Information Theoretic Private Inference in Quantized Models

Netanel Raviv, **Rawad Bitar**, and **Eitan Yaakobi**

***Abstract*—In a *Private Inference* scenario, a server holds a model (e.g., a neural network), a user holds data, and the user wishes to apply the model on her data. The privacy of both parties must be protected; the user's data might contain confidential information, and the server's model is his intellectual property.**

Private inference has been studied extensively in recent years, mostly from a cryptographic perspective by incorporating homomorphic encryption and multiparty computation protocols, which incur high computational overhead and degrade the accuracy of the model. In this work we take a perpendicular approach which draws inspiration from the expansive *Private Information Retrieval* literature. We view private inference as the task of retrieving an inner product of a parameter vector with the data, a fundamental step in most machine learning models.

By combining binary arithmetic with real-valued one, we present a scheme which enables the retrieval of the inner product for models whose weights are either binarized, or given in fixed-point representation; such models gained increased attention recently, due to their ease of implementation and increased robustness. We also present a fundamental trade-off between the privacy of the user and that of the server, and show that our scheme is optimal in this sense. Our scheme is simple, universal to a large family of models, provides clear information-theoretic guarantees to both parties with zero accuracy loss, and in addition, is compatible with continuous data distributions and allows infinite precision.

***Index Terms*—Private inference, Private computation, Private information retrieval.**

I. Introduction

Loss of privacy in the information age raises various concerns, ranging from mental health issues to the future of democracy [2]. Individual users must choose between surrendering their personal data to service providers, or be left without the ability to perform basic day-to-day activities. On the other hand, training machine learning models is a laborious task which requires knowledge and resources. As such, the models themselves are the intellectual property of their owners, and should be kept private as well.

One such setting in which privacy is a concern is the *inference* phase of the machine learning pipeline. In this setting, a server (e.g., a service provider) holds an already-trained model (e.g., a neural network, a logistic/linear regression model, a linear classifier, etc.), and offers the use of this model to users in exchange for a fee. A user that wishes to make use of the model exchanges information with the server in order to facilitate the inference of the model on her input. A *private inference* protocol is one which this exchange of information provides some privacy guarantees for both parties.

Private inference has been studied extensively in recent years, mostly from a cryptographic perspective, by using primitives such as homomorphic encryption [6, 9] and multiparty computing [14, 18]. Each of these techniques has its limitations, such as high overhead for non-polynomial functions, degraded accuracy, multiple rounds, model structure which must be public, etc., see [1] for a thorough review on the topic. In this paper we take a perpendicular direction and develop an information-theoretic approach to private inference. Among the benefits of the latter over the former are resilience against computationally unbounded adversaries, clear privacy guarantees, simplicity, and compatibility with infinite precision computations, as delineated next.

Our approach begins with the simple observation that computation in most machine-learning models relies on first extracting one or more *signals* from the data, i.e., values of the form $\mathbf{w}\mathbf{x}^\intercal$, where $\mathbf{w}$ is a vector of weights associated with the model, $\mathbf{x}$ is a vector of features, and both vectors are usually real-valued. This is the case in linear classification ($\mathrm{sign}(\mathbf{x}\mathbf{w}^\intercal)$), linear regression ($\mathbf{x}\mathbf{w}^\intercal$), logistic regression ($\frac{1}{1+\exp(-\mathbf{x}\mathbf{w}^\intercal)}$), neural networks with $\mathrm{sign}, \mathrm{ReLU}$, or any sigmoid activation functions, and many more.

Next, in order to provide clear information-theoretic guarantees, specifically from the server's side, we leverage the recently popularized notion of *binarized* (or more generally, quantized) models. It has been shown that by restricting the values of $\mathbf{w}$ to ± 1, one can obtain significant gains in terms of the simplicity of implementation [13], as well as in terms of resilience against adversarial perturbations [5, 15, 17], while keeping the models efficiently trainable [10]. From the user's side, we consider the data distribution to be continuous, and provide privacy guarantees in terms of the dimension on which the server knows the data; an explanation for this approach is given in the sequel using the theory of independent components [11].

Finally, we note that our approach draws inspiration from the vast *private information retrieval* (PIR) literature. In PIR, a user wishes to retrieve an entry from a distributed dataset, while keeping the identity of that entry private from (potentially) colluding servers (note the inverted role of "user" and "server" here with respect to our work). Retrieving an

Netanel Raviv is with the department of computer science and engineering at Washington University in St. Louis, MO, USA. Email: netanel.raviv@wustl.edu. Rawad Bitar is with the institute of computer engineering at the Technical University of Munich, Germany. Email: rawad.bitar@tum.de. Eitan Yaakobi is with the department of computer science at the Technion — Israel Institute of Technology, Israel. Email: yaakobi@cs.technion.ac.il.

RB is funded from the Technical University of Munich - Institute for Advanced Studies, funded by the German Excellence Initiative and European Union Seventh Framework Programme under Grant Agreement No. 291763.

