SecPATE: Enhancing Privacy in PATE with Secure Multi-Party Computation

Anh-Tu Tran Thi Hong Ha Nguyen

[1]Academy of Cryptography Techniques, Hanoi, Viet Nam

THE 2nd INTERNATIONAL CONFERENCE ON CRYPTOGRAPHY AND INFORMATION SECURITY
October 30 – 31, 2025

Presentation Overview

Outline

Introduction

- Significant progress has been made in Machine Learning, especially in Deep Learning, leading to remarkable outcomes in recent years.
- These technologies are widely applied in areas such as image recognition, speech processing, fraud detection, healthcare, and autonomous vehicles.
- However, there are concerns regarding the exposure of personal data, vulnerabilities in model security, and the growing need for advanced techniques to protect user privacy [Rigaki, 2023].

Introduction

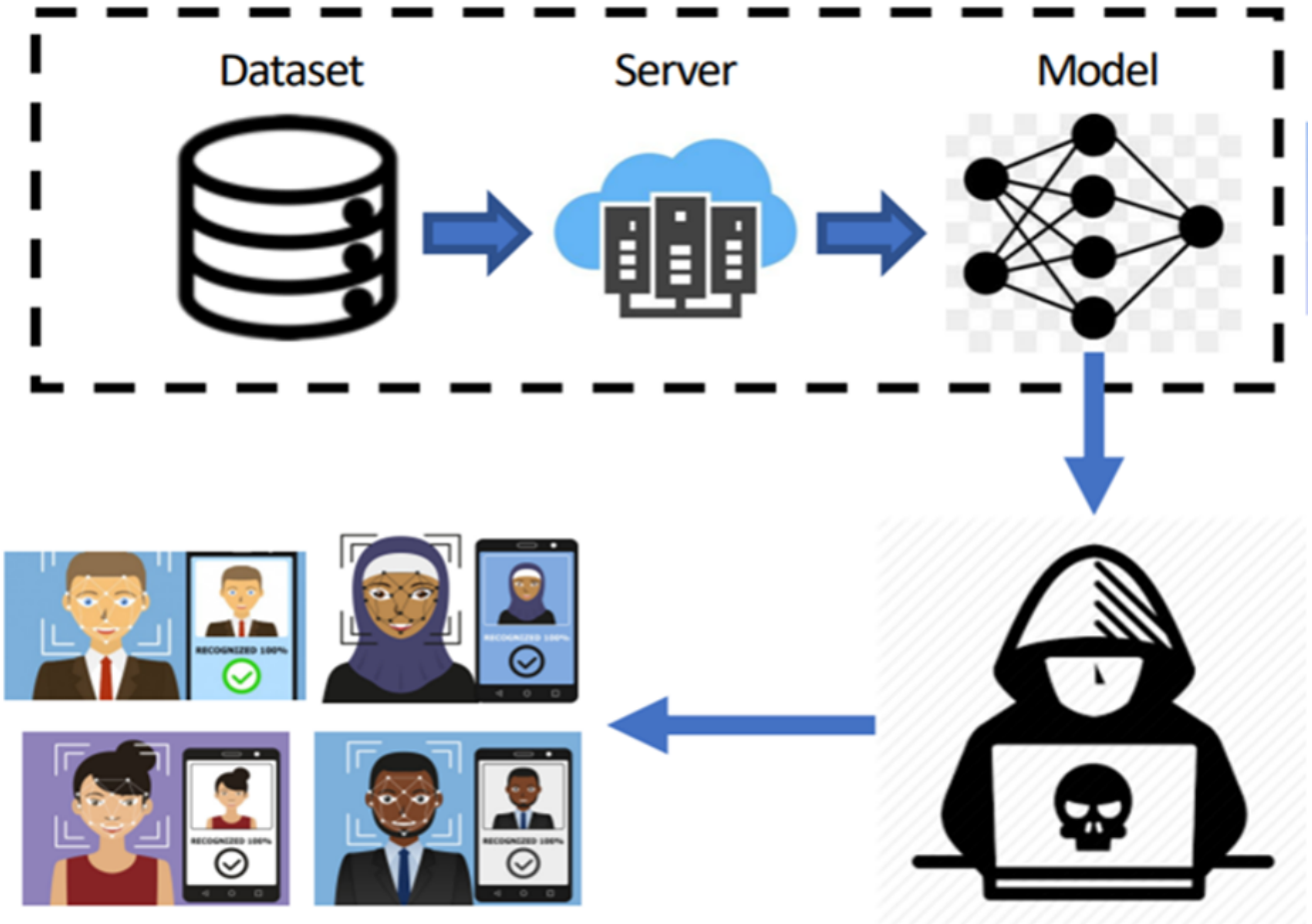
The need for privacy-preserving Machine Learning [Rigaki, 2023]

