used to construct privacy-preserving distributed computation and learning protocols over real/complex datasets. In other words, all the operations including encoding the data symbols to be distributed among the computational servers and recovery of the final outcome from the collected results returned by the servers are over the infinite fields of \mathbb{R}/CC . It is assumed that the servers are *honest-but-curious* meaning that they will not deviate from the protocol but may attempt to infer the data from what they observe throughout the protocol.

In the proposed protocol, the information-theoretic measure of security is no longer *perfect*, comparing to Shamir’s secret sharing scheme over finite fields, as discussed earlier. In order to show the privacy guarantees of the protocol, bounds are provided on how much information about data is revealed to a server/subsets of servers in terms of various notions of security. We also argue that, in a practical setting, this comes at the expense of accuracy of the final outcome of the protocol when all data symbols are represented by floating-point numbers and all operations are also assumed to follow *standard* floating-point operations. More precisely, we provide a fundamental trade-off between the security level of the protocol and the accuracy of the outcome in a practical setting assuming floating-point operations. The proposed protocol is also used in a distributed learning experiment using four servers to train a logistic regression model over MNIST dataset [8]. In this experiment, the amount of information about the dataset and the trained model revealed to each of the servers, in terms of the distinguishing security metric, is less than 10^{-15} and 2×10^{-14} , respectively. It is observed that the accuracy of our protocol closely follows that of the conventional centralized approach, thereby offering a privacy-preserving distributed solution at a negligible cost in terms of the accuracy of the result. Furthermore, it is shown that while approaches based on fixed-point implementations suffer from a sharp transition to the performance of randomly guessing by increasing the size of training dataset, our protocol offers a robust solution that is scalable with the size of the training dataset.

B. Related work

Studying privacy-preserving distributed machine learning algorithms has recently received significant attention in the literature [9]–[14]. There is also an extensive amount of work on secure matrix-matrix multiplication which is a core building block for many machine learning algorithms, see, e.g.,

[15]–[19]. As mentioned earlier, the information-theoretic privacy guarantees of the data in these prior works is based upon Shamir’s secret sharing scheme and its variations. Since Shamir’s scheme needs to be run over a finite field while data symbols are real-valued, a common method is to assume a certain quantization of the data symbols followed by mapping them into elements of a finite field of a large prime size. However, if an overflow occurs, i.e., a computed symbol during the computation process by one of the servers becomes larger than the field size, then a successful recovery of the outcome of the computation can not be guaranteed. In other words, the computation procedure at each of the servers can be regarded as a fixed point computation, which is constrained by conditions guaranteeing no overflow occurs.

There is also another line of work on adopting coded distributed computing protocols for computation over real-valued data [20]–[26]. But these works are mostly focused on the numerical stability of the protocols in the presence of slow or unresponsive servers, also referred to as stragglers, and do not study privacy guarantees for the data. Our focus in this paper is on providing privacy-preserving protocols and straggler servers are not considered.

However, secret sharing and privacy-preserving computation in the analog domain are not discussed in these works.

Another related major line of work concerns with floating-point implementation of secure multi-party computing (MPC) protocols [27]–[29]. Such protocols can be described in high level as follows. In a *standard* floating-point implementation, each number/data symbol is represented by two main components: one represents the most, let’s say b_m , significant bits of the data symbol and the other one represents the power of the exponent of data symbol. Then these two components of the data symbols are secured separately using Shamir’s secret sharing over finite fields. This requires a certain implementation of floating-point operations and does not allow using off-the-shelf readily available floating-point operations that are often optimized to perform computational tasks on badges of data in parallel. As a result, the inefficiency of such protocols poses a major difficulty in their implementation. Furthermore, a major difference between this line of work and our approach is that the parties are allowed to communicate in secure multiparty computing. This is mainly because in this setting each party aims at computing a certain function of data symbols shared between the parties without revealing any information about his/her share of the data. The communication overhead between the parties/servers is another major factor contributing to the inefficiency of these protocols in practical systems. On the other hand, in our approach, different servers are assumed to run in parallel and no communication is required between them. Once finished, the servers return their locally computed results from which the true outcome of the computational task can be computed.

It is worth mentioning that a follow-up work by the authors, that extends the results of this paper by enabling parallelization in computations with privacy guarantees, has been recently published while this paper was undergoing the review process [30]. Building upon the results of this work together with Lagrange coded computing (LCC) [10], we introduced analog

LCC (ALCC) in [30]. ALCC enables parallelizable privacy-preserving coded computing in the analog domain. The scheme leverages the Lagrange interpolation polynomial to encode several secrets/datasets into a single polynomial, thereby allowing the computations to be carried out in a parallel fashion over batches of data. However, the primary distinction of [30] is to illustrate that, even for a fixed noise variance, the choice of a certain set of points in ALCC, referred to as *interpolation points*, leads to a new trade-off between the accuracy and the privacy which is specific to ALCC. Hence, one has to carefully pick these parameters apart from the noise variance to avoid unnecessarily compromising accuracy/privacy in practice. This is a new fundamental trade-off that does not have a counterpart in either analog adaptations of Shamir's scheme, e.g., this work, or LCC with fixed-point implementation over finite fields [10]. It is important to note that neither the encoder nor the decoder of ALCC can be reduced to those of the present work for any choice of parameters. Nevertheless, by choosing $m = n = k = 1$ in ALCC [30], one can compare the provided accuracy bound in [30, Theorem 4] with the one derived in this work and observe that the bound provided in this work outperforms that of [30] for any choice of other parameters in ALCC. Furthermore, the privacy bound in [30] is evaluated numerically, and no closed-form expression is provided for the privacy bound. In comparison, we provide a closed-form expression for the privacy bound in this work.

The rest of this paper is organized as follows. The problem is formulated in Section II followed by the description of the proposed protocol. The accuracy of the the protocol is analyzed in Section III. In Section IV we provide an analysis for the privacy level of data in the protocol by considering two well-known notions of security. Various experimental results are provided in Section V. Finally, the paper is concluded in Section VI.

II. PROBLEM FORMULATION AND THE PROTOCOL

Consider a setup with N computation servers/parties indexed by $1, 2, \dots, N$. Given a data symbol s , also referred to as a *secret*, a D -degree polynomial function of s denoted by $f(s) \stackrel{\text{def}}{=} \sum_{i=0}^D f_i s^i$ needs to be computed by utilizing the computational power of the parties, while the secret remains *private* assuming up to t parties can collude. The notion of privacy will be clarified in Section IV. The secret s is an instance of a continuous random variable S taking values in $[-r, r]$.¹ Other than this constraint on the range of S , no assumption is made on the probability distribution of S .

Remark 1: Note that the computational task of polynomial evaluation is considered in this paper in order to arrive at explicit analytical guarantees. However, in order to apply this setup to a learning experiment, a polynomial approximation of the underlying computation function, e.g., the sigmoid function, can be considered. This will be further discussed in Section V.

In the considered protocol, given the secret s the polynomial $p(x)$ is constructed as follows:

$$p(x) \stackrel{\text{def}}{=} s + \sum_{j=1}^t n_j x^j,$$

where n_j 's are i.i.d., drawn from a zero-mean circular symmetric complex Gaussian distribution with standard deviation $\frac{\sigma_n}{\sqrt{t}}$, denoted by $\mathcal{N}(0, \frac{\sigma_n^2}{t})$, where t is the maximum number of colluding parties. For evaluating the precision of the protocol in practice, the distribution of n_i 's will be truncated, i.e., it is assumed that they are drawn from a truncated Gaussian distribution with a maximum absolute value, denoted by m , for $m \in \mathbb{R}^+$. This will be further clarified in Section III. The shares of the computation parties consist of the evaluation of $p(x)$ over certain complex-valued evaluation points $\omega_1, \dots, \omega_t$, i.e.,

$$y_i = s + \sum_{j=1}^t \omega_i^j n_j \quad (1)$$

is given to server i , for $i \in [N]$. The system of equations in (1) can be written in the matrix form as follows:

$$\mathbf{y}_{N \times 1} = \mathbf{A}_{N \times (t+1)} \mathbf{x}_{(t+1) \times 1}, \quad (2)$$

where $\mathbf{x} = (s, n_1, \dots, n_t)^T$, $\mathbf{y} = (y_1, \dots, y_N)^T$, and

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & \omega_1 & \dots & \omega_1^t \\ 1 & \omega_2 & \dots & \omega_2^t \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N & \dots & \omega_N^t \end{bmatrix}.$$

