PARAAEGIS: PARALLEL PROTECTION FOR FLEXIBLE PRIVACY-PRESERVED FEDERATED LEARNING

Zihou Wu[⋆]*, Yuecheng Li*[⋆]*, Tianchi Liao*[†]*, Jian Lou*[†]*, Chuan Chen*[⋆,*]

[⋆]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[†]School of Software Engineering, Sun Yat-sen University, Zhuhai, China

ABSTRACT

Federated learning (FL) faces a critical dilemma: existing protection mechanisms like differential privacy (DP) and homomorphic encryption (HE) enforce a rigid trade-off, forcing a choice between model utility and computational efficiency. This lack of flexibility hinders the practical implementation. To address this, we introduce ParaAegis, a parallel protection framework designed to give practitioners flexible control over the privacy-utility-efficiency balance. Our core innovation is a strategic model partitioning scheme. By applying lightweight DP to the less critical, low norm portion of the model while protecting the remainder with HE, we create a tunable system. A distributed voting mechanism ensures consensus on this partitioning. Theoretical analysis confirms the adjustments between efficiency and utility with the same privacy. Crucially, the experimental results demonstrate that by adjusting the hyperparameters, our method enables flexible prioritization between model accuracy and training time.

***Index Terms*—** Federated learning, differential privacy, homomorphic encryption

1. INTRODUCTION

Federated learning (FL), a distributed machine learning paradigm [1], has garnered increasing interest on account of its capability to facilitate limited data flow between data silos. However, while FL has distinct advantages, it also faces potential challenges, with those concerning privacy and security being a significant research direction at present. Since this process involves transferring information over a complex network environment, malicious devices can more easily eavesdrop on the link and intercept the transmitted models. Although the local data is not directly included in the model parameters, there is research that demonstrates that it is possible to reconstruct the local training data from the transmitted models [2, 3].

Differential privacy (DP) and homomorphic encryption (HE) are the most widely used to ensure privacy and security. DP provides a formal mathematical framework to measure the privacy of a system. In the context of FL, it typically involves techniques like gradient clipping and noise addition to meet the DP requirements. On the other hand, HE enables computations like addition or multiplication to be performed on encrypted data, without revealing the underlying information. DP tends to reduce model accuracy, with the impact growing as the level of privacy protection increases [4]. HE, while preserving data security, significantly increases both the training time and communication overhead [5]. Consequently, balancing privacy, utility, and efficiency in FL remains a significant challenge, and current research is exploring hybrid approaches that combine different protection methods to address these trade-offs.

Some research has explored the potential of balancing privacy, accuracy and efficiency [6–8]. However, these works do not further investigate the optimization of the implicit trade-offs between privacy, utility, and efficiency. To tackle this issue, we propose ParaAegis, a DP-HE-parallelly-utilizing FL framework, where the model parameters are partitioned into two parts and protected by DP and HE respectively. To address the issue of model partition consistency accross clients, we propose a voting mechanism. In this mechanism, clients upload its local partition to the server, which then counts the occurrences of each index and selects the most frequent ones as the global partition. Our theoretical analysis and emprical results show that the model partition for parallel protection, involving DP and HE, provides significant flexibility on privacy-utility-efficiency trade-offs. Moreover, with a fixed partition ratio, the partition strategy selecting HE part based on the maximum norm further enhances model accuracy without compromising privacy and efficiency. The code is available on `https://anonymous.4open.science/r/ParaAegis/`.

2. PRELIMINARY

In the context of FL, we consider n clients, each of which has a distinct local dataset. The local objective function for the i-th client is denoted by $f_i := \sum_{(\mathbf{x},y)\in\mathcal{D}_i} \ell_i(\mathbf{w};\mathbf{x},y)$, where $\mathbf{w}$ and $(\mathbf{x},y)$ are the parameters and samples respectively. The global objective is the weighted average of the local objectives, weighted by the number of data samples on each client. During the learning process, the clients and the server alter-

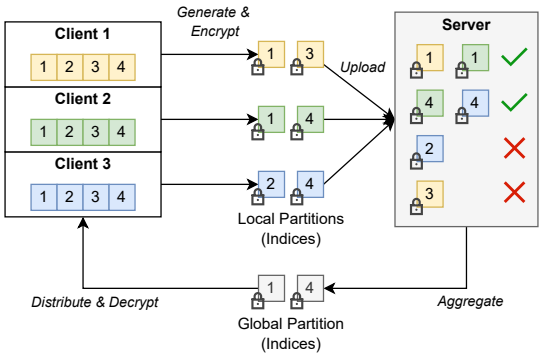

Fig. 1. Illustration of the VOTING mechanism. Each client proposes indices for HE protection. The server aggregates these proposals and selects the top-k most frequently proposed indices (e.g., indices 1 and 4) to form the global partition for all clients.

nately exchange local updates $\mathbf{u}_i$ and global updates $\mathbf{u}$.

