Protecting Data Privacy: From Individual Action to System-Level Safeguards through Compliance and Design

Ziyue Gong
Department of Electrical and Computer Engineering
University of Toronto
joy.gong@mail.utoronto.ca

Abstract

This paper explores the question: “What can individuals do, and what is needed at a broader level, to protect data privacy?” Starting from a personal and bottom-up perspective, the paper draws on course materials to explore practical knowledge and strategies that individuals can adopt. It then shifts focus to systemic, top-down solutions, specifically emphasizing compliance and privacy by design; key topics include standardized privacy policies for compliance, and technical discussions on Membership Inference Attacks (MIA), and its countermeasure Differentially Private Stochastic Gradient Descent (DP-SGD). Through this retrospective and reorganized synthesis, the paper underscores the need for an integrated approach to data privacy that combines individual action with robust system-level safeguards.

Contents

1 Introduction

This paper begins with a simple but pressing question from an individual perspective: How can I protect my own data privacy? To address this, I revisited the course content to identify practical tools and concepts—both technical and legal—that individuals can apply in everyday life. This includes basic privacy-enhancing techniques, informed awareness, and legal recourse drawn from real-world case studies. However, as I progressed, it became clear that individual-level, self-reporting, and reactive strategies are often insufficient. Many people lack the motivation or awareness to protect themselves, underestimate their exposure, or face resource limitations that make privacy preservation impractical in today's digital world.

These limitations prompted a deeper exploration into more structured, proactive solutions rooted in compliance and privacy by design. For compliance by design, reflecting from the AppTrans paper, I examined the challenges of implementing automated tools and proposed the use of standardized privacy policies. For privacy by design in the age of AI, I focused on technical approaches to de-identification and risk mitigation, specifically Membership Inference Attacks (MIA) and Differentially Private Stochastic Gradient Descent (DP-SGD), with a code example for better demonstration.

Together, these perspectives—bottom-up individual actions and top-down systematic frameworks—offer a comprehensive view of what it takes to protect privacy in a digital society, highlighting the import

2 Taking Ownership of Data Privacy: Awareness & Action in the Digital Age

In an era where digital technologies permeate nearly every aspect of daily life, individuals can and should raise awareness, gain knowledge (both legal and technical), and exercise their rights to take proactive steps in defending their personal data. To do so effectively, individuals must first understand how privacy can be compromised, and then take deliberate action to mitigate risks.

2.1 Understanding How Privacy Can Be Compromised

A crucial first step toward defending data privacy is becoming aware of the various ways it can be infringed upon. Device identifiers and location tracking. Whenever a device connects to a Wi-Fi network, it uses its MAC (Media Access Control) address to identify itself. In large installations, multiple Wi-Fi access points (APs) can detect the same device and triangulate its location as it moves around the area [1] Lecture 6, slide 8. However, MAC addresses are only used for identification while the device is on the network, and modern devices such as smartphones and laptops now support MAC randomization. This means the device uses a different MAC address for each network, helping to prevent tracking [1] Lecture 6, slide 9.

Third-party cookies - data stored in a user's browser by a domain other than the one the user is currently visiting. Since third-party services (e.g. advertising) are frequently embedded across multiple unrelated websites, if someone visits websites using the same third-party service, the third-party domain can use the same cookie stored at that user's browser to recognize and monitor the user's browsing activity over time and across sites, without the user being aware that this is happening in the background [1] Lecture 5, slide 6.

Interface manipulation: A deceptive interface design that conceals the true nature of a transaction or user actions. One of the most notable recent examples is the Bybit cold wallet hack—reportedly the most expensive theft in history—was carried out through interface manipulation, which led to the unauthorized transfer of over 400,000 ETH (roughly 1.5 billion USD) to the hackers' account. The attackers modified the wallet signing interface so that while the UI displayed a valid wallet address, the underlying true destination had been altered to that of the hackers' [2].

2.2 Taking Action to Protect Personal Data

With awareness in place, it's now time for actions. Individuals must actively apply their knowledge and make use of the tools and rights available to protect their privacy.