EY is partially supported by the ISF grant 1817/18 and by the Technion Hiroshi Fujiwara cyber security research center and the Israel cyber directorate.

978-1-6654-2159-1/22/$31.00 ©2022 IEEE

entry x_i from a dataset $\mathbf{x} = (x_1, \ldots, x_n)$ while keeping i private can be seen as retrieving $\mathbf{x}\mathbf{e}_i^\intercal$, where $\mathbf{e}_i$ is the i'th unit vector that must remain private. A natural generalization is to retrieve $\mathbf{x}\mathbf{w}^\intercal$, rather than $\mathbf{x}\mathbf{e}_i^\intercal$, for some weight vector $\mathbf{w}$ that must remain private. In this generalization—often referred to as *private computation* [16, 20]—the inner product $\mathbf{x}\mathbf{w}^\intercal$ is computed *over a finite field*, unlike private inference. Another discernible difference is that PIR almost exclusively[1] discusses one user and multiple servers, among which collusion is restricted in some way. This is usually not the case in private inference settings, as both the server and the user are cohesive entities.

This paper is structured as follows. Preliminaries and problem setup are given in Section II, which includes a discussion about the privacy measures and a reduction from binarized weights to fixed-points ones. Our scheme for private inference is given in Section III. A tradeoff between the privacy of the server and that of the user is given in Section IV, under the assumption of polynomial decoding. In Section IV we also provide a simple lower bound on the communication of the protocol, which shows the optimality of the scheme introduced in Section III both in terms of the communication and the privacy guarantees.

II. Preliminaries and Problem Statement

The problem is introduced for the case of a weight vector $\mathbf{w}$ with ± 1 entries; this case is of particular interest for binarized models, that are prominent tool in machine learning as stated above. In the sequel it is shown that the case in which the entries of $\mathbf{w}$ are given in fixed-point representation can be reduced to the case where they are ± 1 with a relatively small loss of privacy. Hence, we focus on ± 1 valued $\mathbf{w}$ throughout.

A. Problem statement

A *server* holds a weight vector $\mathbf{w} \in \{\pm 1\}^n$, randomly chosen from $W = \mathrm{Unif}(\{\pm 1\}^n)$, and a *user* holds a data vector $\mathbf{x} \in \mathbb{R}^n$, randomly chosen from a continuous data distribution X. The end goal is for the server to retrieve $\mathbf{w}\mathbf{x}^\intercal$ (computed over $\mathbb{R}$), while guaranteeing some level of privacy to both parties, defined shortly. Following the computation of $\mathbf{w}\mathbf{x}^\intercal$, the server feeds the result into some pre-trained model m to obtain the inference $m(\mathbf{w}\mathbf{x}^\intercal)$, which is then sent to back to the user. In what follows we focus on the retrieval of $\mathbf{w}\mathbf{x}^\intercal$ and its associated privacy; the remaining parts of the inference (such as weights in inner layers in a neural network), insofar as they are independent of $\mathbf{w}$, remain *perfectly* private. Note that due to the disclosure of $m(\mathbf{w}\mathbf{x}^\intercal)$ to the user, some privacy loss is unavoidable, regardless of the privacy guarantees. For instance, the user may repeatedly query multiple vectors $\{\mathbf{x}_i\}_{i=1}^N$, collect their inferences $\{m(\mathbf{w}\mathbf{x}_i^\intercal)\}_{i=1}^N$ by following the protocol N times, and then train a model similar to m. Such minimal N is known as *sample complexity*, see e.g. [19].

The modeling of the weight vector as a uniform random variable reflects the lack of knowledge the user has about it.

[1] Single-server PIR schemes have very recently been studied [8, 12], yet still in the finite field case.

