le cnam cedric EA4629

Verifiable and Privacy-Preserving Federated Learning through Differential Privacy and Cryptographic Protocols

Rezak AZIZ

Supervisors:

Pr. Samia BOUZEFRANE, Pr. Youakim BADR, Pr. Pierre PARADINAS

Outline

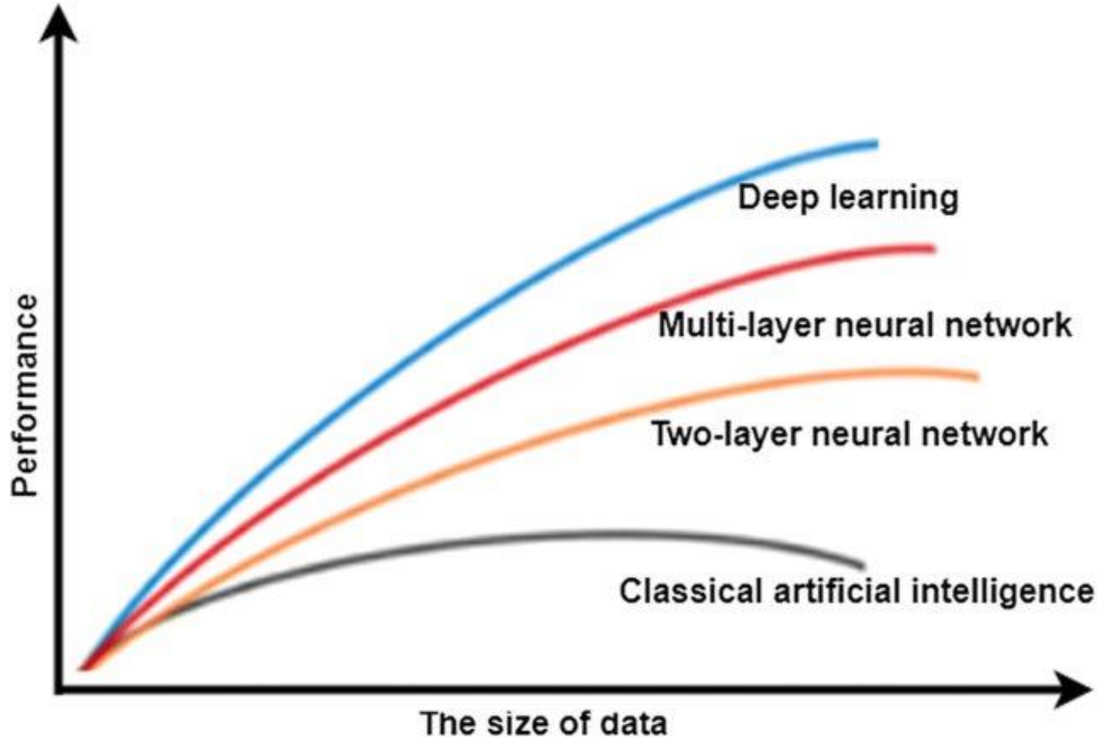
I. Introduction

Understanding the need for verifiable and privacy-preserving Federated Learning

Institutions need better AI…

- Healthcare **(187B $ by 2030)**[1]
  - medical imaging, diagnosis
- Finance **(143.56B $ by 2030)**[2]
  - fraud detection, risk scoring
- Public services
  - traffic, energy, security optimization

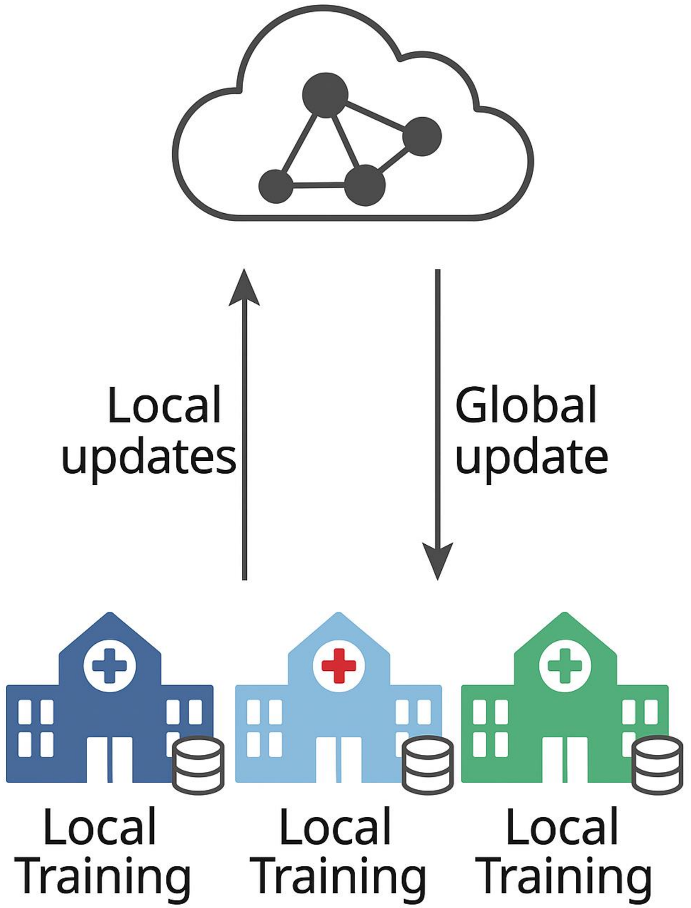
- **GDPR enforced since 2018**
  - **Data Sharing is regulated.**
  - **Max penalty[3]: up to 4% of global revenue.**



Data is DISTRIBUTED and SENSITIVE

1 https://finance.yahoo.com/news/ai-healthcare-market-revenue-worth-152500085.html
2 https://www.opentext.com/media/report/state-of-ai-in-banking-digital-banking-report-en.pdf
3 https://gdpr-info.eu/issues/fines-penalties/

Federated Learning



- **Data stays local** at each client
- **Only model updates are shared** with the server
- Server Aggregates the updates and send the global update for another iteration

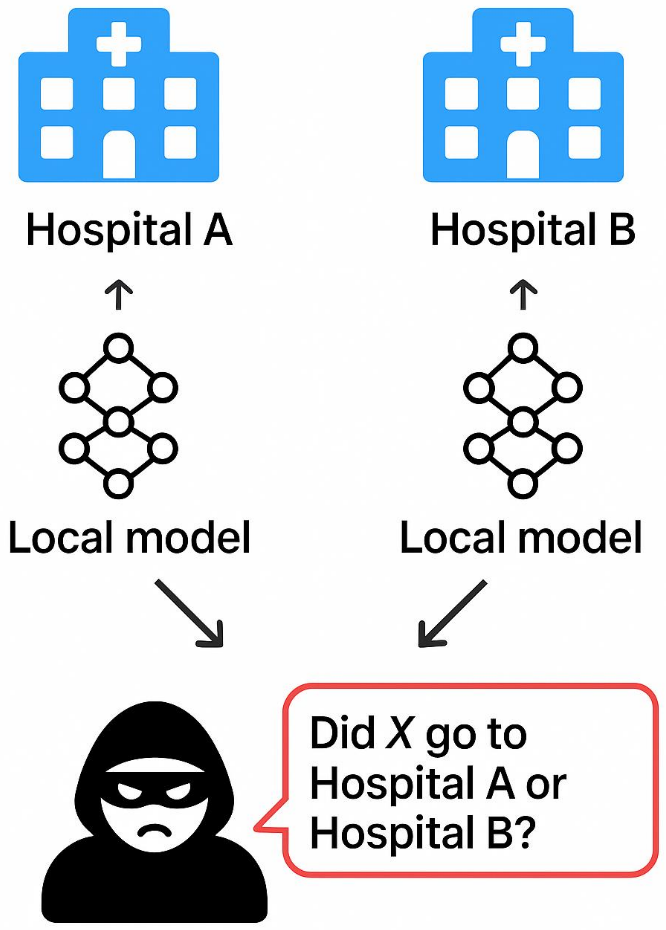
Collaborative training with reduced data exposure.

Data ≠ information

Detect if a person's data was used

- Membership Inference Attack.
- No need for raw data — weights are enough.

- Federated Learning is White box.
- Honest-but-curious server assumption
  - SPOF!
- Iterative communication
  - privacy loss accumulates over rounds



Conflicts with GDPR and Problem Definition

- Art. 5(1) - leakage prevention.
- Art. 25 - Privacy by design.
- Art. 32 - Continuous security of processing.
- Art. 5(2), 24, 30, Recital 74 - Accountability & verifiable compliance.

