FedAvg-FHEMK: Aprendizado Federado Seguro com Criptografia Homomórfica Multi-Chave Eficiente

Gabriel S. Rosa[1], **Hilder V. L. Pereira**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil

{g235988,hilder}@unicamp.br

Abstract. *The adoption of FL exposes risks of information leakage, since it is still possible to recover data from the model updates shared during the protocol. Homomorphic encryption (FHE) provides strong security by enabling computation over encrypted data; however, its application to FL, especially in the multi-key setting is constrained by high computational costs, communication overhead, and interactive protocols. This work proposes FedAvg-FHEMK, a secure federated learning protocol based on FHE with a modified multi-key arrangement, which reduces cost and complexity by requiring only a single communication round per iteration and concentrating computation on quasi-linear operations. The scheme is better suited to stable cross-silo scenarios, as it requires a more expensive initialization phase and does not tolerate client dropout. It guarantees protection against a semi-honest server even under collusion of up to $P - 2$ clients, where P denotes the total number of clients participating in the federation. Experimental results show that FedAvg-FHEMK preserves model accuracy, with a moderate increase in computational cost and a significant reduction in communication cost compared to single-key FHE approaches, indicating a practical trade-off between strong privacy and efficiency in secure federated learning.*

Resumo. *A adoção do FL expõe riscos de vazamento de informação, pois ainda é possível recuperar os dados a partir das atualizações do modelo compartilhadas durante o protocolo. A criptografia homomórfica (FHE) traz segurança ao permitir computação sobre dados cifrados, mas sua aplicação em FL, especialmente no cenário multi-chave, é limitada por altos custos computacionais, sobrecarga de comunicação e protocolos interativos. Este trabalho propõe o FedAvg-FHEMK, um protocolo de aprendizado federado seguro baseado em FHE com arranjo multi-chave modificado, que reduz custo e complexidade ao exigir apenas uma rodada de comunicação por iteração e concentrar o processamento em operações quase lineares. O esquema é mais adequado a cenários* cross-silo *estáveis, pois requer uma fase inicial mais custosa e não tolera abandono de clientes. Ele garante proteção contra servidor semi-honesto mesmo sob colusão de até $P - 2$ clientes, onde P é o número total de clientes participantes da federação. Experimentos mostram que o FedAvg-FHEMK preserva acurácia do modelo, com aumento moderado do custo computacional e redução significativa do custo de comunicação em relação a abordagens de FHE de chave*

única, indicando um compromisso prático entre privacidade forte e eficiência em aprendizado federado seguro.

1. Introdução

O treinamento de modelos de inteligência artificial em larga escala enfrenta um dilema fundamental: dados sensíveis não podem ser compartilhados livremente, mas modelos precisos demandam grandes volumes de informação. Em setores como saúde, finanças e telecomunicações, regulamentações como a Lei Geral de Proteção de Dados (LGPD) no Brasil e o Regulamento Geral sobre a Proteção de Dados (GDPR, do inglês *General Data Protection Regulation*) na Europa impedem a centralização de dados privados, fragmentando bases que poderiam impulsionar modelos mais robustos. O Aprendizado Federado (FL, do inglês *Federated Learning*) emergiu como solução promissora ao permitir treinamento colaborativo sem transferência de dados brutos: clientes treinam localmente e compartilham apenas atualizações de modelo (gradientes ou pesos) com um servidor central que as agrega. Porém, a premissa de privacidade do FL foi severamente abalada por ataques de inversão de gradientes. Pesquisas demonstraram que gradientes compartilhados podem ser explorados para reconstruir dados de treinamento com alta fidelidade, incluindo imagens e textos [Zhu et al. 2019, Geiping et al. 2020]. Ataques mais recentes como Aprendizado para Inversão (LTI, do inglês *Learning to Invert*) [Wu et al. 2023] aprimoraram essas técnicas, evidenciando que a mera descentralização dos dados não garante privacidade.

Tal cenário motivou a busca por soluções criptográficas robustas. A Criptografia Homomórfica Completa (FHE, do inglês *Fully Homomorphic Encryption*) destaca-se por permitir operações matemáticas sobre dados cifrados sem decifrá-los, oferecendo garantias formais de segurança. No entanto, esquemas tradicionais de FHE de chave única criam um ponto único de falha: se a chave é comprometida, toda a privacidade colapsa. Por outro lado, a Criptografia Homomórfica Completa Multi-Chave (MK-FHE, do inglês *Multi-Key Fully Homomorphic Encryption*), onde cada cliente possui sua própria chave, elimina esse risco mas historicamente sofre com custos computacionais e de comunicação proibitivos, tornando-a impraticável para FL em larga escala.