- The performance of Machine Learning models is significantly impacted by the amount of training data available. Larger datasets typically result in higher accuracy for the model (Goodfellow, 2016).
- However, there is a risk of exposing sensitive information, which can lead to security threats, identity theft, and potential misuse of data.

Image: Facebook

Introduction

The need for privacy-preserving Machine Learning



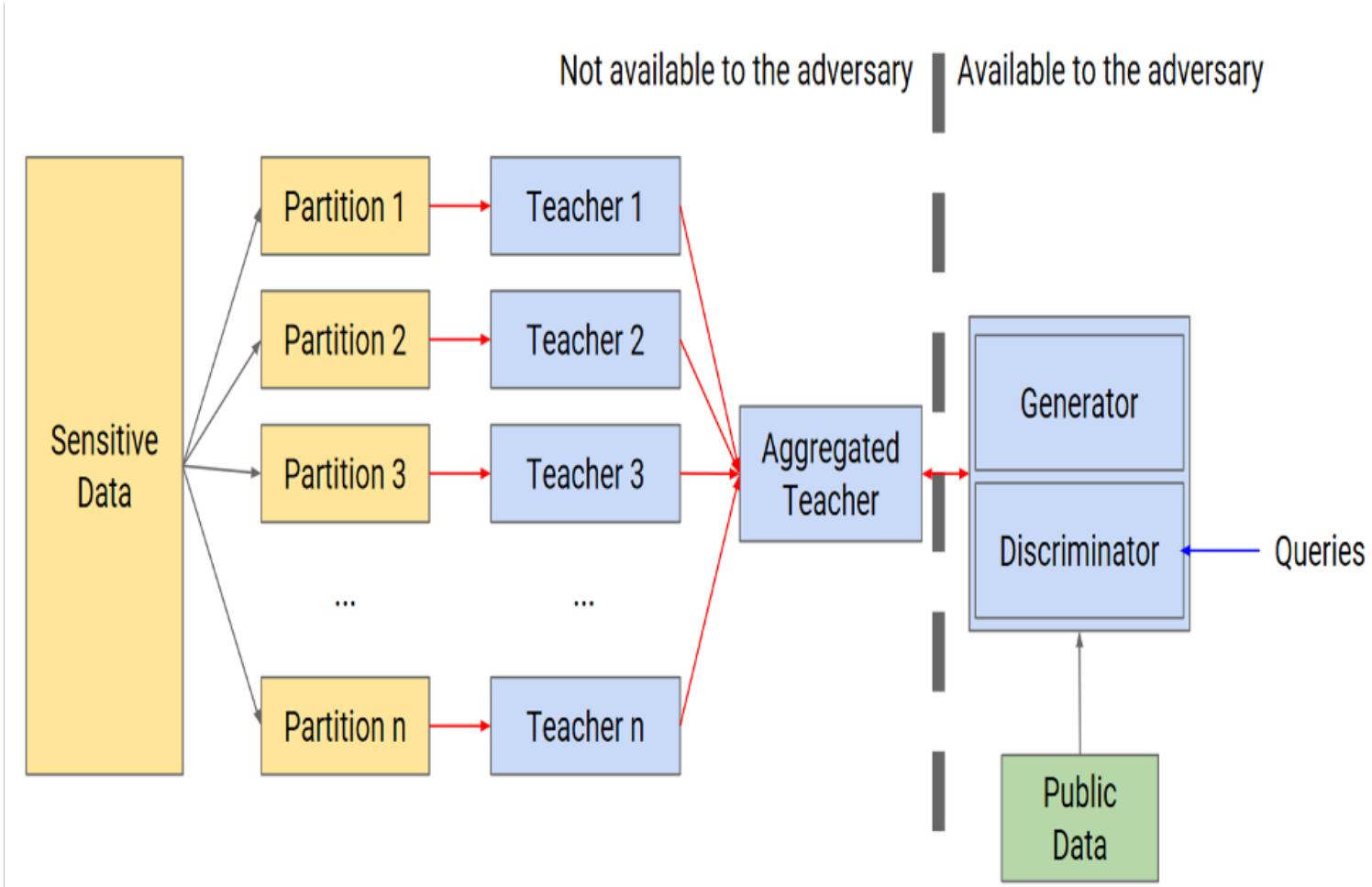
Introduction

The need for privacy-preserving Machine Learning

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.

Introduction

Related work



Advantages:

- Hides local models.
- Local models can be different.

