Priv-PFL: A Privacy-Preserving and Efficient Personalized Federated Learning Approach

Alireza Aghabagherloo*†, Roozbeh Sarenche*†, Maryam Zarezadeh‡§, Bart Preneel*†, and Stefan Köpsell‡§
* *COSIC, Department of Electrical Engineering, KU Leuven, 3001 Leuven, Belgium*
† {*Alireza.aghabagherloo, roozbeh.sarenche, bart.preneel*}*@esat.kuleuven.be*
‡ *Barkhausen Institut, Dresden, Germany*
§ {*maryam.zarezadeh, stefan.koepsell*}*@barkhauseninstitut.org*

***Abstract*—Federated Learning (FL) allows clients to engage in learning without revealing their raw data. However, traditional FL focuses on developing a single global model for all clients, limiting their ability to have personalized models tailored to their specific needs. Personalized FL (PFL) enables clients to obtain their customized models, either with or without a central party. Current PFL research includes mechanisms to detect poisoning attacks, in which a couple of malicious nodes try to manipulate training convergence by submitting misleading data. However, these detection approaches often overlook privacy concerns, as they require clients to share their models with all other clients.**

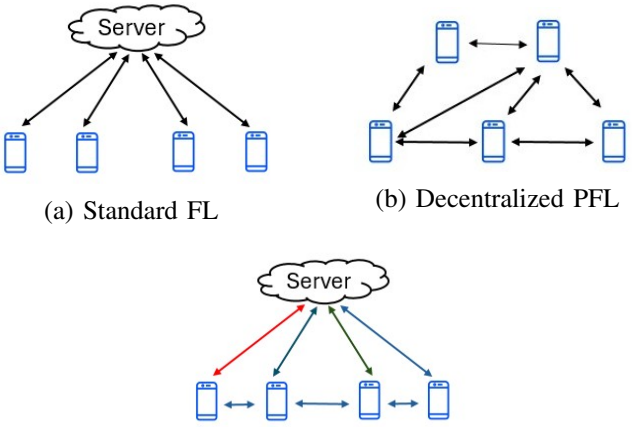
This paper extends BALANCE, a personalized poisoning detection mechanism based on client models and their expectations. Our method enhances both security and privacy by ensuring clients are not required to share their model data with other clients. By leveraging server-assisted PFL and Fully Homomorphic Encryption (FHE), we enable a central party to identify unpoisoned clients from the perspective of individual clients and train personalized models securely. Additionally, we introduce an efficient personalized client selection algorithm that prevents redundant checks and ensures the inheritance of unpoisoned clients.

***Index Terms*—Personalized Federated Learning, Deep Learning**

1. Introduction

Federated Learning (FL) was developed to address learning systems' data privacy and governance challenges. Instead of sharing raw data, each client shares its local model with a central server. This ensures that the local datasets remain inaccessible to other parties. This enhances the privacy of the system while reducing the need for extensive data storage and centralized computational resources, resulting in a more scalable learning system [1].

Despite the advantages of FL, training a single, globally uniform model in traditional FL can create fairness issues, where clients with less representative data are disadvantaged by the global model. Due to the heterogeneous data distributions among clients, the global model may significantly underperform compared to models trained on local datasets, particularly for clients whose data differs from the global distribution. As a result, clients may be less motivated to participate in FL, as they do not receive personalized benefits [3].



(a) Standard FL

(b) Decentralized PFL

(c) Server-assisted PFL

Figure 1: A sketch of different FL architectures [2].

Moreover, conventional FL approaches struggle with unfair detection methods for certain attacks, including poisoning attacks [4]. In poisoning attacks, malicious clients submit corrupted model updates to manipulate the global model's training process. To overcome poisoning attacks, traditional FL approaches aggregate a global model using client updates identified as not poisoned by the global detection mechanism. While this approach may be effective in some scenarios, there are cases where each client needs to detect its own set of unpoisoned models based on their own dataset and expected robustness level. However, conventional FL lacks the flexibility to accommodate such individualized detection mechanisms.