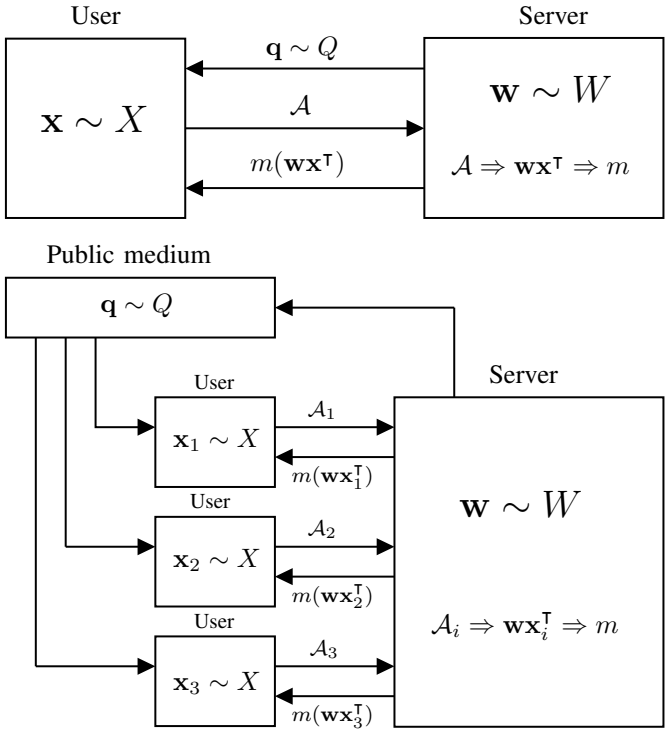

Fig. 1. An illustration of the information-theoretic private inference problem. (*top*) in a two-party setting, a server holds a weight vector $\mathbf{w} \sim W$ and a user holds data $\mathbf{x} \sim X$. The server sends a query $\mathbf{q}$ to the user, which replies with an answer $\mathcal{A}$. This answer is used by the server to extract $\mathbf{w}\mathbf{x}^\intercal$, which is then fed into a model m. The inference $m(\mathbf{w}\mathbf{x}^\intercal)$ is sent back to the user. (*bottom*) Our scheme can also be used in a multi-user single-server case, where each user has her own data $\mathbf{x}_i \sim X$. The query $\mathbf{q}$ is published in a public medium, from which it is downloaded by the users. The protocol then proceeds as in the single-user single-server case. The privacy guarantees from the two-party setting remain.

Modeling the data as taken from a distribution is a common practice in machine learning; while it is normally assumed that the data distribution X is not known (e.g., in PAC learning), here we make the more restrictive assumption that the server knows X, and yet the scheme is compatible with any such X. Inspired by the PIR literature, we focus on protocols of the query-answer form, as follows.

1) The server sends to the user a query $\mathbf{q} \in \{\pm 1\}^d$ for some d, randomly chosen from a distribution Q; this distribution is a deterministic function of W, i.e., $H(Q|W) = 0$, where H is the entropy function.
2) The user computes ℓ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_\ell \in \mathbb{R}^n$ deterministically from $\mathbf{q}$, and sends an answer $\mathcal{A} = \{\mathbf{v}_i\mathbf{x}^\intercal\}_{i=1}^\ell$ to the server.
3) The user combines the elements $\{\mathbf{v}_i\mathbf{x}^\intercal\}_{i=1}^\ell$ to retrieve the value $\mathbf{w}\mathbf{x}^\intercal$. To prove a lower bound in the sequel, it is additionally assumed that this part of the protocol is done by using a polynomial whose coefficients depend only on Q, and not on X.

We note that the structure of the problem allows the server to send *the same* query $\mathbf{q}$ to all future users. This way, the server may post $\mathbf{q}$ in some public forum (e.g., the server's website) and save all future communication with users interested in

inference. The upcoming information theoretic analysis guarantees that $\mathbf{q}$ can be publicly available indefinitely, while the privacy of $\mathbf{w}$ remains protected against any computational power that exists or may exist in the future (up to the inevitable learning attack mentioned earlier). These two interpretation of the problem are illustrated in Fig. 1. The merit of a given protocol is measured by the following quantities.

- **Publication cost**, i.e., the number of bits d published by the server. Notice that this number might change as a function of the value of Q; finding the optimal expected value of d is a *source coding* problem, and this value is lower bounded by $H(Q)$ according to a famous theorem by Shannon [4, Th. 5.3.1].
- **Server-privacy**, measured by the mutual information $I(Q;W)$.
- **User-privacy**, measured by the dimension of the subspace on which $\mathbf{x}$ is revealed, i.e., the parameter[2] ℓ.

A justification for the latter measure for privacy as a figure of merit is given shortly by using the independent components of X. Note that mutual information is largely of no use in this case, since for most reasonable continuous data distributions the entropy in infinite.

Naturally, the quantities d, $I(Q;W)$, and ℓ should be simultaneously minimized. However, it will be shown in the sequel that $I(Q;W)+\ell$ is bounded from below. Hence, one wishes to attain this lower bound with equality, while minimizing the publication cost d as much as possible.

Remark 1. *For a fully comprehensive analysis of the communication of the protocol, one should include ℓ as upload cost. However, it will be clear in the sequel (e.g., Example 2) that usually $\ell \ll d$, and hence it can be neglected. More precisely, our protocol supports infinite precision, and hence the de-facto upload cost depends on the required precision level, chosen by the system designer. In reality, real numbers are normally presented in fixed or floating point notation, which require a constant number of bits. This would bring the upload cost to $O(\ell)$, which can still be neglected if $\ell \ll d$.*

B. From binary weights to fixed-point weights

In this section it is shown that a protocol for the above problem can also be employed in settings where weights are quantized in fixed-point representation rather than in binary form. Consider $\mathbf{w} \in \mathbb{R}^n$ in which each entry w_i is represented using m bits. That is, $\mathbf{w}$ can be seen as a matrix in $\{\pm 1\}^{m\times n}$, in which the i'th column is the fixed point representation of w_i, namely, $w_i = \sum_{j=1}^{m} \frac{1-w_{i,j}}{2} \cdot 2^{j-1}$. For example, the matrix

$$\begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \tag{1}$$

corresponds to the weight vector

$$\mathbf{w} = \left(2^1, 2^0, 2^0+2^1, 2^1+2^2\right) = (2,1,3,6).$$

[2]More precisely, the parameter $\dim \operatorname{span}\{\mathbf{v}_i\}_{i=1}^{\ell}$, but for simplicity of presentation we consider the worst-case assumption, i.e., that the vectors $\mathbf{v}_i$ are independent.