Then server i computes $f(y_i)$ and returns the result, e.g., to a *master* node. The master node then recovers $f(s)$. Conceptually, this can be done by interpolating the polynomial $f(p(x))$ and evaluating it at 0, i.e., the constant coefficient of $f(p(x))$ is equal to $f(s)$. More specifically, let $\mathbf{a} = (f(s), a_1, \dots, a_{\tilde{D}})^T$, where $\tilde{D} = Dt$, denote the vector of all coefficients of the polynomial $f(p(x))$. Let also $\mathbf{z} = (f(y_1), \dots, f(y_N))^T$ and

$$\mathbf{B} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & \omega_1 & \dots & \omega_1^{\tilde{D}} \\ 1 & \omega_2 & \dots & \omega_2^{\tilde{D}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N & \dots & \omega_N^{\tilde{D}} \end{bmatrix}. \quad (3)$$

Then the system of linear equations

$$\mathbf{z}_{N \times 1} = \mathbf{B}_{N \times (\tilde{D}+1)} \mathbf{a}_{1 \times (\tilde{D}+1)}, \quad (4)$$

can be solved for \mathbf{a} in order to recover $f(s)$. Note that $N \geq Dt + 1 = \tilde{D} + 1$ is the necessary and sufficient condition on the number of parties in order to guarantee a successful interpolation of $f(p(x))$, which is of degree \tilde{D} . Equivalently, it is the necessary and sufficient condition for recovery of \mathbf{a} in (4). Throughout the rest of this paper, it is assumed that $N = \tilde{D} + 1$, implying that all shares y_i 's are needed to be returned to the master node for a successful recovery of the computation.

Note that the master node does not need to compute the entire \mathbf{a} in (4) and is only interested in recovering $f(s)$, the first entry of \mathbf{a} . Let $\tilde{\mathbf{b}}$ denote the first row of \mathbf{B}^{-1} , which is

¹Following the convention, random variables are represented by capital letters and their instances are represented by lower case letters.

well-defined due to \mathbf{B} being a Vandermonde matrix. Then the master node only needs to compute $\tilde{\mathbf{b}}\mathbf{z}$ to recover $f(s)$. Since ω_i 's are fixed, $\tilde{\mathbf{b}}$ is computed once, is stored, and then is used every time the protocol is run.

Remark 2: Note that the computational complexity of encoding in the master node is linear with the dataset size, where the dataset is treated as a vector of secrets and $f(\cdot)$ needs to be evaluated over the entries of this vector. Moreover, the complexity of decoding is also linear with the dataset size as the decoder only computes a linear combination of the results returned by the servers. In other words, the computational complexity at the master node does not depend on D , which can be large. It is worth mentioning that the goal of the protocol is not to reduce the *overall* computational complexity of a computation task across all the servers. The protocol in this paper, as well as prior works in the literature, e.g., [13], provide a framework to utilize external computation units in distributed servers while providing privacy guarantees.

To summarize, the protocol is described step-by-step in Algorithm 1 next.

Algorithm 1 Privacy-preserving distributed polynomial evaluation scheme in the analog domain.

Input: Secret s .

Public parameters: $A_{N \times (t+1)}$, $\tilde{\mathbf{b}}_{1 \times N}$.

Output: Evaluation of $f(s)$ in the master node.

Encoding phase (at the master):

Pick i.i.d. $n_j \sim \mathcal{N}(0, \frac{\sigma_n^2}{t})$, for $j = 1, \dots, t$.

Set $\mathbf{x} = (s, n_1, \dots, n_t)^\top$.

Compute $(y_1, \dots, y_N)^\top = \mathbf{A}\mathbf{x}$.

Send y_i to server i .

Computation phase (at server i):

Compute $z_i = f(y_i)$.

Send z_i to the master node.

Decoding phase (at the master):

Set $\mathbf{z} = (z_1, \dots, z_N)^\top$.

Compute $f(s) = \tilde{\mathbf{b}}\mathbf{z}$.

In the next section, the accuracy of the protocol described in Algorithm 1 is analyzed. In theory, if all the computations are done over the complex numbers with infinite precision, then $f(s)$ is computed accurately. In practice, data is represented using a finite number of bits, either as fixed point or floating point. Floating-point representation consists of a fixed-precision part and an exponent part specifying how the fixed-precision part is scaled. Let b_m denote the number of precision bits in the floating-point representation, referred to as *mantissa* and b_e denote the number of bits used to represent the power of the exponent part in the floating-point representation, referred to as *exponent*.

III. ACCURACY ANALYSIS

In this section, accuracy of the computation outcome of the proposed protocol in Section II is characterized in terms of other parameters of the protocol.

In general, in a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{y}$, where \mathbf{x} is the vector of unknown variables, the perturbation in the

solution caused by the perturbation in \mathbf{y} is characterized as follows. Let $\hat{\mathbf{y}}$ denote a noisy version of \mathbf{y} , where the noise can be caused by round-off errors, truncation, etc. Let also $\hat{\mathbf{x}}$ denote the solution to the considered linear system when \mathbf{y} is replaced by $\hat{\mathbf{y}}$. Let $\Delta\mathbf{x} \stackrel{\text{def}}{=} \hat{\mathbf{x}} - \mathbf{x}$ and $\Delta\mathbf{y} \stackrel{\text{def}}{=} \hat{\mathbf{y}} - \mathbf{y}$ denote the perturbation, also referred to as error, in \mathbf{x} and \mathbf{y} , respectively. Then the relative perturbations of \mathbf{x} is bounded in terms of that of \mathbf{y} as follows [31]:

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa_A \frac{\|\Delta\mathbf{y}\|}{\|\mathbf{y}\|}, \quad (5)$$

where κ_A is the condition number of A and $\|\cdot\|$ denotes the l^2 -norm.

As mentioned in Section II, the Gaussian distribution of n_i 's is truncated in practice. This is used to provide a deterministic (non-probabilistic) guarantee on the accuracy of the computation result expressed in the following theorem. It is assumed that the computations at the master and servers do not impose any errors other than precision loss due to finite representation of the results.

Theorem 1: Let $\Delta f(s)$ denote the perturbation of $f(s)$ in the protocol discussed in Section II. Let $mt + r \geq 1$ and $c \stackrel{\text{def}}{=} \sum_{i=0}^D |f_i|$. Then,

$$\Delta f(s) \leq c\sqrt{N}(mt + r)^D \frac{\kappa_B}{\lambda_{\min}} 2^{-b_m}, \quad (6)$$

where t is the maximum number of colluding parties, m is the truncation parameter of the Gaussian distribution, κ_B and λ_{\min} respectively are the condition number and minimum singular value of \mathbf{B} given in (3), b_m is the number of precision bits, and r is the bound on the absolute value of the secret. In particular, by setting $\omega_i = \exp(\frac{2\pi j i}{N})$ for $i \in [N]$, we have

$$\Delta f(s) \leq c(mt + r)^D 2^{-b_m}.$$

Proof: In order to recover $f(s)$, the system of equations $f(p(\omega_i)) = y_i$, for $i \in [N]$, is solved once all y_i 's are returned. This can be considered as a system of linear equations $\mathbf{z} = \mathbf{B}\mathbf{a}$, as described in (4). Since the minimum singular value of \mathbf{B} is λ_{\min} , then

$$\|\mathbf{a}\| \leq \frac{\|\mathbf{z}\|}{\lambda_{\min}}. \quad (7)$$

Furthermore, one can write

$$\|\mathbf{z}\| = \left(\sum_{i=1}^N |f(y_i)|^2 \right)^{\frac{1}{2}} \leq \sqrt{N} \max_{i \in [N]} |f(y_i)| \quad (8)$$

$$\leq c\sqrt{N} \max_{i \in [N]} \max_{j \in [D] \cup \{0\}} y_i^j \quad (9)$$

$$\leq c\sqrt{N}(mt + r)^D, \quad (10)$$

where (8) is by the definition of \mathbf{z} , (9) is by noting that $f(\cdot)$ is a D -degree polynomial and c is equal to the sum of absolute values of the coefficients of $f(x)$, and (10) is by noting that (1) together with $|s| \leq r$, $|n_i| \leq m$ and $r + tm \geq 1$ imply $y_i \leq r + tm$. Observe that $\Delta f(s) \leq \|\Delta\mathbf{a}\|$ as $f(s)$ is a component of \mathbf{a} . This together with (7) and (10) and utilizing the relative

perturbation bound specified in (5) result in

$$\frac{\Delta f(s)\lambda_{\min}}{c\sqrt{N}(mt+r)^D} \leq \kappa_B \frac{\|\Delta z\|}{\|z\|}. \quad (11)$$

Moreover, since b_m is the number of precision bits, we have

$$\frac{\|\Delta z\|}{\|z\|} \leq 2^{-b_m}. \quad (12)$$

Combining (11) with (12) yields

$$\Delta f(s) \leq c\sqrt{N}(mt+r)^D \frac{\kappa_B}{\lambda_{\min}} 2^{-b_m}. \quad (13)$$