As shown in Fig. 1, PFL [2], [5], [6], [7] addresses the challenges of standard FL by enabling local clients to customize the global model to fit their data better with or without the assistance of a server. Instead of creating a single global model, PFL allows each client to adapt the

global model through local fine-tuning or additional training, improving its relevance, performance, and robustness for the specific needs of each user. As part of ongoing research, Fang et al. [2] introduced a personalized decentralized poisoning detection and training algorithm named BALANCE. In BALANCE, each client uses its local model as a reference to assess whether the received model is poisoned or benign.

While BALANCE, as a promising PFL approach, has addressed the challenge of standard FL, it suffers from heavy reliance on clients' computing resources for client selection and the calculation of personalized aggregated models. Another notable concern is the assumption that all clients are honest and do not intend to compromise the privacy of other users. In BALANCE, models from all clients are shared among each other, which creates privacy risks, as certain attacks, such as model inversion attacks [8] or membership inference attacks [9], can potentially extract sensitive information from these models.

1.1. Our contribution

To improve privacy and mitigate the reliance on clients' computing resources in PFL, this work proposes an enhanced server-assisted private PFL (Priv-PFL), with our contributions focused on:

- **Proposing a secure and efficient FHE-based personalized client selection algorithm:**
  We implement an FHE-based benign client selection algorithm to enhance security and privacy in detecting poisoned clients compared to conventional PFL. To improve efficiency, we extend the traditional client selection process by introducing benign client mutual acceptance and inheritance conditions, reducing redundant checks, and enhancing server-side performance. The client inheritance condition allows clients with higher thresholds to inherit benign-detected clients from those with lower thresholds. We experimentally show the efficiency of this approach.
- **Highlighting a secure, personalized model aggregation mechanism for each client:**
  We introduce a secure aggregation method using a semi-honest server, with ongoing work on extending it to an arbitrary server. This aggregates the benign detected models in the client selection algorithm.

The paper is structured as follows: Section 2 reviews BALANCE [2], a decentralized PFL framework. Section 3 explores applying FHE in a server-assisted setting and introduces an optimized secure client selection algorithm. Section 4 explains an ongoing work on secure personalized model aggregation, and Section 5 concludes the paper and discusses the remaining challenges.

2. BALANCE Personalized FL

This section provides an overview of BALANCE, a decentralized framework for PFL, as shown in Fig. 1.b. In BALANCE [2], each client detects benign clients from its perspective and trains a personalized model. This process occurs in two main steps:

Client Selection: In training round t, client $i \in V$ receives the client model $\mathbf{w}_j^{t+\frac{1}{2}}$ from other client $j \in N_i$. To determine whether the received model is benign, client i checks if the model of client j is sufficiently similar to $\mathbf{w}_i^{t+\frac{1}{2}}$. Specifically, client i considers the model of client j benign if the following condition holds:

$$\|\mathbf{w}_i^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq \gamma_i \exp(-\kappa\lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\|. \quad (1)$$

where $\gamma_i > 0$ sets an acceptance threshold for client i, $\kappa > 0$ determines the decay rate and the function $\lambda(t)$ is a monotonically increasing function of t. For simplicity, we introduce the function *decayed acceptance threshold*, denoted by $\sigma_i(t)$, which is defined as $\sigma_i(t) = \gamma_i \exp(-\kappa\lambda(t))$. The benign detected clients by client i are denoted as S_t^i.

Model Aggregation: After selecting the benign models, client i updates its model by combining its intermediate model with the aggregated models:

$$\mathbf{w}_i^{t+1} = \alpha\mathbf{w}_i^{t+\frac{1}{2}} + (1-\alpha)\cdot\frac{1}{|S_t^i|}\sum_{j\in S_t^i}\mathbf{w}_j^{t+\frac{1}{2}}. \quad (2)$$

where α is a weighting factor.

In BALANCE [2], it is proven that, under common assumptions in the literature, poisoning attacks do not negatively affect the convergence rate of their method.

3. Secure and Efficient Implementation of Client Selection in BALANCE

In this section, we proposed a secure client selection approach that extends the BALANCE client selection mechanism using FHE. Then, we introduce an optimized client selection algorithm that eliminates redundant checks.