Daily practices are the most straightforward and intuitive way of action. From the above awareness discussion, to minimize passive tracking and device fingerprinting, individuals can enable MAC address randomization, turn off unnecessary Wi-Fi and Bluetooth access, and avoid open public networks; Similarly, to block third-party cookies and thus limit third-party services to build behavioral profiles without consent, users can install privacy-focused browser extensions, clear cookies regularly and disable unnecessary location services.

Beyond individual practices, however, it is important to recognize the role of legal action and public discourse in shaping more comprehensive privacy protections. In many cases, it is not just individuals' behaviors that determine privacy outcomes, but also how governments and courts define and enforce the boundaries of personal data.The 2016 case of Patrick Breyer v. Bundesrepublik Deutschland [3] , addressed whether dynamic IP addresses could be considered as "personal data." German federal institutions were storing IP addresses and access times in server logs but claimed they could not personally identify users without information from third-party ISPs. The key legal question was whether an IP address constitutes personal data if the site operator cannot directly identify the individual, but could do so with help from another party. The Court of Justice of the European Union (CJEU) ultimately ruled that dynamic IP addresses do count as personal data when the site operator has legal means to potentially identify a user, such as in the event of a cyberattack. This ruling affirmed that data privacy should be assessed not just by what one party knows, but by what could reasonably be inferred or accessed via cooperation with others—highlighting the complex nature of digital identification [1] Lec3, slide14.

A more recent illustration is the 2023 case of Single Resolution Board (SRB) v. European Data Protection Supervisor (EDPS) [4], which considered whether pseudonymous data could be treated as anonymous when shared with third parties. The SRB had collected comments from the public and tagged them with alphanumeric codes, removing direct identifiers before transferring them to an external consultant, Deloitte. The central issue was whether Deloitte could, in practice, identify individuals using the data provided. The court ruled that even pseudonymous data may still be considered personal data, given that the recipient party has a "means reasonably likely to be used" to re-identify individuals (Lec3, slide15). This ruling reinforces the importance of accountability in data handling—not just for those collecting data directly, but also for those downstream in the data chain.

These two legal cases demonstrate how individual legal actions can ignite broader public discourse and lead to court rulings that clarify and establish more detailed privacy standards. Individual legal actions like these raise public awareness, clarify regulatory boundaries, and pressure organizations to handle data with more care.

2.3 Calling for top-down systematic frameworks: Limitation of individual efforts calls

While individual actions/efforts—such as enabling MAC randomization, managing cookies, or invoking legal rights—are important first steps toward protecting personal data, they are often insufficient on their own. Structural limitations persist, even in cases where legal protections are in place. The case of Mr. Gonzalez, who was forced to pursue legal action to request the de-indexing of outdated yet damaging information (a fully repaid debt) from Google's search engine [5], underscores how the current system typically responds only after the harm has already been done. Similarly, third-party cookies and embedded libraries continue to facilitate opaque cross-site tracking, even as users attempt to block them. These examples underscore the reactive nature of existing privacy mechanisms and the imbalance of power and responsibility between users and data handlers.

Notably, beyond legal gaps and technical complexity, a more fundamental problem undermines effective data privacy protection:a lack of incentives from both the data providers and consumers. On the data provider's side—such as everyday app users—privacy is often an afterthought, reduced to a mere checkbox for compliance. Most users don't read consent forms carefully; they simply agree in order to access app features, relying on blind trust that companies or regulators have already ensured their privacy.

On the data consumer's side—namely companies—there are strong incentives to collect, exploit, and monetize user data, with little motivation to prioritize privacy beyond legal requirements.To maximize revenue, developers commonly integrate third-party advertising and analytics libraries that harvest behavioral and demographic data under the guise of enhancing engagement. For example, in the Cadillac Fairview Analytics Expansion case, the company deployed Anonymous Video Analytics (AVA) technology across its malls [6], which detected and numerically encoded faces to estimate age and gender, and tracked movement using mobile device geolocation. Despite claims that the data was anonymous, investigations revealed that unique facial identifiers were retained, contradicting the company's own privacy statements.

