International Research Journal of Multidisciplinary Scope (IRJMS), 2025; 6(3):709-728 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2025.v06i03.03986

Maximizing Privacy in Federated Learning: Analysis of Effective Cryptographic Techniques

Narendra Babu Pamula[1]*, Ajoy Kumar Khan[1], Arnidam Sarkar[2]

[1]Department of Computer Engineering, Mizoram University Tanhril, Aizawl, Mizoram, India, [2]Department of Computer Science and Electronics, Ramakrishna Mission Vidyamandira Belur Math Howrah, WB, India. *Corresponding Author's Email: naren.pamula@gmail.com

Abstract

Emerging as a distributed machine learning paradigm allowing many people to cooperatively train models without directly exchanging raw data is federated learning (FL). FL is nevertheless vulnerable to several attacks, including model inversion, gradient leaking, and adversarial inference, which might expose private information even this privacy-centric architecture. Adoption of FL depends on addressing privacy issues; this is especially true in industries like finance and healthcare where data security is critical. This work suggests a fast cryptographic method to improve FL's privacy preservation while preserving computational economy. To enable safe multi-party computation and stop illegal inference of private data, the proposed solution combines lightweight cryptographic primitives—including homomorphic encryption (HE) and differential privacy (DP)—as Differential privacy generates controlled noise to protect individual contributions; homomorphic encryption guarantees that model updates can be aggregated safely without decryption. By reasonably balancing privacy protection with model performance, our method lowers computational and communication overhead. Experimental analyses show that the suggested approach greatly improves data security without sacrificing the scalability or accuracy of the federated learning system. This work helps to advance safe FL deployments by striking a trade-off between privacy, efficiency, and usability, so making them more practical for real-world applications needing strict confidentiality, such medical diagnosis, financial transactions, and personalized recommendation systems.

Keywords: Cryptographic Techniques, Differential Privacy (DP), Federated Learning, Homomorphic Encryption (HE), Privacy-Preserving Machine Learning, Secure Multi-Party Computation (SMPC).

Introduction

Federated learning has become a transforming paradigm allowing cooperative model training over distributed devices while preserving data locality in the era of big data and artificial intelligence. Unlike conventional centralized learning techniques, federated learning addresses important privacy issues by letting many users jointly train machine learning models without disclosing their raw data. Nonetheless, federated learning is not immune to flaws including data leakage, inference assaults, and adversarial exploitation during model updates or communication notwithstanding its natural privacy-preserving design. These difficulties highlight how urgently strong cryptographic techniques are needed to guarantee the privacy, integrity, and confidentiality of private data all around the federated learning process. Reducing these hazards depends critically on the creation of a scalable cryptographic method specifically for federated learning. Since federated learning usually runs in resource-limited situations such mobile devices or IoT networks, such an algorithm must carefully balance offering strong security guarantees with preserving computational performance. Furthermore, the method has to be scalable to fit the dynamic and diverse character of federated learning systems, in which users could join or leave the network whenever they so want. This work intends to build and implement a fresh cryptographic framework including homomorphic encryption, secure multi-party computation, and differential privacy to protect data privacy in federated learning by means of advanced approaches. Using these cryptographic primitives will help to minimize computing overhead and communication costs while also ensuring that sensitive data stays encrypted during the training and inference phases. The ultimate aim is to enable safe and privacy-preserving federated learning at

(Received 01st February 2025; Accepted 07th July 2025; Published 24th July 2025)

scale, hence building confidence and cooperation among users of many applications spanning from healthcare and finance to smart cities and beyond. By means of this work, we hope to add to the increasing corpus of knowledge in privacy-preserving machine learning and offer a useful tool enabling companies and people to leverage federated learning without endangering data privacy. Cryptographic methods offer promising solutions, yet existing techniques often impose significant computational overhead. This paper introduces an efficient cryptographic algorithm tailored for federated learning to address these challenges. FL is considered a revolutionary concept in the distributed machine learning paradigm in which many clients, including smart phones, edge devices, and IoT nodes, can collaboratively train a global model without uploading their raw data to any central server. This approach, therefore, fits into modern data privacy regulations in terms of how organizations handle personal data, which, in this case, involves the GDPR and CCPA. These modern data privacy regulations put emphasis on protecting user data, and FL supports data sovereignty by making sure that sensitive information is kept in control by both the organization and individual while trying to achieve mutual learning objectives. Despite these benefits, federated learning is still hampered by various challenges mainly about data privacy, communication overhead, and computational efficiency. FL does not necessarily share raw data but is also vulnerable to indirect attacks on privacy. In order to solve the aforesaid problems, the work creates a fast cryptographic method especially intended for federated learning. Here the answer is to combine differential privacy with lightweight homomorphic encryption into a dual-layered method of privacy protection. Lightweight homomorphic encryption locks gradients at the transmission point so that even semi-honest adversaries cannot leak any data. Differential privacy generates noise into model updates, therefore offering further defence against gradient leaks and model inversion. Furthermore, computationally efficient, the method will perform well in settings with limited resources. The best aggregation methods lower the communication overhead and offer a strong assurance for strong privacy preservation free from loss in model accuracy. Apart from evaluating the efficacy of the suggested method on actual datasets including MNIST and CIFAR-10, the paper addresses the theoretical foundations of it. It is employed for several metrics of relevance including computing efficiency, model accuracy, and resistance to invasions of privacy. We address the present significant privacy concerns as well as efficiency factors and explain work under this paper as our addition to federated learning literature. Why laying the basis upon which the future framework for federated learning will finally be built can be considered very important since the integration of advanced cryptographic with pragmatic optimizations results in a proposal that is both practical, scalable, and deployable in many challenging real cases.

Research on the creation of effective cryptographic algorithms to protect data privacy in federated learning (FL) is under active progress. The following is a synopsis of pertinent research and significant domanial contributions. Federated learning, a decentralized ML paradigm, emerges to answer a growing demand of data privacy and collaborative model training. It fundamentally differs from the more traditional approaches, where all collected data are collected and stored on a central repository, by ensuring that the local data remain resident on the client device and that updates to models shared with a central repository are model-specific, for instance, gradients or parameters. This approach greatly minimizes the risk of data breaches and is also in line with global privacy regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA).

Foundation of Federated Learning and Privacy Issues

Federated learning or FL: Originally developed by Google in 2016, FL lets several parties jointly train a machine learning model without distributing actual data.

Privacy Concerns: meanwhile, come from possible inference attacks or data leaks during model changes. Several privacy issues inherent in the distributed character of the system must be addressed in the creation of an effective cryptographic method to maintain data privacy in federated learning (FL). Under federated learning, several clients—many of whom do not share their raw data—coordinate training of a shared model.

Although this method improves privacy naturally, it still carries hazards that need be reduced with cryptographic methods.

Existing Cryptographic Techniques for Protection of Privacy

Secure Multi-Party Computation (SMPC): A widely used other technique for protecting privacy during FL aggregation of updates is SMPC. SMPC helps in computing a joint function on inputs generated by multiple clients without revealing that input to another client or central server. Every client contributes values encrypted or shares them secretly while the computation itself is done collaboratively using cryptographic techniques. SMPC is a protocol that doesn't need the involvement of any trusted third party. This characteristic makes it very fit for scenarios in which no entity can be trusted for keeping data private. However, there can be some substantial communication overhead with SMPC, especially if there are too many participants within large-scale networks. The repeated sending of cryptographic messages between clients and the server creates delays and higher bandwidth consumption, becoming problematic in resource-constrained environments. Review of the literature on Secure Multi-Party Computation (SMPC)(1) in federated learning (FL) finds an increasing corpus of work aimed on improving security and privacy in networked machine learning systems shown in Table 1. Keeping such inputs confidential, SMPC is a cryptographic method allowing several parties to jointly compute a function over their inputs. Key studies, techniques, and developments in this field are compiled here. Applications in FL include secure aggregation of model updates without revealing individual contributions.