Disadvantages:

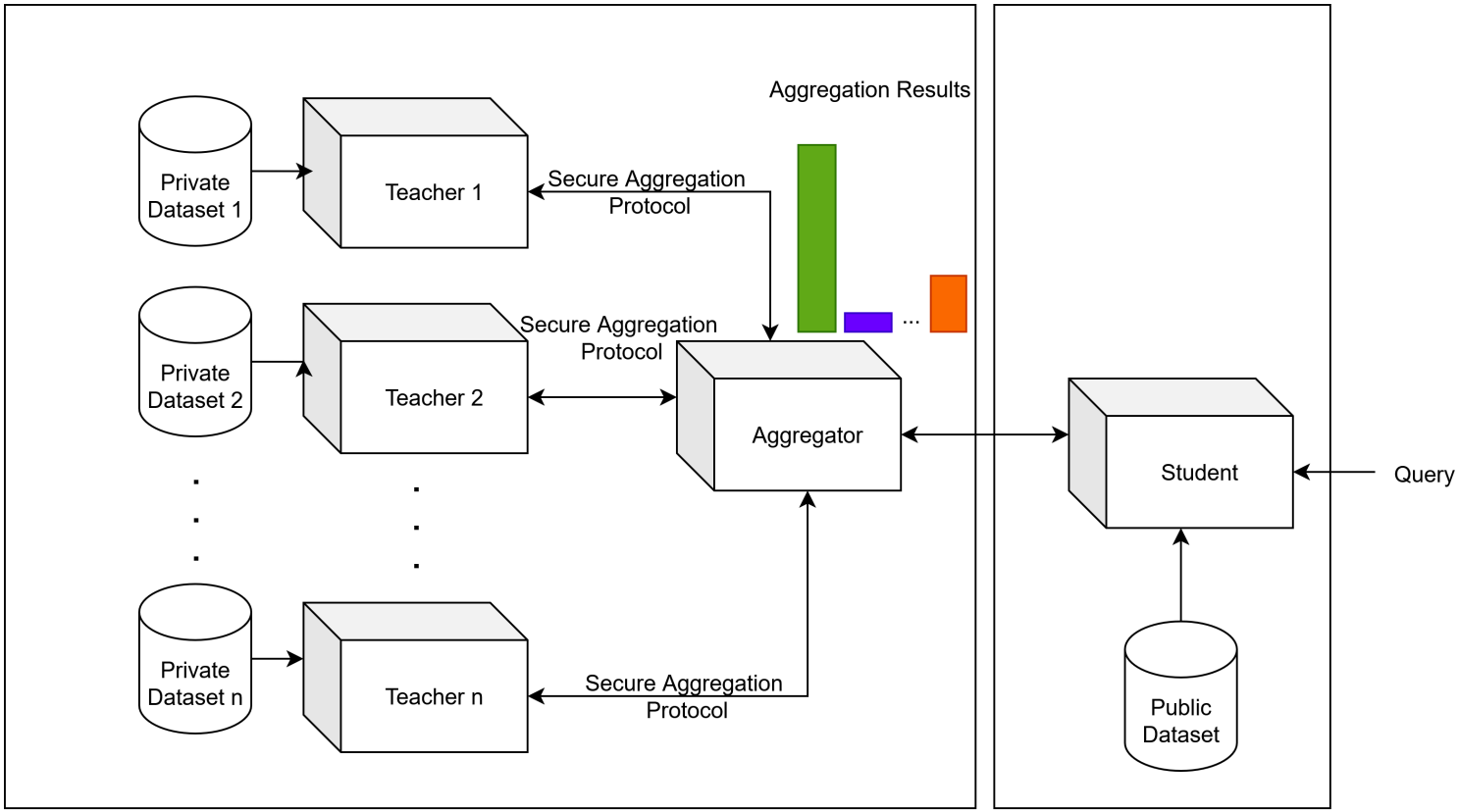
- Lower accuracy due to random noise.
- In the PATE framework, teacher model predictions can be exposed during the aggregation process. In simpler terms, while the aggregation process is intended to ensure privacy by combining the predictions, there is still a vulnerability: if the individual predictions are exposed, they may leak details about the private datasets used by the teachers. This is particularly concerning in sensitive applications such as healthcare, financial data analysis, or personal data processing, where any disclosure of information could lead to severe privacy breaches.