Este trabalho apresenta o FedAvg-FHEMK, um protocolo de FL seguro baseado em MK-FHE modificado que reconcilia privacidade forte com viabilidade prática. Tal abordagem se diferencia ao utilizar um setup distribuído de chaves executado uma única vez, baseado em interpolação polinomial via Transformada Numérica Teórica (NTT, do inglês *Number Theoretic Transform*), permitindo que o servidor compute uma chave agregada sem aprender as chaves individuais. Durante o treinamento, cada iteração requer apenas uma rodada de comunicação entre cliente e servidor, eliminando múltiplas interações típicas de MK-FHE tradicional. As principais contribuições são:

- **Setup eficiente**: Geração distribuída de chaves via interpolação polinomial com NTT, atingindo complexidade quasilinear por cliente em vez de quadrática, como em outros protocolos;
- **Protocolo de rodada única**: Cada iteração de treinamento exige apenas uma mensagem por cliente, reduzindo o custo de comunicação em torno de 75% quando comparado com Criptografia Homomórfica Completa de Chave Única (SK-FHE, do inglês *Single-Key Fully Homomorphic Encryption*);

- **Segurança formal**: Baseado na hipótese de que o problema Aprendizado com Erros em Anéis (RLWE, do inglês *Ring Learning with Errors*), nosso protocolo oferece segurança mesmo no cenário extremo em que $P - 2$ clientes se juntam para atacar o sistema.
- **Viabilidade prática**: Avaliação experimental conduzida sobre as bases de dados: Conjunto de Dados Modificado do Instituto Nacional de Padrões e Tecnologia (MNIST, do inglês *Modified National Institute of Standards and Technology*) e Conjunto de Dados do Instituto Canadense de Pesquisa Avançada (CIFAR, do inglês *Canadian Institute for Advanced Research*), demonstraram que o esquema proposto preserva a acurácia do modelo, com variação inferior a 1% em relação ao treinamento em claro, ao mesmo tempo em que impõe apenas um pequeno overhead computacional.

O restante do artigo está organizado como segue: a Seção 2 apresenta os fundamentos teóricos; a Seção 3 posiciona o FedAvg-FHEMK frente ao estado da arte; a Seção 4 detalha o protocolo proposto; a Seção 7 apresenta resultados experimentais; a Seção 8 discute limitações e direções futuras.

2. Referencial Teórico

Sejam q e N inteiros positivos, com N uma potência de dois. Definimos o anel quociente

$$R_q = \mathbb{Z}_q[X]/\langle X^N + 1\rangle,$$

composto por polinômios de grau estritamente menor que N, com coeficientes reduzidos módulo q. De forma análoga, ao considerar coeficientes inteiros sem redução modular, definimos

$$R = \mathbb{Z}[X]/\langle X^N + 1\rangle.$$

Ao longo deste trabalho, o módulo q é sempre escolhido como um número primo tal que $q \equiv 1 \pmod{2N}$, condição que garante a existência de raízes primitivas de ordem $2N$ em $\mathbb{Z}_q$ e, consequentemente, a correta definição da NTT sobre R_q. Essa propriedade permite a multiplicação eficiente de polinômios em R_q com complexidade $O(N \log N)$, em contraste com o custo quadrático $O(N^2)$ do método ingênuo.

O problema RLWE consiste em recuperar um segredo $s \in R_q$, dados vários pares

$$(a_1, b_1), \ldots, (a_m, b_m) \in R_q^2,$$

onde cada a_i é amostrado uniformemente em R_q e

$$b_i = a_i \cdot s + e_i \bmod q,$$

sendo $e_i \in R$ um polinômio cujos coeficientes são pequenos, tipicamente amostrados a partir de uma distribuição gaussiana discreta. O termo e_i é denominado *ruído* e sua presença é essencial tanto para a segurança do esquema quanto para a indistinguibilidade dos pares RLWE em relação a amostras uniformes. O RLWE é amplamente aceito como um problema computacionalmente difícil, inclusive frente a adversários quânticos, fundamentando a segurança de diversos esquemas criptográficos pós-quânticos.

A FHE permite a realização de operações aritméticas diretamente sobre dados cifrados, sem a necessidade de decifragem intermediária. Entre os esquemas FHE baseados em RLWE, destaca-se o Cheon–Kim–Kim–Song (CKKS), projetado para computação aproximada sobre números reais ou complexos.

No CKKS, a chave secreta é um elemento $sk \in R_q$. Para cifrar uma mensagem $m_i \in R$, o esquema amostra um par RLWE e incorpora a mensagem por meio de um fator de escala Δ, produzindo um criptograma

$$(a_i, b_i) \in R_q^2, \quad \text{onde} \quad b_i = a_i \cdot sk + e_i + \Delta \cdot m_i.$$

A decifragem resulta em uma aproximação de m_i, sendo a precisão controlada pelo fator de escala e pelo crescimento do ruído ao longo das operações homomórficas.

Uma propriedade fundamental do CKKS é sua homomorfia aproximada. A soma de dois criptogramas

$$(a_i, b_i) + (a_j, b_j) = (a_i + a_j, b_i + b_j)$$

resulta em um novo criptograma que cifra $m_i + m_j$ sob a mesma chave secreta, com ruído acumulado $e_i + e_j$. De modo semelhante, a multiplicação de um criptograma por um escalar em claro v,

$$(a_i \cdot v, b_i \cdot v),$$

cifra a mensagem $v \cdot m_i$, com crescimento proporcional do ruído. Operações entre criptogramas, em especial multiplicações, exigem procedimentos adicionais, como relinearização e reescala, para manter o ruído controlado e preservar a precisão numérica.

No contexto de protocolos distribuídos, adotamos o modelo de adversário *semi-honesto*: participantes seguem o protocolo corretamente, mas podem analisar mensagens observadas para inferir informações sensíveis, inclusive via colusão entre subconjuntos de participantes. As garantias específicas do FedAvg-FHEMK sob esse modelo são detalhadas na Seção 4.