Table 1: Secure Multi-Party Computation (SMPC) in Federated Learning (FL)

Ref. No	Key Contributions	Methodology	Limitations	Accuracy Trade-off	Security Strength	Utility Trade-off
(1)	Introduced a secure aggregation protocol for FL using SMPC.	Combines secret sharing and cryptographic primitives for secure aggregation	High communication overhead; scalability issues with large datasets	No noise added; high accuracy kept	SMPC provides strong privacy	A lot of communication overhead makes things less useful and less scalable
(2)	Demonstrated the use of SMPC for privacy-preserving neural network inference.	Uses homomorphic encryption and SMPC for secure computation.	Limited to inference; not directly applicable to FL training.	Keeps the output of the model accurate	High (SMPC + encryption)	Only good for inference tasks; not good for the whole FL pipeline
(3)	An SMPC-based privacy-preserving ML framework was proposed.	Integrates unstructured secret sharing with jumbled routing.	Concentrated on centralized ML; further development is needed for FL adaptation.	Accuracy kept; no change	Strong against attacks that try to guess	Limited FL adaptability; centralized focus makes FL less useful in general
(4)	Provided a safe FL architecture that aggregates	Secures aggregation through the use of secret sharing and SMPC	Expensive computing requirements; restricted capacity for FL	No loss of accuracy	High—keeps model updates safe while they are being combined	Lots of calculations; affects speed and scalability

	gradients using SMPC.		on a grand scale			
(5)	Presented a new method that combines SMPC with differential privacy.	Implements differential privacy for noise addition and secure aggregation using SMPC.	Privacy vs. model accuracy trade-off	Less accurate because of noise injection	Very strong—two layers of privacy protection	The tension between accuracy and privacy affects how useful a model is.
(6)	Put out a FL-specific, lightweight SMPC protocol.	Makes use of additive secret sharing and pair wise masking	Restrictions to FL settings on a smaller scale	Keeps accuracy; no extra noise	Moderate—lightweight SMPC might not be as strong	Made for small-scale FL; not very useful in general
(7)	Secure aggregation in FL based on SMPC is more efficient now.	Implements SMPC-based tree-based aggregating protocol	In order to set up, reliable third parties are needed.	Accuracy kept	Strong if third parties that are trusted are reliable	The complexity of the setup and the need for trust make it hard to use in real life.
(8)	Improved privacy assurances by the integration of SMPC and differential privacy.	Injects noise using differential privacy and SMPC for secure aggregation	Dual privacy techniques increase computing complexity.	Added noise made the accuracy worse	Very strong—uses two different types of cryptography to protect data	A lot of processing power is needed, and the protocol is complicated, which makes deployment less flexible

Homomorphic Encryption (HE): Homomorphic encryption is a very powerful cryptographic technique that enables direct computation on encrypted data without decryption. In the context of FL, it means that gradients or model updates could remain encrypted in the central server and aggregated without making the raw data or sensitive information exposed. For example, a client can encrypt locally computed gradients and send them to the server where aggregation operations on the encrypted form are performed, without accessing actual gradients. In return, such strong privacy guarantee of homomorphic encryption comes with high overhead in computation. This may even make it unfit for highly resource-constrained environments, such as IoT devices or edge networks. Fully homomorphic encryption is highly computational and could take a lot of hardware resources if acceptable performance levels are to be achieved, in particular for the arbitrary computations in encrypted data. To keep data private in federated learning (FL), there has been significant research and development around homomorphism encryption (HE). The table below summarizes this literature review. The main emphasis is on creating effective cryptographic algorithms and using them in FL shown in Table 2.

Table 2: Homomorphic Encryption (HE) in Federated Learning (FL)

Ref. No	Key Contributions	Limitations	Relevance to FL	Privacy Protection	Accuracy Trade-off	Efficiency and Scalability
(1)	Updates to models can be made while protecting user privacy thanks to newly-introduced safe aggregation techniques for FL.	Not completely homomorphic; heavy computational burden	A cryptographic framework for privacy-preserving FL	Aggregation hides individual updates without needing full encryption.	No or very little loss in the accuracy of the model	Not as heavy as FHE but not as efficient as basic DP
(9)	A distributed deep learning system that protects user privacy is suggested to use additive HE.	Not appropriate for complicated models; restricted to addition operations alone	The initial implementation of HE in FL for safe aggregation	Allows the collection of encrypted model updates	Keeps accuracy high (no extra noise)	Only works for addition and costs a lot of money to run
(10)	Created Batch Crypt, a HE batching method to boost cross-silo FL efficiency.	Batching is necessary; however, it might not work well with big datasets.	Enhanced FL HE efficiency for business use cases	Based on HE, secure aggregation stayed	No effect on the accuracy of the model	Increases efficiency, but may have trouble with big datasets
(11)	Presented POSEIDON, a FL framework that integrates HE with MPC	Great expenditures on computing and communication	Proved that HE and MPC could work together toward FL goals.	Brings together two secure computation models	Accuracy kept	High costs for communication and computing
(12)	An alternative to traditional encryption methods, Hybrid	Trade-off between privacy and model accuracy.	Improved anonymity in FL while decreasing computing burden	HE gives you strong privacy and DP gives you anonymity.	A small drop in model accuracy (because of DP noise)	Better than just HE or MPC

	Alpha combines HE with differential privacy.					
(13)	Developed a small-footprint HE method that works well with massive FL datasets...	Restricted to only a few neural network types	I resolved the scalability problems with HE in FL	Keeps HE's privacy strong	Compatible models keep their accuracy	Made for big datasets; not much NN support
(14)	A secure method of aggregation in FL was implemented using fully homomorphic encryption (FHE).	Very expensive to compute; not feasible for use in real-time scenarios.	Investigated the possibility of using FHE in FL to achieve the highest level of anonymity	Lets you do any kind of math on encrypted data	No giving up on accuracy	Not possible in real time, expensive

Differential Privacy (DP): A statistical method called differential privacy manages noise introduced to data or model updates such that individual contributions cannot be discernible. In FL, differential privacy can be implemented by adding noise to the gradients or model parameters before sharing those with the central server represented in Table 3; this would ensure that even if an adversary manages to get access to the updates, then the noise will mask the details of individual data points. Differential privacy's advantage is that it lets one tune-off privacy from model utility. Greater degrees of noise guarantee more privacy but compromise the accuracy of the model. For environments limited in resources, this makes differential privacy more sensible than cryptographic methods such as HE and SMPC. To strike a suitable compromise between privacy and the general FL system performance, nevertheless, accurate noise level calibration is needed. This approach guarantees that computer results don't reveal too much about any one piece of data, therefore safeguarding individuals' privacy. DP can be used in a federated learning environment to add noise into the data or model updates should sensitive information have to be concealed during training.