Outline

Framework Overview

- **Student's Role:** Holds unlabeled data.
- **Teacher Models:** Trained on separate datasets and provide label predictions.
- **What is SecPATE?**: SecPATE is an enhancement to PATE that integrates Secure Multi-Party Computation (SMC) for privacy-preserving aggregation of predictions.
- **How it works**: Teacher models participate in a secure computation protocol, ensuring that no individual predictions are exposed during the aggregation.

Framework Overview



Secure Aggregation Protocol for PATE

Input:

- Each **teacher** T_i has private prediction vector $O_i = \{O_i^{(j)}, 1 \leq j \leq \text{num\_class}\}$.
- Each **teacher** T_i has two private key vectors: $x_i = \{x_i^{(j)}\}, y_i = \{y_i^{(j)}\}$.
- System parameters: the exponential factor (γ), $\mathbb{Z}_{\mathbf{p}}$ and generator g.

Output: Approximate vector sum: $\tilde{O} = \sum_{i=1}^{n} O_i$.

Phase 1: Initialization Phase

- Each **teacher** T_i sends its public key vectors $\{X_i^{(j)}\} = \{\mathbf{g}^{\mathbf{x}_i^{(j)}}\}, \{Y_i^{(j)}\} = \{\mathbf{g}^{\mathbf{y}_i^{(j)}}\}$ to **aggregator**.
- The **aggregator** computes: $X = \left\{\prod_{i=1}^{n} X_i^{(j)}\right\}$; $Y = \left\{\prod_{i=1}^{n} Y_i^{(j)}\right\}$ for $1 \leq j \leq \text{num\_class}$ then sends them back to all **teachers**.

Phase 2: Main phase

- Each **teacher** quantizes their prediction vectors $\tilde{O}_i^{(j)} \leftarrow O_i^{(j)} 10^{\gamma}$, for $1 \leq j \leq \text{num\_class}$.
- Each **teacher** T_i encrypts their quantized prediction vectors: $\left\{V_i^{(j)} = \frac{X^{(j)\mathbf{y}_i^{(j)}}}{Y^{(j)\mathbf{x}_i^{(j)}}} \mathbf{g}^{\tilde{O}_i^{(j)}}\right\}$ for $1 \leq j \leq \text{model\_size}$ and sends to the **aggregator**.
- The **aggregator** then computes $\{V^{(j)}\} = \left\{\prod_{i=1}^{n} V_i^{(j)}\right\}$ for $1 \leq j \leq \text{num\_class}$.
- The **aggregator** performs Shank's algorithm to find $S^{(j)}$ with:
$$g^{S^{(j)}} = V^{(j)} \text{ for } 1 \leq j \leq \text{num\_class}.$$
- The **aggregator** computes the approximate vector sum by:
$$\tilde{O}^{(j)} = \frac{S^{(j)}}{10^{\gamma}}$$