To protect transmitted updates from inversion attacks, some methods propose applying the θ-norm clipping and Gaussian noise injection to provide theoretical guarantees of client-level DP (see [4] for the formal definition):

$$\hat{\mathbf{u}}_i := \frac{\mathbf{u}_i}{\max\{1, \|\mathbf{u}_i\|^2 \theta\}} + \mathbf{z}_i \sim \mathcal{N}(0, \sigma_z \mathbf{I}). \quad (1)$$

DP has a negligible impact on efficiency. However, its adverse effect on utility primarily stems from clipping and noise addition, and some studies have indicated a negative correlation between the magnitude of these operations and model accuracy.

The aforementioned methods often sacrifice model accuracy. An alternative approach, which prioritizes utility at the expense of efficiency, HE [9]. With this cryptographic method, clients encrypt their local updates into ciphertexts $[\![\mathbf{u}_i]\!]$ using a public key. The server then aggregates these ciphertexts in the encrypted domain by leveraging the homomorphic addition primitive, resulting in an encrypted global update $[\![\mathbf{u}]\!]$. Finally, the clients can decrypt this result using the private key. Throughout this process, the cryptographic properties of HE guarantee that all information remains computationally secure. HE has a negligible impact on utility, while its efficiency overhead typically scales linearly with the plaintext length. Among the various HE schemes, the CKKS cryptosystem [10] is particularly efficient for FL, owing to its plaintext space supporting real-valued vectors and its inherent batching capabilities. Therefore, the HE component in this paper is based on the CKKS scheme.

3. METHODOLOGY

In this section, we elaborate on the proposed DP-HE parallel protection method ParaAegis. Our approach is detailed in Alg.1 and motivated by the observation that not all parameters in a deep learning model contribute equally to the learning process. Updates with larger norms tend to be more influential in guiding the model's convergence. However, as shown in Eq.1, these large-norm updates are also the most severely impacted by the clipping and noise injection inherent in DP, which can harm model utility.

Algorithm 1 Proposed ParaAegis

Initialize the global model $\mathbf{w}_0$ and broadcast it to clients
for $t = 1$ to T **do**
  Randomly sample clients $\mathcal{C}_t \subseteq [M]$
  for all $i \in \mathcal{C}_t$ **in parallel do**
    Train the model for K epochs locally;
    Compute the local update $\mathbf{u}_i^t := \mathbf{w}_i^{t,K} - \mathbf{w}_i^t$;
    Select the indices of $\mathbf{u}_i^t$ with the highest $r\%$ norm to construct a partition vector $\mathbf{v}_i^t$;
    Upload $\mathbf{v}_i^t$ to the server;
  end for
  Aggregate the partition $\mathbf{v}^t := \text{VOTING}(\mathbf{v}_i^t)$;
  Send $\mathbf{v}^t$ to clients;
  for all $i \in \mathcal{C}_t$ **in parallel do**
    Divide $\mathbf{u}_i^t$ into $\mathbf{u}_{i,\text{DP}}^t$ and $\mathbf{u}_{i,\text{HE}}^t$ according to $\mathbf{v}^t$;
    Perturb $\mathbf{u}_{i,\text{DP}}^t$ to $\hat{\mathbf{u}}_{i,\text{DP}}^t$ by Eq. 1;
    Encrypt $\mathbf{u}_{i,\text{HE}}^t$ by public key to $[\![\mathbf{u}_{i,\text{HE}}^t]\!]$;
    Upload $\hat{\mathbf{u}}_{i,\text{DP}}^t$ and $[\![\mathbf{u}_{i,\text{HE}}^t]\!]$ to the server;
  end for
  Aggregate updates into $\hat{\mathbf{u}}_{\text{DP}}^t$ and $[\![\mathbf{u}_{\text{HE}}^t]\!]$ seperatedly, then send them to all clients;
  for all $i = 0$ to $M - 1$ **in parallel do**
    Decrypt $[\![\mathbf{u}_{\text{HE}}^t]\!]$ to $\mathbf{u}_{\text{HE}}^t$ by secret key;
    Reorganize $\hat{\mathbf{u}}_{\text{DP}}^t$ and $\mathbf{u}_{\text{HE}}^t$ into $\mathbf{u}^t$ by $\mathbf{v}^t$;
    $\mathbf{w}^{t+1} := \mathbf{w}^t + \mathbf{u}^t$;
  end for
end for

The core tenet of our framework is to partition the local update vector, $\mathbf{u}_i^t$, into two disjoint subsets. A small, high-norm subset, which we denote $\mathbf{u}_{i,\text{HE}}^t$, is protected by HE, preserving its precision. The remaining majority of parameters, $\mathbf{u}_{i,\text{DP}}^t$, are protected by the DP mechanism. This partitioning allows the DP-protected group to use a much smaller clipping threshold, thereby reducing noise and preserving utility. Concurrently, limiting the application of HE to a small subset minimizes its computational and communication overhead.