Can we design a Federated Learning system that is:

Continuously Private

Trust-minimized and Verifiable

II. State Of the Art

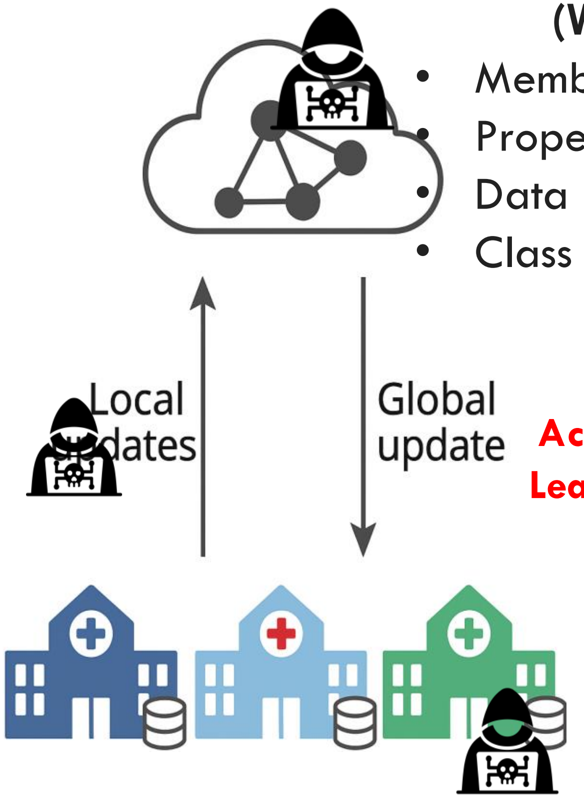
Attacks, Causes, Countermeasures

Where Privacy Leakage Comes From

- **Memorization**
  - Model Capacity
  - Overfitting
- **Regularization**
  - No Formal Privacy Guarantee

Server Attacks (Whitebox)
- Membership Inference
- Property Inference
- Data Reconstruction
- Class Representative

Auxiliary Attacks
- Poisoning
- Backdoors



Accumulated Leakage over round

Client Attacks (Blackbox)
- Membership Inference
- Property Inference
- Data Reconstruction
- Class Representative

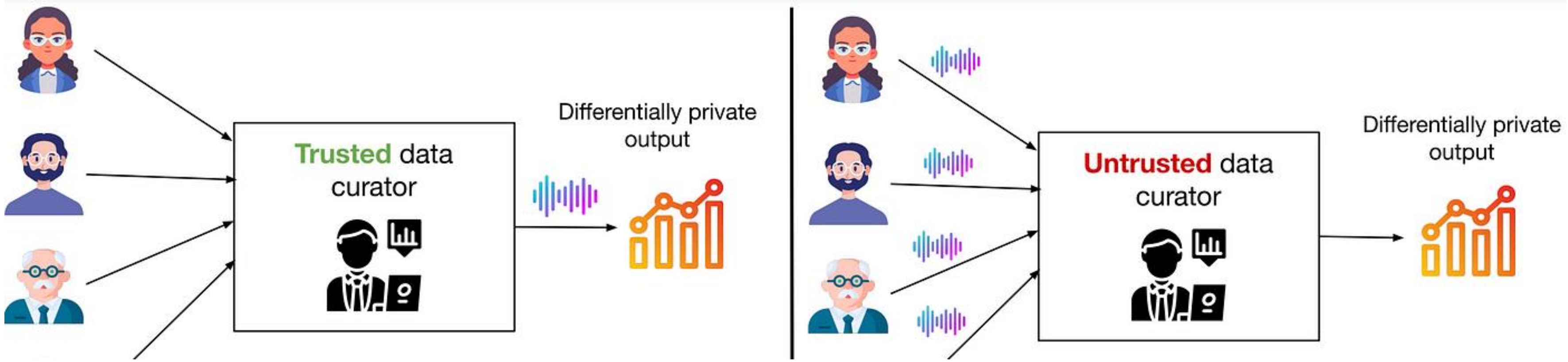
Privacy Enhancing Techniques

Technique	Protects Global Update	Protects Local Updates	Server Trust Assumptions
Secure Aggregation	✗	✓	Honest but Curious
Secret Sharing (SS)	✗	✓	At least one honest
HE	✓(if not decrypted)	✓	Honest but Curious
Central DP	✓	✗	Trusted
Local DP	✓	✓	Untrusted

Definition. *A randomized mechanism $\mathcal{M}$ satisfies ε-* differential privacy *if for all pairs of neighboring datasets D and D' differing in one individual, and for all measurable sets of outputs S, it holds that*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S].$$