Notice that the choice of 2 as the basis of the representation, as well as the range of exponents $\{0,1,\ldots,m-1\}$, are arbitrary.

To see that a protocol for binary valued weight vectors can be used for fixed point ones, we follow several straightforward computation steps, which are omitted, and get

$$\mathbf{w}\mathbf{x}^\intercal = \frac{2^m-1}{2}\sum_{i=1}^{n} x_i - \sum_{j=1}^{m} 2^{j-2}\tilde{\mathbf{w}}_j\mathbf{x}^\intercal, \tag{2}$$

where $\tilde{\mathbf{w}}_j = (w_{j,1},\ldots,w_{j,n})$. Eq. (2) implies that given black-box access to a protocol $A(\mathbf{x},\mathbf{u})$ for retrieving $\mathbf{u}\mathbf{x}^\intercal$ for $\mathbf{x}\in\mathbb{R}^n$ and $\mathbf{u}\in\{\pm1\}^n$, one can use it as follows:

1) The server and the user execute $A(\mathbf{x},\tilde{\mathbf{w}}_j)$ for every $j\in[m]$ to retrieve $\{\tilde{\mathbf{w}}_j\mathbf{x}^\intercal\}_{j=1}^{m}$.
2) The user sends $\sum_{i=1}^{n} x_i$ to the server.
3) The server linearly combines $\sum_{i=1}^{n} x_i$ and $\{\tilde{\mathbf{w}}_j\mathbf{x}^\intercal\}_{j=1}^{m}$ as in (2) to retrieve $\mathbf{w}\mathbf{x}^\intercal$.

It is readily verified that the server-privacy guarantees of the algorithm A are maintained for each $\tilde{\mathbf{w}}_j$, and that if the server learns $\mathbf{x}$ on an ℓ-dimensional subspace in each execution of A, then in the above protocol the server learns $\mathbf{x}$ on a subspace of dimension at most $m\ell+1$. Therefore, in the remainder of the paper we focus on $\{\pm1\}$-valued weight vectors.

C. Privacy for continuous data distribution using independent components

As shown in the sequel, there exists an unavoidable tradeoff between user- and server-privacy. On the one hand, since the distributions W and Q are discrete, the mutual information $I(W;Q)$ quantifies the server-privacy well. On the other hand, since data is often continuous, the mutual information $I(X;\mathcal{A})$ is infinite in most cases. Quantifying user-privacy using the parameter ℓ can be interpreted by considering the *independent components* of the data distribution X.

It is a common practice in the literature [3, 11] to model the data distribution X as a summation $X=\sum_{i=1}^{c} Y_i\mathbf{s}_i$, where the $\mathbf{s}_i$'s are fixed linearly independent vectors in $\mathbb{R}^n$, and the Y_i's are mutually independent random variables in $\mathbb{R}$. In a sense, the variables Y_i are the degrees of freedom of the data distribution, and thus can be used to quantify lost privacy as follows.

Assume that the independent components $\{\mathbf{s}_i\}_{i=1}^{c}$ of X are known to both the server and the user. Additionally, recall that the computation of the vectors $\{\mathbf{v}_i\}_{i=1}^{\ell}$ from the query q is deterministic, and hence these are also known to both the server and the user. From the perspective of the server, receiving $\{\mathbf{v}_i\mathbf{x}^\intercal\}_{i=1}^{\ell}$ effectively reduces the number of independent components of X from c to at least $c-\ell$. We illustrate this via an example.

Example 1. *Assume that X has $c=3$ independent components, i.e., $X=Y_1\mathbf{s}_1+Y_2\mathbf{s}_2+Y_3\mathbf{s}_3$, for some Y_i's and $\mathbf{s}_i$'s. Suppose that $\ell=2$, i.e., the server receives the values*

$$\begin{aligned} \mathbf{v}_1\mathbf{x}^\intercal &= y_1\mathbf{v}_1\mathbf{s}_1^\intercal + y_2\mathbf{v}_1\mathbf{s}_2^\intercal + y_3\mathbf{v}_1\mathbf{s}_3^\intercal \triangleq \lambda_1 \\ \mathbf{v}_2\mathbf{x}^\intercal &= y_1\mathbf{v}_2\mathbf{s}_1^\intercal + y_2\mathbf{v}_2\mathbf{s}_2^\intercal + y_3\mathbf{v}_2\mathbf{s}_3^\intercal \triangleq \lambda_2. \end{aligned} \tag{3}$$

For some $y_i \sim Y_i$, $i \in [3]$. Therefore, in the worst-case the matrix of coefficients in (3) is of full-rank, and the server may infer that (y_1, y_2, y_3) lie in a subspace of degree $c - \ell = 1$. Namely, from the server's perspective, the data distribution X has been reduced to $c - \ell = 1$ independent components, that remain unknown.