3.1. FHE-Based Implementation of BALANCE

BALANCE heavily relies on clients' computing resources and assumes that all clients are honest and do not attempt to extract information from shared models. To address this limitation, we propose an FHE-based server-assisted approach to securely identify unpoisoned clients from each client's perspective without revealing the models. The client selection process follows these steps:

- **Data Encryption by Clients**: Each client i and client j encrypt their vector components.
- **Homomorphic Computation**: The server S processes the encrypted data using FHE without decrypting the data, primarily performing inequality in Ineq. 1 without revealing any information to the server or clients.

By repeating this process across all clients, benign clients can be identified without disclosing any client's model. Appendix A contains the description of FHE, and FHE-based client selection is detailed in Appendix B.

3.2. Optimized Client Selection Algorithm

Although integrating FHE into a server-assisted implementation of BALANCE preserves privacy, FHE introduces a significant computational burden on the server side. To mitigate this issue and improve efficiency, we propose an optimized client selection algorithm derived from the following theorems proven in Appendix C:

Theorem I: If client i detects client j as benign w.r.t. the decayed acceptance threshold $\sigma_i(t)$ and

$$\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \sigma_j(t).\|\mathbf{w}_j^{t+\frac{1}{2}}\|,$$

then client j will also detect client i as benign w.r.t. the decayed acceptance threshold $\sigma_j(t)$.

Theorem II: Let S_i^t denote the set of all benign clients for client i, and let $\texttt{subS}_i^t$ be the set of clients j such that $j \in S_i^t$ and

$$\sigma_i(t)\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \frac{1}{2}\sigma_j(t)\|\mathbf{w}_j^{t+\frac{1}{2}}\|.$$

In this case, all clients in $\texttt{subS}_i^t$ will inherit all clients in S_i^t as benign. This client inheritance condition allows clients with higher thresholds to inherit benign clients from those with lower thresholds.

3.2.1. Optimized Client Selection Algorithm and Experimental Results. Given a set of K clients, Algorithm 1 efficiently determines which clients are benign to each other. In this Algorithm clients are sorted by the norms $\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\|$, with the smallest being processed first. Running the algorithm, which utilizes Theorems I and II, prevents redundant checks. This algorithm reduces the complexity of BALANCE client selection from $O(K(K-1))$ — where K is the number of clients, and each client must operate the Ineq. 1 for all other clients $(K-1)$ — to a lower value between $O(K-1)$ and $O(K(K-1)/2)$. The degree of complexity reduction depends on the decayed acceptance threshold of the clients and their norms.

In a simplified experimental setup, we considered 30 clients with decayed acceptance thresholds uniformly sampled from the range [0.1, 0.5]. The weight norms for each client were randomly assigned to simulate an FL setup, with the models trained on the CIFAR-10 dataset [10], partitioned in a non-IID fashion, as outlined in [11]. The results showed a reduction in the number of times Ineq. 1 needed to be operated, reducing the required comparisons from 870 to 298 on average. This reduction in comparisons is crucial, as evaluating Ineq. 1 in Algorithm 1 involves FHE, as detailed in Section 3.1.

4. Ongoing Work on Secure Implementation of Model Aggregation in BALANCE

We assume clients have a Tamper-Resistant Device (TRD) [12] that securely stores a private key. Each client has a public and a private key (inaccessible even for clients and stored in the TRD), while the server has its own public and private key. The process begins with each client encrypting its model using its public key, and then encrypting it with the server's public key. Upon receiving the encrypted models, the server decrypts them using its private key. The server applies FHE (Section 3) to identify benign clients per client i. It computes and returns the aggregated sum $\sum_{j\in S_i^t} \mathbf{w}_j^{t+\frac{1}{2}}$ using FHE without revealing model details. Clients then decrypt aggregated weights (Eq. 2) with the TRD-stored key. This ensures privacy but assumes the server is semi-honest and would not extract TRD keys—an assumption that, while common, is not entirely realistic [12].

Our ongoing research aims to develop a secure aggregation protocol with an untrusted server, where clients encrypt model updates with their public keys for homomorphic aggregation using RNS-CKKS encryption [13] and collaborative decryption. The main remaining challenge is preventing client-server collusion, which we aim to mitigate the risks leveraging Threshold FHE (TFHE) [14].