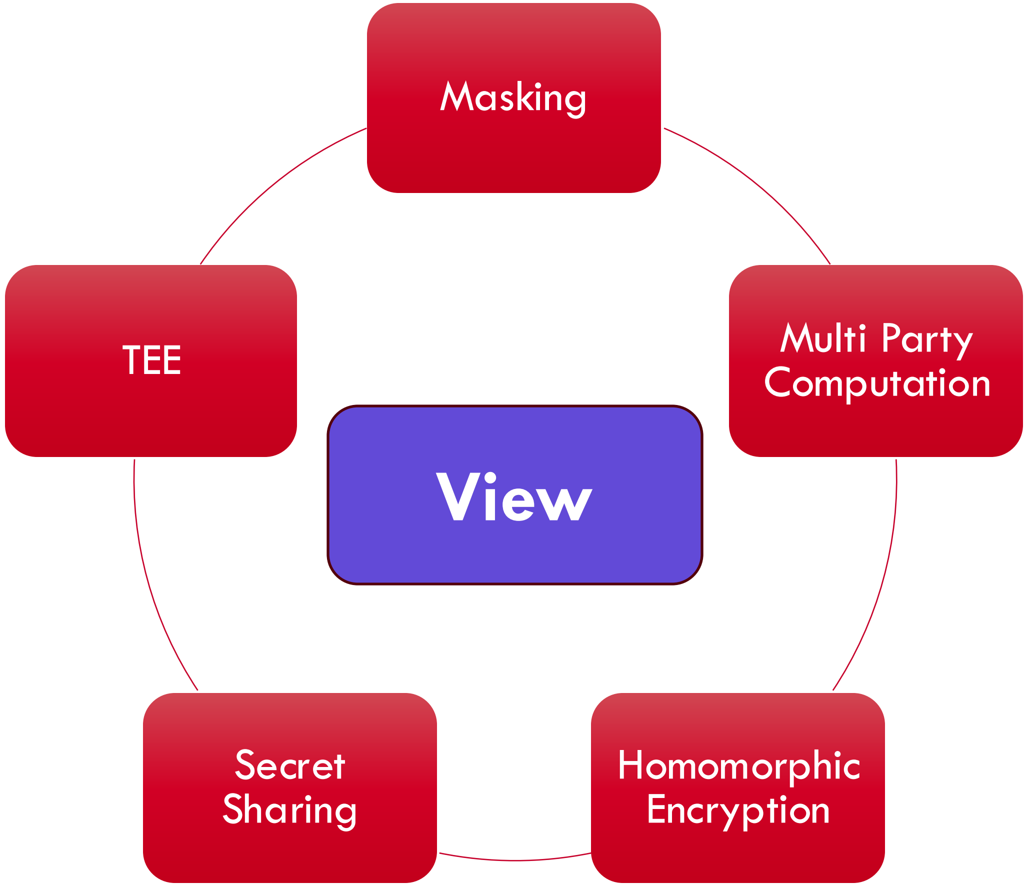
Central vs Local Differential Privacy

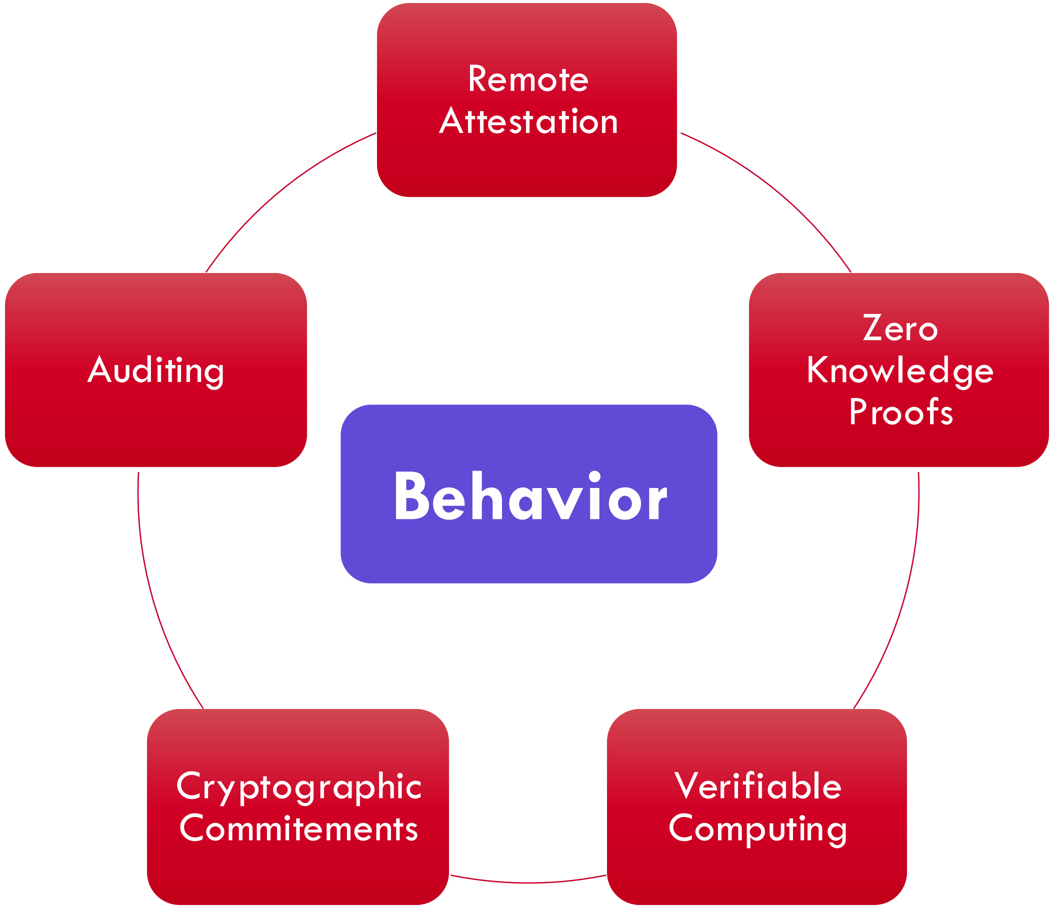


	Central DP	Local DP
Utility	**High**	Low
Security	Low	**High**

- **We need both utility and Security.**
- **CDP shifts the problem from privacy to trust.**

Trust in Federated Learning





Takeaways From the SOTA

- Partial solutions:
  - Secure aggregation/secret sharing : protect updates, not the final model
  - Differential Privacy : formal guarantees, but trust (CDP) or utility loss (LDP)

- Core limitation:
  - Honesty is assumed, not verifiable.

III. Contributions

How to design a unified framework for verifiable and privacy preserving federated learning ?

1
View Trust Reduction
Update Hiding

WISTP'2024 (Published)

2
Behavior Trust Reduction
Verification Protocol for DP

AINA'2025 (Published)

3
Full Trust Reduction
Verification Protocol for PPFL

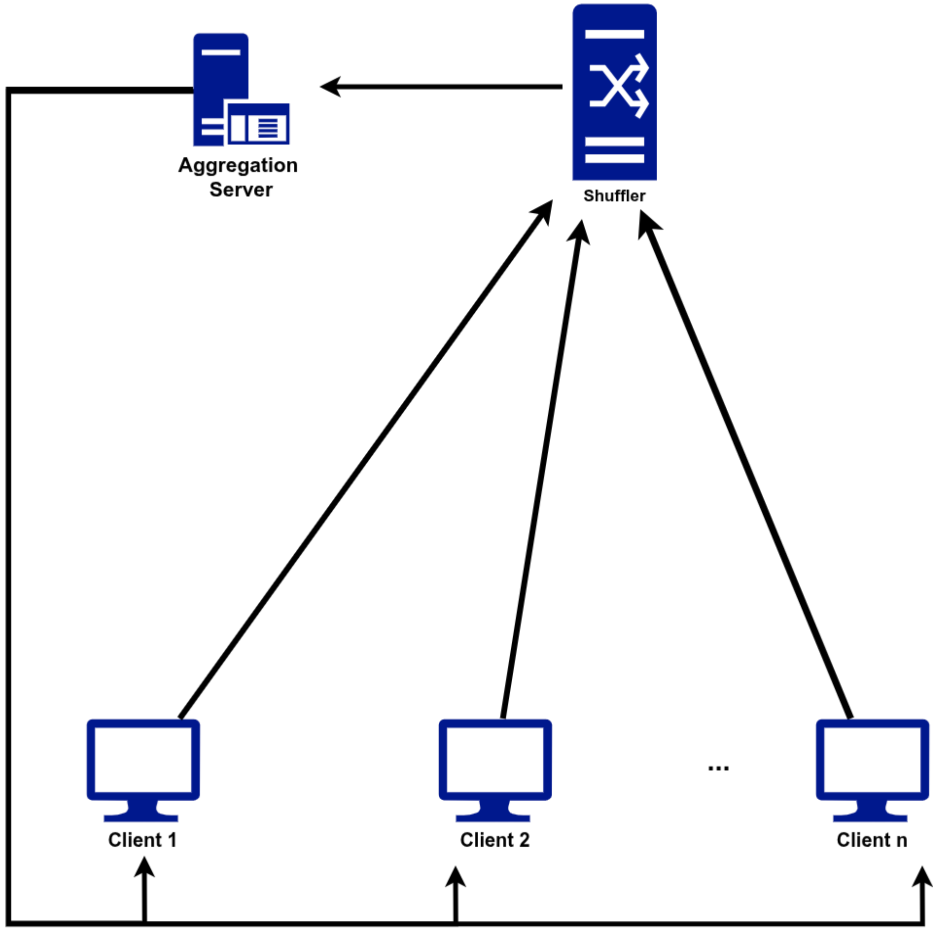
JISA (Under Review)

III-A. Reducing Trust of View

Prevent any entity access to unprotected updates.

Proposed DP Enabled FL Architecture

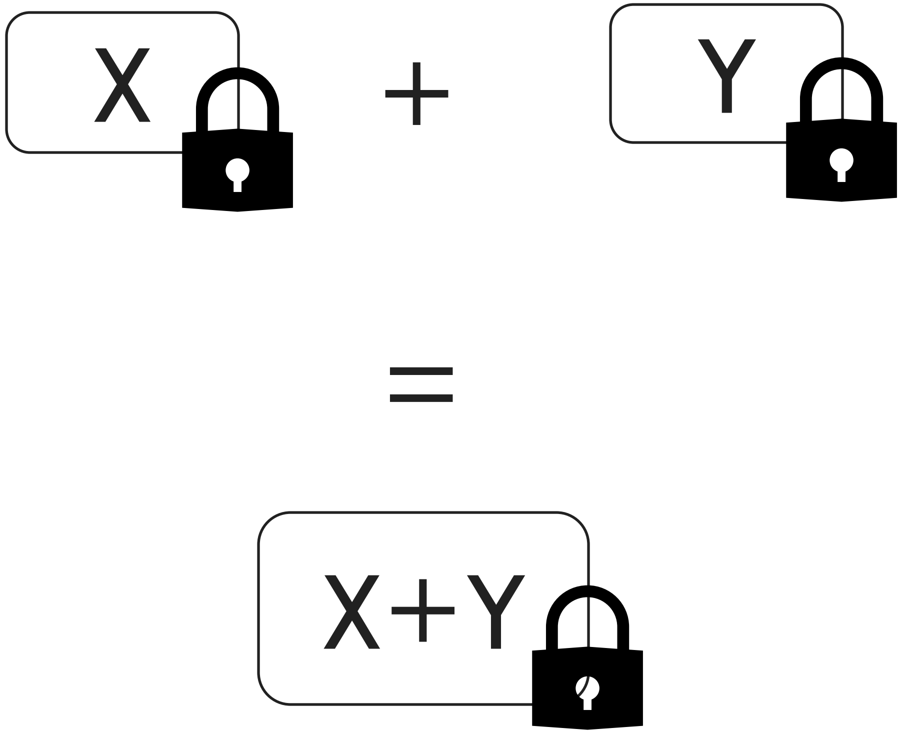
- Introduction of a third entity
- Role:
  - Add DP noise
  - Anonymize client updates
- Objective:
  - Server never sees raw updates
- Key requirement:
  - **Do not shift trust!!**
  - Reduce trust of view for all entities