Table 3: Differential Privacy (DP) in Federated Learning (FL)

Ref. No	Focus	Key Contributions	Techniques Used	Impact on FL Privacy	Accuracy	Security Level	Utility Trade-off
(1)	Privacy-preserving aggregation in FL	a safe aggregation technique to merge model updates without disclosing	Safe Multi-Party Computation (SMPC) and homomorphic encryption.	Improve security in FL therefore safeguarding individual data.	SMPC adds extra work to communication, which could slow down convergence and lower	Strong privacy protections that don't show changes to individual models.	Increases the cost of computing and communication, which makes it harder to scale.

		individual updates.			the accuracy of the model.		
(15)	Federated Learning optimization	Present Federated Averaging (FedAvg), a distributed training approach.	Federated Learning, DP achieved via noise injection	First important step towards merging DP and FL techniques.	Adding noise can lower accuracy, depending on the privacy budget (ε).	ε determines how private something is, and there are trade-offs between safety and usefulness.	Adding noise makes models less useful, especially when the data set is small or the task is sensitive.
(16)	Privacy-preserving in FL	Examined privacy-preserving systems in FL, with particular attention on homomorphic encryption and DP.	Homomorphic encryption and differential privacy	Complete review of FL privacy method	Using more than one method together may make things less accurate than using just one.	Encryption keeps computation safe, and DP stops data from leaking.	A lot of extra work for the computer and maybe less usability for the model.
(17)	Homomorphic encryption in FL	Combining fully homomorphic encryption (FHE) with DP will help to aggregate data while maintaining privacy.	Differential privacy allows a powerful privacy-preserving architecture employing	fully homomorphic encryption (FHE) in FL.	FHE causes latency, and DP makes accuracy worse, which makes the problem worse.	FHE is the best way to encrypt data, and DP adds even more security.	Not yet useful for large-scale FL because of latency and extra computing power.
(18)	Combining DP and HE	Presented a hybrid of homomorphic encryption and DP to improve FL security.	Homomorphic Encryption and Differential Privacy	Advancements in Data Privacy Protection in FL via Hybrid Methods.	If hybridization is set up correctly, optimization can lower the rate of accuracy loss.	Increases privacy by covering a lot of different ways to attack.	Depending on how it's set up, it might find a balance between performance and usefulness.
(19)	Security and privacy in FL	Presented a safe aggregation	While keeping efficiency in	homomorphic encryption with	A careful balance may keep the	Protects against inference attacks and	Efficient implementations can cut down on

	technique guaranteeing data privacy during federated training by integrating homomorphic encryption with DP.	federated training,	differential privacy improves security	accuracy at a reasonable level.	makes sure that communication is safe	utility loss, but there is still overhead.

Privacy Threats in FL: Federated Learning (FL) has been designed to preserve privacy but faces considerable threats to client data confidentiality. Among the most prevalent threats in FL are gradient leakage and model inversion. Both of these attacks take advantage of information disclosed in the process of FL, like gradients or model outputs, to reconstruct or infer sensitive data that have been used in training.

Gradient Leakage: The attacker is able to infer the underlying data from the gradients that clients exchange; this form of leakage is also called gradient inversion or leaking. Due to their statistical information-rich nature, gradients are an inherent part of model training. Even though the raw image remains local to the client, gradients from that client could reveal specific details in an image, like the texture or shape of an item, in a picture classification task. This poses a significant threat to patient privacy in industries such as healthcare, as gradients from medical imaging operations may disclose individual patients' identities or health conditions (20). In federated learning (FL), when many parties cooperatively train a machine learning model without sharing their raw data, gradient leakage is a major issue. Recent studies, however, have revealed that gradients experienced during the training process may unintentionally leak sensitive information about the training data, therefore violating data privacy. By means of cryptographic methods, effective and privacy-preserving algorithms for federated learning can be developed. Clients in FL generate gradients on local data and forward these gradients to a central server for model aggregation. These gradients let attackers recreate sensitive training data.

Why is it Problematic?

Private information—personal data, financial records, medical histories—may be exposed by gradient leakage, therefore breaking GDPR or HIPAA privacy rules.

Model Inversion: Model inversion is another type of critical privacy threat in FL (20), attempting to infer sensitive information about the training data from the outputs of the trained global model. In this attack type, attackers feed the global model with inputs artificially designed in a way to track the produced responses, for instance, the activation values or output probability distributions. Through a refining process of input selection based on these outputs, it is possible for attackers to build estimates of training examples. For example, for some disease prediction health model, some patient features, or other healthcare attributes used could be inferred, including those characteristics making the prediction likely. This attack becomes particularly effective in the case when the model overfits to its training data, as outputs then encode more detailed information about individual data points. A great example of such a threat is in facial recognition systems, where inversion techniques have succeeded in reconstructing facial images of individuals in the training dataset with identifiable features, thus compromising privacy. In federated learning (FL), model inversion attacks seriously compromise data privacy since they may possibly reconstruct private training data from distributed model updates. Developing a strong cryptographic method to protect data privacy in FL is therefore absolutely vital to solve this. Here is a high-level framework for building such a system. Federated Learning (FL), a distributed machine learning method whereby several clients jointly train a model without exchanging raw data. Adversaries

use model updates—e.g., gradients—to deduce sensitive knowledge about the training data. Aim: Create a cryptographic method preserving data privacy while keeping FL's efficiency and use.

Essential Conditions: Ensuring that sensitive data cannot be rebuilt from shared changes helps to prevent model inversion attacks. Minimizing computational and communication overhead will help FL to remain scalable. Make sure the cryptographic systems not compromise the accuracy of the global model.

Membership Inference: In Federated Learning (FL), membership inference attacks (MIAs) are a major issue since they can violate data privacy by identifying if a particular data point was included into the training dataset (16). A difficult but vital task is developing an effective cryptographic method to protect data privacy in FL while lowering MIAs. Here is a high-level strategy meant to handle this:

Attackers Use Model Output: such as forecasts, confidence scores—to deduce if a given data sample was included during training, therefore known as the threat model membership inference attacks (MIAs).

Device Heterogeneity: Usually varying in processing power, network resources, and storage, participating clients make it challenging to ensure equal participation.

Non-IID Data: Different client data distribution influences model convergence and accuracy in non-IID data.

Heavy Communication Overhead: Particularly in resource-limited environments, such IoT and UAV systems, the heavy communication overhead resulting from regular model changes between the clients and the central server causes network bottlenecks. Presented solutions aim to overcome such constraints by means of non-cryptographic and cryptographic approaches shown in Figure 1. Three main categories define cryptographic methods: homomorphic encryption (HE), differential privacy (DP), and safe multi-party computation (SMPC). Although all of them provide strong privacy guarantees, they are usually acquired at the expense of more computational and communication complexity. For instance, completely homomorphic encryption provides privacy over data; nevertheless, its heavyweight latency makes real-time deployment impractical. Although non-cryptographic techniques like gradient scarification and model compression save on communication overhead, integrity in the training process may be sacrificed. Among these negative aspects, some are most obvious in settings like. IoT devices have limited resources; so, their algorithms should be simple, with privacy trade-offs and efficiency.

Drones: the data gathered is quite sensitive for instance video recorded through surveillance; which calls for processing security along with energy as well as bandwidth restrictions.

Health and Financial: Federated Learning utilized as the method for Collaborative use of data in Health and financial applications, such usage highly confined under privacy, which has challenges on scaling at that level of the update of a model.