Algorithm 1 Efficient Client Selection

Input: $\{\mathbf{w}_1^{t+\frac{1}{2}}, \mathbf{w}_2^{t+\frac{1}{2}}, \ldots, \mathbf{w}_K^{t+\frac{1}{2}}\}$,$\{\sigma_1(t), \sigma_2(t), \ldots, \sigma_K(t)\}$
Output: Accepted client sets $\{S_i^t\}$ for each client i.

1. Sort clients by weight norms: Sort clients $1, 2, \ldots, K$ in ascending order of $\sigma_i(t) \cdot \|\mathbf{w}_i^{t+\frac{1}{2}}\|$.
2. Initialize benign client lists: Set each $S_i^t \leftarrow \emptyset$.
for *Client* $i = 1$ *to* K **do**
  3.1. Benign client detection for Client i**:**
  for *each Client* $j \notin S_i^t$ **do**
    if *the weights of* j *satisfy Inequality 1* **then**
      Add Client j to S_i^t
  3.2. Mutual acceptance based on Theorem I:
  for *each Client* $j \in S_i^t$ *and* $i < j \leq K$ **do**
    Add Client i to S_j^t
  3.3. Benign client inheritance based on Theorem II:
  for *each Client* $j \in S_i^t$ **do**
    if $\sigma_j(t) \cdot \|\mathbf{w}_j^{t+\frac{1}{2}}\| \geq 2 \cdot \sigma_i(t) \cdot \|\mathbf{w}_i^{t+\frac{1}{2}}\|$ **then**
      Add all clients in S_i^t to S_j^t
return $\{S_i^t\}$, the list of accepted clients.

5. Conclusion and Future Work

In this paper, we proposed an enhanced server-assisted Priv-PFL framework to address privacy concerns and client computational resource limitations. Our contributions include a secure and efficient FHE-based client selection algorithm and a personalized model training mechanism for each client, improving security and performance.

Since one of the goals of PFL is to facilitate collaboration among nearby clients, the main remaining challenge is integrating the proposed PFL algorithm with a hierarchical cluster-based approach. This should facilitate model convergence, efficient poisoned client detection, and collaboration between cluster heads. Additionally, in our

algorithm, clients share $\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\|$ without encryption, enabling client sorting by weight norms without additional computational cost. While this approach does not pose the privacy risks associated with sharing model weights, the initial step of our client selection algorithm—sorting clients by weight norms—should be integrated with subsequent steps in future work on hierarchical cluster-based Priv-PFL to eliminate the need for unencrypted norm weight sharing. Another interesting topic for future work is the implementation of a server-assisted Priv-PFL that not only withstands the poisoning attack but also mitigates the adversarial examples generated in FL [15], [16].

Acknowledgment

This work was supported by the Flemish Government through the Cybersecurity Research Program with grant number: VOEWICS02 and in part by the Research Council KU Leuven projects IF/C1 From Website Fingerprinting to App Fingerprinting: Inferring private user activity from encrypted network traffic. This paper was also partially supported by the AIDE project funded by the Belgian SPF BOSA under the programme "Financing of projects for the development of artificial intelligence in Belgium" with reference number 06.40.32.33.00.10.". This work also has been partly funded by the German Federal Ministry of Education and Research (project SEMECO-Q1, grant no. 03ZU1210AA). Additionally, the authors of Barkhausen Institut are also financed based on the budget passed by the Saxonian State Parliament in Germany.

References

[1] M. Asad, A. Moustafa, and T. Ito, "Federated learning versus classical machine learning: A convergence comparison," *arXiv preprint arXiv:2107.10976*, 2021.

[2] M. Fang, Z. Zhang, Hairi, P. Khanduri, J. Liu, S. Lu, Y. Liu, and N. Gong, "Byzantine-robust decentralized federated learning," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2874–2888.

[3] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International conference on machine learning*. PMLR, 2019, pp. 4615–4625.

[4] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-Robust federated learning," *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

[5] T. Li, S. Hu, A. Beirami, and V. Smith, "Fair and robust federated learning through personalization," *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, 2021.

