Supplemental for Gradient Disaggregation: Breaking Privacy in Federated Learning by Reconstructing the User Participant Matrix

1. Experiments on ”Honest but Curious” Disaggregation Attack (neural network is updated, not fixed)

Prior experiments assumed that the central server fixed the neural network model. In the following experiments, we eliminate this assumption and update the neural network model with model updates computed from participants (via FedAvg). In this scenario, the attack becomes honest but curious as the attacker no longer has to modify the learning protocol (specifically, via fixing the model), but instead only needs to observe the gradient information and summary analytics collected by the server.

Dataset Size D	Batch Size b	Local Epochs e					
		1	2	4	8	16	32
64	8	1.0	1.0	.30	0.0	0.0	0.0
	16	1.0	1.0	1.0	.35	0.0	0.0
	32	1.0	1.0	1.0	1.0	.35	0.0
128	8	1.0	.52	0.0	0.0	0.0	0.0
	16	1.0	1.0	.46	0.0	0.0	0.0
	32	1.0	1.0	1.0	.49	0.0	0.0

Table 1. ”Honest but curious” gradient disaggregation on FedAvg updates: neural network model is updated with participants’ updates via FedAvg (SGD lr=1e-3) every round. Even with extra added noise from the changing model, our gradient disaggregation attack may reconstruct P exactly in a good portion of settings.

We run our experiments using FedAvg model updates, on Cifar10, with 100 users, 200 rounds, participation rate of .1, constraint granularity of 10, SGD lr of 1e-3, on a LeNet CNN model. Table 1 shows the fraction of P reconstructed across various FedAvg settings. Results show that with lower lr (1e-3), gradient disaggregation can exactly reconstruct P when FedAvg updates are small (1-4 epochs). With more epochs of FedAvg (and larger dataset size or smaller batch size), increased noise prevents reconstruction of P. We additionally tried the experiment with higher lr (1e-2), and disaggregation typically failed due to high noise, even with lower epochs of FedAvg. Our results indicate that under the right set of circumstances, the gradient disaggregation attack can be used in an honest but curious scenario; however, more robust results are achieved if the attacker can fix the neural network model.

2. Experiments on Partial Constraints

We perform experiments showing our gradient disaggregation attack with partial sets of constraints. Specifically, constraint granularity determines the rounds across which number of participations is known (e.g: if granularity is 10, we know how many times each user participated between rounds 0-9, 10-19, 20-29, etc) and we drop a specific fraction of these constraints (e.g: in prior setting, only knowing participation counts between rounds 0-9, 40-49, etc). Testing this scenario shows the degree to which our method works when only partial summary analytics is given. We enforced a time limit of 10 minutes for solving each column

Table 2 shows the fraction of P reconstructed across various proportions of dropped constraints. Results indicate that even when significant proportions of constraints are dropped, P may be exactly recovered with more rounds of collected aggregated updates. Note that, when rounds $<$ users reconstruction fails due to the rank being less than the number of users.

3. Experiments on Inexact ”Noisy” Constraints

We perform experiments showing our gradient disaggregation attack when the constraints are inexact. For example, if analytics specified that a user participated 5 times between rounds 0-10, when the user actually participated 4 times. To handle noisy constraints, we relax our formulation and convert participation constraints into soft constraints:

Users	Rounds	Constraint Fraction									
		1	.9	.8	.7	.6	.5	.4	.3	.2	.1
128	256	1	1	1	1	1	.97	.71	.26	.08	.03
	512	1	1	1	1	1	1	1	1	.99	.38
	1024	1	1	1	1	1	1	1	1	1	.99
	2048	1	1	1	1	1	1	1	1	1	1
	4096	1	1	1	1	1	1	1	1	1	1
256	256	0	0	0	0	0	0	0	0	0	0
	512	1	1	1	.98	.94	.63	.09	.02	0	0
	1024	1	1	1	1	1	1	1	.99	.70	.10
	2048	1	1	1	1	1	1	1	1	1	.96
	4096	1	1	1	1	1	1	1	1	1	1
512	256	0	0	0	0	0	0	0	0	0	0
	512	0	0	0	0	0	0	0	0	0	0
	1024	1	.99	.99	.93	.53	.09	0	0	0	0
	2048	1	1	1	1	1	1	1	.99	.55	.01
	4096	1	1	1	1	1	1	1	1	1	.85
1024	256	0	0	0	0	0	0	0	0	0	0
	512	0	0	0	0	0	0	0	0	0	0
	1024	0	0	0	0	0	0	0	0	0	0
	2048	1	1	.99	.95	.41	.01	0	0	0	0
	4096	1	1	1	1	1	1	.99	.81	.1	0

Table 2. Fraction of P recovered with gradient disaggregation on partial constraint information. With more rounds, we can exactly recover P even when a significant fraction of constraints are missing.

$$\begin{aligned} \min \, \|Nul(G^T_{aggregated})p_k\|^2 + \lambda(C_k p_k - c_k) \\ p_k \in \{0,1\}^n \end{aligned} \tag{1}$$

Where λ is a reweighting factor for the participation constraints.

