Appendix

The organization of this Appendix is given below. Appendix A presents additional details on the experiments. In Appendix B, we present the proofs of security and convergence guarantees of our method. Finally, in Appendix C, we provide the complexity analysis of the proposed DReS-FL framework and compare it with other secure aggregation methods.

A Additional Experimental Details

All experiments are performed by Pytorch on an Intel Xeon Gold 6246R CPU @ 3.40 GHz and a Geforce RTX 3090.

Datasets. We select six image datasets in the experiments, including MNIST, Fashion-MNIST, EMNIST (Balanced), CIFAR-10, CIFAR-100, and SVHN. Specifically, we use the balanced subset of EMNIST to conduct experiments. The extra training samples [1] in the SVHN dataset are not utilized on the experiment. More details of these datasets are summarized in Table 1.

Model structures. The neural network for MNIST, Fashion-MNIST, and EMNIST datasets is a two-layer multi-layer perception (MLP) with 64 hidden units each. For CIFAR-10, CIFAR-100, and SVHN datasets, we resize the input images from 32×32 to 224×224 and adopt the convolutional layers of a pretrained VGG model to extract 25088-dimensional features. To classify the extracted features, we select a two-layer MLP model with 4096 hidden units each.

Baselines. We select algorithm-based methods as baselines, including FedAvg [2], FedAvg with importance sampling (FedAvg-IS) [3, 4], and SCAFFOLD [5]. These methods can be easily combined with secure aggregation methods [6, 7] to protect the privacy of clients' local models. Specifically, we perform secure aggregation for both the local model updates and the control variates in SCAFFOLD. Data-centric approaches [8, 9, 10, 11, 12, 13] are not compared since some of them [8, 9, 10] cannot provide strong privacy guarantees while others [11, 12, 13] do not naturally extend to federated neural network training with multiple clients.

Hyperparameters. We adopt mini-batch SGD with a batch size of 64 to optimize the models in federated training. The communication round is set to be 7×10^4, and the clients perform one local SGD step in each round. The learning rate is initialized as 0.1, and it will decay with a factor of 0.65 after every 1500 rounds. Other parameters in our DRes-FL framework are summarized in Table 2.

B Proofs

B.1 Proof of Theorem 2

According to Theorem 1, the mutual information between the local dataset of client i and the encoded dataset $(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i)$ is zero for $i \in [N]$. By the chain rule of mutual information, the conditional mutual information $I(\overline{\mathbf{X}}_i, \overline{\mathbf{Y}}_i; \widetilde{\mathbf{X}}_i^{(\mathcal{I})}, \widetilde{\mathbf{Y}}_i^{(\mathcal{I})}, \mathbf{w}|\mathbf{w})$ equals zero. Due to the data processing inequality, the central server can infer no information about local dataset $(\overline{\mathbf{X}}_i, \overline{\mathbf{Y}}_i)$ from stochastic gradient $\widetilde{\boldsymbol{g}}(\widetilde{\mathbf{X}}_i^{(\mathcal{I})}, \widetilde{\mathbf{Y}}_i^{(\mathcal{I})}; \mathbf{w})|\mathbf{w})$ beyond the global model $\mathbf{w}$, i.e., $I(\overline{\mathbf{X}}_i, \overline{\mathbf{Y}}_i; \widetilde{\boldsymbol{g}}(\widetilde{\mathbf{X}}_i^{(\mathcal{I})}, \widetilde{\mathbf{Y}}_i^{(\mathcal{I})}; \mathbf{w})|\mathbf{w}) = 0$, for any $i \in [N]$ and index set $\mathcal{I}$.

B.2 Proof of Theorem 4

For simplicity, we denote $\boldsymbol{g}^{(t)} = \frac{1}{bK} \sum_{j=1}^{K} \boldsymbol{g}(\widetilde{\mathbf{X}}_j^{(\mathcal{I}_t)}, \widetilde{\mathbf{Y}}_j^{(\mathcal{I}_t)}; \mathbf{w}^{(t)})$ in this section. The server updates the global model by $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - Q(\eta \boldsymbol{g}^{(t)})$ in each round t after receiving $\deg(\boldsymbol{g})(K+T-1)+1$ uploads from clients. We first provide an important lemma to show that the model update $Q(\eta \boldsymbol{g}^{(t)})$ on the server is an unbiased estimate of $\eta \boldsymbol{g}_e(\mathbf{w}^{(t)})$.