Therefore, the parameter ℓ quantifies the loss of privacy from the continuous data distribution X. Note that one must choose the parameter ℓ while considering the number of independent components c in the data distribution, rather than the dimension of the feature space n.

III. Our Protocol

In what follows let $\mathbb{F}_2$ be the binary field in its $\{\pm 1\}$ representation, i.e., -1 represents the Boolean "one", and 1 represents the Boolean "zero". To prevent ambiguity between $\mathbb{F}_2$-operations and $\mathbb{R}$-operations we use $\oplus, \odot$ for the former, and $+, \cdot$ for the latter, as well as $\mathrm{Span}_{\mathbb{R}}$ and $\mathrm{Span}_{\mathbb{F}_2}$ whenever relevant.

For a privacy parameter $t \in [n]$ let $L \in \{\pm 1\}^{t \times t}$ be an invertible matrix over $\mathbb{R}$, and let $\mathcal{S} = \{S_1, \ldots, S_t\}$ be a partition of $[n]$ (i.e., $S_i \cap S_j = \varnothing$ for every $i \neq j$, $S_i \neq \varnothing$ for all i, and $\cup_i S_i = [n]$). We assume that L and $\mathcal{S}$ are known to all for every t; further details regarding this assumption are given in Remark 2 below. For a given partition $\mathcal{S}$ let $V = V(\mathcal{S}) = \mathrm{Span}_{\mathbb{F}_2}\{\mathbb{1}_{S_i}\}_{i=1}^t$, where $(\mathbb{1}_{S_i})_j = -1$ if $j \in S_i$, and 1 otherwise. In addition, let $M = M(V) \in \mathbb{F}_2^{(n-t)\times n}$ be a Boolean parity-check matrix for V, i.e., $V = \{\mathbf{x} \in \mathbb{F}_2^n | M \odot \mathbf{x}^\intercal = \mathbb{1}\}$ ($\mathbb{1}$ being the zero vector in $\mathbb{F}_2^{n-t}$), and let $B(A, \mathbf{b})$ be any deterministic algorithm that finds a solution $\mathbf{x}$ to the equation $A \odot \mathbf{x}^\intercal = \mathbf{b}$ over $\mathbb{F}_2$; the algorithm B is assumed to be known to all as well. Our protocol proceeds as follows.

1) The server:
   a) Publishes $\mathbf{q}^\intercal \triangleq M \odot \mathbf{w}^\intercal \in \mathbb{F}_2^{n-t}$.
   b) Defines $\mathbf{u} \triangleq B(M, \mathbf{q})$, and keeps it private.
   c) Finds the unique $\ell_1, \ldots, \ell_t \in \{\pm 1\}$ such that $\mathbf{u} = \mathbf{w} \oplus \bigoplus_{r=1}^t (\ell_r \odot \mathbb{1}_{S_r})$, and keeps them private as well. These ℓ_i's exist since $M \odot \mathbf{u}^\intercal = M \odot \mathbf{w}^\intercal$, and therefore $\mathbf{u} \in V \oplus \mathbf{w} = \mathrm{Span}_{\mathbb{F}_2}\{\mathbb{1}_{S_i}\}_{i=1}^t \oplus \mathbf{w}$.
2) The user:
   a) Defines $\mathbf{u} \triangleq B(M, \mathbf{q})$. This is the same vector $\mathbf{u}$ that is found above by the server since B is deterministic.
   b) Defines $\mathbf{v}_i = \mathbf{u} \oplus \bigoplus_{r=1}^t [L_{i,r} \odot \mathbb{1}_{S_r}]$ for each $i \in [t]$. The vectors $\{\bigoplus_{r=1}^t [L_{i,r} \odot \mathbb{1}_{S_r}]\}_{i=1}^t$ can be computed in a pre-processing step since they depend exclusively on $\mathcal{S}$ and L, and do not depend on $\mathbf{w}$ nor on $\mathbf{x}$.
   c) Sends $\mathcal{A} = \{\mathbf{v}_i \mathbf{x}^\intercal\}_{i=1}^t$ to the server (the $\mathbf{v}_i \mathbf{x}^\intercal$'s are computed over $\mathbb{R}$).

To retrieve $\mathbf{w}\mathbf{x}^\intercal$, observe that

$$\begin{aligned}\mathbf{v}_i &= \mathbf{u} \oplus \bigoplus_{r=1}^t [L_{i,r} \odot \mathbb{1}_{S_r}] \\ &= \mathbf{w} \oplus \left[\bigoplus_{r=1}^t \ell_j \odot \mathbb{1}_{S_r}\right] \oplus \left[\bigoplus_{r=1}^t L_{i,r} \odot \mathbb{1}_{S_r}\right] \\ &= \mathbf{w} \oplus \bigoplus_{r=1}^t [(\ell_r \oplus L_{i,r}) \odot \mathbb{1}_{S_r}]. \end{aligned} \tag{4}$$