3. Trabalhos Relacionados

A pesquisa sobre privacidade em FL tem explorado diferentes estratégias para equilibrar segurança e eficiência. Esta seção revisa as principais abordagens existentes e posiciona o FedAvg-FHEMK frente ao estado da arte.

3.1. Criptografia Seletiva e Híbrida

Essa linha de trabalho propõe cifrar apenas parâmetros considerados críticos durante a execução do protocolo, como estratégia para reduzir o custo associado ao uso de criptografia. O FedML-HE [Jin et al. 2024] emprega um mecanismo de consenso colaborativo para identificar camadas sensíveis do modelo, reportando uma redução de até 40$\times$ no overhead computacional para Representações de Codificador Bidirecionais a partir de Transformadores (BERT, do inglês *Bidirectional Encoder Representations from Transformers*), em comparação com a cifragem de 100% dos parâmetros transmitidos. No entanto, a seleção baseada em sensibilidade pode vazar informações sobre a distribuição dos dados subjacentes. O MASKCRYPT [Hu and Li 2024] descentraliza o processo de seleção dos parâmetros a serem cifrados por meio do uso de máscaras, nas quais cada cliente define localmente um subconjunto de parâmetros a proteger; essas máscaras são posteriormente

agregadas no servidor. Embora essa abordagem reduza a dependência de decisões centralizadas, ela ainda resulta na transmissão de parâmetros não cifrados, mantendo superfícies de ataque exploráveis. De forma semelhante, o FAS [Korkmaz and Rao 2025] combina FHE seletiva, Privacidade Diferencial (DP, do inglês *Differential Privacy*) e embaralhamento de bits, alcançando speedups de até 90%. Entretanto, a composição de múltiplos mecanismos de defesa dificulta a análise formal das garantias de segurança do sistema, tornando imprecisa a quantificação do esforço necessário para que um adversário recupere informações dos dados dos clientes.

Limitação fundamental: todas essas abordagens aceitam proteção parcial como trade-off de eficiência, o que resulta em superfícies de ataque residuais. Um adversário com conhecimento do mecanismo de seleção ou das máscaras empregadas pode concentrar seus esforços sobre os parâmetros desprotegidos, tornando essas soluções inadequadas para aplicações com requisitos rigorosos de privacidade.

3.2. Otimização de Dados e Protocolos Híbridos

Outra estratégia combina técnicas de compressão com criptografia. BatchCrypt [Zhang et al. 2020] utiliza quantização agressiva e empacotamento Instrução Única, Múltiplos Dados (SIMD, do inglês *Single Instruction, Multiple Data*) para compactar gradientes, atingindo reduções de 66-101x na comunicação com degradação de acurácia $< 1\%$. Essa abordagem é eficaz para modelos grandes, mas a quantização pode amplificar o ruído criptográfico em cenários sensíveis. A Criptografia Homomórfica Híbrida (HHE, do inglês *Hybrid Homomorphic Encryption*) [Correia et al. 2025] propõe arquitetura assimétrica onde clientes executam cifras simétricas leves, como o AES, e o servidor realiza conversão para FHE assimétrica, transferindo complexidade do cliente para o servidor. Embora reduza custos nos dispositivos, o overhead no servidor cresce linearmente ($\approx$15.621x por cliente), tornando a escalabilidade questionável para grandes federações.**Trade-off central**: essas técnicas redistribuem complexidade entre componentes sem reduzir o custo total.

3.3. Agregação Segura e Protocolos Baseados em MPC

Uma linha de trabalho distinta utiliza computação multi-partes segura (*Secure Multi-Party Computation*, MPC) para agregar gradientes sem revelar contribuições individuais ao servidor. Bonawitz et al. [Bonawitz et al. 2017] propuseram um protocolo de agregação segura prático que tolera dropout de clientes e requer apenas duas rodadas de comunicação por iteração, mas cujo custo de comunicação cresce com $O(P^2)$ por rodada. Bell et al. [Bell et al. 2020] estenderam esse resultado para escalabilidade com grafos esparsos ($O(P \log P)$). Esses esquemas protegem contra um servidor curioso, mas não oferecem garantias contra reconstrução a partir dos gradientes agregados (sujeitos a ataques de inversão), ao contrário do FedAvg-FHEMK, que mantém os gradientes cifrados ao longo de toda a agregação.

Em contraste, o FedAvg-FHEMK mantém gradientes cifrados durante toda a agregação e concentra o custo multi-partes inteiramente na Fase 1 de setup.

3.4. MK-FHE Tradicional

Esquemas clássicos de SK-FHE aplicados ao FL oferecem proteção contra adversários externos ao garantir que o servidor agregador opere apenas sobre dados cifrados. Contudo,

esse modelo assume implicitamente que todos os clientes são honestos. A introdução de um cliente malicioso compromete essa suposição, pois o compartilhamento de uma única chave privada permite que tal cliente decifre informações de outros participantes, fragilizando todo o protocolo. Nesse contexto, surge a MK-FHE, que aumenta significativamente a robustez do sistema ao atribuir uma chave privada distinta a cada cliente, impossibilitando que gradientes individuais sejam decifrados por qualquer outra entidade.