In particular, if the evaluation points are the N -th roots of unity, i.e., $\omega_i = \exp(\frac{2\pi j i}{N})$, the matrix \mathbf{B} turns into a \sqrt{N} times a unitary matrix for which $\kappa_B = 1$ and $\lambda_{\min} = \sqrt{N}$. Hence,

$$\Delta f(s) \leq c(mt+r)^D 2^{-b_m}. \quad (14)$$

Remark 3: The upper bound provided on $\Delta f(s)$ in (13) scales exponentially in the truncation limit of the noise m , with the scaling exponent of D . Note that the ratio of the noise term to $f(s)$ in the computation results of the worker nodes, i.e., $f(y_i)$'s, grows exponentially in the absolute value of the noise which is upper bounded by m . Hence, when the number of decimal precision digits in the implementation is less than $D \log_{10}(\frac{m}{r})$, the precision digits of s are completely lost during the computations performed in the worker nodes. Therefore, one can not recover the true computation outcome in our proposed scheme thereby implying that the exponent D in (13) can not be improved. Note also that the bound in (13) provides a deterministic guarantee on $\Delta f(s)$, but one can also utilize it to obtain probabilistic bounds on the statistics of the computation error in terms of the statistics of the noise by replacing m with a random variable representing the noise value and applying several well-known inequalities such as Markov's inequality, Chebyshev's inequality, etc.

IV. PRIVACY ANALYSIS

In this section, we provide an analysis for the privacy level of data/secret in the proposed distributed computing protocol by considering two well-known notions of security, namely, mutual information security (MIS) and distinguishing security (DS). More specifically, we first consider these metrics of security assuming $t = 1$, i.e., there is no collusion between the computation parties, in Section IV-A. Characterizing the privacy in the presence of t colluding parties when $t > 1$ is studied in Section IV-B. In these two sections it is assumed that the Gaussian distribution of noise terms n_j 's is not truncated, i.e., $m = \infty$. Then, in Section IV-C, the results on the privacy of data are extended to cases with truncated Gaussian distribution for the noise terms.

A. Privacy against a single party

Consider the computational party i for $i \in \{1, 2, \dots, N\}$. Then the amount of information revealed to party i about the

secret s can be measured in terms of the MIS metric, denoted by η_c , and defined as

$$\eta_c \stackrel{\text{def}}{=} \max_i \max_{P_S: |S| < r} I(S; Y_i), \quad (15)$$

where P_S is the probability density function (PDF) of S . DS metric, denoted by η_s , is another metric for security which is defined using the *total variation* (TV) distance metric $D_{\text{TV}}(\cdot, \cdot)$. In general, for any two probability measures P_1 and P_2 on a σ -algebra \mathcal{F} , $D_{\text{TV}}(P_1, P_2)$ is defined as $\sup_{A \in \mathcal{F}} |P_1(A) - P_2(A)|$. While DS metric is often defined for discrete random variables in the cryptography literature, it can be extended to real-valued random variables as follows:

$$\eta_s \stackrel{\text{def}}{=} \max_i \sup_{s_1, s_2 \in \mathbb{D}_S} D_{\text{TV}}(P_{Y_i|S=s_1}, P_{Y_i|S=s_2}), \quad (16)$$

where \mathbb{D}_S is the support of S . Note that both metrics η_c and η_s are non-negative. Also, roughly speaking, the smaller these metrics are the more private the secret s is.

Upper bounding the security metric η_c is discussed next. Since $|S| < r$, as discussed in Section II, we have $E[S^2] \leq r^2$. This together with (15) imply that

$$\begin{aligned} \eta_c &= \max_{P_S: |S| < r} I(S; Y_i) \leq \max_{P_S: E[S^2] \leq r^2} I(S; Y_i) \\ &= \log_2\left(1 + \frac{r^2}{\sigma_n^2}\right), \end{aligned} \quad (17)$$

where the last equality is by the well-known result on the capacity of the additive white Gaussian noise (AWGN) channel [32]. Since the noise variance σ_n^2 can be picked arbitrarily large, one can assume $r = o(\sigma_n)$ to simplify the inequality in (17) as follows:

$$\eta_c \leq \frac{1}{\ln 2} \frac{r^2}{\sigma_n^2} + o\left(\left(\frac{r}{\sigma_n}\right)^2\right). \quad (18)$$

The notion of Hellinger distance, denoted by $H(\cdot, \cdot)$, is useful to bound the DS metric. It is defined as follows:

$$H(P_1, P_2) \stackrel{\text{def}}{=} \frac{1}{2} \int (\sqrt{P_1} - \sqrt{P_2})^2 d\psi. \quad (19)$$

The Hellinger distance can be bounded in terms of the total variation distance as follows [33]:

$$H(P_1, P_2)^2 \leq D_{\text{TV}}(P_1, P_2) \leq \sqrt{2}H(P_1, P_2). \quad (20)$$

Let P_1 and P_2 be the PDFs of two complex Gaussian distributions both with variance σ^2 and means μ_1 and μ_2 , respectively. Also assume that the real and imaginary parts are independent and have identical variances. Then we have [34]

$$H(P_1, P_2) = \sqrt{1 - \exp\left(-\frac{|\mu_1 - \mu_2|^2}{4\sigma^2}\right)}. \quad (21)$$

Using the aforementioned relations, the privacy parameter η_s is bounded in the following theorem.

Theorem 2: The DS metric η_s is bounded as follows:

$$\eta_s \leq \sqrt{2\left(1 - \exp\left(-\frac{r^2}{\sigma_n^2}\right)\right)},$$

where r is the maximum absolute value of the secret s and σ_n^2 is the variance of the noise used in the proposed protocol. In

particular, when $r = o(\sigma_n)$ we have

$$\eta_s \leq \sqrt{2} \frac{r}{\sigma_n} + o\left(\frac{r}{\sigma_n}\right).$$

Proof: Note that the conditional distribution of Y_i given $S = s_i$, specified in (16), is $\mathcal{N}(-s_i, \sigma_n^2)$. Then by using (16) together with (20) and (21) we have

$$\eta_s \leq \sqrt{2(1 - \exp(-\frac{r^2}{\sigma_n^2}))}. \quad (22)$$

In particular, for $r = o(\sigma_n)$, (22) is simplified to

$$\eta_s \leq \sqrt{2} \frac{r}{\sigma_n} + o\left(\frac{r}{\sigma_n}\right). \quad (23)$$

Next, we discuss the relation between the two considered security metrics in the analog domain. It is known that the MIS and DS metrics can be directly related to each other over the space of discrete random variables [35]. In particular, it is shown that [35]:

$$\eta_s \leq \sqrt{2\eta_c}, \quad (24)$$

assuming all the underlying random variables are discrete. We show in the next lemma that this result can be extended to the analog domain.

Lemma 3: The inequality in (24) also holds when the underlying random variables, i.e., the secret as well as observations by parties, are continuous random variables.

Proof: Let X and Y denote two continuous random variables and X^Δ and Y^Δ denote their quantized versions, respectively. Then we have [32]

$$I(X, Y) = \lim_{\Delta \rightarrow 0} I(X^\Delta; Y^\Delta).$$

It can be observed that the same is true for the total variation distance, i.e., $D_{\text{TV}}(X, Y) = \lim_{\Delta \rightarrow 0} D_{\text{TV}}(X^\Delta; Y^\Delta)$. Hence, (24) still holds assuming all the underlying random variables are continuous. ■

Lemma 3 is used to bound η_s later in Section IV-B. Note that one could apply it to derive a bound on η_s using the bound on η_c in (17). However, the resulting bound would be weaker comparing to the result stated in Theorem 2.

Note that the amount of information revealed to a computational party, in terms of either of the security metrics, is an increasing function of $\frac{r}{\sigma_n}$. Furthermore, these metrics approach zero, i.e., the case with the *perfect* privacy ($\eta_s, \eta_c = 0$), as $\sigma_n \rightarrow \infty$. Hence, increasing the noise variance improves the privacy of the scheme, however, this comes at the expense of reducing the precision of the result. This motivates studying the trade-off between the security metrics, as measures of data privacy, and the precision of the computations given a fixed number of bits to represent the floating-point numbers. This is the focus of Section IV-C. In the next section, the results of this section are extended to the case with colluding parties.