[6] D. Hashemi, L. He, and M. Jaggi, "Cobo: Collaborative learning via bilevel optimization," *arXiv preprint arXiv:2409.05539*, 2024.

[7] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9587–9603, 2022.

[8] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[10] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[11] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[12] A. Aghabagherloo, M. Delavar, J. Mohajeri, M. Salmasizadeh, and B. Preneel, "An efficient and physically secure privacy-preserving authentication scheme for vehicular ad-hoc networks (vanets)," *Ieee Access*, vol. 10, pp. 93 831–93 844, 2022.

[13] E. Lee, J.-W. Lee, Y.-S. Kim, and J.-S. No, "Optimization of homomorphic comparison algorithm on rns-ckks scheme," *IEEE Access*, vol. 10, pp. 26 163–26 176, 2022.

[14] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. Rasmussen, and A. Sahai, "Threshold cryptosystems from threshold fully homomorphic encryption," in *Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part I 38*. Springer, 2018, pp. 565–596.

[15] A. Aghabagherloo, R. Gálvez, D. Preuveneers, and B. Preneel, "On the brittleness of robust features: An exploratory analysis of model robustness and illusionary robust features," in *2023 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2023, pp. 38–44.

[16] J. Zhu, J. Yao, T. Liu, Q. Yao, J. Xu, and B. Han, "Combating exacerbated heterogeneity for robust models in federated learning," *arXiv preprint arXiv:2303.00250*, 2023.

[17] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university, 2009.

Appendix

Appendix A: FHE Description

Homomorphic Encryption (HE) is a cryptographic system that enables certain algebraic computations to be performed directly on encrypted data. Fully Homomorphic Encryption (FHE) extends this capability by supporting all types of algebraic operations on encrypted data [17]. This section describes key operations in the FHE-based RNS-CKKS scheme [13].

Public Parameters

The scheme relies on several public parameters: q is the ciphertext modulus, which determines precision and security levels, $\mathbb{R}_2^q$ is the ring of polynomials with coefficients modulo q, a is a uniformly random polynomial sampled from $\mathbb{R}_2^q$, and b is computed as $b = -a \cdot s + e \mod q$, where s is the secret key and e is an error polynomial.

Key Generation

- **Public Key (pk).** The public key, used for encryption, is generated as:

$$\text{pk} = (b, a) \in \mathbb{R}_2^q.$$

  Here, b is computed as $b = -a \cdot s + e \mod q$, where s is the secret key and e is an error polynomial.
- **Secret Key (sk).** The secret key, used for decryption, is a polynomial $s \in \mathbb{R}$, sampled from a predefined secret key distribution.

Encryption

The encryption process transforms plaintext data (real or complex numbers) into ciphertexts using the public key pk.

- **Encoding:** Let $z \in \mathbb{C}^{N/2}$ be the plaintext data. The plaintext is first encoded into a polynomial $m \in \mathbb{R}$ using a scaling factor Δ:

$$m = \text{Ecd}(z; \Delta) = \lfloor \Delta \cdot \tau^{-1}(z) \rceil,$$

  where τ is a field isomorphism that maps polynomials to vectors of complex numbers.
- **Encryption:** The encoded polynomial m is then encrypted as:

$$\text{Enc}(z; \text{pk}) = \text{ct} = (v \cdot \text{pk} + (m + e_0, e_1)) \mod q_j.$$

  Here:
  - v is a random polynomial sampled from the encryption key distribution.
  - e_0, e_1 are error polynomials sampled from the error distribution.
  - q_j is the ciphertext modulus at level j.

Decryption

The decryption process converts ciphertexts back into plaintext data using the secret key sk.