O principal desafio da MK-FHE reside em seus elevados custos computacionais e de comunicação, uma vez que esses esquemas tipicamente exigem múltiplas rodadas interativas para viabilizar o treinamento completo do modelo. López-Alt et al. [López-Alt et al. 2012] introduziram o primeiro esquema funcional de MK-FHE, ao custo de múltiplas rodadas de expansão de textos cifrados, com complexidade de comunicação de $O(P)$ para P clientes. Posteriormente, Chen et al. [Chen et al. 2019] aprimoraram a eficiência por meio de empacotamento SIMD e relinearização otimizada, reduzindo o custo das operações de multiplicação, mas ainda mantendo uma complexidade de comunicação superlinear. Mouchet et al. [Mouchet et al. 2021] apresentaram uma implementação prática baseada na biblioteca Lattigo, demonstrando viabilidade apenas para cenários com poucos participantes ($P \leq 10$), enquanto federações maiores permanecem inviáveis devido ao overhead. Mais recentemente, o MASER [Omar et al. 2025] combina MK-FHE com esparsificação de modelos, empregando consenso de máscaras para selecionar pesos críticos e alcançando reduções de 2–3× na comunicação, ao custo de reintroduzir garantias de privacidade parciais. **Gargalo fundamental**: complexidade de comunicação multiplicativa ou múltiplas rodadas interativas por iteração inviabilizam FL em larga escala.

3.5. Posicionamento do FedAvg-FHEMK

O FedAvg-FHEMK diferencia-se por garantir cifragem total dos parâmetros sob Indistinguibilidade sob Ataque de Texto Plano Escolhido (IND-CPA, do inglês *Indistinguishability under Chosen-Plaintext Attack*) sem interações inter-clientes durante o treinamento. Ao concentrar o custo multi-partes em um setup one-time de complexidade quasilinear, o protocolo evita tanto a proteção parcial da criptografia seletiva quanto as múltiplas rodadas da MK-FHE tradicional. A Tabela 2 (Seção 4) quantifica essas vantagens.

4. Protocolo FedAvg-FHEMK

Esta seção apresenta o FedAvg-FHEMK, esquema de FL que reconcilia segurança MK-FHE com eficiência prática. A ideia central é executar uma única vez um setup distribuído onde clientes colaboram para que o servidor compute uma chave agregada sem aprender chaves individuais. Após o setup, o treinamento ocorre com apenas uma rodada de comunicação por iteração: clientes cifram atualizações locais, servidor agrega e decifra apenas a soma. Ou seja, o protocolo se divide em duas fases: *Fase 1*, setup distribuído, executado uma vez, e *Fase 2*, treinamento iterativo, repetido a cada rodada.

O sistema possui $P+1$ participantes, sendo P clientes e um servidor. Assumimos que todos os participantes são *semi-honestos*.

Cada cliente u possui chave secreta local $sk_u \in R_q$. Em cada rodada de atualização do modelo, cada cliente calcula localmente uma mensagem $m_u \in R$ e o servidor deseja computar a agregação $m_{agg} = \sum_{i=1}^{P} m_i$ sem acesso às mensagens $m_1, ..., m_P$.

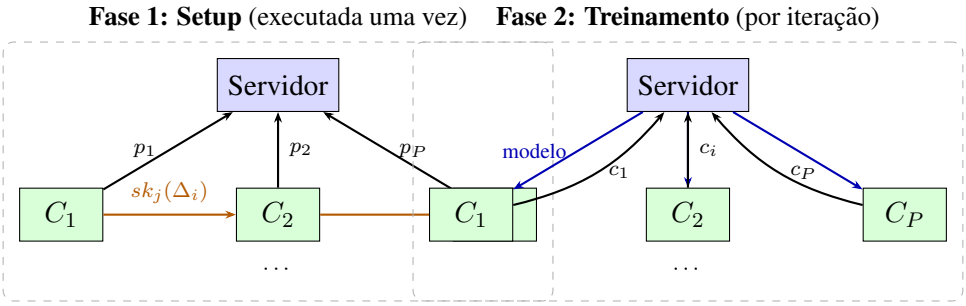

Figura 1. Visão geral do protocolo FedAvg-FHEMK. Fase 1 (esquerda): setup distribuído de chaves via NTT – clientes trocam avaliações polinomiais entre si (setas laranja) e enviam avaliações agregadas p_i ao servidor, que reconstrói sk_{agg}. Fase 2 (direita): cada iteração de treinamento – servidor distribui o modelo, clientes cifram suas atualizações locais c_i e enviam ao servidor, que agrega e decifra com sk_{agg}.