Homomorphic Encryption

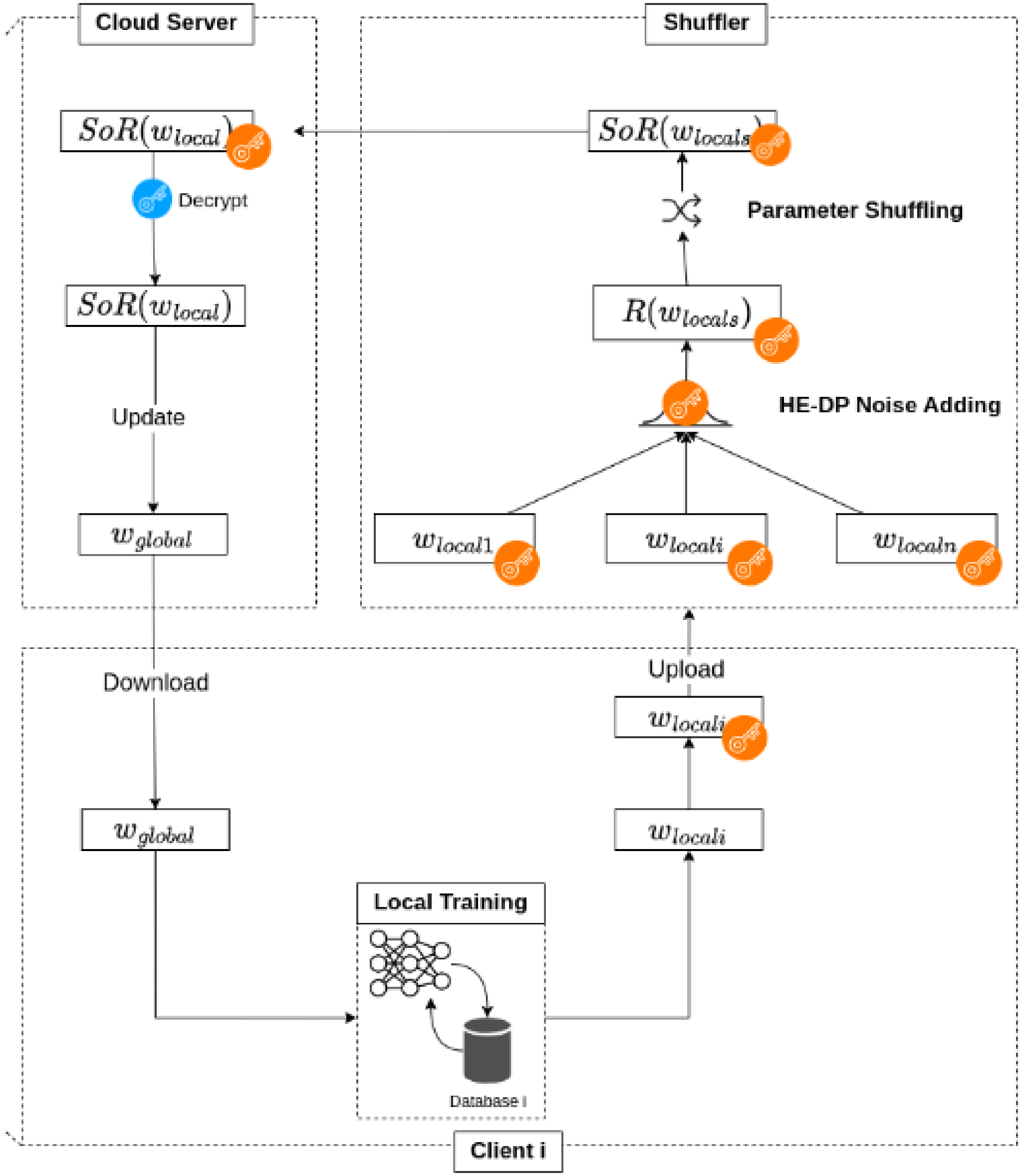
- Perform computation over encrypted data.
- The result will be also encrypted.
- Only authorized part can decrypt the result.

- Computation Overhead Not Suitable.
- Limit the homomorphic to Noise Addition
- **Noise is Independent => Highly parallelizable.**



Proposed Workflow

- Shuffler:
  - No decryption key
  - Assumed to follow the protocol
- Server:
  - Sees only shuffled & DP-protected updates
  - Updates are uninformative individually
- Non-collusion assumption:
  - Server and shuffler do not collude
  - ⇒ No single entity can access raw updates



Complexity Evaluation

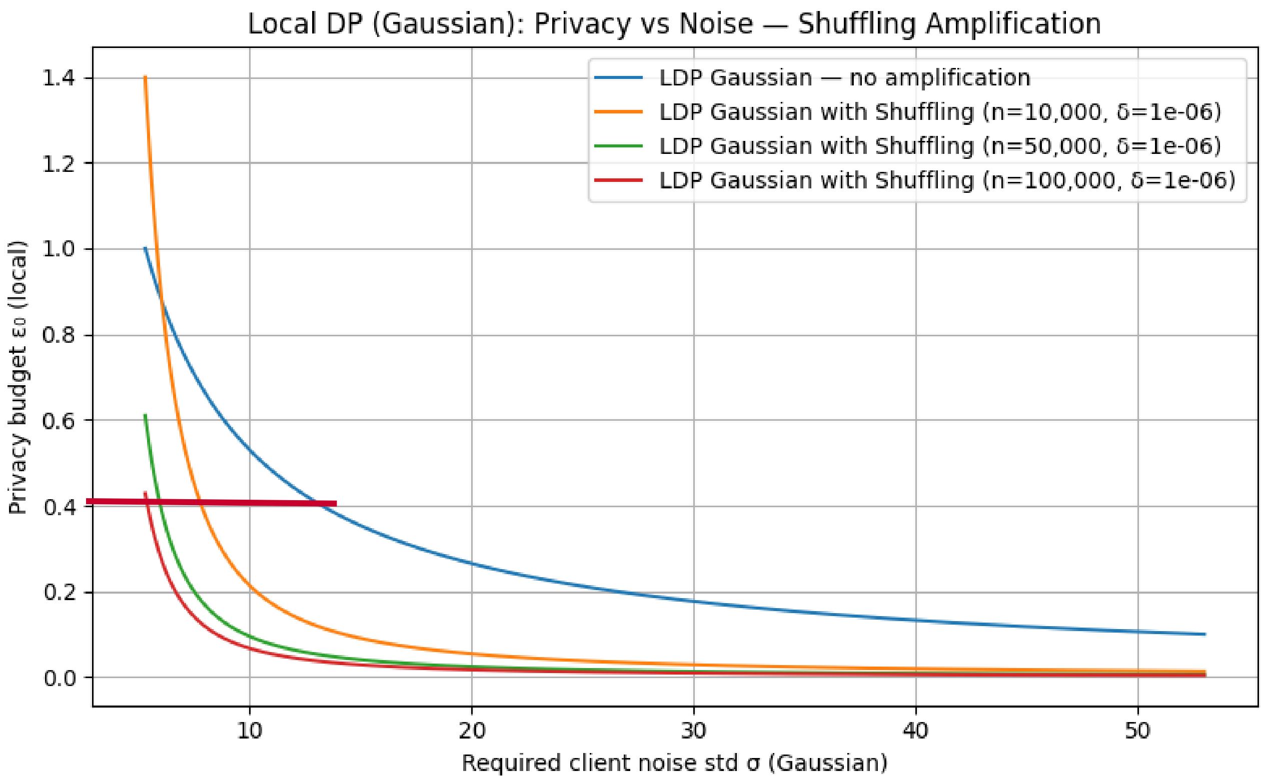
Per-round communication overhead of different schemes.

Scheme	**Communication Overhead**	**Complexity**
Plaintext	$n \times P \times 4$ bytes	$O(nP)$
Secure Aggregation	$n \times P \times 4 + n(n-1) \times 32$ bytes	$O(nP) + O(n^2)$
HE-2048 (Ours)	$n \times P \times (512 + 4)$ bytes	$O(nP)$

- Same asymptotic complexity as plain-text computation.
  - Do not mean same execution time.