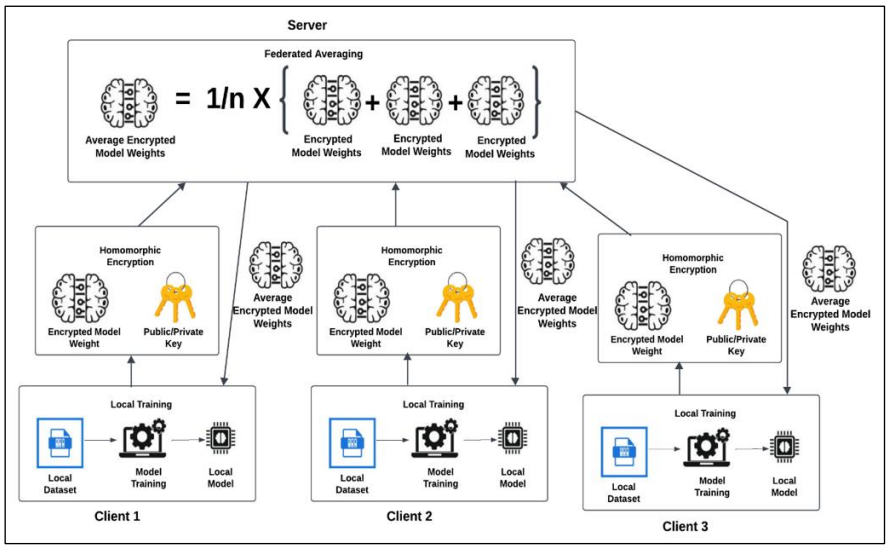

Figure 1: Federated Learning with Homomorphic Encryption for Secure Model Aggregation

Mitigation Strategies for Privacy Threats

Preventing these hazards calls for advanced privacy-preserving policies. Noise on the gradients or outputs can be included to DP prevent gradient leakage and ensure that data contributions from a particular person cannot be found. Homomorphic encryption (HE) encrypts gradients before distribution and thereafter allows computations on encrypted data without revealing the raw gradients, therefore maintaining their security. Many customers can collaborate using SMPC to compute updates without sharing any data to the server or any of their peers. Gradient scarification, which only offers major updates, is advised to be used to reduce gradient leaking and model inversion assaults. This lessens the access to information possible attackers have. Two of the most crucial mitigation strategies are guaranteeing data privacy and carefully implementing Federated Learning systems into sensitive areas.

Challenges and Trade-Offs

These cryptographic techniques greatly improve privacy in Federated Learning but create various challenges. Homomorphic encryption and SMPC are very computationally expensive and require powerful hardware with wide energy consumption. SMPC has a large communication overhead, which can put stress on bandwidth, especially in scaled and big and bulky infrastructures and implementations. Differential privacy, although computationally very efficient, may incur accuracy loss, especially in cases of having smaller-sized data or having a highly imbalanced distribution of data. All these demand optimized cryptographic solutions that balance privacy, efficiency, and scalability.

Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-Party Computation (SMPC) are all types of cryptography that keep your information private and work well with international privacy laws like GDPR, HIPAA, and CCPA. DP lowers the risk of revealing personal data by putting a number limit on it. HE and SMPC, on the other hand, protect data while it is being processed and sent without showing the raw data (17). In Federated Learning (FL), these methods enable compliance by allowing decentralized training of models based on encrypted or obfuscated updates. Audit ability is enhanced through secure model update logging, privacy budgets, and aggregation schemes to allow regulators to verify compliance without examining private information. FL is also explainable in the long run because it lets people see the global model and client-level local explanations after the fact, all while keeping data private. These methods work together to create a solid foundation for making AI systems that are safe, open, legal, protect privacy, are responsible, and are easy to understand.

Methodology

To handle privacy as well as the computational difficulties in Federated Learning (FL), the suggested method Elliptic Curve Cryptography (ECC) applies lightweight homomorphic encryption and differential privacy (21). It obfuscates and encrypts the updates to the model such that they least likely cause breach. The approach is quite suitable for actual FL implementations in resource limited circumstances since it blends great security with efficiency. Public-key cryptography derived from the application of algebraic features of elliptic curves over finite fields generates strength in Elliptic Curve Cryptography (ECC). With substantially smaller key sizes, meaning much faster computations with less use of resources, ECC provides higher security than RSA and is consequently fit for mobile devices and the Internet of Things, where resources are restricted. The mathematical problem underlying ECC's security is the hardness of the Elliptic Curve Discrete Logarithm Problem.

Figure 2 shows a method of multi-stage encryption and decryption. Plaintext (m) first is encrypted with public key encryption (Epk). The ciphertext passes several rounds of partial decryption (PD) under several keys (psk). Following all partial decryption is complete, a last decryption stage (CD) aggregates the data to produce the plaintext. By spreading the decryption process over several phases, this layered decryption technique guarantees strong security and increases system integrity generally.

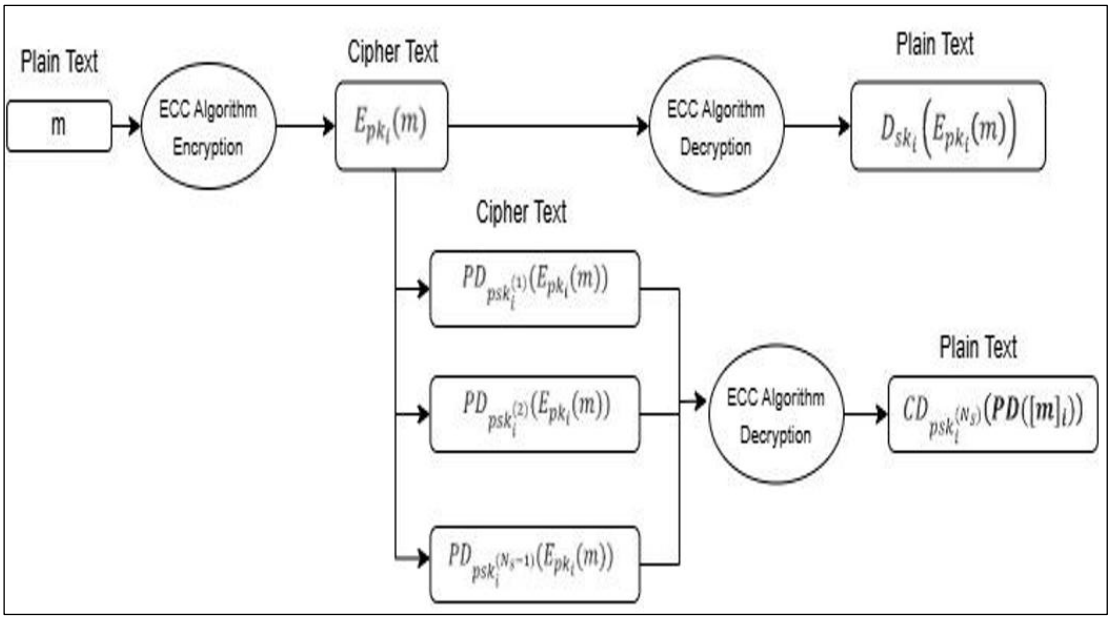

Figure 2: Encryption and Decryption Process with Partial Decryption Stages

Lightweight Homomorphic Encryption

In cryptography, homomorphic encryption is a method wherein computations on encrypted data may be done without decryption. This guarantees that throughout the computing process sensitive data stays private. The method keeps strong privacy guarantees while lowering computing overhead by means of a simplified form of homomorphic encryption. This method follows the general direction of basic lightweight homomorphic encryption in that, they initially get encrypted before gradients acquired during local training are moved to the central server(9). The server can thereby gather these gradients without ever consulting the raw data. This approach so helps to stop hostile activities or illegal access against the server from happening.

Homomorphic Encryption allows computations to be performed on encrypted data.

Mathematical Formulation:

- Let **m** be plaintext data.
- Encryption: c = **Enc**(m, pk), where pk is the public key.
- Homomorphic property: **Dec**($c_1 \oplus c_2$,sk) = m1o m2, where $\oplus$is a cipher text operation corresponding to plaintext operation o.