Figure: Secure Aggregation Protocol for PATE

How SecPATE Works

SecPATE addresses this vulnerability by incorporating SMC into the aggregation phase. Instead of sending raw predictions from each teacher model to a central server, the teacher models participate in a secure computation protocol. This ensures that no individual prediction or sensitive information is exposed during the aggregation.

- Teacher Models: Each teacher makes a prediction based on its own private dataset.
- SMC Protocol: Teacher models collaborate using an SMC protocol to aggregate their predictions without exposing them.
- Final Prediction: The aggregated result is then used to label the student dataset, ensuring that no sensitive information is leaked.

Key Benefits of SecPATE

- Privacy-Preserving: Teacher model predictions are never exposed in plain text.
- Secure Aggregation: Predictions are aggregated within a secure, encrypted domain, preventing any leaks of private data.
- No Accuracy Trade-off: Unlike other privacy-preserving methods, such as differential privacy, SecPATE can maintain the model's accuracy while improving privacy guarantees.

Comparison with Other Methods

- In standard PATE, teacher predictions are aggregated using a voting system, where individual teacher predictions are exposed during the summation phase. This can create a risk of information leakage, especially if the aggregator is compromised or malicious.
- By replacing the traditional aggregation with a secure SMC-based protocol, SecPATE ensures that even during the aggregation phase, no teacher's individual prediction is revealed. This significantly enhances the privacy guarantees of the system.

Outline

Datasets Used: MNIST Dataset

- **MNIST Dataset (Image Classification)**:
  - Task: 10-class image classification
  - Training Data: 60,000 images (partitioned among $N_{clients}$ - teachers)
  - Test Data: 10,000 images, further split into:
    - Public Portion: $\sim$6,667 samples for student training
    - Hold-out Set: $\sim$3,333 samples for final evaluation

Datasets Used: SMS Spam Collection

- **SMS Spam Collection (Text Classification)**:
  - Task: Binary classification for spam detection
  - Training Data: 70% for teacher training (split among $N_{clients}$)
  - Student Training: 20%
  - Evaluation: 10%
  - Preprocessing: All text messages were tokenized and padded to a uniform length of 20.

Model Summary: LSTM Architecture

Layer (type)	**Output Shape**
Embedding (Embedding)	(None, 20, 20)
LSTM (LSTM)	(None, 400)
Dense (Dense)	(None, 1)
Total params: 752,401	
Trainable params: 752,401	
Non-trainable params: 0	

Table: Model Summary of the LSTM Architecture

Model Summary: CNN Architecture

Layer (type)	Output Shape	Param #
Conv2D (Conv2D)	(None, 26, 26, 32)	320
MaxPooling2D (MaxPooling2D)	(None, 13, 13, 32)	0
Conv2D (Conv2D)	(None, 11, 11, 64)	18,496
MaxPooling2D (MaxPooling2D)	(None, 5, 5, 64)	0
Flatten (Flatten)	(None, 1600)	0
Dense (Dense)	(None, 128)	204,928
Dense (Dense)	(None, 10)	1,290
Total params: 225,034		
Trainable params: 225,034		
Non-trainable params: 0		

Table: Model Summary of the CNN Architecture

Experimental Procedure

- **Teacher Model Training**: 5 teacher models trained for 10 epochs (batch size 64, Adam optimizer) on disjoint data subsets.
- **Prediction Aggregation**: Teacher predictions were aggregated using the SecPATE framework with a Secure Vector Sum Protocol (Elgamal-like, 256-bit prime field).
- **Student Model Training**: Student model trained for 10 epochs on securely generated labels with the same architecture as the teachers.
- **Evaluation**: Student model performance evaluated on an unseen test set.