Key Findings : Amplification By Shuffling

- General Theorem:
  - Require shuffling before randomization.
- Not applicable in practice:
  - Require sharing raw updates.
  - Contradict the guarantees local DP.



- Our results:
  - Use the general theorem
  - Allow shuffling before randomization.

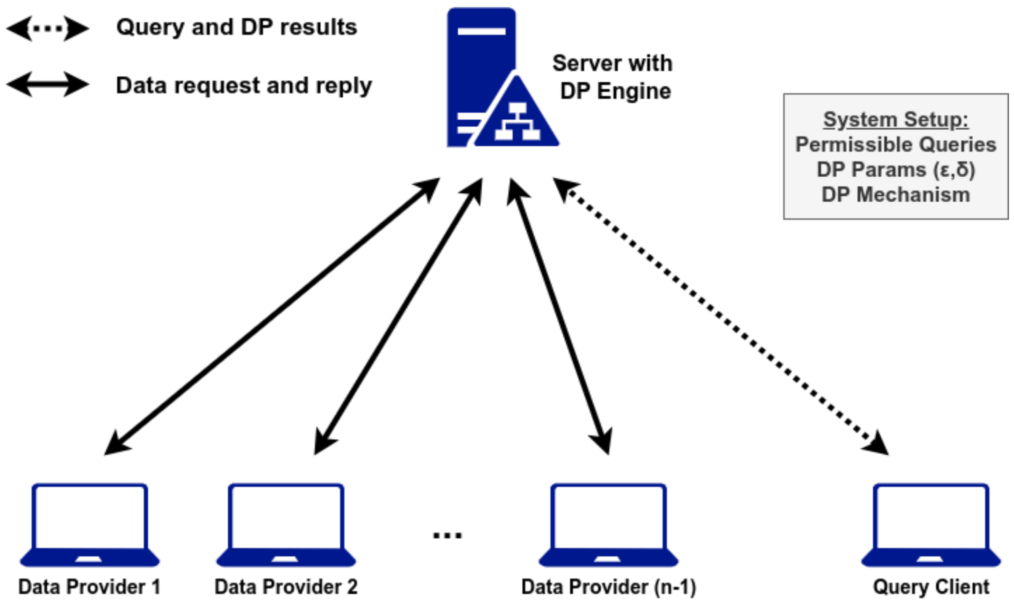
The General Theorem becomes valid under realistic conditions

III-B. Reducing trust of Behavior

How to enforce correct behavior of the server adding Differential Privacy Noise.

Problem Definition

Server claim : "I am Applying DP in your data"



- No control over DP application.
- No verifiability of server behavior
  - DP can be silently bypassed

Privacy is a claim, not a guarantee

- How to verify randomness.
- The verification should be done in without leaking information.

=> Zero Knowledge Proof

How can we guarantee that DP is correctly applied, without relying on trust in the server?

Zero Knowledge Proofs

Let L be an NP language with a witness relation R.

A *Zero-Knowledge Proof system* for (L, R) involves a prover $P(x, w)$ producing a proof π and verifier $V(x, \pi)$ such that:

- **Completeness.** If $x \in L \implies$ the verifier accepts the proof π.
- **Soundness.** If $x \notin L \implies$ the verifier reject the proof π.
- **Zero-Knowledge.** The verifier learns nothing beyond the fact that x is true.

ZKP Constraints :

- **operate over finite fields with modular constraints**

Noise Generation Constraints:

- **involve arithmetic over continuous domains**
- **Involve Complex Operations (log and exp)**

Noise Generation in Finite Fields

- Constraints
  - Mechanism must satisfy ε-DP
  - Avoid complex operations.
- Discrete Laplace Distribution

$$f_p(k) = \mathbb{P}(Y = k) = \frac{1-p}{1+p}\, p^{|k|},$$

$$k \in \mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}.$$

- Why Discrete Laplace?
  - Satisfies ε-DP
  - Generated as $X = G_1 - G_2$, where G_i is Geom(p)
  - No exp/log → lightweight computation

Shift from generating Laplace Distribution → generating geometric Distribution

Noise Sampling Distribution

Algorithm 1 Laplace Noise Generation

Require: b (scale parameter of the Laplace distribution)
Ensure: η (Laplace-distributed noise)

Each data owner provides two independent geometric samples k_i, k'_i.

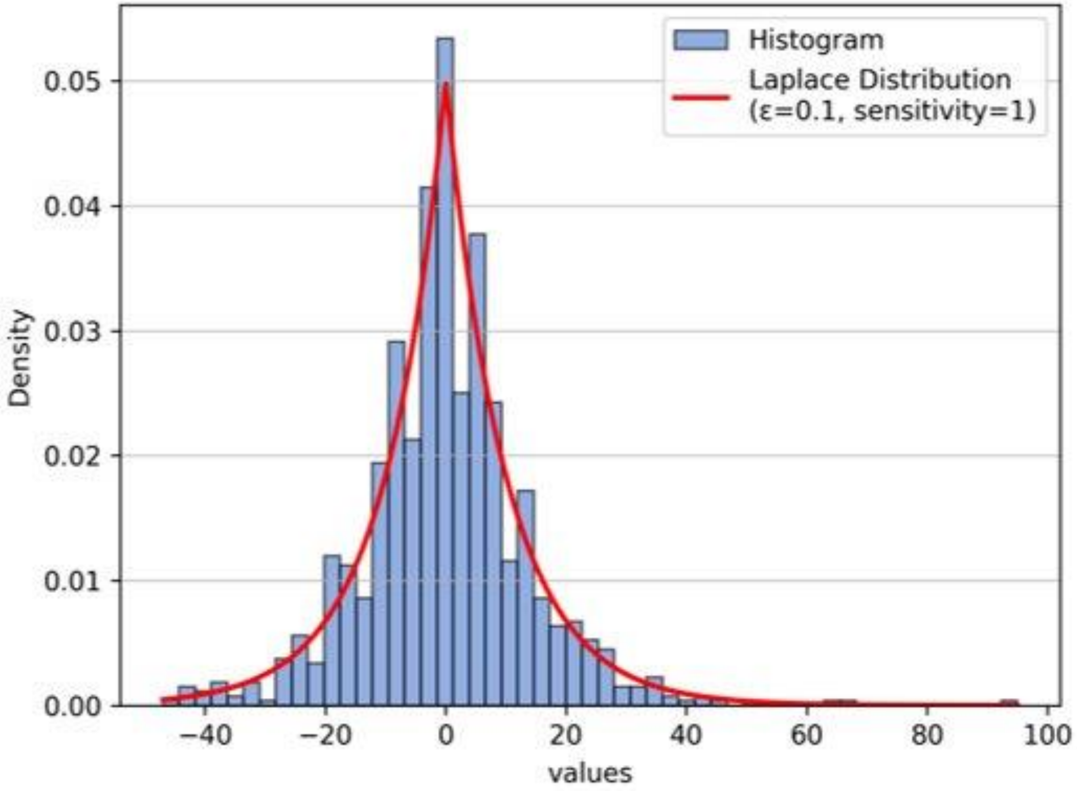
The server computes

$$\boxed{g_1 = \min(k_1, \ldots, k_n)} \text{ and } g_2 = \min(k'_1, \ldots, k'_n)$$

with $p = 1 - \exp(-1/b)$.

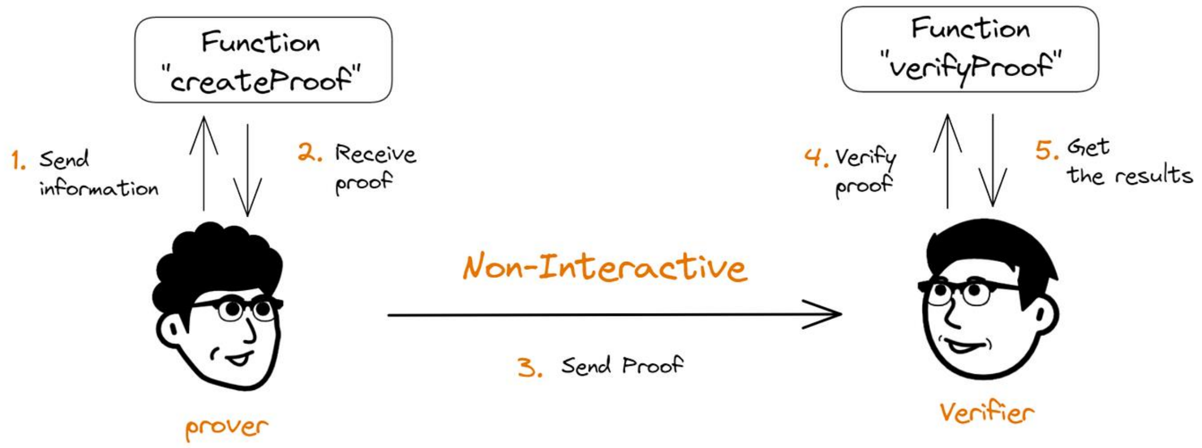
Compute the Laplace noise: $\boxed{\eta = g_1 - g_2.}$

Return: the noise η.