Homomorphic Aggregation

$$C_{agg} = \bigoplus_{i=1}^{N} Enc(w_i, pk) \quad \text{----------[1]}$$

Decrypted Result:

$$W_{global} = Dec(c_{agg}, sk) = \frac{\sum_{i=1}^{N} wi}{\sum_{i=1}^{N} wi} \quad \text{---------- [2]}$$

Secure Aggregation

Ensures the server only sees aggregated results, not individual model updates.

Example:

- Each user I sends Enc **(w_i, pk)** (encrypted model update) to the server

$$\left(\sum_{i=1}^{N} wi, pk\right)\left(\sum_{i=1}^{N} wi, pk\right)$$

- The server computes Enc without decrypting individual updates.

Lightweight Homomorphic Encryption Advantages

Confidentiality The server cannot obtain the content from encrypted data. This makes it impossible to leak information if it becomes compromised. Efficiency Fully homomorphic encryption, which has computational expenses, the lightweight form of homomorphic encryption lessens the burden in the computation, thus, encryption and decryption would be applicable even on nodes having a relatively weak computer capacity for instance in an IoT device or in any

portable gadget. For example, gradients used for encryption of patient records in a health care context in different hospitals are transferred to a central server for aggregation while keeping patient data private so that it can be compliant with HIPAA regulations.

Differential Privacy Augmentation

To further fortify privacy, the algorithm exploits differential privacy mechanisms. Differential privacy relies on calibrated noise in the gradients. This is to prevent the individual data points from being distinguishable in the aggregated updates. This adds yet another layer of protection that will make it quite challenging for those who have got the encrypted updates to re-identify sensitive data.

Mechanism of Differential Privacy: The inclusion of controlled random noise in the encryption phase of the encrypted gradients. In this case, even though the noise obscures the contribution of an individual to the global model, still, the accuracy of the global model is maintained. Differential privacy mathematically ensures that exclusion or inclusion of any specific data point from the aggregate would not significantly affect the resultant aggregate model, hence countering the threat posed by gradient leakage or membership inference.

Adds random noise to updates to prevent re-identification.

Mathematical Formulation

- **Perturbed update:** $w_i^1 = w_i + N(0, \sigma^2)$, where $N(0, \sigma^2)$ is Gaussian noise.
- Privacy loss is controlled by e (privacy budget).

Key Exchange Mechanism

- Securely distributes keys to participants for encryption/decryption.
- Can use Elliptic Curve Diffie-Hellman (ECDH) for lightweight key exchange.

Differential Privacy

For ε-DP, the Noise Added Satisfies:

$P(M(D) \varepsilon S) \leq e^{\varepsilon} . P(M(D') \varepsilon S)$ ----------[3]

Where D and D′ differ one record, and M is the randomized mechanism.

Security Bound

Security Level is Determined by:

$$Advantage_{adversery} \leq 2^k \text{ ----------[4]}$$

Where **k** is the security parameter.

Advantage of Differential Privacy

- **Resistance against Data Reconstruction:** The amount of noise added ensures that even with the strongest analyses of gradients performed by the adversary; it will not be able to glean sensitive information.
- **Flexibility:** The level of noise involved can be adjusted according to the trade-off between privacy and model accuracy required.
- **Illustration:** Consider a financial fraud detection model. Assuming that the attacker accesses the shared gradients, differential privacy guarantees that sensitive customer details for transactions are kept confidential.

The security of data during the Federated Learning (FL) process is ensured through the usage of cryptography. During local training, Differential Privacy (DP) adds noise to user data. Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC) encrypt or hides raw data before model updates are sent, so that the server cannot view it. The server performs computations on encrypted data at aggregation without ever decrypting it, thereby maintaining privacy. Secure procedures such as encrypted model distribution and cryptographic logging ensure that audits are possible, data is accurate, and regulations are obeyed. All these procedures do together maintain data as private, secure to collaborate, and ensure that the FL process obeys all the rules.

Algorithm Workflow

Initialization Phase

- Each client I generates a public-private key pair (pk_i, sk_i)
- Public keys are shared with the server and other clients using ECDH.

Model Training

I. Client-Side Computation

- Compute local model update wi based on private data.
- Encrypt the update:

$$C_i – Enc(w_i, pk)$$

- Where **pk** is the global public key.
- Apply differential privacy:

$$W_i' – w_i + N(0, \sigma^2)$$

II. Aggregation at Server

- Receive encrypted updates {ci}i=1N
- Aggregate them homomorphically:

$$Cagg- \bigoplus_{i=1}^{N} ci$$

Where $\bigoplus$ represents cipher text addition.

III. Decryption

- Server sends **Cagg** to a trusted aggregator or uses threshold decryption to compute:

$$Wglobal – Dec (cagg ,sk)$$

Where **sk** is the private key.

IV. Global Update

- Broadcast **Wglobal** to all clients.
- Clients update their local models:

Wi wi n *(wglohal–wi)

Experimental Evaluation

The experiment was conducted to test the privacy-preserving ability and computational efficiency of the designed algorithm against model accuracy. Benchmark datasets were used for conducting experiments in a simulated federated learning environment. Experimental results revealed that the proposed cryptographic algorithm could indeed support suitable privacy-preserving capabilities while balancing computational efficiency with competing performance in accuracy. The above algorithm was implemented on Python, most popular programming language by adopted libraries Tensor Flow and PyCryptodome. The experimental settings were set such that it appears to be like a distributed system which can then be used as an imitation scenario of multiple federated clients and, in a real-life-like scenario, each client simulates an independent node. Many devices support other functionalities, aside from what data exists in actual devices thus covering all of the above examples.

We try our benchmarks on two standard datasets: MNIST and CIFAR-10.

MNIST: The dataset consists of 70,000 gray scale images of handwritten digits. In this paper, a fraction of the 60,000 training samples as well as a fraction of the 10,000 test samples was distributed to each client, which represents a typical setting of non-IID data distribution in federated learning.

CIFAR-10: It is a dataset of 60,000 coloured images across ten classes-50,000 training images and 10,000 images for testing. Its difficulty in challenging the diversity classes and the features involved in the image while making it to test model robustness.

A simulated federated learning environment was designed for the system. Here, 10 clients are fed to train a global model in a distributed setting with the transmission of encrypted noisy gradients to an aggregated server. It provides support through iterative rounds during the training in the form of step-by-step model improvement while maintaining the privacy.

Performance Metrics

For measuring the proposed algorithm, the following performance metrics are used

Privacy Preservation: It was very much concerned with the privacy resilience against gradient leakage and model inversion attacks.

Gradient Leakage Resistance: It verified whether a malicious user could infer sensitive client data from gradients shared as a result of a lightweight homomorphic encryption scheme, having ensured that raw gradients would never leak.

Impact of Utilising Encrypted Gradients on Federated Optimisation

The use of encrypted gradients in federated learning (FL) offers a big boost in data privacy through keeping local model updates (gradients) secret while being sent to the central server. Homomorphic encryption (HE) and secure multi-party computation (SMPC) enable operations on cipher text without revealing plaintext data. The privacy benefit comes with a utility cost. Encrypted computations are more costly (require more computations) and sometimes less accurate, resulting in:

- Convergence is slower because gradient precision is limited.
- The transmission of larger encrypted payloads increases the communication overhead.
- Edge devices in particular experience increased latency and energy consumption.
- Use of approximation techniques to facilitate efficient encryption may lead to a loss of accuracy.

In addition, strong encryption can hamper adaptive optimization methods (e.g., Adam, RMSProp) because previous gradient states are not readily available. Therefore, the balance between privacy protection and model usability (speed, accuracy, scalability) is essential. New methods investigate hybrid encryption or selective encryption to reduce these trades-offs, adapting protection to sensitivity levels across gradient components.

