

EncCluster: Scalable Functional Encryption in Federated Learning through Weight Clustering and Probabilistic Filters

Vasileios Tsouvalas · Samaneh Mohammadi · Ali Balador · Tanir Ozcelebi · Francesco Flammini · Nirvana Meratnia

Eindhoven University of Technology, The Netherlands | RISE Research Institutes of Sweden, Sweden | Mälardalen University, Sweden

Pervasive and Mobile Computing, 2025

01 Motivation

The problem

Secure federated learning (FL) must hide individual model updates without making communication and client computation impractical. Encrypting every weight is costly, especially as neural networks grow.

Contributions

- **Compact encryption:** Combine weight clustering with decentralized multi-client functional encryption (DMCFE).
- **Mapping compression:** Encode weight-to-centroid assignments with Binary Fuse (BF) filters.
- **Measured trade-offs:** Evaluate accuracy, uploads and encryption time across four architectures.

Threat model

An honest-but-curious (HbC) server aggregates updates, while an HbC third-party authority (TPA) initializes the cryptosystem; both may seek private client information. Clients may collude by sharing their own weights. Secure channels, protected key provisioning and stable connections are assumed.

Protection boundary

DMCFE provides aggregate-only decryption under the stated key assumptions. The server recovers position mappings, so these are not secret from it; BF filters do not inherit cryptographic guarantees. Poisoning and denial-of-service are outside the analysis.

Scope

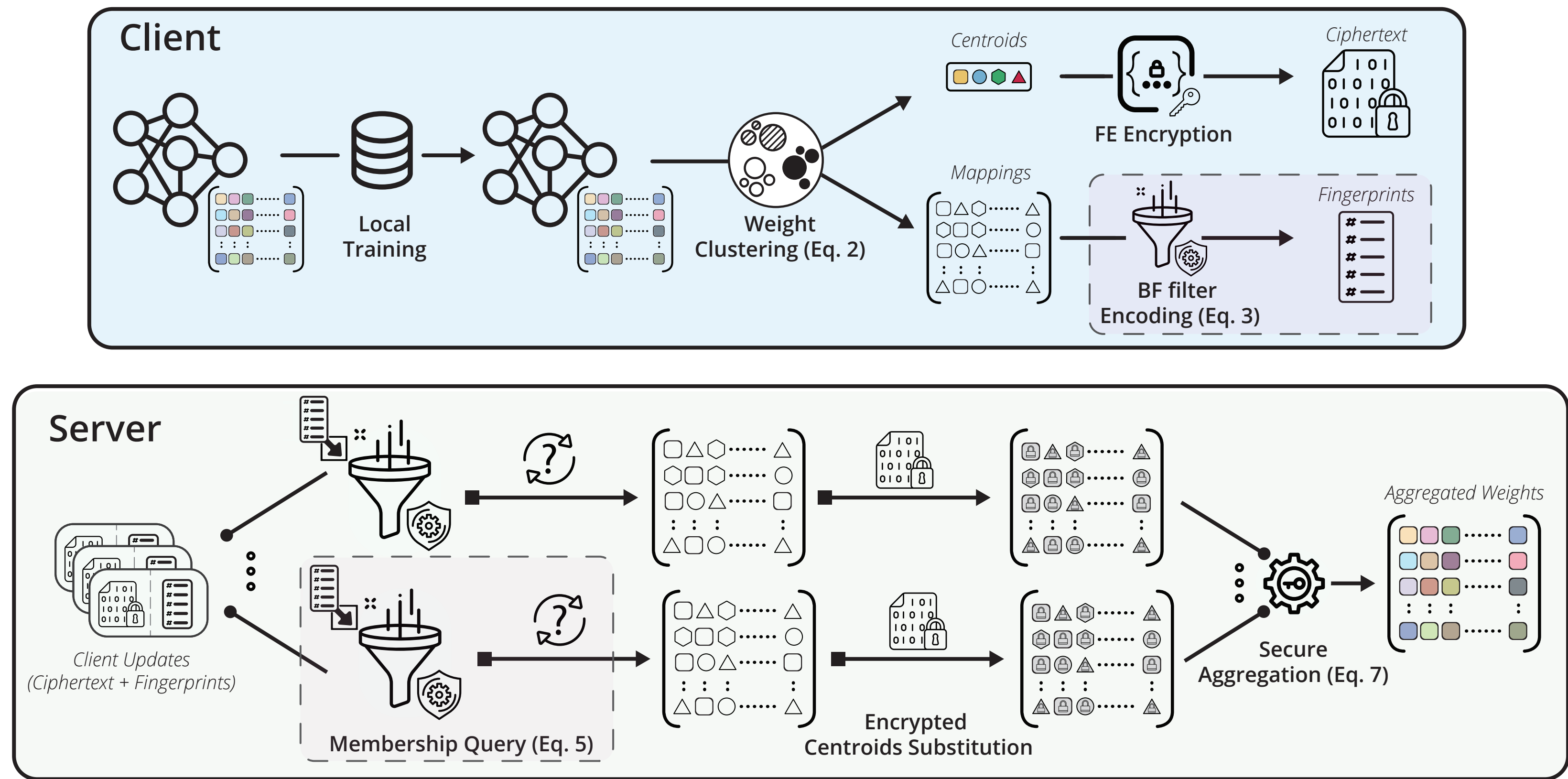
Timings come from a cluster simulation and are hardware-specific; repeated-run uncertainty is unreported. Filter errors do not establish differential privacy.

