GroupSecAgg: Information Theoretic Secure Aggregation with Uncoded Groupwise Keys

Kai Wan*, Yin Yao†, Hua Sun‡, Mingyue Ji†, Giuseppe Caire§
*Huazhong University of Science and Technology, 430074 Wuhan, China, kai_wan@hust.edu.cn
†University of Utah, Salt Lake City, UT 84112, USA, {Xin.Yao,mingyue.ji}@utah.edu
‡University of North Texas, Denton, TX 76203, USA, hua.sun@unt.edu
§Technische Universität Berlin, 10587 Berlin, Germany, caire@tu-berlin.de

***Abstract*—Secure aggregation, which is a core component of federated learning, aggregates locally trained models from distributed users at a central server, without revealing any other information about the local users' data. This paper follows a recent information theoretic secure aggregation problem with user dropouts, where the objective is to characterize the minimum communication cost from the K users to the server during the model aggregation. All existing secure aggregation protocols let the users share and store coded keys to guarantee security. On the motivation that uncoded groupwise keys are more convenient to be shared and could be used in large range of practical applications, this paper is the first to consider uncoded groupwise keys, where the keys are mutually independent and each key is shared by a group of S users. We show that if S is beyond a threshold, a new secure aggregation protocol with uncoded groupwise keys, referred to as `GroupSecAgg`, can achieve the same optimal communication cost as the best protocol with coded keys. The experiments on Amazon EC2 show the considerable improvements on the key sharing and model aggregation times compared to the state-of-the art.**

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279171

I. Introduction

Federated learning (FL) is a distributed machine learning framework, where a central server aims to solve a machine learning problem by the help of distributed clients/users with local data [1]–[4]. One important design criterion of FL is to securely aggregate the users' updated model at the server without letting the server learn the model of each user. This is called secure aggregation. During the secure aggregation process, there two main challenges. First, in a real environment some users may drop out or response slowly due to the network connectivity or computational capability. Second, additional communication among the users and server may be needed to guarantee perfect security and mitigate the effect of the user dropouts. Since a federated learning system usually contains of a massive number of devices, the minimization of the communication cost (i.e., number of bits transmitted bits) is crucial. In this paper, we will apply information theoretic tools to address this two challenges by proposing `GroupSecAgg`, which is based on uncoded groupwise keys (see later). `GroupSecAgg` achieves information theoretically optimal communication cost, which is surprisingly the same as the best protocol with coded keys.

Secure aggregation with user dropouts was originally considered in [5], and generally contains two phases: *offline key sharing* and *model aggregation*, where the user dropouts may happen in either phase or both phases. In the first phase, the users generate random seeds, and secretly share their private random seeds such that some keys are shared among the users. The offline key sharing phase is independent of the users' local training data, and thus can take place during off-peak traffic times when the network is not busy. For example, all of the secure aggregation protocols in [5]–[7] uses offline key sharing.[1] Once the keys are shared among users, during the *model aggregation* phase, they mask the updated models by keys and send masked models to the server, such that the server recovers the aggregated models of the surviving users without learning any information about the users' local models.

It is worth to note that none of the above papers consider the information theoretic security (i.e., perfect security). Recently, the authors in [11] proposed an information theoretic formulation of secure aggregation with user dropouts considered [5]. Under the assumption that the offline key sharing phase is done, the objective is to characterize the optimal communication cost in the model aggregation phase while preserving the information theoretic security of the users' local model. In particular, the authors in [11] formulated a (K, U) information theoretic secure aggregation problem, where K represents the number of users and U represents the minimum number of surviving users.[2] The server aims to compute the element-wise sum of the vector inputs (i.e., updated models) of K users, where the input vector of user k is denoted by W_k and contains L uniform and i.i.d. symbols over a finite field $\mathbb{F}_\mathsf{q}$. Each user k has stored a key Z_k, which can be any random variable independent of $W_1, \ldots, W_\mathsf{K}$. The transmissions (in the model aggregation phase) contains two rounds for the sake of security under user dropouts. In the first round of transmission, each user $k \in \{1, \ldots, \mathsf{K}\}$ sends a coded message X_k as a function of W_k and Z_k to the server. Since some users may drop during its transmission, the server only receives the messages from the users in $\mathcal{U}_1$ where $|\mathcal{U}_1| \geq \mathsf{U}$. Then the server informs the users in the subset $\mathcal{U}_1$ of non-dropped users. In the second round of transmission, after knowing the set $\mathcal{U}_1$, each user $k \in \mathcal{U}_1$ transmits another coded message $Y_k^{\mathcal{U}_1}$ as a function

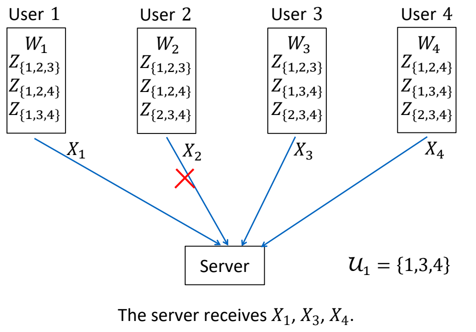
[1]Online key sharing protocols (for example the ones proposed in [8]–[10]) which are beyond the scope of this paper, allow users to communicate some information about the updated models and keys among each other, while in offline protocols users can only share keys.

[2]The problem in [11] only considers one epoch of the learning process.