4.1. Fase 1: Setup Distribuído de Chaves

O desafio desta fase é permitir que o servidor obtenha $sk_{agg} = \sum_{i=1}^{P} sk_i$ sem aprender qualquer sk_i individual. A solução utiliza interpolação polinomial: um polinômio de grau $N-1$ é determinado por N avaliações em pontos distintos. Se cada cliente avalia sua chave em pontos compartilhados e os clientes agregam essas avaliações antes de enviar ao servidor, o servidor reconstrói apenas a soma das chaves (Shamir (SSS) [Mignotte 1982]). O protocolo procede:

1. **Geração local**: Cada cliente i amostra $sk_i \in R_q$.
2. **Avaliação via NTT**: Clientes computam $\vec{v_i} := NTT(sk_i) \in \mathbb{Z}_q^N$, obtendo N avaliações em raízes primitivas da unidade, ou seja, para um $\omega \in \mathbb{Z}_q$, temos $\vec{v_i}[j] = sk_i(\omega^j) \bmod q$.
3. **Agregação peer-to-peer**: As N avaliações são particionadas em P conjuntos disjuntos $\Delta_1, \ldots, \Delta_P$, de modo que cada conjunto contenha aproximadamente N/P pontos. Quando N/P não é um valor inteiro, a partição é realizada de forma balanceada, fazendo com que alguns conjuntos possuam $\lceil N/P \rceil$ elementos, enquanto os demais contenham $\lfloor N/P \rfloor$ elementos. Cada cliente i recebe, de todos os clientes j, as avaliações $sk_j(\delta)$ para todo $\delta \in \Delta_i$ e computa o valor agregado $p_i = \sum_{j=1}^{P} sk_j(\Delta_i)$, onde a notação $sk_j(\Delta_i)$ denota o vetor de avaliações $[sk_j(\delta_1), \ldots, sk_j(\delta_{|\Delta_i|})]$.
4. **Envio ao servidor**: Cada cliente envia p_i (vetor de n/P valores) ao servidor.
5. **Reconstrução**: Servidor aplica NTT inversa sobre os n pares (δ_k, p_k), recuperando $sk_{agg} = \sum_{i=1}^{P} sk_i$.

4.2. Fase 2: Treinamento Iterativo

Após o setup, cada iteração de treinamento executa sequencialmente criptografia, agregação e decifração em rodada única. O esquema baseia-se em cifras aditivas homomórficas: somar criptogramas resulta em um criptograma cifrando a soma das mensagens. O protocolo:

1. **Geração de CRS**: O String de Referência Comum (CRS, do inglês *Common Reference String*) é gerado de forma determinística e distribuída por todos os clientes

a partir de uma *seed* pública compartilhada. Cada cliente utiliza essa *seed* para amostrar localmente o mesmo elemento $CRS \leftarrow \mathcal{U}(R_q)$ segundo a distribuição uniforme. O CRS atua como uma "máscara pública" compartilhada entre todos os participantes.

2. **Criptografia**: Cada cliente i cifra sua atualização: $c_i = sk_i \cdot \text{CRS} + m_i + e_i$, onde $e_i \leftarrow \chi_{error}$ é ruído pequeno (essencial para segurança RLWE).
3. **Agregação**: Servidor recebe $c_1, \ldots, c_P$ e soma:

$$c = \sum_{i=1}^{P} c_i = \left(\sum_{i=1}^{P} sk_i\right) \cdot \text{CRS} + \left(\sum_{i=1}^{P} m_i\right) + \left(\sum_{i=1}^{P} e_i\right) = sk_{agg} \cdot \text{CRS} + m_{agg} + e_{agg}.$$

4. **Decifragem**: Servidor computa $\text{Dec}(c) = c - sk_{agg} \cdot \text{CRS} = m_{agg} + e_{agg}$. O erro agregado e_{agg} é pequeno (soma de P erros pequenos), mantendo precisão numérica.
5. **Atualização do modelo**: Servidor aplica m_{agg} ao modelo global e distribui aos clientes para a próxima rodada.

5. Análise de Complexidade

Esta seção sintetiza os custos computacionais e de comunicação do FedAvg-FHEMK. A Tabela 1 apresenta as operações dominantes em cada fase: todas envolvem multiplicações polinomiais aceleradas via NTT, com complexidade quasilinear $O(N \log N)$, enquanto as demais etapas consistem em somas modulares de custo linear. A Fase 1 apresenta custo de comunicação elevado ($O(P^2 \cdot N \cdot \log q)$) devido à troca peer-to-peer, porém essa sobrecarga ocorre apenas uma vez e é amortizada ao longo das iterações. A Tabela 2 compara o custo por iteração com esquemas tradicionais de MK-FHE, evidenciando a eliminação de rodadas adicionais de comunicação e da descriptografia colaborativa.

6. Garantias de Segurança

O protocolo proposto, FedAvg-FHEMK, oferece garantias formais de segurança supondo que o problema RLWE é difícil. Como a cifra homomórfica tem segurança contra ataques de texto claro escolhido (em inglês, IND-CPA), quando um participante vê um criptograma c_j enviado por um participante j, c_j é indistinguível de aleatório, já que a chave secreta correspondente sk_j só é conhecida por j. Logo, nenhum participante consegue descobrir informações sobre as mensagens dos outros atacando diretamente os criptogramas enviados em cada rodada.

A confidencialidade das chaves secretas individuais sk_i é garantida, pois usamos um protocolo de compartilhamento de segredos do tipo Shamir (SSS) [Mignotte 1982]. No final desse protocolo, o servidor conhece a apenas a chave agregada $sk_{\text{agg}} = \sum_{i=1}^{P} sk_i$, não obtendo informação sobre qualquer sk_i.