This case illustrates a larger industry trend: while there is significant incentive to collect personal data, there is minimal motivation to safeguard it beyond basic legal compliance. In fact, companies often invest more in crafting legally sound but abstract privacy policies; using generalized terms and vague explanations, rather than implementing meaningful, user-focused protections. These policies may satisfy regulatory obligations but fail to provide users with clear insight into how their data is being handled. Moreover, even in cases where users object or withdraw consent, it is unlikely that companies would go as far as, say, retraining machine learning models to exclude their personal data—because doing so incurs high cost with little perceived benefit.

All of this points to the need for systemic change: one that removes the burden from individual users and instead builds privacy directly into the design and governance of systems. This is where the principles of Privacy by Design and Compliance by Design come into play. Privacy by Design advocates for embedding privacy-enhancing technologies and practices at every stage of system development—from initial architecture to deployment and updates. Compliance by Design ensures that legal standards such as GDPR, PIPEDA, and CCPA are not just adhered to superficially but are operationalized into concrete technical and procedural controls.

In the sections that follow, I will discuss these two frameworks in details: beginning with Compliance by Design, which emphasizes standardized privacy policies and the role of technical enforcement, followed by an discussion on the formal privacy-preserving techniques, with a focus on Differential Privacy, illustrated by a code example of Membership Inference Attacks (MIA) and DP-SGD as its countermeasure.

3 Privacy by compliance

3.1 Automated Compliance tools

Ensuring that mobile applications adhere to their stated privacy policies is essential for maintaining user data protection. Tools like AppTrans have been proposed to automate the process of privacy compliance monitoring [7]. This tool works through three steps: it begins with the automated analysis of privacy policies, then audits application data flow (primarily via static code analysis), and finally flags inconsistencies between declared and actual practices. The ultimate goal of AppTrans is to assist regulators in overseeing app behavior and to increase accountability for third-party data usage. An example usage of such tools could be the Facebook-Cambridge Analytica scandal, where Aleksandr Kogan's app "This Is Your Digital Life" collected user data under the guise of academic research but accessed both the installing users' and their friends' data, which were later shared with Cambridge Analytica for political ad targeting. If a tool like AppTrans had been used, it might have flagged such unauthorized data collection and helped prevent the data misuse.

Despite its promising intentions, developing automated compliance tools like AppTrans face implementation challenges. One major obstacle lies in the complexity and ambiguity of privacy policy language. To remain legally compliant while retaining flexibility for future feature additions, many policies are deliberately vague or overly broad. In fact, interpreting such language often requires manual segmentation and annotation by legal experts, which complicates efforts to automate the process [7]. Another limitation is acquiring access to application code. App auditing can be done using static code analysis, which attempts to map data flows without executing the code, but current tools often fail to detect every flow. Meanwhile, live data monitoring, although more comprehensive, poses serious privacy trade-offs, as it involves tracking all user activity in real time—an approach that contradicts the initial intention of a privacy-preserving system.

3.2 Standarized privacy policies

Given these limitations, a more feasible approach may be to focus on automating the policy analysis component by standardizing privacy policies. Standardized policies would benefit all stakeholders—users, developers, and toolmakers alike. For users, it would make consent more meaningful by making policies easier to understand. For app developers, it would clarify what data third-party libraries collect. And for developers of automated tools, it would provide a more reliable foundation to build upon. Currently, most privacy policies are written in dense legal language and are not designed for machine readability or automated analysis.

A useful analogy comes from software development: in writing "user stories," project teams describe app functions using a consistent format that focuses on user perspective rather than technical detail. A typical user story might read: "As a community user, I want to notify my network about icy roads so they don't have the same near misses as me" [8]. This format facilitates clear communication between stakeholders while avoiding jargon.