The vector-based nature of the CKKS scheme necessitates a common partition for all clients, requiring a consensus on the partition indices. We achieve this consensus via a server-aggregated voting mechanism inspired by FL model aggregation (Fig. 1). In this process, each client "votes" for indices of its highest-norm parameters as $\mathbf{v}_i$. However, transmitting these indices in plaintext would pose a significant privacy risk by revealing information about local updates. To mitigate this, indices are encrypted. Since the server's task

is limited to counting votes, standard non-homomorphic encryption is sufficient. The server receives all encrypted local partitions, determines the global partition by selecting the most frequently occurring indices (identifiable as unique ciphertexts), and then distributes this partition to all clients to apply to their respective updates.

The trade-offs inherent in the ParaAegis framework are adjustable through the partition ratio. For simplicity, we define r as the proportion of the total parameters allocated to the HE-protected partition, with the remaining $(1-r)$ portion being protected by DP. This ratio acts as a crucial lever: as r decreases, protection leans more on DP, enhancing efficiency at a potential cost to utility, while a larger r shifts the balance towards HE, improving precision at the expense of computational overhead.

Crucially, in ParaAegis, this partition ratio is not required to be fixed, allowing for a more flexible approach tailored to the training dynamics. This is motivated by the observation that in the early stages of training, backpropagation gradients typically have larger norms and establish the general direction of convergence; interfering with them via noise can slow down the process. Conversely, gradients in later stages tend to have smaller norms and can benefit from the regularizing effect of randomness, making them more tolerant to noise. Inspired by this, we designed a dynamic decay mechanism for the partition ratio, where the number of HE-protected parameters gradually decreases as training progresses. We set an initial HE ratio r_0 and a decay rate $\lambda \in (0,1)$, and after each round, the proportion of HE-protected parameters is updated according to $r_t := \lambda r_{t-1}$. This mechanism allows us to focus on accuracy during the early training stages by minimizing deviations in the convergence path, and then shift emphasis towards efficiency in the later stages by reducing the computational cost of HE, thereby minimizing training time while preserving model utility. We validate the rationale of this design in the subsequent experimental analysis.

4. ANALYSIS

We establish the privacy and convergence guarantees for ParaAegis. Due to space constraints, we present the main theorems. The complete, detailed proofs are available in Appendix A.

Assumption 1. *We assume standard conditions for FL convergence analysis [11]: a smooth loss function f with bounded gradient norms and variances across clients and data samples.*

Theorem 1 (Convergence of ParaAegis)**.** *Under Assumption 1, and if the noise injected satisfies (ε, δ)-DP, the convergence of ParaAegis is bounded by*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\mathbf{w}^t)\|^2] \leq O\left(\frac{1}{T}\right) + \underbrace{C_1 \cdot (1-r)}_{\textit{Clipping Effect}} + \underbrace{\frac{C_2 \cdot (1-r)\ln(1/\delta)}{N^2\epsilon^2}}_{\textit{Privacy Noise}}, \tag{2}$$

where C_1 and C_2 are constants that depend on problem parameters such as the gradient bound and Lipschitz constants, and N is number of clients.

By adjusting the partition ratio r and privacy budget (ε, δ), it is clear that with the privacy and efficiency changing directly, the utility, which is represented by the convergence bound shown in Theorem 1 is affected correspondingly.

5. EXPERIMENTS

Experimental Setup. We conduct experiments on Imagenette [17] dataset and ResNet-18 model [18], whose parameter sizes are sufficient to highlight the advantages of our method in balancing privacy, efficiency, and utility. For the federated learning setup, the total number of clients is $N = 10$. The training proceeds for a total of $T = 50$ global rounds, with each client performing $K = 3$ local epochs per round. Regarding the training hyperparameters, the batch size $|\mathcal{B}|$ is set to 32, the learning rate η is 0.01, and the gradient clipping threshold θ is 1.

Benchmarks. To evaluate the flexibility of Paraaegis with respect to the privacy-utility-efficiency trade-off, we benchmark it against CKKS-FedAvg [13] and DP-FedAvg [12]. We also consider the serial combination of DP and HE [14] as a competitive alternative. Additionally, two methods considering utility-efficiency trade-off with privacy guarantee are included, where the former utilizes amplitude-varying DP [15] with the scale factor set as 0.9, and the latter employs DP, secret sharing and learn-with-errors scheme [16]. To monitor an environment with relatively strict privacy constrains, the privacy budget of DP is set to $(\varepsilon, \delta) = (1, 10^{-5})$ Table 1 records the comparison of classification accuracy and running time between Paraaegis and the benchmark methods under various parameter settings. For a more intuitive presentation of the trade-off between performance and efficiency, we define the ratio of accuracy to training time as the *Efficiency Ratio*, which reflects the quality of this trade-off. Here are the findings observed from Table 1. (1) As the HE proportion r increases, the model accuracy improves accordingly, but the time overhead also increases. (2) DP-FedAvg and CKKS-FedAvg can be considered as extreme cases of ParaAegis. In their trade-offs between utility and efficiency, each method sacrifices one aspect to gain an advantage in the other. This also demonstrates that ParaAegis's parallel combination of DP and HE provides more flexibility in adjusting