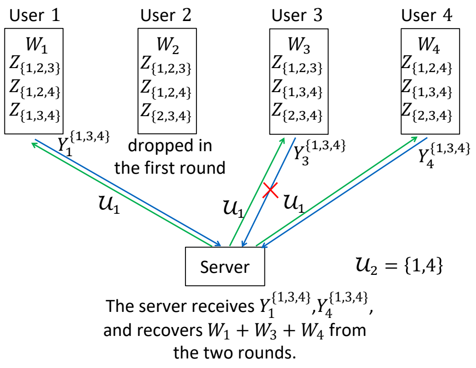
of $(W_k, Z_k, \mathcal{U}_1)$ to the server. Due to the user dropouts in the second round, letting $\mathcal{U}_2$ denote the set of surviving users in the second round with $\mathcal{U}_2 \subseteq \mathcal{U}_1$ and $|\mathcal{U}_2| \geq \mathsf{U}$, the server receives $Y_k^{\mathcal{U}_1}$ where $k \in \mathcal{U}_2$. By receiving $(X_k : k \in \mathcal{U}_1)$ and $(Y_k^{\mathcal{U}_1} : k \in \mathcal{U}_2)$, the server should recover the element-wise sum $\sum_{k \in \mathcal{U}_1} W_k$ without getting any other information about $W_1, \ldots, W_{\mathsf{K}}$ even if the server can receive $(X_k : k \in [\mathsf{K}] \setminus \mathcal{U}_1)$, $(Y_k^{\mathcal{U}_1} : k \in \mathcal{U}_1 \setminus \mathcal{U}_2)$ (e.g., the users are not really dropped but too slow in the transmission). Since the identity of the dropped users in each round is not known a priori by the users unless they receive the list of surviving users from the server, we should design $(X_k : k \in \{1, \ldots, \mathsf{K}\})$ and $(Y_k^{\mathcal{U}_1} : k \in \mathcal{U}_1)$ for any sets $\mathcal{U}_1, \mathcal{U}_2$ where $\mathcal{U}_2 \subseteq \mathcal{U}_1 \subseteq \{1, \ldots, \mathsf{K}\}$ and $|\mathcal{U}_1| \geq |\mathcal{U}_2| \geq \mathsf{U}$, while minimizing the communication rates by the users in two rounds. The communication rates in these two rounds forms a capacity region. It was shown in [11] that the minimum numbers of symbols that each user needs to send are L over the first round, and L/U over the second round, which can be achieved simultaneously by a novel secure aggregation protocol. Another secure aggregation protocol was proposed in [12] for the above problem, which achieves the optimal communication rates in the two-round model aggregation phase but with a less amount of needed keys than that of [11].

In this paper, we focus on the secure aggregation problem with uncoded groupwise keys and information theoretic security as illustrated in Fig. 1, where the constraint of uncoded groupwise keys means that, the keys are independent among each other and each key is stored by a set of users. By defining a system parameter $\mathsf{S} \in \{1, \ldots, \mathsf{K}\}$, for each $\mathcal{V} \subseteq \{1, \ldots, \mathsf{K}\}$ where $|\mathcal{V}| = \mathsf{S}$, there exists a key $Z_{\mathcal{V}}$ shared by the users in $\mathcal{V}$, which is independent of other keys. Uncoded groupwise keys may be preferred in practice since they can be generated with low complexity and shared conveniently, and find a wide range of applications besides secure aggregation in federated learning.[3] Note that all existing secure aggregation protocols fail to satisfy this constraint when $\mathsf{S} < \mathsf{K}$, due to the coded keys shared among users. Our objective is to characterize the capacity region of the numbers of transmissions by the users in two rounds of the model aggregation phase.

a) Main Contributions: In this paper, we first formulate the new information theoretic secure aggregation problem with uncoded groupwise keys. Then our main contributions on this new model are as follows. When $\mathsf{S} > \mathsf{K} - \mathsf{U}$, we propose a new secure aggregation protocol `GroupSecAgg`, which achieves exactly the same capacity region as in [11]; this means that, when $\mathsf{S} > \mathsf{K} - \mathsf{U}$, secure aggregation with uncoded groupwise key sharing has no loss on the communication efficiency. It is also interesting to see that by increasing S above $\mathsf{K} - \mathsf{U} + 1$ yields no reduction in the transmission cost; i.e., $\mathsf{S} = \mathsf{K} - \mathsf{U} + 1$ is sufficient and no larger value of S provide improvements. The main technical challenge of the proposed protocol based on linear coding is to determine the coefficients of the keys in the two round transmissions, satisfying the encodability, decodability, and security constraints. We overcome these challenges by designing new interference alignment strategies.[4] Note that, to achieve the optimal rates region by our proposed protocol, not all the keys $Z_{\mathcal{V}}$ where $\mathcal{V} \subseteq \{1, \ldots, \mathsf{K}\}$ and $|\mathcal{V}| = \mathsf{S}$ are needed during the transmission. The number of needed keys is either $\mathcal{O}(\mathsf{K})$ or $\mathcal{O}(\mathsf{K}^2)$, where each key has $(\mathsf{K} - \mathsf{U} + 1)\mathsf{L}/\mathsf{U}$ symbols. Finally, we implement `GroupSecAgg` into an Amazon EC2 cluster, and demonstrate the reduction of the running time compared to the original secure aggregation protocol in [5] (referred to as `SecAgg`), and the best existing information theoretic secure aggregation protocol with offline key sharing in [12] (referred to as `LightSecAgg`). Experimental results show that the proposed `GroupSecAgg` reduces the communication time in the model aggregation by up to 53% compared to `SecAgg`, and reduces the key sharing time up to 31.7% compared to `LightSecAgg`.


User 1
User 2
User 3
User 4
W_1 $Z_{\{1,2,3\}}$ $Z_{\{1,2,4\}}$ $Z_{\{1,3,4\}}$
W_2 $Z_{\{1,2,3\}}$ $Z_{\{1,2,4\}}$ $Z_{\{2,3,4\}}$
W_3 $Z_{\{1,2,3\}}$ $Z_{\{1,3,4\}}$ $Z_{\{2,3,4\}}$
W_4 $Z_{\{1,2,4\}}$ $Z_{\{1,3,4\}}$ $Z_{\{2,3,4\}}$
X_1
X_2
X_3
X_4
Server
$\mathcal{U}_1 = \{1,3,4\}$
The server receives X_1, X_3, X_4.

(a) First round.