Since the ± 1 representation of $\mathbb{F}_2$ is used, the $\oplus$ operation between $\mathbb{F}_2$ elements is identical to the $\cdot$ operation over $\mathbb{R}$ (i.e., $x \oplus y = x \cdot y$ for every $x, y \in \{\pm 1\}$). Therefore, (4) implies that $v_{i,j} = w_j \ell_r L_{i,r}$ for every $j \in [n]$, where $v_{i,j}$ is the j'th element of $\mathbf{v}_i$, and where $r \in [t]$ is the unique integer such that $j \in S_r$; this integer is unique since $\mathcal{S}$ is a partition. Hence,

$$\begin{aligned}\mathbf{v}_i \mathbf{x}^\intercal &= \sum_{j=1}^n v_{i,j} x_i = \sum_{r=1}^t \sum_{j \in S_r} x_j w_j \ell_r L_{i,r} \\ &= \sum_{r=1}^t \ell_r L_{i,r} \sum_{j \in S_r} x_j w_j,\end{aligned}$$

and therefore the following $\mathbb{R}$-linear equation system holds.

$$\begin{bmatrix} \mathbf{v}_1 \mathbf{x}^\intercal \\ \vdots \\ \mathbf{v}_t \mathbf{x}^\intercal \end{bmatrix} = \begin{bmatrix} L_{1,1} & \cdots & L_{1,t} \\ \vdots & \ddots & \vdots \\ L_{t,1} & \ldots & L_{t,t} \end{bmatrix} \begin{bmatrix} \ell_1 & & \\ & \ddots & \\ & & \ell_t \end{bmatrix} \begin{bmatrix} \sum_{j \in S_1} x_j w_j \\ \vdots \\ \sum_{j \in S_t} x_j w_j \end{bmatrix}$$

Therefore, the final step of the protocol is as follows.

3) The server computes

$$\begin{aligned}\mathbb{1} \cdot \begin{bmatrix} \ell_1 & & \\ & \ddots & \\ & & \ell_t \end{bmatrix} \cdot L^{-1} \cdot \begin{bmatrix} \mathbf{v}_1 \mathbf{x}^\intercal \\ \vdots \\ \mathbf{v}_t \mathbf{x}^\intercal \end{bmatrix} &= \sum_{i=1}^t \sum_{j \in S_i} x_j w_j \\ &= \sum_{i=1}^n x_i w_i = \mathbf{w}\mathbf{x}^\intercal. \end{aligned} \tag{5}$$

Theorem 1. *For any $t \in [n]$, the above scheme has publication cost $n - t$, server-privacy $I(Q; W) = n - t$, and user-privacy $\ell = t$.*

Proof. The claims regarding user-privacy and publication cost are immediate from the definitions; a discussion about how to disclose the partition $\mathcal{S}$ and the invertible matrix L without increasing the publication cost is given shortly. It remains to show that $I(Q; W) = n - t$. Since $I(Q; W) = H(W) - H(W|Q) = n - H(W|Q)$, it suffices to show that $H(W|Q) = t$. Notice that disclosing $Q = \mathbf{q}$ reveals the identity of the coset of V in which $\mathbf{w}$ resides. Since $\mathbf{w}$ is uniformly distributed over $\{\pm 1\}^n$, and since cosets of V are of size 2^t, it follows that $W|Q = \mathbf{q}$ is uniform over a set of size 2^t, and hence $H(W|Q = \mathbf{q}) = t$. This implies that $H(W|Q) = t$ by a straightforward computation. □

Remark 2. *Since the matrix L and the partition $\mathcal{S} = \{S_1, \ldots, S_t\}$ do not depend on $\mathbf{w}$ nor on $\mathbf{x}$, they can be assumed public, and discarded from the publication cost. This, however, requires deterministic protocols by which the user and the server can agree on L and $\mathcal{S}$ without communicating. The partition $\mathcal{S}$ can simply be determined as S_1 being the first $\lfloor n/t \rfloor$ elements, S_2 being the next $\lfloor n/t \rfloor$ elements, and*

so on, where S_t is the last $\lfloor n/t \rfloor + n \bmod t$ elements. The matrix L can be computed from the Sylvester construction of a Hadamard matrix; this is a simple deterministic construction of a $\{\pm 1\}$-matrix H of size $2^m \times 2^m$ that satisfies $HH^\intercal = 2^m I$, and hence it is invertible over $\mathbb{R}$. Specifically, Let m be the smallest integer such that $t \leq 2^m$, and construct a $2^m \times 2^m$ matrix H by the Sylvester construction. Then, since H is invertible over $\mathbb{R}$, its first t rows are linearly independent over $\mathbb{R}$, and hence must have a $t \times t$ invertible submatrix. Both the server and the user define L as the first $t \times t$ invertible submatrix—in lexicographic order—of the first t rows of H.