B. Privacy against colluding parties

Let t denote the number of colluding parties. The aim is to ensure the privacy of data against any subset of t colluding computational parties. To this end, an upper bound on the

amount of information revealed about the data/secret to the colluding parties is derived. Let $A = \{i_1, \dots, i_t\}$ denote the set of indices for the colluding parties. Then the MIS metric is the mutual information between S and all shares Y_i 's for $i \in A$, in the worst case, i.e.,

$$\eta_c = \max_A I(S; Y_{i_1}, \dots, Y_{i_t}). \quad (25)$$

Next it is shown that this can be upper bounded using the known results on the capacity of a single-input multiple-output (SIMO) channel under power constraints [36], similar to how the upper bound in (17) is obtained using the capacity result of AWGN channel. The input-output relation of a SIMO channel with t receive antennas is as follows:

$$\mathbf{y} = \mathbf{h}x + \mathbf{n},$$

where $x \in \mathbb{C}$ is the transmit signal, $\mathbf{y} \in \mathbb{C}^{t \times 1}$ denotes the vector of received signal, $\mathbf{h} \in \mathbb{C}^{t \times 1}$ denotes the channel coefficient vector, and, $\mathbf{n} \in \mathbb{C}^{t \times 1}$ is the zero-mean complex Gaussian noise vector. The real and the imaginary parts of \mathbf{n} are assumed to be independent. Let \mathbf{N} denote the noise correlation matrix of the SIMO channel, i.e., $\mathbf{N} = \mathbb{E}[\mathbf{n}\mathbf{n}^T]$. Then the capacity is given by [36]

$$C = \log_2(1 + p \|\mathbf{h}\|^2 \nu), \quad (26)$$

where p is the power of transmitted signal and ν is the maximum eigenvalue of \mathbf{N}^{-1} . Consider a SIMO channel with $\mathbf{h}_{t \times 1} = \mathbf{1} \stackrel{\text{def}}{=} (1, \dots, 1)^T$ and the correlated noise terms of $\tilde{\mathbf{n}}_i \stackrel{\text{def}}{=} \sum_{j=1}^t \omega_j^i \mathbf{n}_j$. Then the secret s is mapped to the input of this channel. It can be observed that the shares given to t servers can be mapped to the received symbols in this SIMO channel. Then the average input power is bounded by r^2 , where r is the maximum absolute value of s . Consequently, the capacity of the aforementioned SIMO channel with input power r^2 is an upper bound on the amount of information revealed to the t colluding parties. Note that the coefficients ω_j 's are the N -th roots of the unity, as discussed in Section III. Then it can be observed that $E[\tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_k^*] = -\frac{\sigma_n^2}{t}$, for $j \neq k$, and $E[\tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_j^*] = \sigma_n^2$. Then, similar to (17), one can write

$$\eta_c \leq \log_2\left(1 + \frac{r^2}{\sigma_n^2} t \tilde{\nu}\right), \quad (27)$$

where $\tilde{\nu}$ is the maximum eigenvalue of $\tilde{\mathbf{N}}^{-1}$, where $\tilde{\mathbf{N}} = \frac{t+1}{t} \mathbf{I}_{t \times t} - \frac{1}{t} \mathbf{1}\mathbf{1}^T$. Note that $\tilde{\mathbf{N}}$ has $t-1$ eigenvalues equal to $\frac{t+1}{t}$ and the last one is equal to $\frac{1}{t}$. This implies that $\tilde{\nu} = t$. Substituting this in (27) yields:

$$\eta_c \leq \log_2\left(1 + \frac{r^2 t^2}{\sigma_n^2}\right), \quad (28)$$

providing an upper bound on the amount of information revealed to t colluding parties in terms the MIS metric η_c . In particular, for $r = o(\sigma_n)$ we have

$$\eta_c \leq \frac{t^2}{\ln 2} \frac{r^2}{\sigma_n^2} + o\left(\frac{r^2}{\sigma_n^2}\right). \quad (29)$$

Note that (29) is reduced to (18) for $t = 1$.

Let η_s denote DS metric for this case. By Lemma 3 together

with (28) we have

$$\eta_s \leq \sqrt{2 \log_2(1 + t^2 \frac{r^2}{\sigma_n^2})}. \quad (30)$$

In particular, for $r = o(\sigma_n)$

$$\eta_s \leq \sqrt{\frac{2}{\ln 2}} t \frac{r}{\sigma_n} + o(\frac{r}{\sigma_n}). \quad (31)$$

C. Privacy results with truncated noise

The results provided on the security metrics so far are derived by assuming the additive noise terms n_j 's are drawn from a Gaussian distribution. While this assumption is valid in theory, such terms need to be truncated in practice as they can not be arbitrarily large. Furthermore, as shown in Section III, in order to provide guarantees on the accuracy of the computations, n_j 's need to be bounded, i.e., $|n_j| \leq m$ for some $m \in \mathbb{R}^+$. In this section, we extend the results on bounding the security metrics in the proposed protocol to the case where n_j 's are drawn from a truncated Gaussian probability distribution. To simplify the computation, it is assumed that the truncation threshold is $m = \alpha \frac{\sigma_n}{\sqrt{t}}$, where $\alpha \in \mathbb{R}^+$ and $\frac{\sigma_n}{\sqrt{t}}$ is the standard deviation of the Gaussian distribution.

First, the effect of truncation on the total variation distance metric is analyzed in the general case with t colluding parties. In particular, we show that the change in the DS metric is exponentially small in terms of α . Let η'_s denote the DS metric after truncation of noise terms. Let Ω denote a t -dimensional complex vector space associated with $(y_{i_1}, \dots, y_{i_t})$, where y_{i_j} 's are defined in (1). Let P_{Y_t} and Q_{Y_t} denote the PDFs of $\mathbf{Y}_t \stackrel{\text{def}}{=} (Y_{i_1}, \dots, Y_{i_t})$ given $s = r$ and $s = -r$, respectively, when the noise terms are not truncated. Similarly, \tilde{P}_{Y_t} and \tilde{Q}_{Y_t} are defined when the noise terms are truncated. Also, Let $B_1 = \{\mathbf{y}_t \in \Omega : \tilde{P}_{Y_t}(\mathbf{y}_t) \neq 0\}$, $B_2 = \{\mathbf{y}_t \in \Omega : \tilde{Q}_{Y_t}(\mathbf{y}_t) \neq 0\}$ and $B_{12} = B_1 \cap B_2$.

Note that $\tilde{P}_{Y_t}(\mathbf{y}_t) = \frac{1}{\rho} P_{Y_t}(\mathbf{y}_t)$ and $\tilde{Q}_{Y_t}(\mathbf{y}_t) = \frac{1}{\rho} Q_{Y_t}(\mathbf{y}_t)$, for $\mathbf{y}_t \in B_1$ and $\mathbf{y}_t \in B_2$, respectively, and are zero otherwise, where ρ is given by

$$\rho = \Pr[(|n_1| < m, \dots, |n_t| < m)] \quad (32)$$

$$= \Pr[|n_1| < m]^t \geq (1 - 2 \exp(-\frac{\alpha^2}{2}))^t, \quad (33)$$

where the inequality is by bounding the tail distribution function of the standard normal distribution. One can observe that the TV distance in (16) is maximized when $s_1 = r$ and $s_2 = -r$. Then, using an alternative definition of the total variation distance when the probability measures are over \mathbb{R} we can write:

$$\eta'_s = \frac{1}{2} \int_{\Omega} |\tilde{P}_{Y_t}(\mathbf{y}_t) - \tilde{Q}_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t \quad (34)$$

$$= \frac{1}{2} \int_{B_{12}} |\tilde{P}_{Y_t}(\mathbf{y}_t) - \tilde{Q}_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t \quad (35)$$

$$+ \frac{1}{2} \int_{B_{12}^c} |\tilde{P}_{Y_t}(\mathbf{y}_t) - \tilde{Q}_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t. \quad (36)$$

The term in (35) is bounded as follows:

$$\int_{B_{12}} |\tilde{P}_{Y_t}(\mathbf{y}_t) - \tilde{Q}_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t = \frac{1}{\rho} \int_{B_{12}} |P_{Y_t}(\mathbf{y}_t) - Q_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t \quad (37)$$

$$\leq \frac{1}{\rho} \int_{\Omega} |P_{Y_t}(\mathbf{y}_t) - Q_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t = \frac{1}{\rho} \eta_s, \quad (38)$$

where (38) is by noting that $B_{12} \subset \Omega$. In order to derive an upper bound on the term in (36) note that

$$\int_{B_{12}^c} |\tilde{P}_{Y_t}(\mathbf{y}_t) - \tilde{Q}_{Y_t}(\mathbf{y}_t)| d\mathbf{y}_t \quad (39)$$

$$\leq \int_{B_{12}^c} \tilde{P}_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t + \int_{B_{12}^c} \tilde{Q}_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t \quad (40)$$

$$= 2 \int_{B_{12}^c} \tilde{P}_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t, \quad (41)$$

where (41) is due to symmetry. An upper bound on the term in (41) is derived in the following lemma.