User 1
User 2
User 3
User 4
W_1 $Z_{\{1,2,3\}}$ $Z_{\{1,2,4\}}$ $Z_{\{1,3,4\}}$
W_2 $Z_{\{1,2,3\}}$ $Z_{\{1,2,4\}}$ $Z_{\{2,3,4\}}$
W_3 $Z_{\{1,2,3\}}$ $Z_{\{1,3,4\}}$ $Z_{\{2,3,4\}}$
W_4 $Z_{\{1,2,4\}}$ $Z_{\{1,3,4\}}$ $Z_{\{2,3,4\}}$
$Y_1^{\{1,3,4\}}$
dropped in the first round
$Y_3^{\{1,3,4\}}$
$Y_4^{\{1,3,4\}}$
$\mathcal{U}_1$
$\mathcal{U}_1$
$\mathcal{U}_1$
Server
$\mathcal{U}_2 = \{1,4\}$
The server receives $Y_1^{\{1,3,4\}}, Y_4^{\{1,3,4\}}$, and recovers $W_1 + W_3 + W_4$ from the two rounds.

(b) Second round.

Fig. 1: $(\mathsf{K}, \mathsf{U}, \mathsf{S}) = (4, 2, 3)$ information theoretic secure aggregation problem with uncoded groupwise keys.

[3]For example, the uncoded pairwise key shared among each two users are independent of the other keys and thus can guarantee the information theoretic secure communication between these two users, while the other users (who may collude) are eavesdropper listening to the communication [13]. However, the pairwise coded keys used in the protocol [12] cannot guarantee secure communication between any two users, because the coded key shared by these two users are correlated to other keys stored by the other users.

[4]Interference alignment was originally proposed in [14] for the wireless interference channel, which aligns the undesired packets (i.e., interference) by each user such that their linear space dimension is reduced.

b) Notations: Calligraphic symbols denote sets, bold symbols denote vectors and matrices, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b]:=\{a,a+1,\dots,b\}$ and $[n]:=[1:n]$; $\mathbb{F}_{\mathsf{q}}$ represents a finite field with order q; $\mathbf{e}_{n,i}$ represents the vertical n-dimensional unit vector whose entry in the i^{th} position is 1 and 0 elsewhere; $\mathbf{I}_n$ represents the identity matrix of dimension $n\times n$; let $\binom{\mathcal{X}}{y}=\{\mathcal{S}\subseteq\mathcal{X}:|\mathcal{S}|=y\}$ where $|\mathcal{X}|\geq y>0$. In the rest of the paper entropies will be in base q, where q represents the field size.

II. System Model

We formulate a $(\mathsf{K},\mathsf{U},\mathsf{S})$ information theoretic secure aggregation problem with uncoded groupwise keys as illustrated in Fig. 1, which contains one epoch of federated learning process among K users and one server. For each $k\in[\mathsf{K}]$, user k holds one input vector (i.e., updated model) W_k composed of L uniform and i.i.d. symbols over a finite field $\mathbb{F}_{\mathsf{q}}$, where L is large enough. Ideally, the server aims to compute the element-wise sum of input vectors of all users. However, due to the user dropouts, the server may not be able to recover the sum of all input vectors. Hence, we let the server compute the sum of the input vectors from the surviving users, where the number of surviving users is at least U. In this paper, we mainly deal with the user dropouts and thus we assume that $\mathsf{U}\in[\mathsf{K}-1]$. In addition, by the secure aggregation constraint, the server must not retrieve any other information except the task from the received symbols. In order to guarantee the security, the users must share some secrets (i.e., keys) which are independent of the input vectors. Different from the secure aggregation problem in [11] which assumes that the keys could be any random variables shared among users, in this paper we consider uncoded groupwise keys, where the keys are independent of each other and each key is shared among S users where $\mathsf{S}\in[\mathsf{K}]$. For each set $\mathcal{V}\in\binom{[\mathsf{K}]}{\mathsf{S}}$, there exists a key $Z_{\mathcal{V}}$ shared by the users in $\mathcal{V}$ and independent of other keys. Thus $H\left((Z_{\mathcal{V}}:\mathcal{V}\in\binom{[\mathsf{K}]}{\mathsf{S}}),(W_1,\dots,W_{\mathsf{K}})\right)=\sum_{\mathcal{V}\in\binom{[\mathsf{K}]}{\mathsf{S}}}H(Z_{\mathcal{V}})+\sum_{k\in[\mathsf{K}]}H(W_k)$. We define $Z_k:=\left(Z_{\mathcal{V}}:\mathcal{V}\in\binom{[\mathsf{K}]}{\mathsf{S}},k\in\mathcal{V}\right)$, as the keys accessible by the user $k\in[\mathsf{K}]$. The whole secure aggregation procedure contains the following two rounds.

First round. In the first round, each user $k\in[\mathsf{K}]$ generates a message X_k as a function of W_k and Z_k, without knowing the identity of the dropped users. The communication rate of the first round R_1 is defined as the largest transmission load among all users normalized by L, i.e., $\mathsf{R}_1:=\max_{k\in[\mathsf{K}]}|X_k|/\mathsf{L}$. User k then sends X_k to the server.

Some users may drop in the first round transmission, and the set of surviving users after the first round is denoted as $\mathcal{U}_1$, where $\mathcal{U}_1\subseteq[\mathsf{K}]$ and $|\mathcal{U}_1|\geq\mathsf{U}$. Thus the server receives X_k where $k\in\mathcal{U}_1$.

Second round. In the second round, the server first sends the list of the surviving users (i.e., the set $\mathcal{U}_1$) to each user in $\mathcal{U}_1$. Then each user $k\in\mathcal{U}_1$ participates in the second round transmission by generating and sending a message $Y_k^{\mathcal{U}_1}$ as a function of W_k, Z_k, and $\mathcal{U}_1$. The communication rate of the second round R_2 is defined as the largest transmission load among all $\mathcal{U}_1$ and all users in $\mathcal{U}_1$ normalized by L, i.e., $\mathsf{R}_2:=\max_{\mathcal{U}_1\subseteq[\mathsf{K}]:|\mathcal{U}_1|\geq\mathsf{U}}\ \max_{k\in\mathcal{U}_1}|Y_k^{\mathcal{U}_1}|/\mathsf{L}$.