Table 1: Details of the datasets

	MNIST	Fashon-MNIST	EMNIST	CIFAR-10	CIFAR-100	SVHN
No. of classes	10	10	47	10	100	10
No. of training samples	60,000	60,000	112,800	50,000	50,000	73,257
No. of test samples	10,000	10,000	18,800	10,000	10,000	26,032
Image size	28×28	28×28	28×28	32×32	32×32	32×32
License	Creative Commons Attribution-Share Alike 3.0 License	MIT License	Apache License 2.0	MIT License	MIT License	CC0:Public Domain License

Table 2: Hyperparameters for our DReS-FL method

Parameters	MNIST	Fashion-MNIST	EMNIST	CIFAR-10	CIFAR-100	SVHN
Maximum L2-norm for gradient clipping	2×10^4	2×10^4	5×10^6	2×10^4	1×10^9	2×10^4
Prime number p	$10^{31}+33$	$10^{31}+33$	$10^{51}+121$	$10^{31}+33$	$10^{71}+273$	$10^{31}+33$
Parameter l in data transformation	4	4	4	2	2	2

Lemma 1. *(Unbiased and variance-bounded model update) In the t-th round, the model update $Q(\eta \boldsymbol{g}^{(t)})$ has the following properties:*

$$\mathbb{E}\left[Q(\eta \boldsymbol{g}^{(t)})\right] = \eta \boldsymbol{g}_e(\mathbf{w}), \tag{1}$$

$$\mathbb{E}\left[\|Q(\eta \boldsymbol{g}^{(t)}) - \eta \boldsymbol{g}_e(\mathbf{w})\|^2\right] \leq (\gamma^2+1)\eta^2\frac{\sigma^2}{bK} + \gamma^2\eta^2\left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2. \tag{2}$$

Proof. According to Assumption 2-3, we directly obtain that $\mathbb{E}\left[Q(\eta \boldsymbol{g}^{(t)})\right] = \mathbb{E}\left[\eta \boldsymbol{g}^{(t)}\right] = \boldsymbol{g}_e(\mathbf{w})$. Since the batch sampling and rounding operation cause independent errors, the variance is upper bounded as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left\|Q(\eta \boldsymbol{g}^{(t)}) - \eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] \\
&= \mathbb{E}\left[\left\|Q(\eta \boldsymbol{g}^{(t)}) - \eta \boldsymbol{g}^{(t)}\right\|^2\right] + \mathbb{E}\left[\left\|\eta \boldsymbol{g}^{(t)} - \eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] \\
&\overset{(a)}{\leq} \gamma^2\mathbb{E}\left[\left\|\eta \boldsymbol{g}^{(t)}\right\|^2\right] + \mathbb{E}\left[\left\|\eta \boldsymbol{g}^{(t)} - \eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] \\
&= \gamma^2\eta^2\left[\mathbb{E}\left[\left\|\boldsymbol{g}^{(t)} - \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] + \left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] + \eta^2\mathbb{E}\left[\left\|\boldsymbol{g}^{(t)} - \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] \\
&\overset{(b)}{\leq} (\gamma^2+1)\eta^2\frac{\sigma^2}{bK} + \gamma^2\eta^2\left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2,
\end{aligned}$$

where (a) follows Assumption 3. (b) is due to Assumption 2 and the independence of mini-batch sampling noises among clients. □

With Lemma 1, we prove Theorem 4 as follows:

Proof. The model update in the t-th iteration can be expressed as $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - Q(\eta \boldsymbol{g}^{(t)})$. According to the Taylor's expansion, we have:

$$
\begin{aligned}
&\mathbb{E}\left[\ell(\mathbf{w}^{(t+1)})\right] - \mathbb{E}\left[\ell(\mathbf{w}^{(t)})\right] \\
&\leq -\mathbb{E}\left\langle \boldsymbol{g}_e(\mathbf{w}^{(t)}), Q(\eta \boldsymbol{g}^{(t)})\right\rangle + \frac{L}{2}\mathbb{E}\left[\left\|Q(\eta \boldsymbol{g}^{(t)})\right\|^2\right] \\
&\overset{(c)}{=} -\mathbb{E}\left\langle \boldsymbol{g}_e(\mathbf{w}^{(t)}), \eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\rangle + \frac{L}{2}\mathbb{E}\left[\left\|Q(\eta \boldsymbol{g}^{(t)})\right\|^2\right] \\
&\overset{(d)}{=} -\eta \left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2 + \frac{L}{2}\mathbb{E}\left[\left\|Q(\eta \boldsymbol{g}^{(t)}) - \eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] + \frac{L}{2}\mathbb{E}\left[\left\|\eta \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right] \\
&\overset{(e)}{\leq} -\left(\eta - \frac{\eta^2 L}{2}\right)\left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2 + \frac{L}{2}\left((\gamma^2+1)\eta^2\frac{\sigma^2}{bK} + \gamma^2\eta^2\left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2\right) \\
&= -\left(\eta - \frac{\eta^2 L}{2} - \frac{\eta^2\gamma^2 L}{2}\right)\left\|\boldsymbol{g}_e(\mathbf{w}^{(t)})\right\|^2 + \frac{\eta^2 L\sigma^2}{2bK}(\gamma^2+1),
\end{aligned}
$$

where (c) follows Lemma 1, (d) holds according to the fact that $\mathbb{E}\left\langle \nabla Q(\eta \boldsymbol{g}^{(t)}) - \eta \boldsymbol{g}_e(\mathbf{w}^{(t)}), \boldsymbol{g}_e(\mathbf{w}^{(t)})\right\rangle = 0$, and (e) is due to Assumption 2-3. If it holds that $\eta - \frac{\eta^2 L}{2} - \frac{\eta^2\gamma^2 L}{2} > 0$, we summarize the above inequality over $t = 1, 2, \ldots, \tau'$ to conclude the proof. $\square$

C Complexity Analysis and Comparison

In this part, we analyze the communication and computational complexities of the proposed DReS-FL framework with respect to the parameters $(N, T, K, \tau, d_w, b_g)$. Parameter N is the number of clients, and T denotes the privacy threshold in Lagrange coding [14]. Parameter K denotes the number of shards in the local datasets. A large value of K reduces the communication overhead in secret data sharing and the local computation loads of clients. In the federated training, parameter τ corresponds to the number of communication rounds. Parameters d_w and b_g denotes the model size and the global batch size, respectively. Before training starts, each client's computation cost for Lagrange coding and communication complexity for data sharing are $\mathcal{O}(N\log^2(K+T)\log\log(K+T))$ and $\mathcal{O}(N/K)$, respectively. In each round of federated training, the local computation complexity is $\mathcal{O}(d_w b_g/K)$, and the model uploading cost is $\mathcal{O}(d_w)$. Besides, the communication overhead of the server for model distributing is $\mathcal{O}(Nd_w)$, and the model decoding complexity by polynomial interpolation is $\mathcal{O}(R\log^2 R\log\log Rd_w)$, where R denotes the minimum uploads needed for gradient decoding. The model uploading cost of each client is $\mathcal{O}(d_w)$ and the communication overhead of global model downloading is $\mathcal{O}(Nd_w)$.

Different from our method, secure aggregation approaches [6, 15, 16, 17, 18, 7] generate random masks to protect the local model parameters. In each round, clients first share random-seeds [6, 15, 16, 18] or coded masks [7] with each other which allows for aggregating the masked models at the server. As some clients may drop out of the training process unexpectedly, the surviving clients upload the shared information belonging to the dropped clients to reconstruct the aggregated model. The main drawback of such approaches is that clients need to generate new masks in each round, and thus the extra costs are proportional to the number of training rounds. In comparison, our method introduces extra costs in secret data sharing before the training starts, and its communication and computational complexities are independent of the training round τ. In the scenario that the training round is very large, the proposed DReS-FL method could reduce the latency compared with the secure aggregation protocols. The complexities comparisons with FedAvg and the well-known LightSecAgg [7] method are summarized in Table 3 and 4.