Se $P - 1$ clientes colaborarem com o servidor, compartilhando suas respectivas chaves secretas, digamos $sk_2, ..., sk_P$, o servidor consegue encontrar a chave faltante, digamos, sk_1, pois já possui sk_{agg}. Em outras palavras, o servidor pode calcular $sk_1 = sk_{\text{agg}} - \sum_{i=2}^{P} sk_i$. Isso quebra a segurança do protocolo, pois dá ao servidor o poder de decifrar as mensagens enviadas pelo usuário detentor de sk_1. No entanto, se houver uma colusão de no máximo $P - 2$ clientes, então há equação $sk_{\text{agg}} = \sum_{i=1}^{P} sk_i$ tem pelo

Tabela 1. Custos computacionais e de comunicação do protocolo FedAvg-FHEMK por fase.

Fase	**Participante**	**Operação / Conteúdo**	**Complexidade / Volume**
Custo computacional			
Fase 1	Cliente	Avaliação de $sk_i(x)$ via NTT	$O(N \log N)$
	Cliente	Agregação local de avaliações	$O(N)$
	Servidor	Interpolação de $sk_{\text{agg}}(x)$ via NTT	$O(N \log N)$
Fase 2	Cliente	Criptografia $sk_i \cdot CRS$	$O(N \log N)$
	Servidor	Soma homomórfica de P ciphertexts	$O(P \cdot N)$
	Servidor	Descriptografia com sk_{agg}	$O(N \log N)$
Custo de comunicação			
Fase 1	Cliente $\leftrightarrow$ Cliente	Avaliações parciais de $sk_i(\delta)$	$O(P^2 \cdot N \cdot \log q)$
	Cliente $\rightarrow$ Servidor	Avaliações agregadas p_i	$O(N \cdot \log q)$
Fase 2	Cliente $\rightarrow$ Servidor	Ciphertext $c_i \in \mathcal{R}_q^N$	$O(P \cdot N \cdot \log q)$ / iter.

Tabela 2. Comparação do custo por iteração entre o FedAvg-FHEMK e MK-FHE com descriptografia colaborativa.

Métrica	**FedAvg-FHEMK**	**MK-FHE**
Criptografia (Cliente)	$O(N \log N)$	$> O(N \log N)$
Agregação (Servidor)	$O(P \cdot N)$	$O(P \cdot N)$
Descriptografia	$O(N \log N)$ (centralizada)	$O(P \cdot N)$ + interação
Rodadas de Comunicação	1	2–3
Tolerância a Dropout	Não	Sim (threshold)

menos duas variáveis livres, logo, infinitas soluções, portanto, o sigilo das chaves faltantes se mantém independentemente do poder computacional dos participantes, i.e., se trata de segurança baseada em teoria da informação (este é basicamente o mesmo argumento de segurança do SSS).

Limitações. O limiar $P - 2$ é estrutural e deriva do SSS com reconstrução da soma. A extensão para thresholds ajustáveis e a proteção contra adversários ativos (e.g., *model poisoning*) são discutidas na Seção 8.

7. Avaliação Experimental

Esta seção apresenta a avaliação experimental do FedAvg-FHEMK em cenários de aprendizado federado com diferentes níveis de complexidade e distribuição de dados.

7.1. Configuração Experimental

Os experimentos foram conduzidos utilizando os datasets MNIST (dígitos manuscritos 28×28) e CIFAR-10 (imagens coloridas 32×32×3). Para MNIST, empregou-se uma rede Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*) com 2 camadas ocultas; para CIFAR-10, utilizou-se ResNet-20. Foram avaliados três cenários: MNIST, CIFAR-10 Independente e Identicamente Distribuído (IID, do inglês *Independent and Identically Distributed*) (distribuição uniforme) e CIFAR-10 Non-IID (heterogeneidade de dados entre clientes). Os baselines incluem: Média Federada (FedAvg, do inglês *Federated Averaging*) em texto claro [McMahan et al. 2017], FHE de chave única [Sav et al. 2020], e M-FHE de chave única [Hu and Li 2024] (criptografia seletiva com 10%, 20%, 40% e 80% dos parâmetros protegidos). As métricas coletadas foram acurácia final, tempo médio por rodada e custo de comunicação por rodada.

Todos os experimentos foram realizados com $P = 10$ clientes e 20 rodadas de treinamento, com 1 época local por rodada. Os parâmetros criptográficos do FedAvg-FHEMK são: grau polinomial N definido como a próxima potência de dois acima do tamanho do modelo (e.g., $N = 2^{17} = 131072$ para o MNIST/MLP; $N = 2^{19} = 524288$ para o CIFAR-10/ResNet-20), e módulo $q = 4179340454199820289$ (primo de 61 bits). O código dos experimentos está disponível em `https://github.com/Sr-Souza-dev/FedAvg-FHEMK.git`.

7.2. Resultados no MNIST

A Tabela 3 apresenta os resultados completos para o dataset MNIST. O tempo de treino local varia entre 3,2s e 3,6s dependendo do método, refletindo diferenças na manipulação dos pesos antes e após o treinamento. A introdução de mecanismos criptográficos não afeta o custo computacional do aprendizado local propriamente dito.

Tabela 3. Resultados no dataset MNIST com $P = 10$ clientes, $N = 131072$, $q = 4{,}18 \times 10^{18}$ (61 bits).