Notation

Dirichlet distribution with concentration a ($\text{Dir}(a)$) · independent and identically distributed (IID) · functional encryption (FE) · percentage points (pp)

02 EncCluster

Encrypt centroids instead of every model weight



Technical core

$$\mathcal{L}_{\text{cluster}} = \sum_{i=1}^d (\theta_i - z_{a_i})^2$$

θ_i : weight; z : centroids; a_i : assigned cluster. Approximate many weights with a few shared values before encryption.

$$\theta_j^{r+1} = \frac{1}{|D|} \text{Dec}(\{\hat{\theta}_{n,j}^r\}_n, \text{dk}_f)$$

Encrypted weights are preweighted by client sample count. Divide the decrypted sum by total samples $|D|$; dk_f authorizes aggregation.



Paper



Code

03 Evaluation

0.284×

Upload volume / FedAvg

CIFAR-10 · full participation

2.04 s

Client encryption time

A10 / 96-core cluster simulation

Lower encrypted communication and computation

Method	IID Δ (pp)	Non-IID Δ (pp)	Upload / FedAvg	Encrypt (s)
FedAvg + clustering	-0.12	-0.71	0.034	—
DeTrust-FL	-0.09	-0.10	3.743	329.53
SEFL	-0.98	-1.74	2.581	7.64
EncCluster (Ours)	-0.32	-0.79	0.284	2.04

CIFAR-10 · ResNet-20 · 30 clients · full participation · 100 rounds. $\text{Dir}(10)/\text{Dir}(0.1)$, 128 centroids, 256-bit keys. Deltas vs. FedAvg; timings from an A10 / 96-core cluster simulation.

Architecture coverage

Architecture	C10 Δ (pp)	C100 Δ (pp)	Encrypt (s)
ConvNeXt-Tiny	-0.36	-1.29	2.02
ResNet-20	-0.32	-1.19	2.03
MobileNet	-0.24	-0.97	2.08
ConvMixer-256/8	-0.19	-1.02	2.12

IID CIFAR-10/100 · 10 clients · full participation · 100 rounds. Accuracy changes use each architecture's FedAvg baseline; same cluster hardware.

Takeaway

EncCluster reduces uploads to **0.284× FedAvg** with a **0.32-point IID accuracy decrease** on CIFAR-10. Client encryption takes **2.04 s** in the A10 / 96-core cluster simulation, versus 7.64 s for SEFL.