Table 3: Computational complexity comparison

	Preparation	Iterative training (τ rounds)		
	Lagrangian coding	Generating coded random masks	Local model update	Global model aggregation
FedAvg	—	—	$\mathcal{O}\left(\tau d_w b_g / N\right)$	$\mathcal{O}(\tau N d_w)$
FedAvg with LightSecAgg	—	$\mathcal{O}\left(\frac{\tau d_w N^2 \log N}{R-T}\right)$	$\mathcal{O}\left(\tau d_w b_g / N\right)$	$\mathcal{O}\left(\frac{\tau d_w R \log R}{R-T}\right)$
DReS-FL	$\mathcal{O}(N^2 \log^2(K+T) \log\log(K+T))$	—	$\mathcal{O}\left(\tau d_w b_g / K\right)$	$\mathcal{O}(\tau d_w R \log^2 R \log\log R)$

Table 4: Communication complexity comparison

	Preparation	Iterative training (τ rounds)			
	Data sharing	Coded masks sharing among clients	Local model uploading	Coded masks uploading	Global model downloading
FedAvg	—	—	$\mathcal{O}(\tau d_w)$	—	$\mathcal{O}(\tau N d_w)$
FedAvg with LightSecAgg	—	$\mathcal{O}\left(\frac{\tau N^2 d_w}{R-T}\right)$	$\mathcal{O}(\tau d_w)$	$\mathcal{O}\left(\frac{\tau d_w R}{R-T}\right)$	$\mathcal{O}(\tau N d_w)$
DReS-FL	$\mathcal{O}(N^2/K)$	—	$\mathcal{O}(\tau d_w)$	—	$\mathcal{O}(\tau N d_w)$

References

[1] Netzer, Y., T. Wang, A. Coates, et al. Reading digits in natural images with unsupervised feature learning. 2011.

[2] McMahan, B., E. Moore, D. Ramage, et al. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[3] Ren, J., Y. He, D. Wen, et al. Scheduling for cellular federated edge learning with importance and channel awareness. *IEEE Transactions on Wireless Communications*, 19(11):7690–7703, 2020.

[4] Kairouz, P., H. B. McMahan, B. Avent, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[5] Karimireddy, S. P., S. Kale, M. Mohri, et al. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[6] Bonawitz, K. and Ivanov, Vladimir and Kreuter, Ben and Marcedone, Antonio and McMahan, H Brendan and Patel, Sarvar and Ramage, Daniel and Segal, Aaron and Seth, Karn. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. 2017.

[7] Yang, C.-S., J. So, C. He, et al. Lightsecagg: Rethinking secure aggregation in federated learning. *arXiv preprint arXiv:2109.14236*, 2021.

[8] Zhao, Y., M. Li, L. Lai, et al. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[9] Jeong, E., S. Oh, H. Kim, et al. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[10] Zhang, L., B. Shen, A. Barnawi, et al. Feddpgan: federated differentially private generative adversarial networks framework for the detection of covid-19 pneumonia. *Information Systems Frontiers*, 23(6):1403–1415, 2021.

[11] Mohassel, P., Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[12] So, J., B. Güler, A. S. Avestimehr. Codedprivateml: A fast and privacy-preserving framework for distributed machine learning. *IEEE Journal on Selected Areas in Information Theory*, 2(1):441–451, 2021.

[13] So, J., B. Guler, S. Avestimehr. A scalable approach for privacy-preserving collaborative machine learning. *Advances in Neural Information Processing Systems*, 33:8054–8066, 2020.

[14] Yu, Q., S. Li, N. Raviv, et al. Lagrange coded computing: Optimal design for resiliency, security, and privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1215–1225. PMLR, 2019.

[15] Bonawitz, K., V. Ivanov, B. Kreuter, et al. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. 2017.

[16] Kadhe, S., N. Rajaraman, O. O. Koyluoglu, et al. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020.

[17] Jahani-Nezhad, T., M. A. Maddah-Ali, S. Li, et al. Swiftagg+: Achieving asymptotically optimal communication load in secure aggregation for federated learning. *arXiv preprint arXiv:2203.13060*, 2022.

[18] Bell, J. H., K. A. Bonawitz, A. Gascón, et al. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1253–1269. 2020.