Some users may also drop in the second round, and the set of surviving users after the second round is denoted as $\mathcal{U}_2$, where $\mathcal{U}_2\subseteq\mathcal{U}_1$ and $|\mathcal{U}_2|\geq\mathsf{U}$. Thus the server receives $Y_k^{\mathcal{U}_1}$ where $k\in\mathcal{U}_2$.

Decoding. The server should recover $\sum_{k\in\mathcal{U}_1}W_k$ from $(X_{k_1}:k_1\in\mathcal{U}_1)$ and $(Y_{k_2}^{\mathcal{U}_1}:k_2\in\mathcal{U}_2)$, i.e.,

$$H\Big(\sum_{k\in\mathcal{U}_1}W_k\Big|(X_{k_1}:k_1\in\mathcal{U}_1),(Y_{k_2}^{\mathcal{U}_1}:k_2\in\mathcal{U}_2)\Big)=0, \qquad (1)$$

for each $\mathcal{U}_1\subseteq[\mathsf{K}]$ and $\mathcal{U}_2\subseteq\mathcal{U}_1$ where $|\mathcal{U}_1|\geq|\mathcal{U}_2|\geq\mathsf{U}$.

Threat model and security constraint. We consider a threat model where the server is honest but curious. In addition, we also consider that the users may be not really dropped but too slow in the transmission. Hence, the security constraint imposes that after receiving all messages sent by the users including the dropped users, the server cannot get any other information about the input vectors except $\sum_{k\in\mathcal{U}_1}W_k$, i.e.,

$$I\Big(W_1,\dots,W_{\mathsf{K}};X_1,\dots,X_{\mathsf{K}},(Y_k^{\mathcal{U}_1}:k\in\mathcal{U}_1)\Big|\sum_{k\in\mathcal{U}_1}W_k\Big)=0, \qquad (2)$$

for each $\mathcal{U}_1\subseteq[\mathsf{K}]$ where $|\mathcal{U}_1|\geq\mathsf{U}$.

Objective. A rate tuple $(\mathsf{R}_1,\mathsf{R}_2)$ is achievable if there exist uncoded groupwise keys $\left(Z_{\mathcal{V}}:\mathcal{V}\in\binom{[\mathsf{K}]}{\mathsf{S}}\right)$ and a secure aggregation protocol satisfying the decodability and security constraints (1) and (2). Our objective is to determine the capacity region (i.e., the closure of all achievable rate tuples) of the considered problem, denoted by $\mathcal{R}^\star$.

A converse bound from [11]. By removing the uncoded groupwise constraint on the keys in our considered problem, we obtain the information theoretic aggregation problem in [11]. Hence, the converse bound on the capacity region in [11] is also a converse bound for our considered problem, which leads to the following lemma.

Lemma 1 ([11]). *For the* $(\mathsf{K},\mathsf{U},\mathsf{S})$ *information theoretic secure aggregation problem with uncoded groupwise keys, any achievable rate tuple* $(\mathsf{R}_1,\mathsf{R}_2)$ *satisfies* $\mathsf{R}_1\geq1,\ \mathsf{R}_2\geq1/\mathsf{U}$.

Obviously, the capacity region of the $(\mathsf{K},\mathsf{U},\mathsf{S}_1)$ information theoretic secure aggregation problem with uncoded groupwise keys covers that of the $(\mathsf{K},\mathsf{U},\mathsf{S}_2)$ problem, where $\mathsf{S}_1>\mathsf{S}_2$.

III. Main Results

We first present the main result of this paper. The converse bound for Theorem 1 is directly from Lemma 1. For the achievability, we propose a `GroupSecAgg` based on linear coding and interference alignment, whose details are described in the extended version of this paper [15, Section IV].

Theorem 1. *For the* $(\mathsf{K},\mathsf{U},\mathsf{S})$ *information theoretic secure aggregation problem with uncoded groupwise keys, when* $\mathsf{S}>\mathsf{K}-\mathsf{U}$, *we have*

$$\mathcal{R}^\star=\{(\mathsf{R}_1,\mathsf{R}_2):\mathsf{R}_1\geq1,\mathsf{R}_2\geq1/\mathsf{U}\}. \qquad (3)$$

When $\mathsf{S} > \mathsf{K} - \mathsf{U}$, `GroupSecAgg` achieves the same capacity region as the optimal secure aggregation protocol without any constraint on the keys in [11], [12]. It is also interesting to see that increasing S above $\mathsf{K} - \mathsf{U} + 1$ will not reduce the communication cost.

There are totally $\binom{\mathsf{K}}{\mathsf{S}}$ subsets of $[\mathsf{K}]$ with cardinality S. By the problem setting, we can use at most $\binom{\mathsf{K}}{\mathsf{S}}$ keys each of which is shared by S users. However, we do not need to use generate all these $\binom{\mathsf{K}}{\mathsf{S}}$ keys in our proposed secure scheme for Theorem 1. The number of needed keys by the proposed secure aggregation scheme for Theorem 1 is K when $\mathsf{U} \leq \mathsf{K} - \mathsf{U} + 1$ and is $\mathcal{O}(\mathsf{K}^2)$ when $\mathsf{U} > \mathsf{K} - \mathsf{U} + 1$, where each key has $(\mathsf{K} - \mathsf{U} + 1)\mathsf{L}/\mathsf{U}$ symbols. Note that if coded key assignment is allowed, the secure aggregation scheme in [11] needs to generate U coded keys with L/U symbols for each group of users $\mathcal{V} \subseteq [\mathsf{K}]$ where $|\mathcal{V}| \in [\mathsf{U} : \mathsf{K}]$, where each user in the group stores a linear combination of these U coded keys; for each pair of users $\mathcal{V} \subseteq [\mathsf{K}]$ where $|\mathcal{V}| = 2$, the secure aggregation scheme in [12] lets each user in the pair generate a coded key with L/U symbols and share it to the other user in the pair. Note that for the case $\mathsf{S} \leq \mathsf{K} - \mathsf{U}$, a converse bound is proposed in [15], showing that $\mathsf{R}_1 = 1$ is not sufficient for the security. It is one of our on-going works to characterize the exact capacity region for the case $\mathsf{S} \leq \mathsf{K} - \mathsf{U}$.