Zk SNARKs For Laplace Destribution

- ZK SNARK is a zero knowledge proofs.
  - Succint
  - Non Interactive
- The only thing to define is the invariant to verify



$$\exists g, r_g, \{k_i, r_i\}_{i=1}^{n} \text{ such that } \begin{cases} C_g = \mathsf{Com}(g; r_g), & \\ C_{k_i} = \mathsf{Com}(k_i; r_i), & \forall i \in \{1, \dots, n\}, \\ \prod_{i=1}^{n}(k_i - g) = 0, & \text{(membership condition)}, \\ k_i - g \geq 0, & \forall i \in \{1, \dots, n\} \text{ (minimality} \end{cases}$$

Minimality Condition

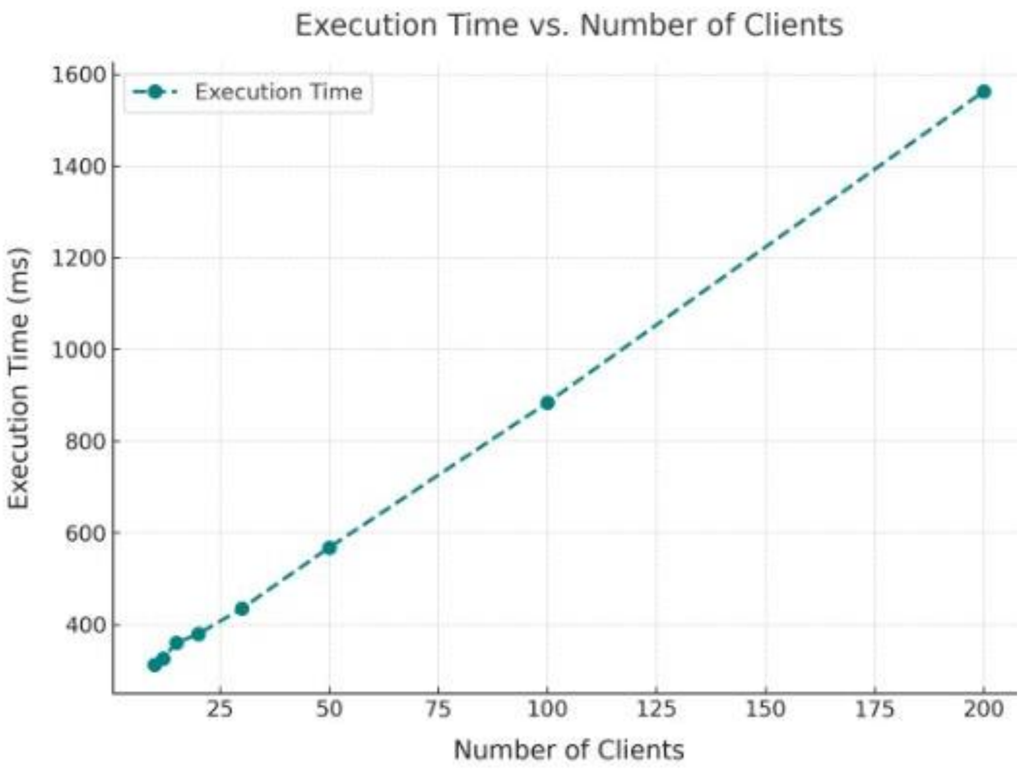
$$\exists\, g_1, g_2, r_{g_1}, r_{g_2} \text{ such that } \begin{cases} C_{g_1} = \mathsf{Com}(g_1; r_{g_1}), \\ C_{g_2} = \mathsf{Com}(g_2; r_{g_2}), \\ C_\eta = \mathsf{Com}(\eta; r_\eta), \\ \eta = g_1 - g_2. \end{cases}$$

Difference Condition

Time Evaluation

Performance Metrics for Each Phase

Phase	**Server Time (ms)**	**Verifier Time (ms)**	**# Constraints**
Stage 1	49.01	3.66	-
Stage 2 : π_{min}	404.92	11.32	9216
Stage 2: π_{noise}	94.83	11.62	2338
Stage 3: $\pi_{compose}$	183.69	10.43	3344
Stage 4: $\pi_{addition}$	98.76	9.37	2338
Overall	**1433.03**	-	**17236**



Direction Choices

- Summary:
  - Reduce trust in server behavior
  - DP becomes verifiable rather than trust-based

Reflection

- Initially, the goal was to combine this with Contribution 1.
- Circuits depend on number of clients.
  - Requires a trusted setup

  >> implications on system
- Computation Overhead (More if added to HE).
- Communication Overhead due to distributed generation.
- Limited DP Mechanism : only discrete ones.

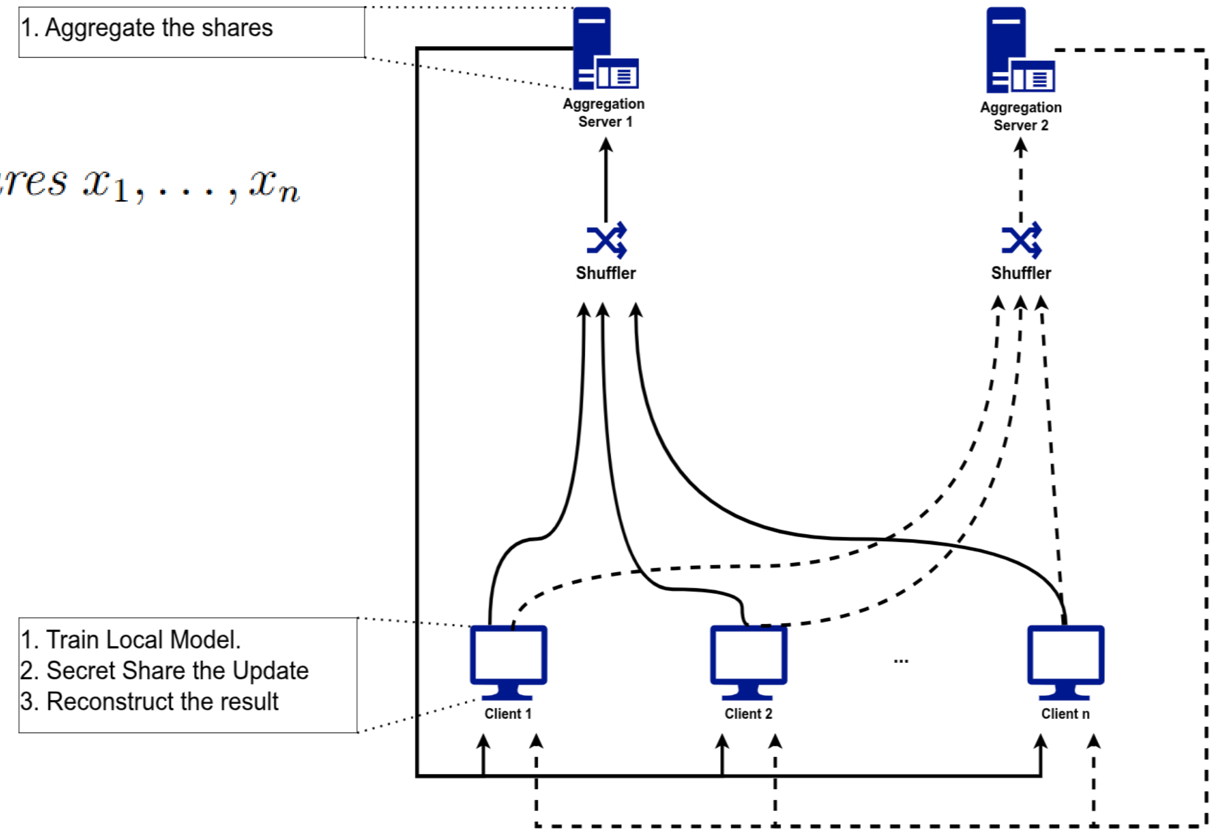
III-C. Reducing Both Trust of View and of Behavior in PPFL

Can we achieve these guarantees with lighter, more efficient techniques ?