Category	Algorithm	Accuracy (%)			Time (s)			Efficiency Ratio		
		IID	Dir(1)	Dir(0.5)	IID	Dir(1)	Dir(0.5)	IID	Dir(1)	Dir(0.5)
No Privacy	FedAvg [1]	80.93	79.38	77.98	3571	3665	3699	2.27	2.17	2.11
Baselines	DP-FedAvg [12]	20.28	17.12	12.87	3007	3084	3112	0.67	0.56	0.41
	CKKS-FedAvg [13]	81.14	79.28	77.63	18527	14971	14568	0.44	0.53	0.53
	Serial [14]	10.54	11.29	12.35	19956	16718	16370	0.05	0.07	0.08
	Varying DP [15]	24.21	21.76	19.85	2494	2534	2789	0.97	0.86	0.71
	Stevens et al. [16]	18.78	17.07	13.75	3635	3708	3543	0.52	0.46	0.39
ParaAegis-Static	1%	67.85	63.48	57.16	6149	6006	5577	1.1	1.06	1.02
	5%	73.03	70.07	66.19	6888	6962	6099	1.06	1.01	1.09
	10%	75.63	73.63	71.1	7776	7706	6379	0.97	0.96	1.11
	20%	77.53	75.37	73.39	9530	9565	7153	0.81	0.79	1.03
ParaAegis-Dynamic	$5\% \times 0.99$	72.09	68.91	64.4	5561	5536	5299	**1.3**	**1.24**	**1.22**
	$10\% \times 0.99$	74.5	71.75	68.5	6238	6210	5679	1.19	<u>1.16</u>	<u>1.21</u>
	$10\% \times 0.95$	68.82	65.13	58.97	5619	5628	5400	<u>1.22</u>	<u>1.16</u>	1.09
	$10\% \times 0.9$	70.87	67.01	60.33	8762	8732	7073	0.81	0.77	0.85

Table 1. Comparison on Performance across Different Algorithms With ResNet-18. The Efficiency Ratio is calculated as (Accuracy / Time) $\times 100$. For ParaAegis-Static, the parameter is the fixed ratio r. For ParaAegis-Dynamic, it is $r_0 \times \lambda$.

Strategy	Max (Proposed)	Min	Rand
Accuracy (%)	63.64	12.40	15.78

Table 2. Comparison on Accuracy across Partition Strategies

θ	0.01	0.1	1	10
Acc.	68.99	69.11	71.88	65.16
N	5	10	25	50
Acc.	72.49	65.16	56.40	48.14

Table 3. Comparison on Accuracy across Clipping Thresholds or # of Clients

the efficiency-utility trade-off for model training. (3) The serial combination of the two protection methods results in a simultaneous reduction in both accuracy and efficiency, making it unsuitable for scenarios that require a balance between efficiency and utility. (4) Two extra methods (Varying DP and Steven et al.) perform relatively worse due to their complete reliance on differential privacy, whereas ParaAegis can avoid this drawback. (5) In the dynamic variant, selecting appropriate parameters r and λ can achieve better accuracy in less time compared to the static version, resulting in an improved privacy-efficiency-utility trade-off.

Ablation of Partition Strategies. In this experiment, we fix the partition ratios and compare them across different partition strategies to demonstrate the advantages of the partition strategy our proposed. The partitioning strategies involved in the experiment are as follows. *Max*, as the strategy adopted in ParaAegis, selects the parameter set with the highest norm from local updates of each client as the HE part; *Min*, in contrast to *Max*, selects the parameter set with the lowest norm from the local updates of each client as the HE part; *Rand*, the client randomly selects a certain number of parameters as the DP part. To facilitate observing the differences, we fixed the partition ratio at $r = 0.1$ and number of global rounds at $T = 20$. Observed from Table 2 we know that the *Min* and *Rand* strategies exhibit similarly poor and unacceptable convergence behaviors, while only *Max* achieves well-behaved accuracy. This phenomenon indicates that the *Max* strategy effectively selects the most important parameters for HE protection, shielding them from the perturbations introduced by noise.

Hyperparameter Sensitivities. We evaluate ParaAegis's sensitivity to two key hyperparameters: the DP clipping threshold θ and the number of clients N. First, we determined the optimal clipping threshold by testing θ. The results in Table 3 indicate that $\theta = 1$ yields the best accuracy. This optimal value is significantly smaller than in typical DP-FL settings [19], which demonstrates that our voting mechanism effectively reduces the norm of the update component subjected to DP noise. Second, we examined the impact of client scale by varying N. The observed decrease in accuracy with a larger N (Table 3) indicates that reaching a consensus becomes more difficult with increased client diversity. Consequently, ParaAegis is best suited for cross-silo FL environments, a finding consistent with other HE-based FL frameworks [20].