A. Outline description of `GroupSecAgg`

To present the proposed `GroupSecAgg`, we only need to focus on the case where $\mathsf{S} = \mathsf{K} - \mathsf{U} + 1$, because a secure aggregation protocol for the case $\mathsf{S} = \mathsf{K} - \mathsf{U} + 1$ could also work for the case $\mathsf{S} > \mathsf{K} - \mathsf{U} + 1$ (see the reason provided at the end of Section II). The construction structure of `GroupSecAgg` is as follows.

- Since the length of each input vector W_i where $i \in [\mathsf{K}]$ is large enough, as explained in [11], we can consider blocks of symbols of W_i as an element of a suitably large field extension and consider operations such as element wise sum as operations over the field extension. Hence, without loss of generality, in `GroupSecAgg` we can assume that q is large enough. We then divide each input vector W_i where $i \in [\mathsf{K}]$ into U non-overlapping and equal-length pieces, where the j^{th} piece denoted by $W_{i,j}$ contains L/U symbols on $\mathbb{F}_{\mathsf{q}}$. In addition, for each $\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}$ and each $k \in \mathcal{V}$,[5] we let $Z_{\mathcal{V},k}$ denote a vector of L/U uniform i.i.d. symbols on $\mathbb{F}_{\mathsf{q}}$. Then, we define a key $Z_{\mathcal{V}} = (Z_{\mathcal{V},k} : k \in \mathcal{V})$ and let $Z_{\mathcal{V}}$ be shared by all users in $\mathcal{V}$.
- In the first round, each user $k \in [\mathsf{K}]$ sends
$$X_{k,j} = W_{k,j} + \sum_{\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}} : k \in \mathcal{V}} a_{\mathcal{V},j} Z_{\mathcal{V},k}, \ \forall j \in [\mathsf{U}], \quad (4)$$
where $a_{\mathcal{V},j} \in \mathbb{F}_{\mathsf{q}}$ is a coefficient to be designed and $a_{\mathcal{V},j} Z_{\mathcal{V},k}$ represents multiplying each element in $Z_{\mathcal{V},k}$ by $a_{\mathcal{V},j}$. Each $X_{k,j}$ contains L/U symbols, and thus $X_k = (X_{k,1}, \ldots, X_{k,\mathsf{U}})$ contains L symbols, which leads to $\mathsf{R}_1 = 1$. We let $\mathbf{a}_{\mathcal{V}} := [a_{\mathcal{V},1}, \ldots, a_{\mathcal{V},\mathsf{U}}]^{\mathsf{T}}$. By security, W_k should be perfectly protected by the keys in $X_k = (X_{k,1}, \ldots, X_{k,\mathsf{U}})$. So by denoting the sets $\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}$ where $k \in \mathcal{V}$ by $\mathcal{S}_k(1), \ldots, \mathcal{S}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}-1}\right)$, we aim to have that the coefficients matrix (whose dimension is $\mathsf{U} \times \binom{\mathsf{K}-1}{\mathsf{S}-1}$)
$$\left[\mathbf{a}_{\mathcal{S}_k(1)}, \ldots, \mathbf{a}_{\mathcal{S}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}-1}\right)}\right] \text{ has rank equal to } \mathsf{U}, \ \forall k \in [\mathsf{K}]. \quad (5)$$
If the constraints in (5) are satisfied, with the fact that the keys are independent of the input vectors, the server cannot get any information about $W_1, \ldots, W_{\mathsf{K}}$ even if the server receives all $X_1, \ldots, X_{\mathsf{K}}$.
Since the set of surviving users after the first round is $\mathcal{U}_1$, the server receives X_k where $k \in \mathcal{U}_1$, and thus can recover $\sum_{k \in \mathcal{U}_1} X_{k,j}$, which is equal to
$$\sum_{k \in \mathcal{U}_1} W_{k,j} + \sum_{\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}} : \mathcal{V} \cap \mathcal{U}_1 \neq \emptyset} \left(a_{\mathcal{V},j} \sum_{k_1 \in \mathcal{V} \cap \mathcal{U}_1} Z_{\mathcal{V},k_1} \right) \quad (6)$$
$$= \sum_{k \in \mathcal{U}_1} W_{k,j} + \sum_{\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}} \left(a_{\mathcal{V},j} \sum_{k_1 \in \mathcal{V} \cap \mathcal{U}_1} Z_{\mathcal{V},k_1} \right), \ \forall j \in [\mathsf{U}], \quad (7)$$
where (7) follows since $\mathsf{S} = \mathsf{K} - \mathsf{U} + 1 > \mathsf{K} - |\mathcal{U}_1|$. Hence, the server still needs to recover $\sum_{\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}} \left(a_{\mathcal{V},j} \sum_{k_1 \in \mathcal{V} \cap \mathcal{U}_1} Z_{\mathcal{V},k_1} \right)$ for each $j \in [\mathsf{U}]$ in the next round. We can treat
$$Z_{\mathcal{V}}^{\mathcal{U}_1} := \sum_{k_1 \in \mathcal{V} \cap \mathcal{U}_1} Z_{\mathcal{V},k_1}, \ \forall \mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}, \quad (8)$$
as one coded key, which can be encoded by all users in $\mathcal{V} \cap \mathcal{U}_1$ and contains L/U uniform and i.i.d. symbols. **Thus by the construction of the first round transmission, we only need to transmit linear combinations of coded keys in the second round, such that the server can recover $\sum_{\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}} a_{\mathcal{V},j} Z_{\mathcal{V}}^{\mathcal{U}_1}$ for each $j \in [\mathsf{U}]$.**
- In the second round, we denote the sets in $\binom{[\mathsf{K}]}{\mathsf{S}}$ by $\mathcal{S}(1), \ldots, \mathcal{S}\left(\binom{\mathsf{K}}{\mathsf{S}}\right)$, and for each $k \in [\mathsf{K}]$ denote the sets in $\binom{[\mathsf{K}] \setminus \{k\}}{\mathsf{S}}$ by $\overline{\mathcal{S}}_k(1), \ldots, \overline{\mathcal{S}}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}}\right)$. Thus the server should recover
$$\mathbf{F} = [F_1; \ldots; F_{\mathsf{U}}] \quad (9)$$
$$= \left[\mathbf{a}_{\mathcal{S}(1)}, \ldots, \mathbf{a}_{\mathcal{S}\left(\binom{\mathsf{K}}{\mathsf{S}}\right)}\right] \left[Z^{\mathcal{U}_1}_{\mathcal{S}(1)}; \ldots; Z^{\mathcal{U}_1}_{\mathcal{S}\left(\binom{\mathsf{K}}{\mathsf{S}}\right)}\right], \quad (10)$$
where each F_j, $j \in [\mathsf{U}]$, contains L/U symbols.
Note that each user $k \in \mathcal{U}_1$ cannot encode $Z_{\mathcal{V}}^{\mathcal{U}_1}$ where $\mathcal{V} \in \binom{[\mathsf{K}] \setminus \{k\}}{\mathsf{S}}$. If the U-dimensional vectors $\mathbf{a}_{\mathcal{V}}$ where $\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}$ satisfy the constraints that
$$\left[\mathbf{a}_{\overline{\mathcal{S}}_k(1)}, \ldots, \mathbf{a}_{\overline{\mathcal{S}}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}}\right)}\right] \text{ has rank } \mathsf{U} - 1, \quad (11)$$