- **Decryption:** Given ciphertext $\text{ct} = (\text{ct}_0, \text{ct}_1)$, the decrypted polynomial $\tilde{m}$ is computed as:

$$\tilde{m} = \langle \text{ct}_0, \text{sk} \rangle \mod q_0.$$

  Here, the operation $\langle \cdot, \cdot \rangle$ denotes the inner product.
- **Decoding:** The decrypted polynomial $\tilde{m}$ is then decoded back into the plaintext vector z:

$$z = \text{Dcd}(\tilde{m}; \Delta) = \Delta^{-1} \cdot \tau(\tilde{m}).$$

Homomorphic Operations

- **Addition (Add).** The homomorphic addition of two ciphertexts ct_1 and ct_2 is performed component-wise:

$$\text{Add}(ct_1, ct_2) = (c_{10} + c_{20}, c_{11} + c_{21}) \mod q_j.$$

- **Multiplication (Mult).** The homomorphic multiplication of two ciphertexts ct_1 and ct_2 involves:

  1) Computing the tensor product:

$$d_0 = c_{10} \cdot c_{20} \mod q_j,$$
$$d_1 = c_{10} \cdot c_{21} + c_{11} \cdot c_{20} \mod q_j,$$
$$d_2 = c_{11} \cdot c_{21} \mod q_j.$$

  2) Relinearizing the result using the evaluation key evk:

$$ct_{\text{mult}} = \text{Relin}(d_0, d_1, d_2; \text{evk}).$$

- **Scalar Multiplication.** Given a ciphertext $ct = (c_0, c_1)$ and a scalar $\alpha \in \mathbb{R}$, the operation is:

$$\alpha \cdot ct = (\alpha \cdot c_0, \alpha \cdot c_1) \mod q_j.$$

Comparison Operation

The comparison operation compares two encrypted numbers a and b without decrypting them. It is defined as:

$$\text{comp}(a, b) = \begin{cases} 1 & \text{if } a > b, \\ \frac{1}{2} & \text{if } a = b, \\ 0 & \text{if } a < b. \end{cases}$$

The comparison function is implemented using the sign function $\text{sgn}(x)$, approximated via a minimax composite polynomial:

$$\text{sgn}(x) \approx p_k \circ p_{k-1} \circ \cdots \circ p_1(x),$$

where p_i are polynomials of degree d_i, and $\circ$ denotes function composition.

The approximation satisfies:

$$\max_{x \in D} |p_k \circ p_{k-1} \circ \cdots \circ p_1(x) - \text{sgn}(x)| \leq 2^{-\alpha},$$

where α is the precision parameter.

Appendix B: FHE-Based Implementation of BALANCE

Specifically, assume the server wants to verify whether client i accepts client j as benign based on the condition in Ineq. 1:

$$\sum_{k=1}^{n} \left(\left(w_{i,k}^{t+1/2} \right)^2 - 2 w_{i,k}^{t+1/2} w_{j,k}^{t+1/2} + \left(w_{j,k}^{t+1/2} \right)^2 \right) \leq \gamma \cdot \exp(-\kappa \cdot \lambda(t)) \cdot \sum_{k=1}^{n} \left(w_{i,k}^{t+1/2} \right)^2$$

The verification process involves the following steps:

Data Encryption by Clients. Each client i and client j encrypts their vector components using a public key pk, generated by a trusted party, ensuring that only the trusted party can decrypt the results.

- Client i encrypts its vector w_i^t:

$$\psi(w_{i,k}^{t+1/2}) = \text{Encrypt}(p_k, w_{i,k}^{t+1/2}), \quad k = 1, 2, \ldots, n$$

Client j encrypts $w_{i,k}^{t+1/2}$ as well, and then encrypted vectors are sent to server S.

Homomorphic Computation. Server S processes the encrypted data using FHE without decrypting it.

1. Compute the encrypted squares for both vectors:

$$\psi(w_{i,k}^t)^2 = \text{Evaluate}(\text{Mult}, p_k, \psi(w_{i,k}^t), \psi(w_{i,k}^t))$$
$$\psi(w_{j,k}^t)^2 = \text{Evaluate}(\text{Mult}, p_k, \psi(w_{j,k}^t), \psi(w_{j,k}^t))$$

2. Compute the encrypted cross-term:

$$\psi(2 w_{i,k}^t w_{j,k}^t) = \text{Evaluate}(2 \cdot \text{Mult}, p_k, \psi(w_{i,k}^t), \psi(w_{j,k}^t))$$

3. Compute the sum of all terms:

$$\psi(result) = \sum_{k=1}^{n} \left(\psi(w_{i,k}^t)^2 - \psi(2w_{i,k}^t w_{j,k}^t) + \psi(w_{j,k}^t)^2\right)$$

4. Server S verifies whether the inequality holds:

$$\psi(result) \leq \gamma \cdot \exp(-\kappa \cdot \lambda(t)) \cdot \sum_{k=1}^{n} \psi(w_{i,k}^t)^2$$

To conduct this comparison check, the server utilizes the RNS-CKKS scheme [13], as described in Appendix A.