Lemma 4: We have

$$\int_{B_{12}^c} \tilde{P}_{Y_t}(\mathbf{y}_t) \leq (2 \exp(-\frac{1}{2}(\alpha - \frac{2r\sqrt{t}}{\sigma_n})^2))^t.$$

Proof: Let P'_{Y_t} denote the PDF of the random vector \mathbf{Y}_t given $s = r$ and assuming n_i 's are drawn from a truncated Gaussian distribution with threshold $m - 2r$. Similar to the definition of B_1 , let $B'_1 \stackrel{\text{def}}{=} \{\mathbf{y}_t \in \Omega : p'_{Y_t}(\mathbf{y}_t) \neq 0\}$. Since the equations relating y_i 's and n_i 's in (1) are linear, then it can be observed that $B'_1 \subset B_{12}$. Then we have

$$\int_{B_{12}^c} \tilde{P}_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t \leq \int_{B_{12}^c} \frac{1}{\rho} P_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t \quad (42)$$

$$\leq \frac{1}{\rho} \int_{B_{12}^c} P_{Y_t}(\mathbf{y}_t) d\mathbf{y}_t \leq \frac{1}{\rho} (2 \exp(-\frac{1}{2}(\alpha - \frac{2r\sqrt{t}}{\sigma_n})^2))^t \quad (43)$$

where (42) holds because $\tilde{P}_{Y_t}(\mathbf{y}_t)$ is either equal to $\frac{1}{\rho} P_{Y_t}(\mathbf{y}_t)$ or zero, the first inequality in (43) holds since $B'_1 \subset B_{12}$ implies $B_{12}^c \subset B_{12}^c$, and the second one is by bounding the tail distribution function of the standard normal distribution ■

Theorem 5: The DS metric for the case where n_j 's are drawn from a truncated Gaussian distribution with truncation level $\alpha \frac{\sigma_n}{\sqrt{t}}$ satisfies the following inequality:

$$\eta'_s \leq \frac{1}{\rho} \eta_s + \frac{1}{\rho} (2 \exp(-\frac{1}{2}(\alpha - \frac{2r\sqrt{t}}{\sigma_n})^2))^t,$$

where $\rho \geq (1 - 2 \exp(-\frac{\alpha^2}{2}))^t$.

Proof: The proof follows by (34) together with bounding (35) using (38) and (36) using the result of Lemma 4, respectively. ■

Theorem 5 implies that picking, for instance, $\alpha = 10$ with $t = 10$, and already having a very small $\frac{r}{\sigma_n}$ is sufficient to obtain almost the same bound on the DS metric as in the case where the noise terms are not truncated. Hence, truncation of the noise terms in (1) does not compromise the privacy of data in the protocol as long as α is picked sufficiently large.

In order to obtain a similar result for the MIS metric, a result on the capacity of channels with additive truncated Gaussian

$\log_{10}(\sigma_n)$	5	11	14
$\log_{10}(\Delta f(s))$	-9.44	-4.44	-0.45
$\log_{10}(\eta_s)$	-2.36	-8.36	-11.36

TABLE I: Demonstration of the trade-off between DS security metric and accuracy. The upper bound on η_s in (30) is calculated versus the upper bound on $\Delta f(s)$ obtained in Theorem 1 by at $\sigma_n = 10^5, 10^{10}, 10^{14}$. Other parameters are $c = 1, t = 1, D = 1, \alpha = 10, r = 255$ and $b_m = 52$.

noise is needed. This problem is studied recently, see, e.g., [37]. In particular, it is shown that the capacity of AWGN channel is *robust* against truncation of the noise. More specifically, it is shown that the change in the capacity by truncating the noise is $O(\exp(-\frac{\alpha^2}{2}))$ [37]. Hence, the MIS metric is increased by at most $O(\exp(-\frac{\alpha^2}{2}))$ when truncating the noise, mimicking the result derived for the DS metric in Theorem 5.

In Table I, the trade-off between the privacy and the accuracy of our protocol is demonstrated using the theoretical results obtained in Section III and Section IV. It can be observed that increasing the variance of the noise σ_n improves the privacy but at the same time reduces the accuracy of the computations.

Remark 4: Note that the results provided in this section are derived over the continuous space of complex numbers. The well-known data processing inequality for mutual information guarantees that the same inequalities still hold when Algorithm 1 is utilized using floating-point implementation in practice.

V. EXPERIMENTS

In this section, we demonstrate experiment results on the performance of our proposed protocol when applied to a certain learning algorithm. First, it is shown that the accuracy of the results obtained by using our protocol in a distributed setting closely follows that of a conventional centralized approach, thereby providing almost the same accuracy as in the centralized approach. Second, the performance of our protocol is compared with that of the state-of-the-art schemes employing fixed-point numbers by quantizing the data and mapping it to finite field elements. In particular, we compare our protocol with CodedPrivateML [13] in terms of accuracy and run time.

The problem of training a logistic regression (LR) model over GISETTE [38] dataset is considered. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ denote a dataset consisting of m samples with d features and $\mathbf{l} \in \{0, 1\}^m$ denote the corresponding label vector. The task is to compute the model parameters (weights) $\mathbf{h} \in \mathbb{R}^d$ by iteratively minimizing the cross entropy function using the following parameter update equation:

$$\mathbf{h}^{(j+1)} = \mathbf{h}^{(j)} - \frac{\beta}{m} \mathbf{X}^T (g(\mathbf{X}\mathbf{h}^{(j)}) - \mathbf{l}), \quad (44)$$

where $\mathbf{h}^{(j)}$ is the estimated parameters in iteration j , β is the learning rate, and $g(x) \stackrel{\text{def}}{=} \frac{1}{1 + \exp(-x)}$ is the sigmoid function that operates element-wise over the vector inputs. For each data point $\mathbf{x}_i \in \mathbb{R}^{1 \times d}$, the estimated probability of l_i being equal to 1 is $g(\mathbf{x}_i \mathbf{h})$. All experiments are performed in MATLAB and the considered problem is the binary classification between the highly confusable digits 4 and 9 over GISETTE dataset. The train dataset consists of 6000 samples with 5000 attributes. Our

protocol for the distributed training of the LR model is inspired by Algorithm 1 and is described in Algorithm 2. This protocol is implemented using the default double-precision floating-point (FLP) representation in MATLAB with 64 bits, where $b_m = 52$, $b_e = 11$, and the other bit is reserved for the sign.

Algorithm 2 Privacy-preserving distributed training of logistic regression model in the analog domain.

Input: Dataset $\mathbf{X} \in \mathbb{R}^{m \times d}$, the number of iterations k and α .

Public parameters: $(\omega_1, \dots, \omega_N)$, $\tilde{\mathbf{b}}_{1 \times N} = (\tilde{b}_1, \dots, \tilde{b}_N)$.

Output: Parameter vector \mathbf{h} for the logistic regression model.

Encoding dataset (at the master):

Pick i.i.d. $N_j \in \mathbb{R}^{m \times d}$ with entries independently drawn from $\mathcal{N}(0, \frac{\sigma_n^2}{t})$ truncated at $\alpha \frac{\sigma_n}{\sqrt{t}}$, for $j = 1, \dots, t$.

for $i \in [N]$ **do**

 Compute $\tilde{\mathbf{X}}_i = \mathbf{X} + \sum_{j=1}^t \omega_i^j N_j$.

end

Send $\tilde{\mathbf{X}}_i$ to server i .

Computation of \mathbf{h} iteratively:

Set $\mathbf{h}^{(0)} = \mathbf{0}$.

for $j \in \{0, \dots, k-1\}$ **do**

Encoding phase (at the master):

 Pick i.i.d. $\mathbf{n}_j \in \mathbb{R}^{1 \times d}$ with entries independently drawn from $\mathcal{N}(0, \frac{\sigma_n^2}{t})$ truncated at $\alpha \frac{\sigma_n}{\sqrt{t}}$, for $j = 1, \dots, t$.

 Compute $\tilde{\mathbf{h}}_i^{(j)} = \mathbf{h}^{(j)} + \sum_{h=1}^t \omega_i^h \mathbf{n}_h$.

 Send $\tilde{\mathbf{h}}_i^{(j)}$ to server i .

Computation phase (at server i):

 Compute $\mathbf{z}_i = \tilde{\mathbf{X}}_i^T \tilde{\mathbf{X}}_i \tilde{\mathbf{h}}_i^{(j)}$.

 Send \mathbf{z}_i to the master node.

Decoding phase (at the master):

 Compute $\mathbf{u}^{(j)} = \sum_{i=1}^N \tilde{b}_i \mathbf{z}_i$.