[5]Recall that $\binom{\mathcal{X}}{y} = \{\mathcal{S} \subseteq \mathcal{X} : |\mathcal{S}| = y\}$ where $|\mathcal{X}| \geq y > 0$.

for each $k \in [\mathsf{K}]$, then the matrix $\left[\mathbf{a}_{\overline{\mathcal{S}}_k(1)}, \ldots, \mathbf{a}_{\overline{\mathcal{S}}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}}\right)}\right]$ contains exactly one linearly independent left null space vector. **To achieve** (11)**, we will propose some interference alignment techniques to align the** U**-dimensional vectors of the** $\binom{\mathsf{K}-1}{\mathsf{S}}$ **unknown keys to a linear space spanned by** $\mathsf{U}-1$ **linearly independent vectors.**

Thus we can let each user $k \in \mathcal{U}_1$ transmit $Y_k^{\mathcal{U}_1} = \mathbf{s}_k \mathbf{F}$, where $\mathbf{s}_k$ represents the left null space vector of $\left[\mathbf{a}_{\overline{\mathcal{S}}_k(1)}, \ldots, \mathbf{a}_{\overline{\mathcal{S}}_k\left(\binom{\mathsf{K}-1}{\mathsf{S}}\right)}\right]$. By construction, in $Y_k^{\mathcal{U}_1}$ the coefficients of the coded keys which cannot be encoded by user k are 0. Note that $Y_k^{\mathcal{U}_1}$ contains L/U symbols, which leads to $\mathsf{R}_2 = 1/\mathsf{U}$.

For the decodability, from any set of surviving users after the second round $\mathcal{U}_2 \subseteq \mathcal{U}_1$ **where** $|\mathcal{U}_2| \geq \mathsf{U}$**, we should recover** $F_1, \ldots, F_{\mathsf{U}}$ **from the second round transmission**; i.e., we aim to have

$$\text{any } \mathsf{U} \text{ vectors in } \{\mathbf{s}_k : k \in \mathcal{U}_1\} \text{ are linearly independent.} \tag{12}$$

Thus from (7) and (12), the server can recover $F_1, \ldots, F_{\mathsf{U}}$ and then recover $\sum_{k \in \mathcal{U}_1} W_{k,j}$ for all $j \in [\mathsf{U}]$; thus it can recover $\sum_{k \in \mathcal{U}_1} W_k$.

In addition, for the security constraint, by construction we have $H\left(Y_k^{\mathcal{U}_1} : k \in \mathcal{U}_1\right) = \mathsf{L}$, which follows since each $Y_k^{\mathcal{U}_1}$ where $k \in \mathcal{U}_1$ is in the linear space spanned by $F_1, \ldots, F_{\mathsf{U}}$, where each F_j, $j \in [\mathsf{U}]$, contains L/U symbols. Intuitively, from $(X_k : k \in [\mathsf{K}])$, the server cannot get any information about $W_1, \ldots, W_{\mathsf{K}}$. In addition with $(Y_k^{\mathcal{U}_1} : k \in \mathcal{U}_1)$ whose entropy is L, the server can at most get L symbols information about $W_1, \ldots, W_{\mathsf{K}}$, which are exactly the symbols in $\sum_{k \in \mathcal{U}_1} W_k$. Hence, the protocol is secure. The rigorous proof on the security constraint in (2) can be found in [15, Appendix A].

We conclude that the achieved rates are $(\mathsf{R}_1, \mathsf{R}_2) = (1, 1/\mathsf{U})$, coinciding with Theorem 1.

For what said above, it is apparent that the key challenge in `GroupSecAgg` is to design the U**-dimensional vectors** $\mathbf{a}_{\mathcal{V}}$ **where** $\mathcal{V} \in \binom{[\mathsf{K}]}{\mathsf{S}}$**, such that the constraints in** (5), (11), **and** (12) **are satisfied. As showed above, if such constraints are satisfied, `GroupSecAgg` is valid.**

Next, we use an example to illustrate the main idea of the proposed protocol.

Example 1 $((\mathsf{K}, \mathsf{U}, \mathsf{S})) = (4, 3, 2))$**.** Consider the $(\mathsf{K}, \mathsf{U}, \mathsf{S}) = (4, 3, 2)$ information theoretic secure aggregation problem with uncoded groupwise keys. For each $\mathcal{V} \in \binom{[4]}{2}$, we generate a key $Z_{\mathcal{V}} = (Z_{\mathcal{V},k} : k \in \mathcal{V})$ shared by users in $\mathcal{V}$, where each $Z_{\mathcal{V},k}$ contains $\mathsf{L}/3$ uniform and i.i.d. symbols over $\mathbb{F}_{\mathsf{q}}$. We also divide each input vector W_k where $k \in [4]$ into three pieces, $W_k = (W_{k,1}, W_{k,2}, W_{k,3})$, where each piece contains $\mathsf{L}/3$ uniform and i.i.d. symbols over $\mathbb{F}_{\mathsf{q}}$.