Método	Acurácia	Tempo Cliente (s)			Tempo Servidor (s)		Comunicação
		Treino	Encrypt	Decrypt	Agregação	Decrypt	(MB)
FedAvg (baseline)	0.9411	3.63	0.00	0.00	0.01	0.00	0.17
FHE single-key	0.9433	3.62	0.08	0.02	0.28	0.00	6.00
M-FHE 10%	0.9392	3.21	0.02	0.01	0.11	0.00	1.15
M-FHE 20%	0.9415	3.38	0.03	0.01	0.11	0.00	2.14
M-FHE 40%	0.9352	3.36	0.05	0.02	0.17	0.00	3.10
M-FHE 80%	0.9416	3.52	0.08	0.03	0.27	0.00	5.03
FedAvg-FHEMK	**0.9407**	**3.22**	**0.91**	**0.00**	**0.54**	**0.89**	**1.00**

O FedAvg-FHEMK apresenta overhead concentrado na cifragem do cliente (0,91s), eliminando completamente a descriptografia no lado do cliente. Esse comportamento reflete a proposta central do protocolo, que remove a dependência de interações adicionais dos clientes após o envio das atualizações cifradas. No servidor, a agregação (0,54s) e descriptografia final (0,89s) são realizadas de forma autônoma, redistribuindo o custo computacional de maneira estrutural. O custo de setup (Fase 1) é de apenas 0,05s, amortizado ao longo das rodadas. O custo de comunicação é reduzido para 1,00 MB por rodada, economia de 83% frente ao FHE single-key (6,00 MB), com acurácia equivalente ao baseline (0,9407 vs. 0,9411), evidenciando que o ruído criptográfico não compromete a convergência do modelo.

7.3. Resultados no CIFAR-10 IID

A Tabela 4 apresenta os resultados para o cenário CIFAR-10 IID, caracterizado por um modelo de maior profundidade e custo computacional. Nesse contexto, o tempo de treino local varia entre 16s e 17s por rodada entre os diferentes métodos.

Tabela 4. Resultados no dataset CIFAR-10 IID com $P = 10$ clientes, $N = 524288$, $q = 4{,}18 \times 10^{18}$ (61 bits).

Método	Acurácia	Tempo Cliente (s)			Tempo Servidor (s)		Comunicação
		Treino	Encrypt	Decrypt	Agregação	Decrypt	(MB)
FedAvg (baseline)	0.7004	17.12	0.00	0.00	0.07	0.00	1.05
FHE single-key	0.6944	17.30	0.48	0.17	1.81	0.00	34.00
M-FHE 10%	0.7224	17.05	0.07	0.01	0.24	0.00	4.94
M-FHE 20%	0.7214	16.74	0.09	0.02	0.32	0.00	7.84
M-FHE 40%	0.6958	16.43	0.26	0.04	0.73	0.00	14.63
M-FHE 80%	0.7000	16.65	0.33	0.09	1.33	0.00	27.21
FedAvg-FHEMK	**0.6962**	**17.19**	**4.78**	**0.00**	**3.43**	**4.19**	**8.00**

O cenário CIFAR-10 IID confirma as tendências observadas no MNIST: o overhead concentra-se na cifragem do cliente (4,78s) e na descriptografia do servidor (4,19s), enquanto a comunicação é reduzida em 76% frente ao FHE single-key (8,00 vs. 34,00 MB). O FedAvg-FHEMK elimina completamente a etapa de descriptografia no cliente, consistente com a proposta de remover interações pós-envio. A acurácia (0,6962) permanece dentro de 1% do baseline em claro (0,7004), confirmando que o ruído criptográfico não compromete a convergência mesmo em modelos mais profundos. O custo de setup (0,29s) é mais elevado que no MNIST em razão do maior grau polinomial N, mas continua amortizável ao longo das rodadas.

7.4. Análise de Escalabilidade

Para avaliar o impacto do número de clientes no desempenho do protocolo, executamos o FedAvg-FHEMK e o FedAvg no dataset MNIST com $P \in \{5, 10, 20\}$ clientes, mantendo os demais parâmetros fixos (20 rodadas, 1 época local). A Tabela 5 resume os resultados.

Tabela 5. Análise de escalabilidade do FedAvg-FHEMK no MNIST: impacto do número de clientes P sobre acurácia, tempo por rodada, comunicação e custo de setup (Fase 1).

P	Acurácia	Tempo/rodada (s)	Comunicação (MB)	Setup (s)
5	0.9633	9.18	1.00	0.05
10	0.9407	7.76	1.00	0.05
20	0.8966	6.52	1.00	0.06

O custo de comunicação permanece constante (1,00 MB) independentemente de P, pois cada cliente envia um único ciphertext de tamanho fixo determinado pelo grau polinomial N. O setup cresce de forma marginal (0,05s para 0,06s), confirmando a complexidade quasilinear. A redução de acurácia com o aumento de P (de 96,33% com 5 clientes para 89,66% com 20) é esperada pela fragmentação dos dados e também observada no FedAvg sob as mesmas condições.

7.5. Resultados no CIFAR-10 Non-IID e Discussão

No cenário CIFAR-10 Non-IID, caracterizado por heterogeneidade na distribuição de dados entre os clientes, o FedAvg-FHEMK mantém acurácia dentro de 1% do baseline e custos de comunicação idênticos ao cenário IID, confirmando que o overhead é determinado pelas operações criptográficas e não pela distribuição dos dados. O custo de comunicação permanece estável e significativamente inferior ao do FHE single-key, o que é particularmente relevante em cenários Non-IID, onde o número de rodadas tende a ser maior.