 Update $\mathbf{h}^{(j+1)} = \mathbf{h}^{(j)} - \frac{\beta}{2m} [\frac{1}{2} \mathbf{u}^{(j)} + \mathbf{X}^T (\mathbf{1} - 2\mathbf{l})]$.

end

Return $\mathbf{h} = \mathbf{h}^{(k)}$.

Next, we describe the steps in Algorithm 2 in details. In the beginning, the data matrix \mathbf{X} is encoded element-wise using the analog counterpart of Shamir's encoder, same as in (1), and then the secret shares are sent to the servers. Let $\tilde{\mathbf{X}}_i$ denote the share sent to server i , for $i \in [N]$. The initial parameter vector is set to the all-zero vector, i.e., $\mathbf{h}^{(0)} = \mathbf{0}$. Let k denote the total number of iterations for updating the model parameters using (44) in the experiment. In the j -th iteration, for $j \in \{0, \dots, k-1\}$, the master node encodes $\mathbf{h}^{(j)}$ element-wise, again same as in (1), and sends the shares to the servers. Let $\tilde{\mathbf{h}}_i^{(j)}$ denote the share of $\mathbf{h}^{(j)}$ sent to server i . The server i then computes $\tilde{\mathbf{X}}_i^T \tilde{\mathbf{X}}_i \tilde{\mathbf{h}}_i^{(j)}$ and returns the result to the master node. Next, the master node recovers $\mathbf{X}^T \mathbf{X} \mathbf{h}^{(j)}$ by computing a linear combination of the returned results, same as in the decoding phase in Algorithm 1, and utilizes it to update the vector of parameters according to (44) with the sigmoid function substituted by its 1-degree polynomial approximation, i.e., $g(x) \approx \frac{1}{2} + \frac{x}{4}$. This procedure is continued till the desired number of iterations is passed and the last update of the parameter vector is returned as the final result of the protocol. It is worth mentioning that the data

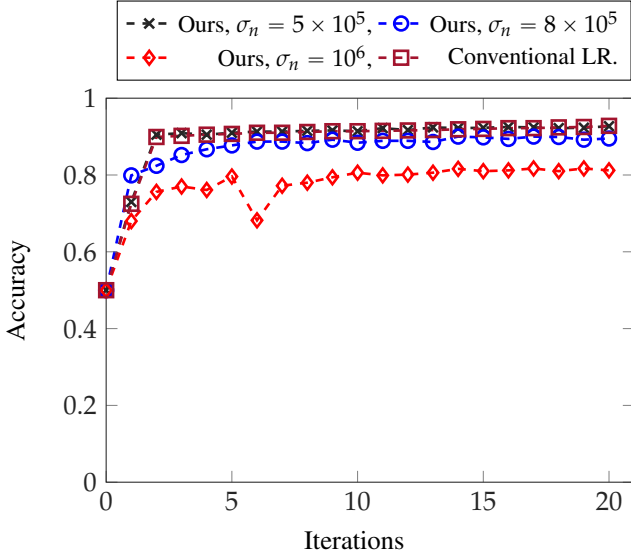


Fig. 1: Comparison between the accuracy of our distributed learning protocol for several values of σ_n and the conventional centralized logistic regression (LR).

matrix \mathbf{X} is secret-shared only once at the beginning and the same shares are used at each iteration by the servers while the parameter vector \mathbf{h} is updated and secret-shared in each iteration.

The vector of model parameters \mathbf{h} for the training dataset is computed using Algorithm 2 as well as using the conventional centralized method. The number of servers $N = 4$ and $t = 1$ are assumed. Note that in the centralized method the sigmoid function is not approximated while in our implementation it is approximated with a degree-1 polynomial. Then the accuracy of the predictions are determined over the GISETTE validation dataset in both approaches. The result is shown in Figure 1. It can be observed that the accuracy of our protocol with $\sigma_n = 5 \times 10^5$ closely follows that of the conventional centralized approach. The accuracy of our protocol is also demonstrated for $\sigma_n = 8 \times 10^5, 10^6$ to experimentally illustrate the implications of the aforementioned privacy-accuracy trade-off in Section IV.

It this setting with honest-but-curious servers, as mentioned in Section I-A, the servers may attempt to infer the data by accumulating all received shares during all iterations. Since \mathbf{n}_h 's in Algorithm 2 are picked independently in each iteration, the leakage of information for the model in terms of DS metric is bounded by $k\eta_s$, where k is the number of iterations and η_s is characterized in (23) for $t = 1$ and in (30) for $t > 1$. Furthermore, the privacy guarantee for the dataset in terms of the DS metric is given by (23) for $t = 1$ and by (30) for $t > 1$, regardless of the value of k since the dataset is encoded and sent to the servers only once during the protocol. Hence, given all the parameters in the described experiment, the privacy guarantee in our protocol in terms of the DS metric is $\eta_s \leq 2.5 \times 10^{-4}$ for the model and $\eta_s \leq 10^{-5}$ for the dataset. These hold by utilizing (23), where $\sigma_n = 5 \times 10^5$ is picked, and setting $l \leq 20$ in all experiments while noting that $d = 784$ and the maximum absolute value of data is $r = 1$ in the normalized GISETTE

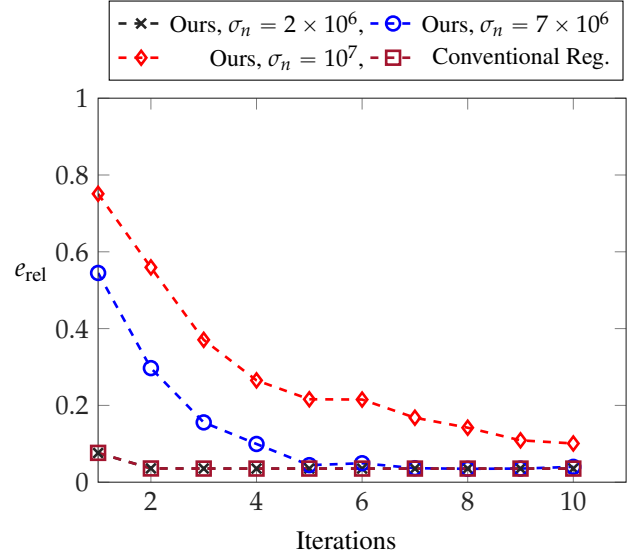


Fig. 2: Comparison between the relative error e_{rel} of our distributed learning protocol for several values of σ_n and the conventional centralized linear regression.

dataset.

Next, we consider the problem of training a linear regression model over Combined Cycle Power Plant dataset [39]. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ denote a dataset consisting of m samples with d features and $\mathbf{l} \in \mathbb{R}^m$ denote the corresponding label vector. The task is to compute the model parameters (weights) $\mathbf{h} \in \mathbb{R}^d$ using the following parameter update equation:

$$\mathbf{h}^{(j+1)} = \mathbf{h}^{(j)} - \frac{\beta}{m} \mathbf{X}^T (\mathbf{X} \times \mathbf{h}^{(j)} - \mathbf{y}), \quad (45)$$

where $\mathbf{h}^{(j)}$ is the estimated parameters in iteration j and β is the learning rate. The dataset contains 9568 samples that is split to train and test datasets with size 7568 and 2000, respectively. The relative error of the predictions over the test dataset is defined as $e_{\text{rel}} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|}{\|\mathbf{y}\|}$, where \mathbf{y} is the vector of true labels in the test dataset and $\hat{\mathbf{y}}$ is the vector of predicted labels. The relative error in our protocol is compared to that of the centralized conventional linear regression algorithm in Figure 2. The number of servers $N = 4$ and $t = 1$ are assumed. It can be observed that the accuracy of our protocol with $\sigma_n = 5 \times 2 \times 10^6$ closely follows that of the conventional centralized approach. The accuracy of our protocol is also demonstrated for $\sigma_n = 7 \times 10^6, 10^7$ to experimentally illustrate the aforementioned privacy-accuracy trade-off in the problem of training a linear regression model.