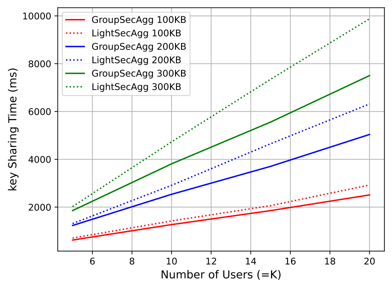
In the first round, each user $k \in [4]$ transmits $X_{k,j} = W_{k,j} + \sum_{\mathcal{V} \in \binom{[4]}{2} : k \in \mathcal{V}} a_{\mathcal{V},j} Z_{\mathcal{V},k}$, $\forall j \in [3]$. Now we select the 3-dimensional vectors $\mathbf{a}_{\{1,2\}}, \mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}, \mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}$, and $\mathbf{a}_{\{3,4\}}$ as follows,

$$\begin{aligned}
&\mathbf{a}_{\{1,2\}} = [1,0,0]^{\mathrm{T}},\ \mathbf{a}_{\{1,3\}} = [0,1,0]^{\mathrm{T}}, \mathbf{a}_{\{1,4\}} = [0,0,1]^{\mathrm{T}},\\
&\mathbf{a}_{\{2,3\}} = \mathbf{a}_{\{1,2\}} - \mathbf{a}_{\{1,3\}} = [1,-1,0]^{\mathrm{T}},\\
&\mathbf{a}_{\{2,4\}} = \mathbf{a}_{\{1,2\}} - \mathbf{a}_{\{1,4\}} = [1,0,-1]^{\mathrm{T}},\\
&\mathbf{a}_{\{3,4\}} = \mathbf{a}_{\{1,3\}} - \mathbf{a}_{\{1,4\}} = [0,1,-1]^{\mathrm{T}}.
\end{aligned} \tag{13}$$

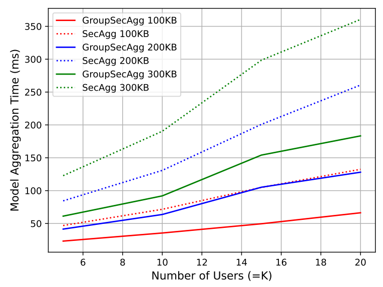
We next show that by the above choice the constraints in (5), (11), and (12) are satisfied.

For user 1, the matrix $[\mathbf{a}_{\{1,2\}}, \mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}] = \mathbf{I}_3$ has rank 3, where we recall that $\mathbf{I}_3$ represents the identity matrix with dimension 3×3. Hence, the constraint in (5) is satisfied for user 1. Thus W_1 is perfectly protected by $(Z_{\{1,2\},1},, Z_{\{1,3\},1}, Z_{\{1,4\},1})$ from X_1. For user 2, the matrix $[\mathbf{a}_{\{1,2\}}, \mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}]$ has rank 3. Hence, the constraint in (5) is satisfied for user 2. Thus W_2 is perfectly protected by $(Z_{\{1,2\},2},, Z_{\{2,3\},2}, Z_{\{2,4\},2})$ from X_2. Similarly, the constraints in (5) are also satisfied for users $3, 4$.

In the second round, we consider the case $\mathcal{U}_1 = [4]$, where the server should recover $W_1 + \cdots + W_4$. Defining the coded keys as in (8), the server needs to further recover $[F_1; F_2; F_3]$, which is equal to $[\mathbf{a}_{\{1,2\}}, \mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}, \mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}, \mathbf{a}_{\{3,4\}}]$ $\left[Z^{[4]}_{\{1,2\}}; Z^{[4]}_{\{1,3\}}; Z^{[4]}_{\{1,4\}}; Z^{[4]}_{\{2,3\}}; Z^{[4]}_{\{2,4\}}; Z^{[4]}_{\{3,4\}}\right]$. For user 1 who cannot encode $Z^{[4]}_{\{2,3\}}, Z^{[4]}_{\{2,4\}}, Z^{[4]}_{\{3,4\}}$, it can be seen that the sub-matrix $[\mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}, \mathbf{a}_{\{3,4\}}]$ has rank 2, equal to the rank of $[\mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}]$, since $\mathbf{a}_{\{2,3\}} - \mathbf{a}_{\{2,4\}} = -\mathbf{a}_{\{3,4\}}$ (i.e., we align the three vectors $\mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}, \mathbf{a}_{\{3,4\}}$ into the linear space spanned by $\mathbf{a}_{\{2,3\}}$ and $\mathbf{a}_{\{2,4\}}$); thus the constraint in (11) is satisfied for user 1. Hence, the left null space of $[\mathbf{a}_{\{2,3\}}, \mathbf{a}_{\{2,4\}}, \mathbf{a}_{\{3,4\}}]$ contains exactly one linearly independent 3-dimensional vector, which could be $[1, 1, 1]$. Thus we let user 1 compute $Y_1^{[4]} = [1, 1, 1][F_1; F_2; F_3] = F_1 + F_2 + F_3$. For user 2, who cannot encode $Z^{[4]}_{\{1,3\}}, Z^{[4]}_{\{1,4\}}, Z^{[4]}_{\{3,4\}}$, it can be seen that the sub-matrix $[\mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}, \mathbf{a}_{\{3,4\}}]$ has rank 2, equal to the rank of $[\mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}]$, since $\mathbf{a}_{\{3,4\}} = \mathbf{a}_{\{1,3\}} - \mathbf{a}_{\{1,4\}}$; thus the constraint in (11) is satisfied for user 2. Hence, the left null space of $[\mathbf{a}_{\{1,3\}}, \mathbf{a}_{\{1,4\}}, \mathbf{a}_{\{3,4\}}]$ contains exactly one linearly independent 3-dimensional vector, which could be $[1, 0, 0]$. Thus we let user 2 compute $Y_2^{[4]} = [1, 0, 0][F_1; F_2; F_3] = F_1$. Similarly, the constraints in (11) are satisfied for users $3, 4$; thus we let user 3 compute $Y_3^{[4]} = [0, 1, 0][F_1; F_2; F_3] = F_2$, and let user 4 compute $Y_4^{[4]} = [0, 0, 1][F_1; F_2; F_3] = F_3$. It can be seen that any 3 of $Y_1^{[4]}, Y_2^{[4]}, Y_3^{[4]}, Y_4^{[4]}$ are linearly independent; thus the constraint in (12) is satisfied. Hence, for any $\mathcal{U}_2 \in \binom{[4]}{3}$, the server can recover F_1, F_2, F_3 from the second round. Thus from the two round transmissions, the server can recover $W_1 + \cdots + W_4$. Since the constraints in (5), (11), and (12) are satisfied, the proposed protocol is secure for the case $\mathcal{U}_1 = [4]$. Similarly, for the other value of $\mathcal{U}_1$, by taking the same vectors in (13), the proposed protocol is also secure. In conclusion, the achieved rates of `GroupSecAgg` are $(\mathsf{R}_1, \mathsf{R}_2) = (1, 1/3)$, coinciding with Theorem 1. $\square$