By repeating this process across all clients, benign clients can be identified without disclosing any client's model.

Appendix C: Proof of Theorem I and Theorem II

Theorem I: If client i detects client j as benign w.r.t. its acceptance threshold and

$$\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \sigma_j(t).\|\mathbf{w}_j^{t+\frac{1}{2}}\|,$$

then client j will also detect client i as benign w.r.t. the acceptance threshold γ_j.

Proof:

Assume that client i detects client j as benign. This means that

$$\|\mathbf{w}_i^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq 2\gamma \exp(-\kappa\lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\|.$$

Now, assuming $\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \sigma_j(t).\|\mathbf{w}_j^{t+\frac{1}{2}}\|$, we can apply following inequality:

$$\begin{aligned}\|\mathbf{w}_i^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| &= \|\mathbf{w}_j^{t+\frac{1}{2}} - \mathbf{w}_i^{t+\frac{1}{2}}\| \\ &\leq \gamma_i \exp(-\kappa\lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\| \\ &\leq \gamma_j \exp(-\kappa\lambda(t))\|\mathbf{w}_j^{t+\frac{1}{2}}\|.\end{aligned}$$

Thus, client j will detect client i as benign, as the difference between their weights satisfies the condition for being benign.

□

Theorem II: Let S_i^t denote the set of all benign clients for client i, and let $\texttt{subS}_i^t$ be the set of clients j such that $j \in S_i^t$ and

$$\sigma_i(t)\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \frac{1}{2}\sigma_j(t)\|\mathbf{w}_j^{t+\frac{1}{2}}\|.$$

In this case, all clients in $\texttt{subS}_i^t$ will inherit all clients in S_i^t as benign.

Proof:

Let n and j be clients such that $n, j \in S_t^j$. Assume the following conditions:

$$\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \frac{1}{2}\sigma_j(t).\|\mathbf{w}_j^{t+\frac{1}{2}}\| \quad (1)$$

From the given inequalities for n and j:

$$\|\mathbf{w}_i^{t+\frac{1}{2}} - \mathbf{w}_n^{t+\frac{1}{2}}\| \leq \gamma_i \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\| \quad (2)$$

$$\|\mathbf{w}_i^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq \gamma_i \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\| \quad (3)$$

Now, applying the triangle inequality to the difference between $\mathbf{w}_n^{t+\frac{1}{2}}$ and $\mathbf{w}_j^{t+\frac{1}{2}}$:

$$\|\mathbf{w}_n^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq \|\mathbf{w}_n^{t+\frac{1}{2}} - \mathbf{w}_i^{t+\frac{1}{2}}\| + \|\mathbf{w}_j^{t+\frac{1}{2}} - \mathbf{w}_i^{t+\frac{1}{2}}\| \quad (4)$$

From equations (2), (3), and (4), we have:

$$\begin{aligned}\|\mathbf{w}_n^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq \gamma_i \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\| \\ + \gamma_i \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\|\end{aligned}$$

Simplifying the right-hand side:

$$\|\mathbf{w}_n^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq 2\gamma_i \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_i^{t+\frac{1}{2}}\|$$

Since $\sigma_i(t).\|\mathbf{w}_i^{t+\frac{1}{2}}\| \leq \frac{1}{2}\sigma_j(t).\|\mathbf{w}_j^{t+\frac{1}{2}}\|$ (from equation 1), we have:

$$\|\mathbf{w}_n^{t+\frac{1}{2}} - \mathbf{w}_j^{t+\frac{1}{2}}\| \leq \gamma_j \cdot \exp(-\kappa \cdot \lambda(t))\|\mathbf{w}_j^{t+\frac{1}{2}}\|$$

Thus, client j will accept client n as benign.

□