Inspired by this idea, privacy policies could adopt a structured syntax that states clearly what kind of data is collected, for what purpose, and whether it is shared with third parties. For instance: "We collect your device location to provide real-time weather updates. This data is used only for core app functionality and is not shared with third parties." Statements like this would be far easier for both users and machines to interpret, eliminating much of the ambiguity of most current policy documents. A similar approach is exemplified in the Securiti Privacy Policy Template [9], which provides a structured and customizable outline for organizations to specify what data they collect, how it is used, who it is shared with, and what user rights apply. Importantly, to add on this template, beyond listing data collection practices, policies should also detail the entire lifecycle/flow of the data—including how it is processed, stored, and transferred. These aspects are often missing from traditional policies but are essential for achieving true transparency and user trust.

The importance of standardized and transparent privacy policies is not limited to Canada—it is also strongly emphasized in other industries and countries, underscoring its broader significance. For example, according to the Federal Trade Commission (FTC)—a regulator in the trade and consumer protection sector in the United States, —overly long, inconsistent and full-of legal jargon privacy policies, not only confuse users but also hinder informed consent and regulatory oversight. To address this, the agency advocates for privacy disclosures written in clear, consistent, and potentially machine-readable formats—comparable to standardized nutrition labels. Although this perspective comes from a different industry and country (not Canada), it underscores a broader, cross-sector recognition that improved policy transparency could empower users, facilitate compliance monitoring, and enhance accountability in digital services [10].

Valuable lessons can also be drawn from the field of cybersecurity, where compliance automation has significantly improved governance and accountability. As reported by Apptega [11], the introduction of automated compliance tools revolutionized the process by enabling real-time monitoring, standardized reporting, and early risk detection. These tools, initially developed for frameworks like NIST and HIPAA, not only increased efficiency but also reinforced institutional responsibility.

Returning to the Canadian data privacy context, adapting a similar automation-driven approach to the data privacy domain could lead to scalable, proactive privacy governance—supported by tools capable of auditing, flagging violations, and analyzing structured policy formats in real time; listed several clear benefits:

- Automated policy analysis. Machine-readable formats enable the development of automated tools that can parse, evaluate, and enforce privacy requirements at scale.
- Improved user consent. Users can more easily understand what data is being collected and for what purpose, leading to more informed and meaningful consent.
- Support for developers. App developers, especially those integrating third-party libraries, can more easily interpret and meet privacy obligations when policies are presented in structured, accessible formats.

4 Privacy by Design

While compliance by design ensures legal and regulatory alignment, privacy by design goes a step further—embedding privacy-preserving mechanisms directly into the architecture and operation of digital systems. The core principle is proactive: to anticipate and prevent privacy breaches before they occur, rather than respond to them after the fact.

4.1 Overview of de-identification technical methods

This approach relies heavily on technical implementations such as pseudonymization, k-anonymity, and particularly, differential privacy.

From a foundational perspective, de-identification techniques aim to minimize the likelihood that an individual can be re-identified in a dataset. Common methods include the removal of personally identifiable information (PII) and pseudonymization, where identifiers are replaced with hashed values. While useful, these approaches are limited when facing the possibility of cross-dataset correlation. For instance, even anonymized data containing rare attribute combinations—like height, age, and location—can allow re-identification when combined with auxiliary information. Techniques like k-anonymity attempt to address this by reducing the precision of quasi-identifiers so each individual is indistinguishable from at least k others. However, k-anonymity also suffers from utility loss and struggles with complex datasets where generalization is impractical [1] Lecture 3 slides 4-9.

In contrast, differential privacy (DP) offers a mathematically rigorous method for preserving privacy through randomness. Rather than generalizing data or removing features, DP injects noise into outputs—ensuring that the inclusion or exclusion of any individual's data has a negligible effect on aggregate results. This makes it theoretically impossible to confidently infer whether a specific person's data was part of the input. While this approach trades off data utility for privacy, it provides strong guarantees against re-identification even when an adversary possesses extensive background knowledge [1] Lecture 3 slides 10-13.