6. CONCLUSION

In this paper, we propose a novel parallel hybrid protective method for FL. Our method partition the model parameters into two part, both of which are protected by DP and HE respectively, and is capable to effectively trade off privacy, utility and efficiency flexibly via adjusting partition ratio r and decay rate λ (in dynamic variant).

7. 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] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," *Advances in neural information processing systems*, vol. 33, pp. 16937–16947, 2020.

[3] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[4] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.

[5] J. Ma, S. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.

[6] A. G. Sébert, M. Checri, O. Stan, R. Sirdey, and C. Gouy-Pailler, "Combining homomorphic encryption and differential privacy in federated learning," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, 2023, pp. 1–7.

[7] C. Hu and B. Li, "Maskcrypt: Federated learning with selective homomorphic encryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 1, pp. 221–233, 2024.

[8] Yuecheng Li, Lele Fu, Tong Wang, Jian Lou, Bin Chen, Lei Yang, Jian Shen, Zibin Zheng, and Chuan Chen, "Clients collaborate: Flexible differentially private federated learning with guaranteed improvement of utility-privacy trade-off," in *Forty-second International Conference on Machine Learning*, 2025.

[9] R. L. Rivest, "Cryptography," in *Algorithms and complexity*, pp. 717–755. Elsevier, 1990.

[10] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International conference on the theory and application of cryptology and information security*. Springer, 2017, pp. 409–437.

[11] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *International Conference on Machine Learning, ICML 2022*, 2022.

[12] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[13] F. Qiu, H. Yang, L. Zhou, C. Ma, and L. Fang, "Privacy preserving federated learning using ckks homomorphic encryption," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2022, pp. 427–440.

[14] X. Zhang, D. Huang, and Y. Tang, "Secure federated learning scheme based on differential privacy and homomorphic encryption," in *International Conference on Intelligent Computing*. Springer, 2024, pp. 435–446.

[15] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, "Amplitude-varying perturbation for balancing privacy and utility in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1884–1897, 2023.

[16] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1379–1395.

[17] J. Howard, "Imagenette: A smaller subset of 10 easily classified classes from imagenet," `https://github.com/fastai/imagenette`, 2019, Accessed: 2024-09-04.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] N. Ponomareva, S. Vassilvitskii, Z. Xu, B. McMahan, A. Kurakin, and C. Zhang, "How to dp-fy ml: A practical tutorial to machine learning with differential privacy," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 5823–5824.

[20] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, D. Sánchez, A. Flanagan, and K. E. Tan, "Achieving security and privacy in federated learning systems: Survey, research challenges and future directions," *Engineering Applications of Artificial Intelligence*, vol. 106, pp. 104468, 2021.

[21] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[22] C. Dwork, A. Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[23] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2021.

A. DETAILED THEOREM ANALYSIS

In this section, we provide detailed mathmatical expression of privacy-utility-efficiency trade-off.

A.1. Privacy Analysis

On account for the Gaussian noise mechanism in proposed method, the privacy budget can be calculated by moment accountant theory [21], which is mainly Lem. 1.

Lemma 1. *There exists constant c_1, c_2 such that for any $\varepsilon \leq c_1 q^2 T$, where $q = \frac{n}{N}$, FedAvg is (ε, δ)-DP for any $\delta > 0$ if we choose*

$$\sigma_z \geq c_2 \frac{n^2 T \ln(1/\delta)}{N^2 \varepsilon^2}.$$

Remark 1. *Moreover, noise amplitude can be expressed by privacy budget [12, 15], i.e.,*

$$\sigma_z^2 = (\frac{\Delta f}{\varepsilon})^2 \cdot 2qT \ln(1/\delta),$$

where Δf is the sensitivity of the object function. In the context of this paper, sensitivity refers to the maximum difference in the DP part norm between two consecutive updates uploaded by a client. According to related researchs, sensitivity can be defined as

$$\Delta f = \min_i \frac{2\theta}{|\mathcal{D}_i|}.$$

The aforementioned privacy budget refers solely to the DP part of the local update, and concerning HE part we can utilized the properties of cryptographic algorithms. Due to computational security of current public key cryptosystems [9], the probability that an adversary without the key can obtain any information, including membership information related to the definition of DP, from the encrypted update is negligible, as long as the key length is sufficiently long. Thus, the privacy of ParaAegis relies entirely on the DP part, which is similar to the barrel effect.

We conducts the proof of Theorem 1.