Table 3 shows the results of gradient disaggregation when constraint noise $\mathcal{N}(0, \mu)$ is added to each constraint. As indicated, even in the presence of noise, gradient disaggregation may exactly recover P with enough rounds.

Number of Users	Rounds	Constraint Noise		
		.3	.5	1
32	128	1	1	.63
	256	1	1	.97
	512	1	1	1
	1024	1	1	1
	2048	1	1	1
64	128	1	1	.34
	256	1	1	.97
	512	1	1	1
	1024	1	1	1
	2048	1	1	1
128	128	0	0	0
	256	1	1	.7
	512	1	1	.99
	1024	1	1	1
	2048	1	1	1
256	128	0	0	0
	256	0	0	0
	512	1	1	.48
	1024	1	1	1
	2048	1	1	1

Table 3. Fraction of P recovered with gradient disaggregation when constraints are inexact/noisy. Participation rate=.1, λ=.1, constraint granularity = 10. Reconstruction is more successful with more rounds.

4. Extended Experiments on Participation Rate

We provide extended data on gradient disaggregation against various parameter values of participation rate. Participation rate is the proportion of users that participate in sending model updates per round and impacts how many updates are summed to yield the aggregated update that is observed by the central server. Unless stated, we enforced a time limit of 10 minutes for solving each column; we use a constraint granularity of 10. We show our extended results in Table 4.

Users	Rounds	Participation Rate		
		.1	.2	.4
128	256	1	1	.05
	512	1	1	1
	1024	1	1	1
	2048	1	1	1
	4096	1	1	1
256	256	0	0	0
	512	1	.73	0
	1024	1	1	1
	2048	1	1	1
	4096	1	1	.31
512	256	0	0	0
	512	0	0	0
	1024	1	.34	0
	2048	1	1	1
	4096	1	1	.03
1024	256	0	0	0
	512	0	0	0
	1024	0	0	0
	2048	1	.02	0
	4096	1	.48	0
	5120*	1	1	0

Table 4. Fraction of P recovered via gradient disaggregation for various participation rates. * indicates settings where the time limit for solving each column was increased to 60 minutes (vs 10 minutes). Generally, using more rounds facilitates more successful reconstruction; note that with larger number of users and rounds, success rate decreased due to exceeding the 10 minute time limit.

5. Extended Experiments on Constraint Granularity

We provide extended data on gradient disaggregation against various parameter values of constraint granularity. Constraint granularity is how precise summary statistics capture user partipation frequency (see main paper for details). Unless stated, we enforced a time limit of 10 minutes for solving each column; we use a constraint granularity of 10. We show our extended results in Table 5.

Users	Rounds	Constraint Granularity			
		10	20	40	80
128	256	1	1	.99	.95
	512	1	1	1	1
	1024	1	1	1	1
	2048	1	1	1	1
	4096	1	1	1	1
256	256	0	0	0	0
	512	1	.94	.27	.02
	1024	1	1	.97	.44
	2048	1	1	1	1
	4096	1	1	1	.86
512	256	0	0	0	0
	512	0	0	0	0
	1024	1	.5	0	0
	2048	1	1	.98	.24
	4096	1	1	.38	.035
1024	256	0	0	0	0
	512	0	0	0	0
	1024	0	0	0	0
	2048	1	.23	0	0
	4096	1	.91	0	0

Table 5. Fraction of P recovered via gradient disaggregation for various constraint granularities.

6. Extended Experiments on FedAvg Updates (Cifar10)

We provide extended experiments on gradient disaggregation on FedAvg. We show our results in Table 6 and 7.

User Dataset Size	Batch Size	Local Epochs						
		1	2	4	8	16	32	64
384	8	1	1	1	1	1	1	1
	16	1	1	1	1	1	1	1
	32	1	1	1	1	1	1	1
	64	1	1	1	1	1	1	1
384 (mom.=.9)	8	.96	.88	.78	.97	.83	.19	.01
	16	1	1	1	1	1	1	1
	32	1	1	1	1	1	1	1
	64	1	1	1	1	1	1	1
384 (fraction=.9)	8	.98	1	1	1	.99	1	1
	16	1	1	1	1	1	1	1
	32	1	1	1	1	1	1	.99
	64	1	1	1	1	1	1	1
384 (fraction=.8)	8	1	1	1	.99	.83	.92	1
	16	.91	.95	.99	1	.97	.84	.95
	32	.99	.99	1	1	1	1	1
	64	1	1	1	1	1	1	1

Table 6. Fraction of P recovered on FedAvg updates using larger LeNet model (last hidden layer size=512), across various settings (mom.= SGD momentum, fraction=fraction of data sampled from the 384 examples to perform FedAvg over). Users=100, rounds=200, participation rate=.1, constraint granularity=10.

Number of Users	Batch Size	Local Epochs					
		1	2	4	8	16	32
512	16	1	1	1	.996	1	1
1024	16	.998	1	.999	1	1	1

Table 7. Fraction of P recovered on FedAvg updates using larger LeNet model (last hidden layer size=512), With more users. (rounds=200, participation rate=.1, constraint granularity=10). With many users, P is still reconstructable.