Next, the problem of training a one layer neural network with softmax activation function over MNIST dataset is considered [40]. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ denote a dataset consisting of m samples with d features. Let $\mathbf{x}_i \in \mathbb{R}^{d \times 1}$ denote the i 'th sample in \mathbf{X} and $\mathbf{y}_i \in \mathbb{R}^{1 \times K}$ denote the corresponding categorical label vector, where K denotes the number of classes in the dataset which is equal to 10 for the MNIST dataset. The task is to compute the model parameters $\mathbf{W} \in \mathbb{R}^{d \times K}$ using the following parameter

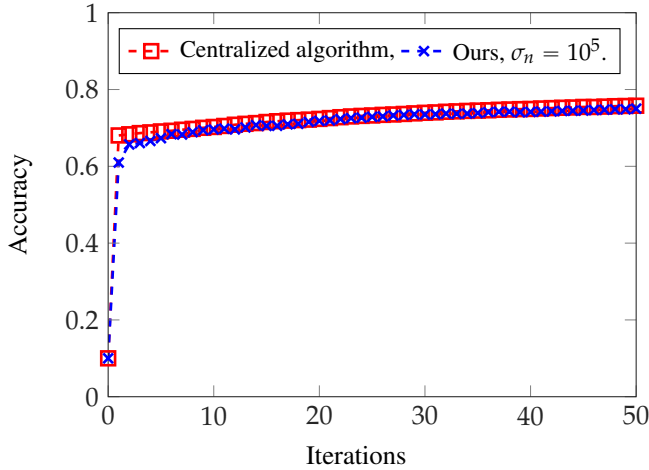


Fig. 3: Comparison between the accuracy of our distributed learning protocol and the conventional centralized algorithm for multi-class softmax classifier.

update equation:

$$\mathbf{W}^{(j+1)} = \mathbf{W}^{(j)} - \frac{\beta}{m} \sum_{i=1}^m \mathbf{x}_i (s(\mathbf{x}_i^T \mathbf{W}) - \mathbf{y}), \quad (46)$$

where $\mathbf{W}^{(j)}$ is the estimated parameters in iteration j , β is the learning rate, and $s(\mathbf{x})$ is the softmax function. The accuracy in our protocol is compared to that of the centralized conventional algorithm in Figure 3. The number of servers $N = 4$ and $t = 1$ are assumed. It can be observed that the accuracy of our protocol with $\sigma_n = 10^5$ closely follows that of the conventional centralized approach. Note that in our protocol the softmax function is substituted by its 1-degree polynomial approximation.

In the last experiment, the accuracy of a fixed-point (FXP) implementation, according to the protocol proposed in CodedPrivateML [13], is simulated in a similar scenario with $N = 4$ and $t = 1$, and is compared with that of our protocol. In CodedPrivateML, the data is first quantized and then mapped to the elements of a finite field. Then, instead of encoding the data using Shamir’s scheme, it utilizes Lagrange interpolation polynomial to encode K data matrices/secrets simultaneously to leverage parallelization in computations. However, in order to have a fair comparison with our results, we pick $K = 1$ in CodedPrivateML that is equivalent to the baseline scheme where the data is encoded using Shamir’s scheme after quantization. In other words, for the considered case, our protocol utilizes the analog counterpart of Shamir’s scheme in the complex domain whereas CodedPrivateML leverages the conventional Shamir’s scheme over finite field. All other parameter are picked according to what is reported in [13], which also uses 64 bits to represent elements of the finite field. The problem of training a logistic regression (LR) model over MNIST dataset for binary classification between the digits 3 and 7 is considered. Figure 4 demonstrates that the accuracy of CodedPrivateML (fixed point) is significantly dropped to around 0.5, equivalent to that of a random guessing, when the size of dataset exceeds

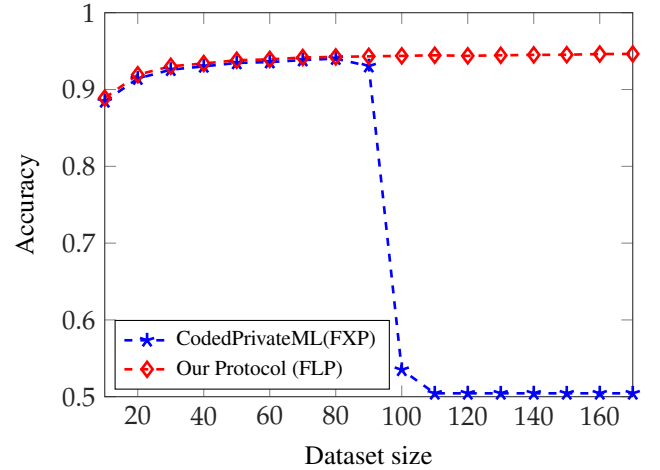


Fig. 4: Comparison between the accuracy of our protocol and CodedPrivateML [13] implementations. The number of iterations in both cases is 15.

100. Note that the original train and test datasets consist of 12396 and 2038 samples, respectively. In order to observe the performance with small dataset sizes, we pick a dataset with equal data points labeled with 3 and 7 in each experiment. Also, we run the experiment 1000 times by picking different sets of samples and the average accuracy is reported in Figure 4.

This comparison demonstrates the superiority of our proposed protocol in the analog domain and implemented using floating point numbers comparing to the state-of-the-art distributed computing and learning schemes employing quantization followed by computations over a finite field. In other words, our protocol is *robust* with respect to the size of the training dataset while the fixed-point implementations suffer significantly from wrap-around error as the size of dataset passes a certain threshold depending on the prime number picked as the size of underlying finite field.

One major advantage of CodedPrivateML over MPC-based approaches is that it provides an order of magnitude speed up, based on the experiment results reported in [13]. The reason is that in CodedPrivateML, there is no communication between computation parties thereby improving the communication complexity of the scheme significantly, compared with the state-of-the-art cryptographic approaches. This advantage is preserved in our protocol as well, since no communication is needed between the parties.

Note that in order to avoid the wrap-around error in the fixed-point implementation each computation party should stop the computation before the the wrap-around threshold is passed and divide the computation task into smaller subtasks. Then, it needs to send back all the computation results associated to each subtask to the master node in order to guarantee recovery of the computation result. This results in an excess communication and computation overhead compared with our protocol. Moreover, since the threshold is not known *a priori*, one always needs to check if the wrap-around is occurred during the computation process. These factors slow down CodedPrivateML when the dataset is large and one wants to avoid

the errors due to wrap-around. In Table II, the computation times of CodedPrivateML and our protocol are compared for the experiment discussed in this section for different dataset sizes, while discarding the delay in CodedPrivateML due to frequently checking wrap-around errors and communication overhead. It shows that for the same level of accuracy of the results, our approach with the floating-point implementation also outperforms the fixed-point implementations while preserving the speed up advantage compared with the MPC-based schemes.

Dataset size	CodedPrivateML	Our Protocol
1000	0.72	0.25
2000	1.49	0.52
3000	2.49	0.80
4000	3.87	1.09
5000	5.94	1.38

TABLE II: Comparison of the run times between the fixed-point and the floating-point implementations. The times are reported in seconds. The experiments are done on a Macbook pro with 3.5 GHz dual-core intel core i7 CPU and 16 GB memory.

VI. CONCLUSION AND DISCUSSIONS

In this paper, we tackled the critical problems of privacy-preserving computation and learning over a real-valued dataset using distributed honest-but-curious servers. To this end, we proposed a protocol that utilizes a counterpart of Shamir’s secret sharing scheme in the analog domain. In order to measure the privacy level of the data, the conventional notion of distinguishing security is extended to the analog domain and privacy guaranties for the proposed scheme are characterized based on this security metric. The well-known connection between the DS and the MIS measures of security is extended from the discrete domain to the continuous domain. This is then utilized to bound the DS metric of our protocol using well-known results on the capacity of SIMO channel with correlated noise. Furthermore, the accuracy of the outcome of the computation is characterized assuming a floating-point implementation of the protocol. In our experiments, we illustrated that the accuracy of the predictions for the logistic regression model over the MNIST dataset derived by our protocol closely follows that of the conventional centralized approach. Finally, we showed that our protocol is robust with respect to the size of the training dataset, i.e., there is almost no accuracy loss as the size of the training dataset grows large, while the performance of the fixed-point implementations in prior work significantly diminishes due to overflow errors.

There are several directions for future work. Extending the proposed protocol in this paper to scenarios with straggler servers is an interesting direction for future research. More specifically, in our protocol it is assumed that all the servers successfully finish their assigned tasks, while a certain number of servers, referred to as stragglers, may be slow or may not respond at all in practice [41]–[46]. The main challenge in this direction is to pick the parameters of the protocol and to design the decoder that is better than the naive and numerically unstable approach of solving a system of linear equations in

the analog domain. Another direction is to adopt the proposed protocol in this paper to perform computational tasks in distributed fashion for other applications, such as distributed optimization and mechanism design [47]–[51], while keeping the data private. Generalizing Algorithm 1 in order to simultaneously compute multiple evaluations of a polynomial in a single-shot is another future direction. To this end, techniques for multi-user secret sharing can be utilized [52]. Obtaining such results can potentially lead to privacy-preserving multi-task learning protocols, i.e., protocols that train multiple models over a dataset in a single round.