Proof. Due to the cryptographic properties of homomorphic encryption, the amount of information that can be extracted from the HE part is computationally negligible, and thus the privacy budget consumed by HE protection is also nearly zero. In parallel, according to Lemma 1, we establish the relationship between the noise magnitude in the DP part and the corresponding privacy budget. Furthermore, by the parallel composition theorem of DP [22], the overall privacy budget for gradient exposure is determined solely by the privacy budget allocated to the DP part. □

A.2. Detailed Assumptions

The following assumptions is utilized in the convergence of the FL problem under proposed hybrid protection, which are detailed version of Assumption 1.

Assumption 2 (Lipschitz Smooth)**.** *There exists L such that for any $\boldsymbol{x}, \boldsymbol{y}, i$ it holds that $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x}-\boldsymbol{y}\|$.*

Assumption 3 (Bounded Local Variance)**.** *There exists σ_l such that for any t, k, i it holds that $\left\|\boldsymbol{g}_i^{t,k} - \nabla f_i(w_i^{t,k})\right\|^2 \leq \sigma_l^2$.*

Assumption 4 (Bounded Global Variance)**.** *There exists σ_g such that for any i it holds that $\nabla f_i(\boldsymbol{w}) - \nabla f(\boldsymbol{w}) \leq \sigma_g^2$.*

Assumption 5 (Bounded Gradient)**.** *There exists G such that for any t, k, i it holds that $\left\|\boldsymbol{g}_i^{t,k}\right\|^2 \leq G^2$.*

A.3. Proof for Theorem 1

In this subsection, we outline the necessary preparation work required before delving into the detailed proof of convergence.

Firstly, we clarify the definitions of the DP part and the HE part, i.e., for any vector $\boldsymbol{x}$ and any $l \in [\dim \boldsymbol{x}]$, its HE part $\boldsymbol{x}_{\text{HE}}$ and DP part $\boldsymbol{x}_{\text{DP}}$ are defined as

$$\begin{aligned}\boldsymbol{x}_{\text{HE}}[l] &= \boldsymbol{x}[l] \cdot 1\{l \in \boldsymbol{v}\}\\ \boldsymbol{x}_{\text{DP}}[l] &= \boldsymbol{x}[l] \cdot 1\{l \notin \boldsymbol{v}\}\end{aligned}$$

We list the clipping coefficient, its mean, and the difference, found by Zhang et al. [11].

$$\begin{aligned}\alpha_i^t &:= \min\left(1, \frac{\theta}{\sum_{k=0}^{K-1}\left\|\breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k}\right\|}\right),\\ \bar{\alpha}^t &:= \frac{1}{N}\sum_{i=1}^N \alpha_i^t,\ \tilde{\alpha}^t := \frac{1}{N}\sum_{i=1}^N \left|\alpha_i^t - \bar{\alpha}^t\right|.\end{aligned}$$

The local update can be easily represented in the following form based on the above definition.

$$\boldsymbol{u}_i^t := \boldsymbol{w}^{t+1} - \boldsymbol{w}^t = -\eta\sum_{k=0}^{K-1}\left(\alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} + \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k}\right)$$

By Lipschitz smoothness, we have

$$\begin{aligned}&\mathbb{E}\left[f(\boldsymbol{w}^{t+1})\right] - \mathbb{E}\left[f(\boldsymbol{w}^t)\right]\\ \leq&\mathbb{E}\left[\left\langle\nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1}-\boldsymbol{w}^t\right\rangle\right] + \frac{L}{2}\mathbb{E}\left[\left\|\boldsymbol{w}^{t+1}-\boldsymbol{w}^t\right\|^2\right]\\ =&\mathbb{E}\left[\left\langle\nabla f(\boldsymbol{w}^t), \frac{1}{n}\sum_{i\in\mathcal{C}}\boldsymbol{u}_i^t\right\rangle\right] + \frac{L}{2}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i\in\mathcal{C}}\left(\boldsymbol{u}_i^t + \boldsymbol{z}_i\right)\right\|^2\right]\\ =&\underbrace{\left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{n}\sum_{i\in\mathcal{C}}\boldsymbol{u}_i^t\right]\right\rangle}_{A_1} + \underbrace{\frac{L}{2}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i\in\mathcal{C}}\boldsymbol{u}_i^t\right\|^2\right]}_{A_2} + \underbrace{\frac{L\sigma_z^2\theta^2 rd}{2n^2}}_{\text{caused by noise}}\end{aligned} \tag{*}$$

where $\boldsymbol{z}_i = [1\{l \in \boldsymbol{v}_i\}\cdot\mathcal{N}(0,\sigma_z)]_l$ and last term is caused by noise.