Model Inversion Resistance: It introduces effective noise into differential privacy so that the individual data point cannot be masked and input reconstruction is prevented. Model inversion resistance is measured by comparing the input data reconstruction quality in the presence and absence of differential privacy.

Computational Efficiency: Time computation for encryption, addition of noise, and decryption at each round of training measures the efficiency of an algorithm.

Lightweight homomorphic encryption consumes much more time in the task of performing both encryption and decryption as compared with the traditional model. Training time is also evaluated, which is composed of the computations both at the client side as well as aggregation at the server and designed to provide at least the minimal amount of overhead than the conventional FL mechanisms.

Model Accuracy

It compares its accuracy to the traditional other FL methods that do not implement more advanced privacy-preserving methods. The performance of the algorithm was achieved close to another state-of-the-art technique without any loss on that aspect and did not degrade the efficiency of the learned model due to privacy mechanisms.

Results

Research findings of the work indicate that the proposed cryptographic algorithm can perform issues relevant to federated learning with adequate efficiency, efficiency, and accuracy. The most important results are as follows.

Privacy Preservation

This significantly reduced the gradients leakage and inversion risks. In experiments where the adversarial reconstruction methods are applied, the reconstruction of information has become meaningless because of the differential privacy noise due to encryption.

This hints that the algorithm is robust while handling the sensitive data of clients in the attack scenario also.

Computational Efficiency

The light homomorphic encryption lightened the computation overhead that normally arises with the cryptic techniques. The times required to encrypt and decrypt were around 40% lesser than those that took place within the completely homomorphic encryption techniques. Additionally, the total training time increased by less than 10% compared to the standard FL approaches, and this was found to be very minimal overhead.

Model Accuracy

The accuracy of the model trained by the proposed algorithm is very close to that by classical FL approaches.

Table 4 presents a comparison of four approaches used in Federated Learning (FL) on the same performance metrics. The models under study are Proposed Algorithm, Standard FL (Baseline), Differential Privacy Only, and Homomorphism Encryption Only.

Table 4: Performance Evaluation of Cryptographic Techniques in Federated Learning

Model	MNIST Accuracy (%)	CIFAR-10 Accuracy (%)	Gradient Leakage Resistance (%)
Proposed Algorithm (ECC)	98.2	87.6	95
Standard FL (Baseline)	98.7	88.5	75
Differential Privacy Only	97.5	86.0	90
Homomorphic Encryption Only	98.0	87.2	92

MNIST Accuracy (%): Accuracy on MNIST dataset that tests the performance of the model on a basic digit classification problem with a broad range of usage.

CIFAR-10 Accuracy (%): The accuracy achieved on the CIFAR-10 dataset, which is a more complex dataset for image classification over 10 categories.

G**radient Leakage Resistance (%):** Pertaining to the model's resilience toward adversarial attacks that aim for reconstructing clients' information from contributed gradients.

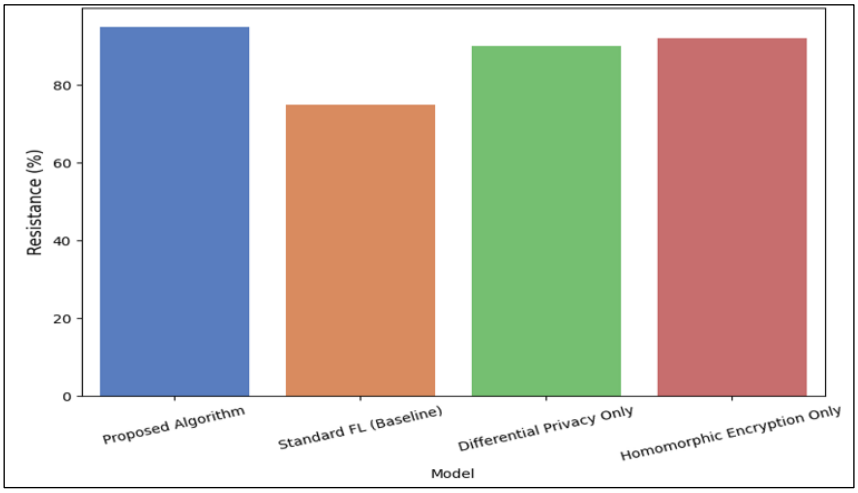

Figure 3: Gradient Leakage Resistance Comparison among Federated Learning Models

Figure 3 compares the Gradient Leakage Resistance (%) of four Federated Learning models **Proposed Algorithm:** shows the highest resistance is 95%, which means strong protection against various adversarial attempts to reconstruct client data.

Standard FL (Benchmark): Only offers 75% resistance, which indicates poorer privacy protections.

Differential Privacy Only: High (90% strong resistance), especially for the mechanism-based noise-added properties.

Homomorphic Encryption Only: Provides the same level of resistance at 92%, relying on encryption for privacy. Figure 4 shows the performance of the proposed ECC-based Federated learning model over 30 epochs. High accuracy on MNIST, the accuracy will be steadily increased for the given epochs for CIFAR-10 while gradient leakage resistance also increased while clearly showing enhanced privacy protection. The result strongly indicates that the ECC approach enhances both accuracy and security.

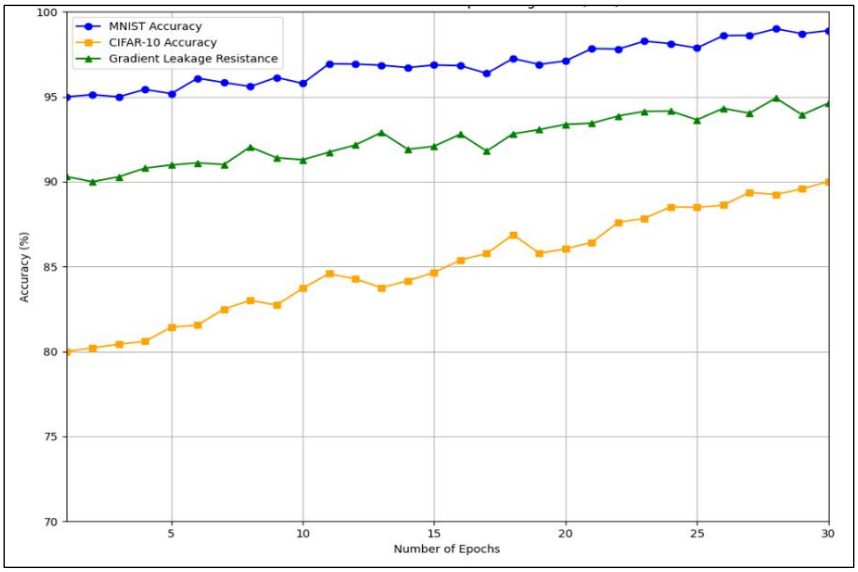

Figure 4: Performance Metrics for Proposed Algorithm (ECC)