(a) `GroupSecAgg` v.s. `LightSecAgg`: $\mathsf{U} = (\mathsf{K}+1)/2$



(b) `GroupSecAgg` v.s. `SecAgg`: $\mathsf{U} = (\mathsf{K}+1)/2$

Fig. 2: The key sharing time and the model aggregation time of `GroupSecAgg` versus `LightSecAgg` and `SecAgg`, respectively.

IV. Experimental Results

We implement our proposed secure aggregation scheme (which is referred to as `GroupSecAgg` for the sake of simplicity) in Python2.7 by using the MPI4py library over the Amazon EC2 cloud, which is then compared to the original secure aggregation scheme in [5] (referred to as `SecAgg`), and the best existing information theoretic secure aggregation scheme with offline key sharing in [12] (referred to as `LightSecAgg`). We compare the key sharing times of `GroupSecAgg` and `LightSecAgg`, since the communication costs in the model aggregation phase of these two schemes are the same. Since `SecAgg` provides computational security instead of information theoretic security, the total size of needed keys is much smaller in `SecAgg`. Thus we compare the model aggregation times of `GroupSecAgg` and `SecAgg`.

Amazon EC2 Setup. The Amazon EC2 `t2.large` and `t2.xlarge` instances are selected, where we take one specific `t2.xlarge` instance as the server and all the other instances are users. The Amazon EC2 T2 instances have a 3.0 GHz Intel Scalable Processor, and all instances which we use in this experiment have the same capacity of computation, memory and network resources. The transmission speed is up to 100MB/s between the server and users. By setting the field size q as 7, we generate the input vectors uniformly i.i.d. over $\mathbb{F}_7$, and consider the three sizes of each input vector (100KB, 200KB, 300KB) as suggested in [5]. In the offline key sharing phase, we consider that each two users have a private link to communicate as in [12]; thus between each two users, we use the `MPI.send` command.

`GroupSecAgg` v.s. `LightSecAgg`. We first compare our `GroupSecAgg` with `LightSecAgg`, by considering the case where $\mathsf{U} = (\mathsf{K}+1)/2$ illustrated in Fig. 2a. Compared to `LightSecAgg`, `GroupSecAgg` reduces the key sharing time by at least 16.5% and at most 31.7% in Fig. 2a.

`GroupSecAgg` v.s. `SecAgg`. We then compare our `GroupSecAgg` with `SecAgg`, by considering the case where $\mathsf{U} = (\mathsf{K}+1)/2$ illustrated in Fig. 2b. Compared to `SecAgg`, `GroupSecAgg` reduces the model aggregation time by at least 48% and at most 53% in Fig. 2b.

Acknowledgement: The work of K. Wan was partially funded by the National Natural Science Foundation of China NSFC-12141107. The work of X. Yao and M. Ji was supported in part by National Science Foundation (NSF) CAREER Award 2145835. The work of H. Sun was supported in part by NSF Awards 2007108 and 2045656. The work of G. Caire was partially funded by the ERC Advanced Grant N. 789190, CARENET.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60.

[4] H. B. McMahan *et al.*, “Advances and open problems in federated learning,” *Foundations and Trends® in Machine Learning*, vol. 14, no. 1, 2021.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in *2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[6] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, “Secure single-server aggregation with (poly) logarithmic overhead,” in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1253–1269.

[7] B. Choi, J.-y. Sohn, D.-J. Han, and J. Moon, “Communication-computation efficient secure aggregation for federated learning,” *arXiv:2012.05433*, Dec. 2020.

[8] J. So, B. Güler, and A. S. Avestimehr, “Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning,” *IEEE Journal on Selected Areas in Info. Theory*, vol. 2, no. 1, pp. 479–489, 2021.

[9] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, “FastSecAgg: Scalable secure aggregation for privacy-preserving federated learning,” *arXiv:2009.11248*, Sep. 2020.

[10] T. Jahani-Nezhad, M. A. Maddah-Ali, S. Li, and G. Caire, “SwiftAgg+: Achieving asymptotically optimal communication load in secure aggregation for federated learning,” *arXiv:2202.04169*, Mar. 2022.

[11] Y. Zhao and H. Sun, “Information theoretic secure aggregation with user dropouts,” *arXiv:2101.07750*, Jan. 2021.

[12] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E. Ali, B. Guler, and S. Avestimehr, “LightSecAgg: a lightweight and versatile design for secure aggregation in federated learning,” *arXiv:2109.14236*, Feb. 2022.

[13] C. E. Shannon, “Communication theory of secrecy systems,” *in The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[14] V. R. Cadambe and S. A. Jafar, “Interference alignment and degrees of freedom of the k-user interference channel,” *IEEE Trans. Infor. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[15] K. Wan, H. Sun, M. Ji, and G. Caire, “On the information theoretic secure aggregation with uncoded groupwise keys,” *arXiv:2204.11364*, App. 2022.