Hardware and Software: Intel Xeon CPU, NVIDIA V100 GPUs, Python, TensorFlow 2.x, NumPy.

Model Utility (Accuracy Analysis)

We compared the accuracy of student models trained using the PATE framework (with and without explicit DP noise) against non-private baselines, and subsequently compared these to student models trained with SecPATE.

- For the MNIST task, the student model trained using labels from SecPATE achieved an accuracy of 98.17% on the unseen test set.
- For the SMS Spam Collection task, the student model achieved an accuracy of 94.98% in binary text classification.
- SecPATE maintains high utility while offering improved privacy guarantees.

Task	Accuracy (%)	Precision	Recall	F1-Score
MNIST	98.17	0.98	0.98	0.98
LSTM	94.98	0.94	0.72	0.81

Table: Performance of MNIST and LSTM models on the remaining 1/3 test set

Accuracy Comparison and Findings

We compared the performance of different privacy-preserving methods, including SecPATE, Differential Privacy (DP) noise, no privacy protection, and the case where PATE is not applied.

Method	**MNIST (%)**	**SMS Spam (%)**
SecPATE	98.17	94.98
DP Noise ($\epsilon = 0.1$)	95.28	90.30
No Protection	98.17	94.98
No PATE	99.12	98.56

Table: Comparison of Accuracy for SecPATE

- SecPATE achieves 98.17% on MNIST and 94.98% on SMS Spam, similar to models with no privacy protection.
- Applying DP noise ($\epsilon = 0.1$) results in accuracy drop, highlighting the privacy-accuracy trade-off.
- Rounding in SecPATE has negligible impact on accuracy when γ is set between 3 and 5.

SecPATE: Enhanced Privacy Protection

SecPATE strengthens privacy by addressing the potential exposure of teacher predictions in standard PATE:

- In standard PATE, teacher predictions are aggregated directly, risking exposure of sensitive data. SecPATE uses a Secure Aggregation protocol to ensure predictions are never exposed in plaintext.
- SecPATE ensures that only the cryptographically derived sum of votes is revealed, preventing direct observation attacks.
- It offers resilience against insider threats, as the aggregator cannot access individual teacher keys or predictions, even in cases of collusion.
- SecPATE integrates with Differential Privacy (DP), ensuring the final student model maintains ϵ-DP privacy while preserving model accuracy.

Computational Overhead Analysis

SecPATE's cryptographic protocols introduce computational overhead, but it remains manageable for practical deployment.

- **Time Complexity**: - Secure aggregation scales with the number of teachers (N) and output classes. Initialization phase: $O(N \cdot \text{num\_class})$ (e.g., 50 teachers, 10 classes = 0.52 seconds). Main phase involves intensive server-side computation, but still scalable.
- **Overhead**: Compared to plaintext summation (`np.sum`), SecPATE introduces noticeable overhead, but it is manageable, especially in offline or batch processing.
- **Scalability**: SecPATE scales well with more teachers and output classes, though discrete logarithm computation is sensitive to the prime field size.
- **Memory Usage**: Low memory usage, primarily for storing public keys and encrypted vectors.

Overhead is acceptable for most practical applications.

Outline

Conclusion

We introduced SecPATE, an enhancement to the PATE framework, addressing the critical vulnerability of exposing teacher predictions. By integrating Secure Multi-Party Computation (SMC), SecPATE ensures that teacher model predictions remain confidential, significantly improving privacy protection.
Key Points:

- SecPATE preserves PATE's core advantages: high utility and accuracy.
- It enhances privacy by protecting both teacher predictions and sensitive training data.
- SecPATE bridges the gap between privacy and model performance, setting a new standard for privacy-preserving machine learning.

SecPATE provides a robust, trustworthy solution for real-world applications where data sensitivity is critical, and it opens up new avenues for future research in secure machine learning.

Thank you!