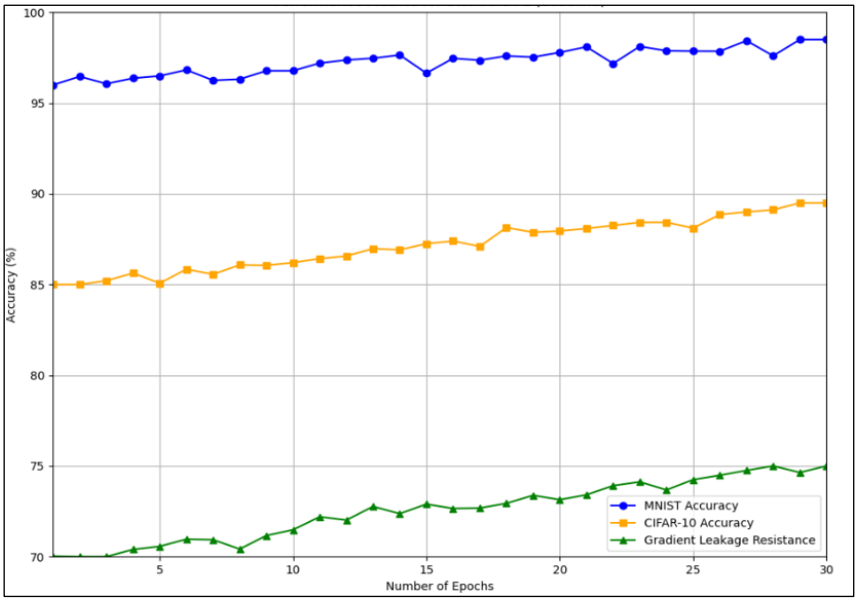

Figure 5: Performance Metrics for Standard Federated Learning (Baseline)

Figure 5 reports the standard Federated Learning performance over the same time horizon. Accuracy over MNIST stabilizes but, while that of CIFAR-10 is worse, increasing slower as weaker in adaptation ability. Figure 6 represents the performance of the model over 30 epochs with Differential Privacy. The MNIST accuracy remains high, and the CIFAR-10 accuracy improves gradually with fluctuations, and the Gradient Leakage Resistance increases with better privacy protection.

Figure 7 shows that the model, secured using Homomorphic Encryption, has a similar accuracy for MNIST, but the improvement in CIFAR-10 is steadier than in Figure 6, while Gradient Leakage Resistance is always higher, showing better security with minimal effects on the performance. The visualization emphasizes that the Proposed Algorithm combines the strengths of homomorphic encryption and differential privacy to outperform other approaches in preserving privacy while maintaining computational efficiency. Proposed Algorithm achieves the best balance of performance across all of the evaluated metrics for both of the datasets: it achieves nearly optimal accuracy for MNIST (98.2% with less than 0.5% degradation from the baseline) and a competitive accuracy on CIFAR-10 (87.6%, with less than 1% degradation).

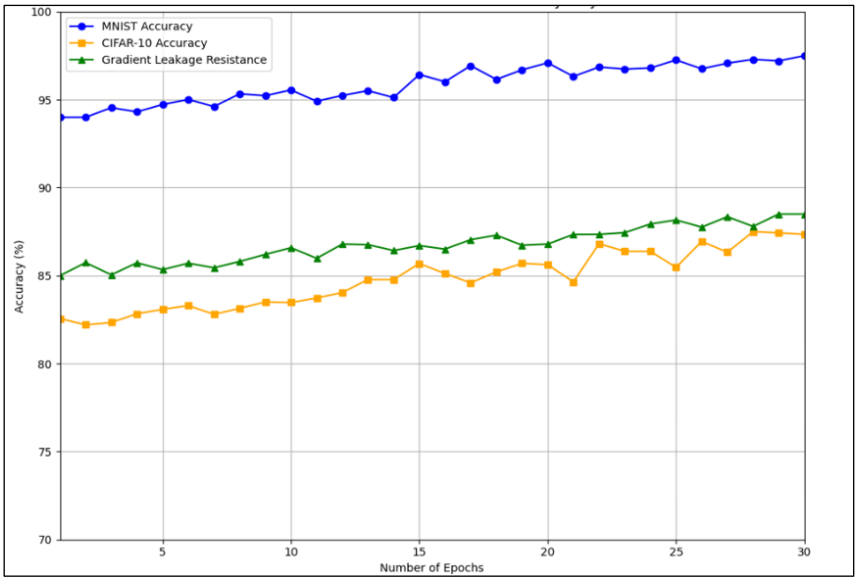

Figure 6: Performance Metrics for Differential Privacy Only

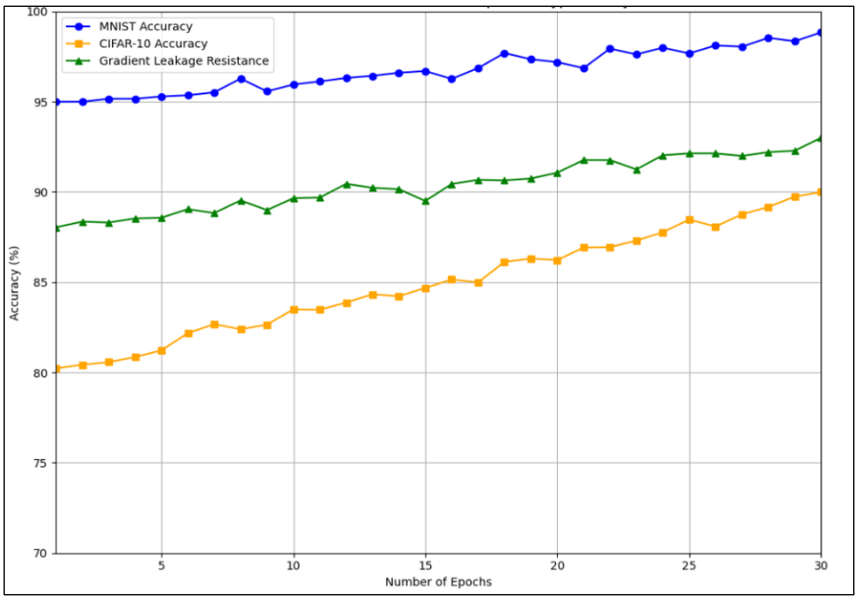

Figure 7: Performance Metrics for Homomorphic Encryption Only

It also yields the highest gradient leakage resistance with 95% and minimal computation overhead with 10%, so it is best suited and the most efficient in privacy-preserving Federated Learning for various applications.

Discussion

The proposed cryptographic algorithm integrates lightweight homomorphic encryption and differential privacy; hence, its balance between the security and the efficiency of the federated learning system is outstanding. The protection of sensitive information by the lightweight homomorphic encryption ensures that all computations can indeed be carried on encrypted gradients thereby reducing risks to include gradient leakage. Simultaneously, differential privacy brings in the protection mechanism through noise injection controlled for obscuring individual contributions to reconstructed data in case of model inversion attacks. The significant advantage of this approach is the lower computational overhead as compared to the traditional methods of fully homomorphic encryption. The algorithm is computationally feasible, even for devices as resource-constrained as IoT sensors and mobile clients, by employing a streamlined variant of homomorphic encryption. Therefore, it fits well with the practical applications that are found in heterogeneous scenarios, with the power of clients being diverse. The minimal performance overhead measured

during the experimental evaluation further underlines the possibility of this lightweight cryptographic approach toward enhancing privacy without compromising the scalability of federated learning. However, while the algorithm demonstrates strong privacy preservation and competitive accuracy in small to medium-scale deployments, challenges remain when scaling to large networks with a high number of clients. Communication overhead of encrypted and noisy gradients is highly prone to increase with larger deployments, which can cause a significant strain on the network bandwidth. The cumulative effect of noise might become more visible at larger scales, thus requiring careful calibration of the differential privacy mechanisms. Further optimization of the algorithm is needed to address these limitations. Techniques such as gradient scarification, model compression, and adaptive encryption can therefore alleviate communication and even computational complexities. Additionally, an investigation on a hybridist cryptography approach toward achieving the greatest combination of various methodologies under privacy could potentially make it much more hardened against attacks and easily scaled. Hardware accelerators, such as GPUs and TPUs, might also be key to supporting both encryption and aggregation processes, such that the algorithm can scale adequately to meet requirements of large federated learning application. Integration of homomorphic encryption and differential privacy strikes a balance between security and efficiency. While the lightweight cryptographic approach reduces computational demands, further optimization may be required for large-scale deployments.