Example 2. *To illustrate the applicability of the above scheme, consider a dataset of the order of magnitude of MNIST, say $n = 1000$ continuous features, that is classified by a neural network with 10 binarized neurons in its first layer. Iterate the above protocol with privacy parameter $t = 10$ for each weight vector W_i $i \in [10]$. The guarantee $I(Q; W_i) = n - t$ implies that from the perspective of the user, each $\mathbf{w}_i$ has 2^{10} equiprobable values. Assuming independence, this leaves 2^{100} equiprobable values for the first-layer weights of the network, and the remaining inner layers of the network remain perfectly concealed. From the perspective of the server, $\mathbf{x}$ is revealed on a subspace of dimension at most 100; if X has c independent component (see Section II-C), this leaves $\max\{c - 100, 0\}$ degrees of uncertainty from the perspective of the server. The overall publication cost is $10 \cdot (n - t) = 9900$ bits, and the upload cost, which is negligible, is $10t = 100$ real values.*

IV. Fundamental Privacy Tradeoff, and Publication Cost Lower Bound

Clearly, one wishes to minimize both $I(Q; W)$ and ℓ in order to guarantee maximum privacy for both parties. In this section we show that both quantities cannot be minimized simultaneously, and their sum is bounded from below. The above scheme attains this lower bound, and is therefore optimal in this sense. Then, we derive a simple lower bound on the publication cost, and show that the scheme is optimal in this sense as well. The proof of the former bound requires the following simple lemma.

Lemma 1. *[7] For an $\mathbb{R}$-subspace S of dimension ℓ we have that $|S \cap \{\pm 1\}^n| \leq 2^\ell$.*

Proof. Let S be an ℓ-dimensional $\mathbb{R}$-subspace, and let $M \in \mathbb{R}^{\ell \times n}$ be a matrix whose row-span is S. Since M has a reduced row-echelon form, it follows that S can be written as $S = \{(\mathbf{v}, L(\mathbf{v})) | \mathbf{v} \in \mathbb{R}^\ell\}$ (up to a permutation of entries) for some linear transform $L : \mathbb{R}^\ell \to \mathbb{R}^{n-\ell}$. It follows that

$$S \cap \{\pm 1\}^n = \{(\mathbf{v}, L(\mathbf{v})) | \mathbf{v} \in \{\pm 1\}^\ell \text{ and } L(\mathbf{v}) \in \{\pm 1\}^{n-\ell}\}$$

which readily implies that $|S \cap \{\pm 1\}^n| \leq |\{(\mathbf{v}, L(\mathbf{v})) | \mathbf{v} \in \{\pm 1\}^\ell\}| = 2^\ell$. □

The following theorem assumes that the decoding at the server's side is done using a polynomial. That is, there exists a polynomial f_Q, which depends only on Q, such that $\mathbf{x}\mathbf{w}^\intercal = f_Q(\mathbf{x}\mathbf{v}_1^\intercal, \ldots, \mathbf{x}\mathbf{v}_\ell^\intercal)$ for every $\mathbf{x}$.

Theorem 2. $I(W; Q) + \ell \geq n$.

Proof. Since $I(W; Q) = H(W) - H(W|Q) = n - H(W|Q)$, it suffices to show that $H(W|Q) \leq \ell$. First, observe that since the vectors $\mathbf{v}_i$ are a deterministic function of Q, it follows that $H(W|Q) = H(W, \{\mathbf{v}_i\}_{i=1}^\ell | Q)$. Second, we assume polynomial decoding, i.e., that there exists $f_Q : \mathbb{R}^\ell \to \mathbb{R}$ such that $\mathbf{x}\mathbf{w}^\intercal = f_Q(\mathbf{x}\mathbf{v}_1^\intercal, \ldots, \mathbf{x}\mathbf{v}_\ell^\intercal)$. Since the scheme must be valid for any data distribution, it must be valid for every $\mathbf{x}$. That is, the polynomial $f_Q(\mathbf{x}\mathbf{v}_1^\intercal, \ldots, \mathbf{x}\mathbf{v}_\ell^\intercal) - \mathbf{x}\mathbf{w}^\intercal$, seen as a polynomial in the n variables $x_1, \ldots, x_n$, must be the zero polynomial.

Denote $f_Q(y_1, \ldots, y_\ell) = \sum_{\mathbf{d} \in \mathbb{N}^\ell} f_{\mathbf{d}} \mathbf{y}^{\mathbf{d}}$, where $\mathbf{y}^{\mathbf{d}} = y_1^{d_1} \cdots y_\ell^{d_\ell}$, and $f_{\mathbf{d}} \in \mathbb{R}$ for every $\mathbf{d}$. It follows that for each $i \in [n]$, the coefficient of x_i in $f_Q(\mathbf{x}\mathbf{v}_1^\intercal, \ldots, \mathbf{x}\mathbf{v}_\ell^\intercal) - \mathbf{x}\mathbf{w}^\intercal$ is $\sum_{r=1}^\ell f_{\mathbf{e}_r} v_{r,i} - w_i$, where $\mathbf{e}_i$ is the i'th unit vector of length ℓ. Setting these coefficients to zero yields the linear equation $\sum_{r=1}^\ell f_{\mathbf{e}_r} \mathbf{v}_r = \mathbf{w}$, and therefore $\mathbf{w} \in \mathrm{Span}_{\mathbb{R}}\{\mathbf{v}_i\}_{i=1}^\ell$.