For A_1, we transform the expectation over sampled clients into an expectation over all clients.

$$\begin{aligned}A_1 &= \left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{n}\sum_{i\in\mathcal{C}}\boldsymbol{u}_i^t\right]\right\rangle\\ &= \left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\boldsymbol{u}_i^t\right]\right\rangle\\ &= \left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N(\boldsymbol{u}_i^t - \bar{\boldsymbol{u}}_i^t)\right]\right\rangle\\ &\quad + \left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\bar{\boldsymbol{u}}_i^t\right]\right\rangle\end{aligned} \tag{**}$$

Then, we focus on the first term of (**) and get

$$\begin{aligned}&\left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N(\boldsymbol{u}_i^t - \bar{\boldsymbol{u}}_i^t)\right]\right\rangle\\ \overset{i}{\leq}&\left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\sum_{k=0}^{K-1}\eta|\alpha_i^t - \bar{\alpha}^t|\boldsymbol{g}_{i,\text{DP}}^{t,k}\right]\right\rangle\\ =&\frac{1}{N}\sum_{i=1}^N\sum_{k=0}^{K-1}\eta\mathbb{E}\left[|\alpha_i^t - \bar{\alpha}^t|\left\langle f(\boldsymbol{w}^t), \boldsymbol{g}_{i,\text{DP}}^{t,k}\right\rangle\right]\\ \overset{ii}{\leq}&\eta\mathbb{E}\left[\tilde{\alpha}^t\right]KG^2\bar{\rho}_t^2\end{aligned}$$

where (i) is derived from the definition of $\boldsymbol{u}_i^t$ and $\bar{\boldsymbol{u}}_i^t$, and (ii) is conducted by the Bounded Gradient Assumption.

Bounding the second term of (**), we have

$$\begin{aligned}&\left\langle\nabla f(\boldsymbol{w}^t), \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\bar{\boldsymbol{u}}_i^t\right]\right\rangle\\ =&\mathbb{E}\left[\left\langle\nabla f(\boldsymbol{w}^t), \frac{1}{N}\sum_{i=1}^N\breve{\boldsymbol{u}}_i^t\right\rangle\right]\\ =&\frac{-\eta\beta_t K}{2}\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 - \frac{\eta\beta_t}{2K}\mathbb{E}\left[\left\|\frac{1}{\eta\beta_t N}\sum_{i=1}^N\breve{\boldsymbol{u}}_i^t\right\|^2\right]\\ &+\underbrace{\frac{\eta\beta_t}{2}\mathbb{E}\left[\left\|\sqrt{K}\nabla f(\boldsymbol{w}^t) - \frac{1}{\eta\beta_t N\sqrt{K}}\sum_{i=1}^N\breve{\boldsymbol{u}}_i^t\right\|^2\right]}_{A_3}\end{aligned}$$

where the second equation comes from that $\langle\boldsymbol{x},\boldsymbol{y}\rangle = -\frac{1}{2}\|\boldsymbol{x}\|^2 - \frac{1}{2}\|\boldsymbol{y}\|^2 + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2$ holds for any vector $\boldsymbol{x}, \boldsymbol{y}$. Noting $\breve{\boldsymbol{g}}_i^t := \nabla f_i(\boldsymbol{w}_i^t)$ and $\breve{\boldsymbol{g}}_i^{t,k} := \nabla f_i(\boldsymbol{w}_i^{t,k})$, we decompose A_3 subsequently:

$$
\begin{aligned}
A_3 =& K\mathbb{E}\left[\left\|\nabla f(\boldsymbol{w}^t) - \frac{1}{KN}\sum_{i=1}^{N}\sum_{k=0}^{K-1}(\alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} + \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k})\right\|^2\right] \\
=& K\mathbb{E}\left[\left\|\frac{1}{KN}\sum_{i=1}^{N}\sum_{k=0}^{K-1}(\breve{\boldsymbol{g}}_i^t - \alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} - \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k})\right\|^2\right] \\
\leq& \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\breve{\boldsymbol{g}}_i^t - \alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} - \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k}\right\|^2\right] \\
\leq& \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\breve{\boldsymbol{g}}_i^t - \breve{\boldsymbol{g}}_i^{t,k}\right\|^2\right] \\
&+ \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left[\underbrace{\left\|\breve{\boldsymbol{g}}_i^{t,k} - \alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} - \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k}\right\|^2}_{A_4}\right] \\
\leq& \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\left(L^2\mathbb{E}\left[\left\|\boldsymbol{w}^t - \boldsymbol{w}_i^{t,k}\right\|^2\right] + G^2\right) \\
\leq& L^2 5K^2\eta^2(\sigma_l^2 + 6K\sigma_g^2) + L^2 30Q^3\eta^2\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 + KG
\end{aligned}
$$

where the first inequality is derived from Jensen's inequality, the third is derived from L-smoothness of f, and the last one is conducted from Lemma 3 of [23], i.e., it holds for any k that $\frac{1}{N}\mathbb{E}\left[\left\|\boldsymbol{w}^t - \boldsymbol{w}_i^{t,k}\right\|^2\right] \leq 5K^2\eta^2(\sigma_l^2 + 6K\sigma_g^2) + 30Q^3\eta^2\left\|\nabla f(\boldsymbol{w}^t)\right\|^2$.