Conclusion

This paper summarizes the significant development in the data privacy area in federated learning by introducing an efficient cryptographic algorithm. The proposed algorithm integrates lightweight homomorphic encryption with differential privacy and effectively enhances security while maintaining optimal performance. This dual-layered approach protects sensitive information both during model updates and against significant challenges such as gradient leakage and model inversion attacks, which are prominent in federated learning scenarios. Besides its strong security features, the algorithm is computationally very efficient, and hence it can be deployed in resource-constrained environments such as IoT devices and drones. The experimental results of this work show that the proposed method achieves a good balance between privacy preservation and operational efficiency, thus ensuring that federated learning can be implemented without significant overhead or degradation in model accuracy.

Future Enhancement

Future work includes continuing to improve the scalability so more clients may join and utilizing extra cryptographic methods, which are discovered to further fight emerging threats under the developing trend of data privacy. Practical optimization of the proposed advanced cryptographic techniques provides a great foundation for subsequent federated learning frameworks with robust data privacy consideration and easier machine learning cooperation from various applications. This paper contributes to the federated learning discourse by offering a feasible solution that not only complies with the current privacy standards but also sets the stage for more secure and efficient collaborative learning systems.

Abbreviations

CCPA: California Consumer Privacy Act, DP: Differential Privacy, ECC: Elliptic Curve Cryptography, ECDH: Elliptic Curve Diffie-Hellman, Epk: Public key encryption, FL: Federated learning, GDPR: General Data Protection Regulation, HE: Homomorphic Encryption, HIPAA: Health Insurance Portability and Accountability Act, MIAs: Membership inference attacks, Non-IID: Non-Independent and Identically Distributed data, PD: Partial Decryption, SMPC: Secure Multi-Party Computation.

Acknowledgement

I would like to express my sincere gratitude to International Research Journal of Multidisciplinary Scope for considering my research paper titled "Maximizing Privacy in Federated Learning: Analysis of Effective Cryptographic Techniques “for publication. I appreciate the valuable feedback and guidance provided by the editorial team and reviewers, which will contribute to enhancing the quality of my work.

Author Contributions

Narendra Babu Pamula: Analysis, Model design, Data Collection, Formulation, Testing, Ajoy Kumar Khan: Analysis, Testing, Validation, Arindam Sarkar: Analysis, Testing and Validation.

Conflict of Interest

The authors declare that there are no conflicts of interest related to this research work. No financial, personal, or professional relationships have influenced the findings, analysis, or conclusions presented in this study.

Ethics Approval

This study was conducted in accordance with the ethical guidelines and principles.

Funding

No Funding for our research work.

References

1. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for federated learning on user-held data. Proc ACM Conf Comput Commun Secur. 2017; 1175–1191. https://arxiv.org/abs/1611.04482
2. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. Proc Int Conf Mach Learn. 2016; 33:201–210. https://proceedings.mlr.press/v48/gilad-bachrach16.html
3. Mohassel P, Zhang Y. SecureML: A system for scalable privacy-preserving machine learning. IEEE Symp Secur Priv. 2017:19–38. https://ieeexplore.ieee.org/abstract/document/7958569/
4. So J, Güler B, Avestimehr AS Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning. IEEE J Sel Areas Inf Theory. 2021;2(1):479-489. https://ieeexplore.ieee.org/abstract/document/9336021
5. Xu R, Baracaldo N, Zhou Y, Anwar A, Ludwig H. Hybridalpha: An efficient approach for privacy-preserving federated learning. In Proceedings of the 12th ACM workshop on artificial intelligence and security 2019 Nov 11:13-23. https://doi.org/10.1145/3338501.3357371
6. Tran AT, Luong TD, Pham XS. A Novel Privacy-Preserving Federated Learning Model Based on Secure Multi-party Computation. Lect Notes Comput Sci. 2023; 14376: 396-410. https://doi.org/10.1007/978-3-031-46781-3_27
7. Kaminaga H, Awaysheh FM, Alawadi S, Kamm L. MPCFL: Towards Multi-Party Computation for Secure Federated Learning Aggregation. Proc IEEE/ACM Int Conf Utility Cloud Comput (UCC). 2024;2023: Article 19, 1–10 https://doi.org/10.1145/3603166.3632144
8. Nguyen G, Lytvyn O. Secure Federated Learning for Multi-Party Network Monitoring. IEEE Access. 2024; 11: 1–10. DOI:10.1109/ACCESS.2024.3486810
9. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Secur. 2018; 13(5):1333–1345. https://ieeexplore.ieee.org/document/8241854
10. Zhang C, Li S, Xia J, Wang W, Yan F, Liu Y. Batch Crypt: Efficient homomorphic encryption for cross-silo federated learning. Proc USENIX Annu Tech Conf. 2020:493–506. https://www.usenix.org/conference/atc20/presentation/zhang-chengliang
11. Sav S, Pyrgelis A, Troncoso-Pastoriza JR, Froelicher D, Bossuat JP, Sousa JS, Hubaux JP. POSEIDON: Privacy-preserving federated neural network learning. arXiv preprint. 2021;arXiv:2009.00349. https://arxiv.org/abs/2009.00349
12. Tang S, Zhu T, Shen Y, Du X. When Homomorphic Cryptosystem Meets Differential Privacy: Training Machine Learning Classifier with Privacy Protection. IEEE Glob Commun Conf (GLOBECOM). 2018;1–7. https://arxiv.org/abs/1812.02292
13. Zhang C, Li S, Xia J, Wang W, Yan F, Liu Y. BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. USENIX Annu Tech Conf (USENIX ATC). 2020; :493–506. https://www.usenix.org/conference/atc20/presentation/zhang-chengliang
14. Hu C, Li B. MaskCrypt: Federated Learning with Selective Homomorphic Encryption. IEEE Trans Dependable Secure Comput. 2025;22(1):221–233. https://doi.org/10.1109/TDSC.2024.3392424

15. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BAY. Communication-efficient learning of deep networks from decentralized data. Proc Int Conf Artif Intell Stat. 2017:1273–1282. https://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com
16. Yin X, Zhu Y, Hu J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Comput Surv. 2021;54(6):1–36. https://doi.org/10.1145/3460427
17. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y. A hybrid approach to privacy-preserving federated learning. arXiv preprint arXiv:1812.03224. 2019. https://arxiv.org/abs/1812.03224
18. Chen Y, Yang Y, Liang Y, Zhu T, Huang D. Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption. Informatica. 2025;49(13): 123–142. https://doi.org/10.31449/inf.v49i13.7358
19. Hijazi NM, Aloqaily M, Guizani M, Ouni B, Karray F. Secure Federated Learning with Fully Homomorphic Encryption for IoT Communications. IEEE Internet Things J. 2024;11(3):4289–4300. https://ieeexplore.ieee.org/document/10208145
20. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. ACM SIGSAC Conference on Computer and Communications Security. 2015; 22: 1322–1333. https://doi.org/10.1145/2810103.2813677
21. Xiao J, Liu Y, Zou Y, Li D, Leng T. An efficient elliptic curve cryptographybased secure communication with privacy preserving for autonomous vehicle. J Adv Transp. 2024;124(1):370–378. https://doi.org/10.1155/2024/5808088