Additive Secret Sharing

Definition. *A value x is decomposed into random shares $x_1, \ldots, x_n$ such that*

$$x = \sum_{i=1}^{n} x_i,$$



Benefits:

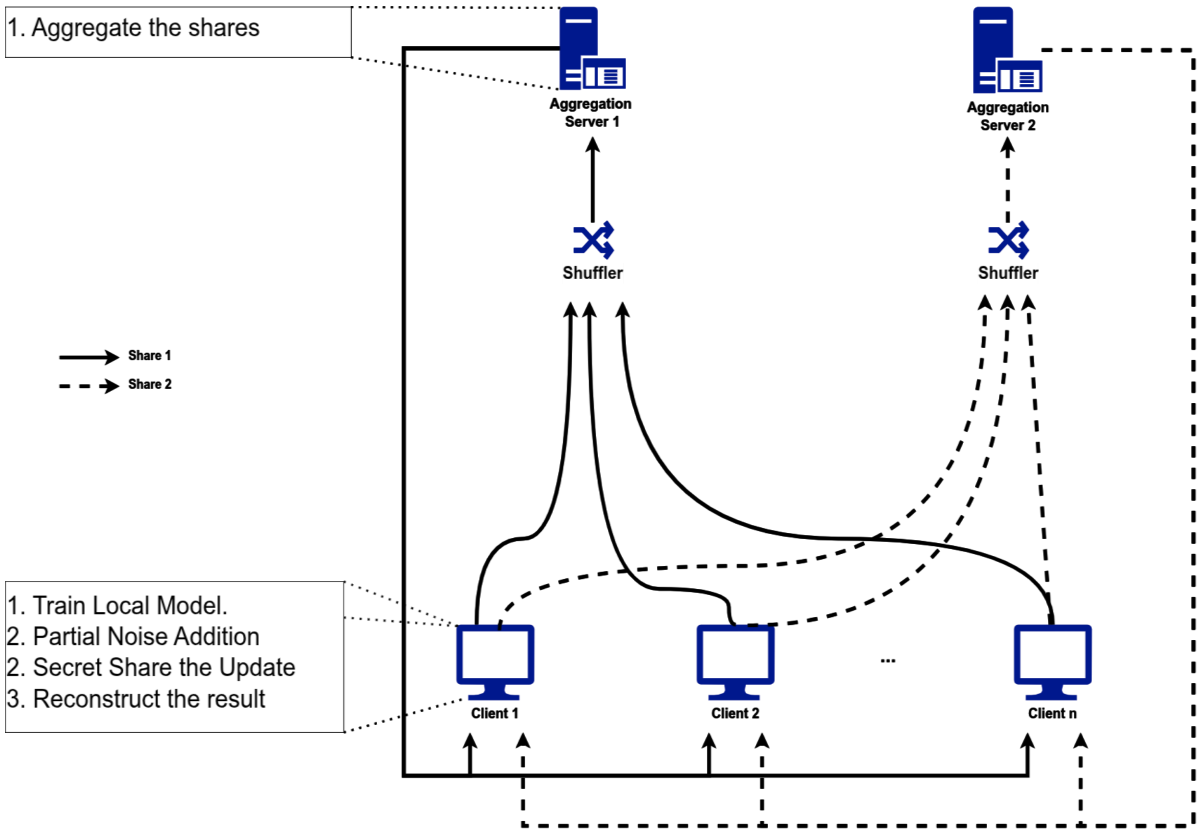
- No need for heavy encrypted computation.
- Everything done on **real numbers**, much faster
- Built-in **homomorphic properties** → easy aggregation

Differential Privacy Layer

- After reconstruction, the global model is unprotected.

$$Y_i = (0, \ldots, \underbrace{X_i}_{j_i\text{-th position}}, \ldots, 0), \quad \text{with } X_i \sim \mathrm{Lap}\left(\frac{\Delta}{\varepsilon}\right)$$

$$W_i^{noisy} = W_i + Y_i$$



Pedersen Commitments

Commit to a value without revealing it.

Definition. *A* Pedersen commitment *to a message m is computed as*

$$C = g^m h^r,$$

where r is a random value.

Properties :

- Hiding
- Binding
- Homomorphism

Verifying a Commitment

- Normally: reveal m and r
- Verifier recomputes C to check correctness

Final Architecture

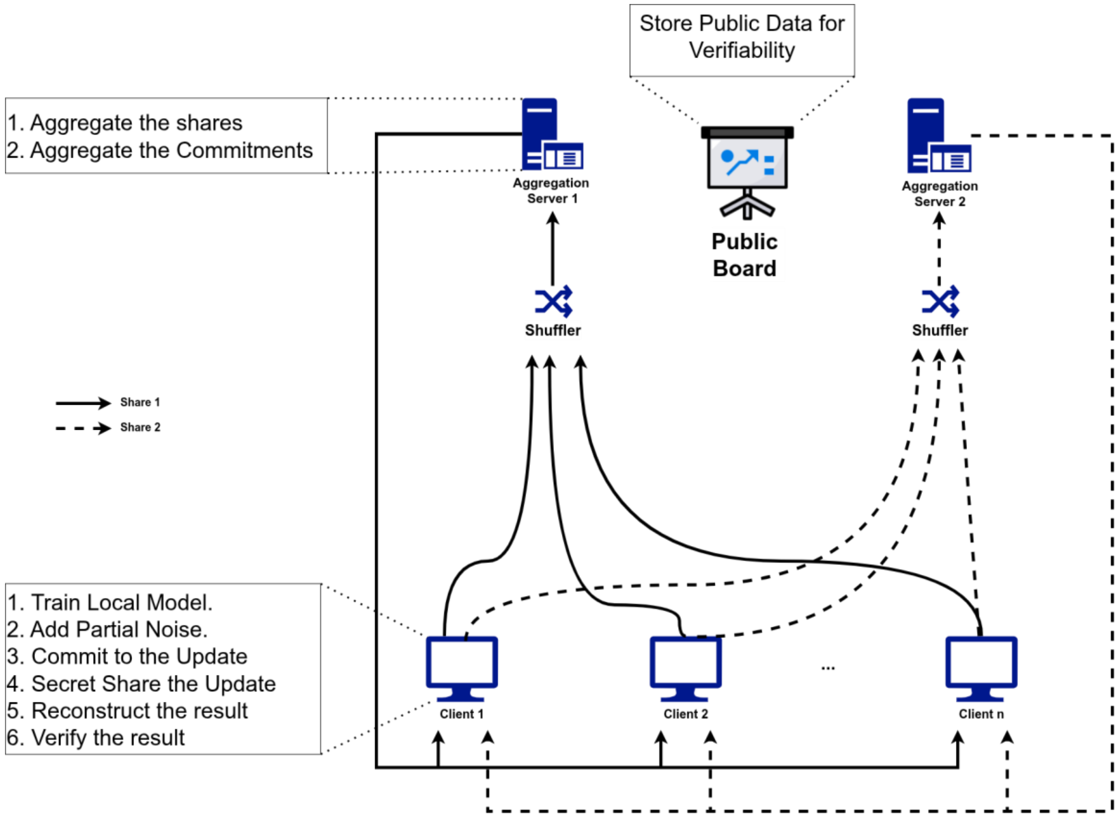
$$C = \prod_{i=1}^{n} C_i = g^{\tilde{\mathbf{w}}} h^R \quad \text{where } R = \sum_{i=1}^{n} r_i$$

Verifying a Commitment

- Normally: reveal w and R

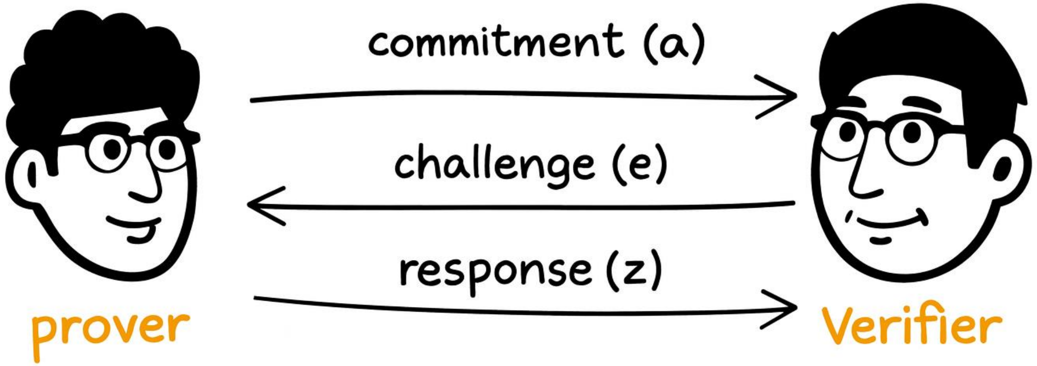
Challenges

- R is secret shared among Clients.
- Revealing the shares of R break the system.
- If we know r_i we can reverse C_i.