REFERENCES

- [1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [2] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” *Health information science and systems*, vol. 2, no. 1, p. 3, 2014.
- [3] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, “Big data: the management revolution,” *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.
- [4] C. E. Shannon, “Communication theory of secrecy systems,” *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [5] S. Goldwasser and S. Micali, “Probabilistic encryption,” *Journal of computer and system sciences*, vol. 28, no. 2, pp. 270–299, 1984.
- [6] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [7] M. Ben-Or, S. Goldwasser, and A. Wigderson, “Completeness theorems for non-cryptographic fault-tolerant distributed computation,” in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 1–10.
- [8] Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database,” 2010.
- [9] S. Wagh, D. Gupta, and N. Chandran, “SecureNN: Efficient and private neural network training,” *IACR Cryptology ePrint Archive*, vol. 2018, p. 442, 2018.
- [10] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, “Lagrange coded computing: Optimal design for resiliency, security, and privacy,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 1215–1225.
- [11] M. Dahl, J. Mancuso, Y. Dupis, B. Decoste, M. Giraud, I. Livingstone, J. Patriquin, and G. Uhma, “Private machine learning in tensorflow using secure computation,” *arXiv preprint arXiv:1810.08130*, 2018.
- [12] A. Barak, D. Escudero, A. P. Dalaskov, and M. Keller, “Secure evaluation of quantized neural networks,” *IACR Cryptology ePrint Archive*, vol. 2019, p. 131, 2019.
- [13] J. So, B. Guler, A. S. Avestimehr, and P. Mohassel, “CodedPrivateML: A fast and privacy-preserving framework for distributed machine learning,” *arXiv preprint arXiv:1902.00641*, 2019.
- [14] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, “Cryptflow: Secure tensorflow inference,” *arXiv preprint arXiv:1909.07814*, 2019.
- [15] Q. Yu and A. S. Avestimehr, “Entangled polynomial codes for secure, private, and batch distributed matrix multiplication: Breaking the “cubic” barrier,” *arXiv preprint arXiv:2001.05101*, 2020.
- [16] M. Aliasgari, O. Simeone, and J. Kliewer, “Private and secure distributed matrix multiplication with flexible communication load,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2722–2734, 2020.
- [17] R. G. D’Oliveira, S. El Rouayheb, and D. Karpuk, “GASP codes for secure distributed matrix multiplication,” *IEEE Transactions on Information Theory*, vol. 66, pp. 4038–4050, 2020.
- [18] R. Bitar, Y. Xing, Y. Keshkarjahromi, V. Dasari, S. E. Rouayheb, and H. Seferoglu, “Private and rateless adaptive coded matrix-vector multiplication,” *arXiv preprint arXiv:1909.12611*, 2019.
- [19] H. A. Nodehi and M. A. Maddah-Ali, “Secure coded multi-party computation for massive matrix operations,” *arXiv preprint arXiv:1908.04255*, 2019.

- [20] M. Fahim and V. R. Cadambe, "Numerically stable polynomially coded computing," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 3017–3021.
- [21] A. Ramamoorthy and L. Tang, "Numerically stable coded matrix computations via circulant and rotation matrix embeddings," *arXiv preprint arXiv:1910.06515*, 2019.
- [22] A. B. Das and A. Ramamoorthy, "Distributed matrix-vector multiplication: A convolutional coding approach," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 3022–3026.
- [23] N. Charalambides, H. Mahdaviyar, and A. O. Hero III, "Numerically stable binary gradient coding," *arXiv preprint arXiv:2001.11449*, 2020.
- [24] M. Soleymani, M. V. Jamali, and H. Mahdaviyar, "Coded computing via binary linear codes: Designs and performance limits," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 879–892, 2021.
- [25] T. Jahani-Nezhad and M. A. Maddah-Ali, "Berrut approximated coded computing: Straggler resistance beyond polynomial computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [26] M. Soleymani, R. E. Ali, H. Mahdaviyar, and A. S. Avestimehr, "Approxifier: A model-agnostic approach to resilient and robust prediction serving systems," *arXiv preprint arXiv:2109.09868*, 2021.
- [27] S. Setty, V. Vu, N. Panpalia, B. Braun, A. J. Blumberg, and M. Walfish, "Taking proof-based verified computation a few steps closer to practicality," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 253–268.
- [28] M. Aliasgari, M. Blanton, Y. Zhang, and A. Steele, "Secure computation on floating point numbers," in *NDSS*, 2013.
- [29] O. Catrina, "Towards practical secure computation with floating-point numbers," in *3rd Annual International Conference on Cryptography and Information Security*, 2018.
- [30] M. Soleymani, H. Mahdaviyar, and A. S. Avestimehr, "Analog lagrange coded computing," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 283–295, 2021.
- [31] J. W. Demmel, *Applied numerical linear algebra*. Siam, 1997, vol. 56.
- [32] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [33] C. Kraft, "Some conditions for consistency and uniform consistency of statistical procedures," *University of California Publication in Statistics*, vol. 2, pp. 125–141, 1955.
- [34] L. Pardo, *Statistical inference based on divergence measures*. Chapman and Hall/CRC, 2018.
- [35] M. Bellare, S. Tessaro, and A. Vardy, "A cryptographic treatment of the wiretap channel," *Advances in Cryptology – CRYPTO*, 2012.
- [36] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [37] M. Egan, S. M. Perlaza, and V. Kungurtsev, "Capacity sensitivity in continuous channels," 2017.
- [38] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," *Advances in neural information processing systems*, vol. 17, 2004.
- [39] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [41] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.
- [42] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "A unified coding framework for distributed computing with straggling servers," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016, pp. 1–6.
- [43] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.
- [44] A. Reiszadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4227–4242, 2019.
- [45] M. Aliasgari, J. Kliewer, and O. Simeone, "Coded computation against straggling decoders for network function virtualization," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 711–715.
- [46] M. V. Jamali, M. Soleymani, and H. Mahdaviyar, "Coded distributed computing: Performance limits and code designs," in *2019 IEEE Information Theory Workshop (ITW)*. IEEE, 2019, pp. 1–5.
- [47] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for unicast transmission," in *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018, pp. 1–6.
- [48] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 20–27.
- [49] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for network resource allocation problems," *IEEE Transactions on Network Science and Engineering*, 2019.
- [50] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," *arXiv preprint arXiv:1806.02246*, 2018.
- [51] N. Heydaribeni and A. Anastasopoulos, "Distributed mechanism design for multicast transmission," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4200–4205.
- [52] M. Soleymani and H. Mahdaviyar, "Distributed multi-user secret sharing," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1141–1145.



distributed storage and computing systems, wireless networks, and machine learning.



He received the NSF career award in 2020. He also received Best Paper Award in 2015 IEEE International Conference on RFID, and the 2013 Samsung Best Paper Award. He also received two Silver Medals at the International Mathematical Olympiad in 2002 and 2003, and two Gold Medals at Iran National Mathematical Olympiad in 2001 and 2002. His main area of research is coding and information theory with applications to wireless communications, storage systems, security, and privacy.



distributed computing and machine learning, secure and private computing, and blockchain systems.

Dr. Avestimehr has received a number of awards for his research, including the James L. Massey Research & Teaching Award from IEEE Information Theory Society, an Information Theory Society and Communication Society Joint Paper Award, a Presidential Early Career Award for Scientists and Engineers (PECASE) from the White House (President Obama), a Young Investigator Program (YIP) award from the U. S. Air Force Office of Scientific Research, a National Science Foundation CAREER award, the David J. Sakrison Memorial Prize, and several Best Paper Awards at Conferences. He has been an Associate Editor for IEEE Transactions on Information Theory and a general Co-Chair of the 2020 International Symposium on Information Theory (ISIT).

Mahdi Soleymani (Student Member, IEEE) received his B.S. and M.S. degrees in Electrical Engineering at Sharif University of Technology, Tehran, Iran, in 2014 and 2016, respectively. He is currently pursuing his Ph.D. degree in Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. He received the Honourable Mention Award at the International Physics Olympiad (IPhO) in 2010 and a Gold Medal at Iran National Physics Olympiad in 2009. His research interests lie in the area of coding and information theory with applications to distributed storage and computing systems, wireless networks, and machine learning.

Hessam Mahdaviyar (Member, IEEE) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan Ann Arbor. He received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and the Ph.D. degrees from the University of California San Diego (UCSD), La Jolla, in 2009, and 2012, respectively, all in electrical engineering. He was with the Samsung US R&D between 2012 and 2016, in San Diego, US, as a staff research engineer.

A. Salman Avestimehr (Fellow, IEEE) is a Professor and director of the Information Theory and Machine Learning (vITAL) research lab at the Electrical and Computer Engineering Department of University of Southern California. He received his Ph.D. in 2008 and M.S. degree in 2005 in Electrical Engineering and Computer Science, both from the University of California, Berkeley. Prior to that, he obtained his B.S. in Electrical Engineering from Sharif University of Technology in 2003. His research interests include information theory and coding theory, and large-scale