De forma geral, os resultados demonstram que o FedAvg-FHEMK preserva a utilidade do modelo com perdas de acurácia inferiores a 1%, reduz comunicação em 76–83% frente a FHE single-key e concentra o custo computacional no servidor, eliminando descriptografia no cliente.

8. Conclusão

Este trabalho apresentou o FedAvg-FHEMK, protocolo de FL que reconcilia garantias criptográficas rigorosas (MK-FHE) com custos competitivos a métodos de proteção parcial. As contribuições principais são: (i) setup distribuído via interpolação polinomial com NTT ($O(N \log N)$ por cliente), executado uma única vez; (ii) protocolo de rodada única por iteração, reduzindo comunicação em 76%–83% vs. FHE de chave única; (iii) garantias formais IND-CPA sob RLWE, com resistência a colusões de até $P - 2$ clientes.

Resultados experimentais: Avaliações em MNIST (MLP) e CIFAR-10 (ResNet-20), sob distribuições IID e Non-IID, confirmam robustez do protocolo. No MNIST, o FedAvg-FHEMK alcança 94,07% de acurácia (baseline: 94,11%) com 1,00 MB/rodada (vs. 6,00 MB em FHE single-key, redução de 83%). No CIFAR-10 IID, atinge 69,62% (baseline: 70,04%) com 8,00 MB/rodada (vs. 34,00 MB, redução de 76%). A análise de escalabilidade com $P \in \{5, 10, 20\}$ clientes demonstra custo de comunicação constante e setup de crescimento marginal.

Limitações e aplicabilidade: O setup inicial é custoso, tornando o FedAvg-FHEMK mais adequado para cenários cross-silo de longa duração onde o custo se amortiza. A ausência de tolerância a dropout requer estabilidade dos participantes. O protocolo garante confidencialidade, mas não aborda integridade dos gradientes: adversários maliciosos podem enviar atualizações envenenadas (*model poisoning*) sem violar as garantias criptográficas. A integração com mecanismos de detecção de anomalias ou Byzantine-robustness é requisito para aplicações em ambientes adversariais.

Direções futuras: (1) **Federações dinâmicas**: threshold cryptography e setup incremental para tolerância a dropout e entrada/saída de clientes sem reexecutar setup completo; (2) **Segurança contra adversários ativos**: extensão para modelo malicioso com provas de conhecimento zero e Byzantine-fault tolerance; (3) **Escalabilidade**: avaliação em federações maiores ($P > 20$) com análise do impacto nos custos; (4) **Aceleração**: exploração de aceleradores criptográficos (GPUs, FPGAs) para redução do overhead temporal.

O FedAvg-FHEMK demonstra que MK-FHE pode ser prática quando co-desenhada com o fluxo de treinamento, viabilizando FL seguro em domínios com requisitos rigorosos de conformidade.

Referências

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. (2020). Secure single-server aggregation with sublinear overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1253–1269.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.

Chen, H., Dai, W., Kim, M., and Song, Y. (2019). Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 395–412.

Correia, P., Silva, I., Amorim, I., Maia, E., et al. (2025). Federated learning: An approach with hybrid homomorphic encryption. *arXiv preprint arXiv:2509.03427*.

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

Hu, C. and Li, B. (2024). Maskcrypt: Federated learning with selective homomorphic encryption. *IEEE Transactions on Dependable and Secure Computing*, 22(1):221–233.

Jin, W., Yao, Y., Han, S., Joe-Wong, C., Ravi, S., Avestimehr, S., and He, C. (2024). Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system. *arXiv preprint arXiv:2303.10837*.

Korkmaz, A. and Rao, P. (2025). A selective homomorphic encryption approach for faster privacy-preserving federated learning. *arXiv preprint arXiv:2501.12911*.

López-Alt, A., Tromer, E., and Vaikuntanathan, V. (2012). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Mignotte, M. (1982). How to share a secret. In *Workshop on cryptography*, pages 371–375. Springer.

Mouchet, C., Troncoso-Pastoriza, J., Bossuat, J.-P., and Hubaux, J.-P. (2021). Multiparty homomorphic encryption from ring-learning-with-errors. *Proceedings on Privacy Enhancing Technologies*, 2021(4):291–311.

Omar, A. A., Yang, X., Choo, E., and Ardakanian, O. (2025). Efficient privacy-preserving cross-silo federated learning with multi-key homomorphic encryption. *arXiv preprint arXiv:2505.14797*.

Sav, S., Pyrgelis, A., Troncoso-Pastoriza, J. R., Froelicher, D., Bossuat, J.-P., Sousa, J. S., and Hubaux, J.-P. (2020). Poseidon: Privacy-preserving federated neural network learning. *arXiv preprint arXiv:2009.00349*.

Wu, R., Chen, X., Guo, C., and Weinberger, K. Q. (2023). Learning to invert: Simple adaptive attacks for gradient inversion in federated learning. In *Uncertainty in Artificial Intelligence*, pages 2293–2303. PMLR.

Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., and Liu, Y. (2020). BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In *2020 USENIX annual technical conference (USENIX ATC 20)*, pages 493–506.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.