ZKP of Commitment Opening

Does Combined client commitments open to server's announced result ?

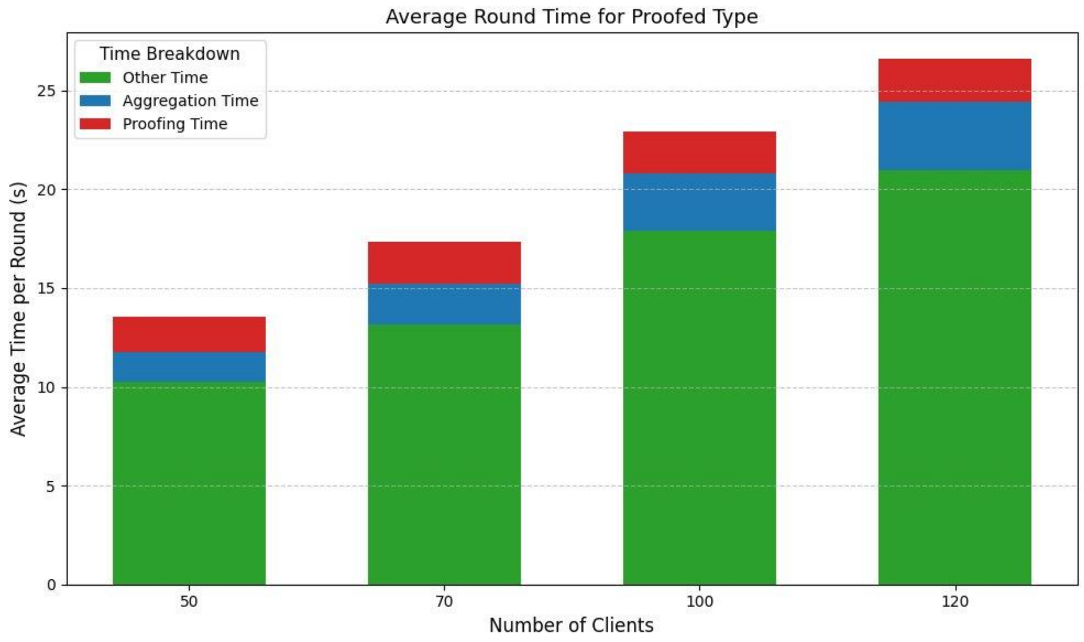


Sigma Protocols

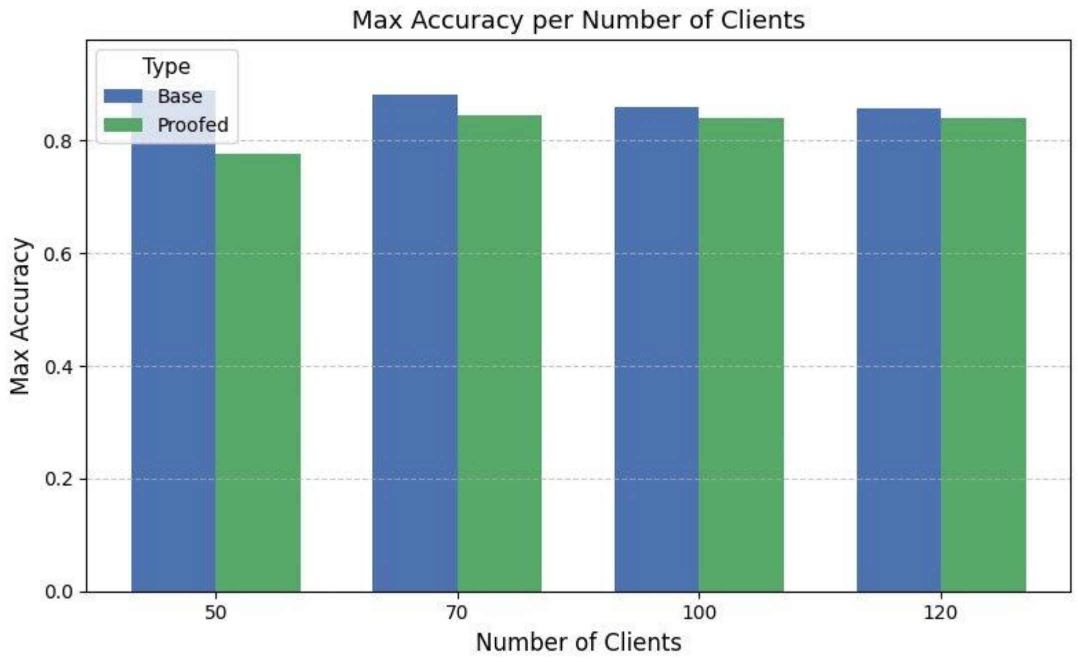
- Commitment $$A = \prod_{i=1}^{n} A_i = h^{\rho}$$
- Challenge $$c = H(A, C, \tilde{\mathbf{w}})$$
(Fiat-Shamir heuristic)
- Response $$z = \sum_{i=1}^{n} z_i = \rho + c \cdot \tilde{\mathbf{w}} \mod q$$
- Verify : $$h^z \stackrel{?}{=} A \cdot (h^{\tilde{\mathbf{w}}})^c$$

Time Perfomance

- We measure time per training round as the number of clients increases
- Server time is divided into:
  - Proofing time
  - Aggregation time
  - Other time (mostly waiting for clients)



Accuracy Evaluation



IV. Takeways

Conclusion and Perspectives

Global Summary

- This thesis turns gives verifiable guarantees in Federated Learning.
  - Privacy becomes enforceable, not declarative.
  - Addresses GDPR accountability (Art. 5, 25, 32)

1
View Trust Reduction
Raw updates never exposed

DP + HE + Shuffling

2
Behavior Trust Reduction
DP application is verifiable

DP+Zk-SNARKs

3
Full Trust Reduction
FL is private and Verifiable

SS + ZK proofs

Publications

- Aziz, R., Banerjee, S., Bouzefrane, S., & Le Vinh, T. (2023). Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm. Future internet, 15(9), 310.
- Aziz, R., Banerjee, S., & Bouzefrane, S. (2024). Privacy Preserving Federated Learning: A Novel Approach for Combining Differential Privacy and Homomorphic Encryption. In IFIP International Conference on Information Security Theory and Practice (pp. 162-177). Cham: Springer Nature Switzerland.
- Aziz, R., Badr, Y., & Bouzefrane, S. (2025). Enhancing Trust in Central Differential Privacy Using zk-SNARKs and Cryptographic Hashes. In International Conference on Advanced Information Networking and Applications (pp. 163-176). Cham: Springer Nature Switzerland.
- Aziz, R., Badr, Y., Banerjee, S., & Bouzefrane, S. (**Under Review at JISA**). ProoFed: A Distributed Differential Privacy framework for Federated Learning based on Secret Additive Sharing and Verifiable Protocols

Limitations and Perspectives

Limitations :

- Computation and Communication Overhead.
- Limited scope of experiments:
  - Fixed Number of Clients.
  - Synchronisation Assumption.
  - No Privacy Leakage Tracking.
- Limited Scope of Attacks

Future Work

- Robustness Against Byzantine Attacks
  - Adapt robust aggregation techniques to the framework.
  - Verifiability on the Client Side
- Asynchronous Secure FL

Open Direction :

- Interpretable Privacy Guarantees

Thank you for your attention

I am happy to answer your questions.