Now, for $\mathbf{q} \in \mathrm{Supp}(Q)$, we bound the support size of the random variable $W, \{\mathbf{v}_i\}_{i=1}^\ell | Q = \mathbf{q}$ from above. Since any $\mathbf{w}$ in this support is in the $\mathbb{R}$-span of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$, it follows that the size of this support cannot be larger than the maximum possible number of $\{\pm 1\}^n$ vectors in an $\mathbb{R}$-subspace of dimension ℓ. Formally, it is readily verified that

$$|\mathrm{Supp}(W, \{\mathbf{v}_i\}_{i=1}^\ell | Q = \mathbf{q})| \leq \max_{S | \dim_{\mathbb{R}}(S) = \ell} |S \cap \{\pm 1\}^n|.$$

Lemma 1 implies that $\max_{S | \dim_{\mathbb{R}}(S) = \ell} |S \cap \{\pm 1\}^n| \leq 2^\ell$, and hence the entropy $H(W, \{\mathbf{v}_i\}_{i=1}^\ell | Q = \mathbf{q})$ cannot be larger than that of $H(\mathrm{Unif}([2^\ell]))$, which is ℓ ([4, Thm. 2.6.4]). Therefore,

$$\begin{aligned} H(W|Q) &= H(W, \{\mathbf{v}_i\}_{i=1}^\ell | Q) \\ &= \sum_{\mathbf{q} \in \mathrm{Supp}(Q)} \Pr(Q = \mathbf{q}) H(W, \{\mathbf{v}_i\}_{i=1}^\ell | Q = \mathbf{q}) \\ &\leq \ell \cdot \sum_{\mathbf{q} \in \mathrm{Supp}(Q)} \Pr(Q = \mathbf{q}) = \ell. \end{aligned}$$ □

We now turn to provide a lower bound on the publication cost d. As mentioned in the problem definition, the number d of published bits is in fact a random variable that depends on the value of Q. Therefore, the problem of minimizing the expected value of d is a source coding problem, and the expected value of d is lower bounded by $H(Q)$. Recall that $I(Q; W) \geq n - \ell$ by Theorem 2. Therefore, by the symmetry of mutual information, we have

$$I(W; Q) = H(Q) - H(Q|W) \geq n - \ell,$$

and hence $H(Q)$ is lower bounded by $n - \ell$. This implies that the expected publication cost is also bounded by $n - \ell$. This, alongside Theorem 2, implies the optimality of our scheme.

Corollary 1. *The scheme in Section III is optimal both in its privacy guarantees and its publication cost. In addition, the scheme allows infinite precision in retrieving $\mathbf{x}\mathbf{w}^\intercal$ (theoretically, up to numerical errors that might arise, say, in the inversion of L in (5)), and hence incurs zero accuracy loss in applying the server's model.*

REFERENCES

[1] F. Boemer, R. Cammarota, D. Demmler, T. Schneider, and H. Yalame, "MP2ML: A mixed-protocol machine learning framework for private inference," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–10.

[2] Center for Humane Technology, *Ledger of harms*, https://ledger.humanetech.com/.

[3] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd. New-York: Wiley, 2006.

[5] A. Galloway, G. W. Taylor, and M. Moussa, "Attacking binarized neural networks," in *International Conference on Learning Representations*, 2018.

[6] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, PMLR, 2016, pp. 201–210.

[7] C. Groenland and T. Johnston, "Intersection sizes of linear subspaces with the hypercube," *Journal of Combinatorial Theory, Series A*, vol. 170, p. 105 142, 2020.

[8] A. Heidarzadeh, N. Esmati, and A. Sprintson, "Single-server private linear transformation: The joint privacy case," *arXiv preprint arXiv:2106.05220*, 2021.

[9] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv preprint arXiv:1711.05189*, 2017.

[10] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 4114–4122.

[11] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[12] S. Kadhe, A. Heidarzadeh, A. Sprintson, and O. O. Koyluoglu, "Single-server private information retrieval schemes are equivalent to locally recoverable coding schemes," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 391–402, 2021.

[13] B. McDanel, S. Teerapittayanon, and H. Kung, "Embedded binarized neural networks," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, 2017, pp. 168–173.

[14] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2017, pp. 19–38.

[15] N. Raviv, S. Jain, P. Upadhyaya, J. Bruck, and A. A. Jiang, "Codnn–robust neural networks from coded classification," in *2020 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2020, pp. 2688–2693.

[16] N. Raviv and D. A. Karpuk, "Private polynomial computation from lagrange encoding," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 553–563, 2019.

[17] N. Raviv, A. Kelley, M. Guo, and E. Vorobeychik, "Enhancing robustness of neural networks through Fourier stabilization," in *International Conference on Machine Learning*, 2021.

[18] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar, "XONN: Xnor-based oblivious deep neural network inference," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 1501–1518.

[19] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[20] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.