Additionally, the detail of finding a upper bound of A_4 can be described as

$$
\begin{aligned}
A_4 =& \left\|\breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} - \alpha_i^t \breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k}\right\|^2 = (1-\alpha_i^t)^2\left\|\breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k}\right\|^2 \\
\leq& (1-\alpha_i^t)^2(1-\rho_t^2)G^2 \leq G^2
\end{aligned}
$$

where the first equality comes from the definition that $\breve{\boldsymbol{g}}_{i,\text{DP}}^{t,k} + \breve{\boldsymbol{g}}_{i,\text{HE}}^{t,k} = \breve{\boldsymbol{g}}_i^{t,k}$ and the second inequality is derived from the fact that $\alpha_i^t \leq 1, \rho_t \leq 1$.

Now we turn our attention to A_2 in *.

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i\in\mathcal{C}}\boldsymbol{u}_i^t\right\|^2\right] \\
=& \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i\in\mathcal{C}}\sum_{k=0}^{K-1}\eta(\alpha_i^t \boldsymbol{g}_{i,\text{DP}}^{t,k} + \boldsymbol{g}_{i,\text{HE}}^{t,k})\right\|^2\right] \\
=& \mathbb{E}\left[\frac{1}{n}\sum_{i\in\mathcal{C}}\sum_{k=0}^{K-1}\eta^2\beta^2\left\|\boldsymbol{g}_i^{t,k}\right\|^2\right] \\
\leq& \frac{\eta^2 G^2}{n^2}\mathbb{E}\left[\sum_{i\in\mathcal{C}}\sum_{k=0}^{K-1}(\beta_i^{t,k})^2\right] = \frac{\eta^2 G^2\mathbb{E}\left[\bar{\beta}_t^2\right]}{n}
\end{aligned}
$$

Sum up the equations, we get

$$
\begin{aligned}
\mathbb{E}\left[f(\boldsymbol{w}^{t+1})\right] \leq& f(\boldsymbol{w}^t) + \eta\tilde{\alpha}^t(1-\tilde{\rho}_t^2)KG^2 \\
&-\frac{\eta\bar{\beta}_t K}{2}\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 - \frac{\eta\bar{\beta}_t}{2K}\mathbb{E}\left[\left\|\frac{1}{\eta N\bar{\beta}_t}\sum_{i=1}^{N}\breve{\boldsymbol{u}}_i^t\right\|^2\right] \\
&+\frac{\eta\bar{\beta}_t}{2}\left(5L^2K^2\eta^2(\sigma_l^2 + 6K\sigma_g^2) + 30L^2K^3\eta^2\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 + K\right. \\
&+\frac{\eta^2 LG^2K}{2n} + \frac{L\sigma_z^2\theta^2 d(1-r)}{2n^2}
\end{aligned}
$$

Simplify it by setting $\eta \leq \frac{1}{\sqrt{60}KL}$:

$$
\begin{aligned}
f(\boldsymbol{w}^{t+1}) \leq& f(\boldsymbol{w}^t) - \frac{\eta\bar{\beta}_t K}{4}\left\|\nabla f(\boldsymbol{w}^t)\right\|^2 + \tilde{\alpha}^t(1-\tilde{\rho}_t^2)KG^2 \\
&+\frac{5}{2}\eta^3\bar{\beta}_t L^2K^2(\sigma_l^2 + 6K\sigma_g^2) + \frac{1}{2}\eta_l^2\bar{\beta}_t KG^2 \\
&+\frac{\eta^2 LG^2K}{2n} + \frac{L\sigma_z^2\theta^2 d(1-r)}{2n^2}
\end{aligned}
$$

After taking the sum from $t=0$ to $T-1$, dividing both side by $\frac{\eta KT}{4}$, substuting σ_z with ε and δ by Lemma 1, and rearranging, we obtain that

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\bar{\beta}_t\left\|\nabla f(\boldsymbol{w}^t)\right\|^2\right] \leq \frac{4}{\eta KT}\mathbb{E}\left[f(\boldsymbol{w}^0) - f(\boldsymbol{w}^T)\right] \\
&+4G^2\mathbb{E}_t\left[\tilde{\alpha}^t(1-\bar{r}^t)\right] + 10\eta^2L^2K(\sigma_l^2 + 6k\sigma_g^2)\mathbb{E}_t\left[\bar{\beta}_t\right] \\
&+2G^2\mathbb{E}_t\left[\bar{\beta}_t\right] + \frac{\eta^2 LG^2K}{2n} + \frac{2TL\theta^2 d(1-r)\ln(1/\delta)}{N^2D_{\min}^2\varepsilon^2}.
\end{aligned}
$$

By omitting the terms irrelavent to r and (ε, δ), we get Theorem 1.