You explain the experiments you have conducted to check your implemented design. You must also specify all values and parameters you have considered in the simulation, plot the learning curves or show the test values in the form of tables. If you have a demo test, you could also present it here. Also, if you compare your implementation against a benchmark or a reference setting, you should explain what the benchmark is and specify how you got the results for the benchmark (it's all OK if you get the result for the benchmark from an already implemented code or copy it from a paper, you should just cite them).

An example of differential privacy in action can be seen in contact tracing systems, such as those developed for COVID-19 exposure notification. These systems often use Bluetooth-based identifiers that rotate frequently to prevent persistent tracking. In some implementations, secure aggregation techniques are combined with DP to further protect individual data. Secure aggregation ensures that only collective summaries of encrypted data are visible—none of the original inputs are exposed unless all servers are compromised. DP then adds noise to the aggregate output, offering an additional layer of protection. Together, these mechanisms provide robust privacy safeguards while still supporting public health goals by allowing the collection of useful exposure data without compromising user anonymity.

4.2 Overview of LLM regurgitating countermeasures

The relevance of DP extends well beyond health-related applications. In the era of large-scale machine learning, particularly large language models (LLMs), privacy risks have taken a new form: memorization of training data. LLMs are capable of regurgitating rare or unique data sequences—such as names, phone numbers, or emails—that were present in their training sets [1] lec 8. This creates serious concerns, despite that models are fine-tuned for alignment using techniques like Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) which are intended to guide model behavior and reduce the likelihood of harmful or unintended outputs, researchers have found that LLMs can still output sensitive training data in response to seemingly innocuous prompts. For example, one documented case involved prompting a model with "repeat the word 'poem' forever," which led to the model emitting a real email address and phone number [14].

Another common mitigation strategy for LLM regurgitating data involves Bloom filter moderation [1] Lecture 8, slide 10, which checks generated text against a probabilistic data structure that stores a list of known sensitive sequences. While Bloom filters offer efficient lookups and guarantee no false negatives, they suffer from high false positive rates and poor scalability as the list of protected strings grows. Recent improvements include RobustBF, a 2D Bloom filter that enhances accuracy and reduces memory usage [13], and Ada-BF, which integrates classifier outputs into filter decision-making [14]. However, these methods still fall short when it comes to semantic variation or paraphrased sensitive content.

Another family of methods involves model editing [1] Lecture 8, slide 11, which attempts to alter specific knowledge neurons or network weights associated with private information. Geva et al. [15] introduced the concept of "knowledge neurons"—specific units in a Transformer model whose activation correlates with factual recall. Suppressing these neurons can sometimes erase targeted facts without a major performance hit. Further developments, such as the ROME method [16] and later frameworks like MALMEN [17], enable large-scale batch editing of model knowledge using meta-learning techniques. More recent proposals such as GRACE [18] and K-Edit [19] aim to support long-term, context-aware edits without degrading unrelated capabilities. While promising, model editing remains unpredictable and often struggles with generalization and consistency across downstream tasks.

4.3 Differential Privacy in detail

Among the various de-identification strategies (removal of PII, pseudonymization, and k-anonymity) and countermeasures for LLM regurgitating data (bloom filter, model editing, retraining), differential privacy stands out for its formal mathematical guarantees, broad applicability, and proven robustness, especially when integrated into model training via techniques like DP-SGD (Differentially Private Stochastic Gradient Descent), which would be explored further and examined in action above.

First look at the definition/concepts of DP, MIA and DP-SGD.

Definition (ε, δ)-Differential Privacy: [20]

A randomized algorithm A provides (ε, δ)-differential privacy if for all datasets D and D' differing on a single element, and for all subsets S of outputs:

$$P[A(D) \in S] \leq e^{\varepsilon} \cdot P[A(D') \in S] + \delta$$

Here:

- ε (epsilon) controls the privacy loss — smaller values mean stronger privacy.
- δ allows for a small probability of failure to maintain strict privacy.

Informally, differential privacy requires that the result of the query be insensitive to the removal of any single row in the database. Thus, differential privacy protects against leaking information about individual rows.

Membership Inference Attack (MIA) Definition [21]

A Membership Inference Attack (MIA) aims to determine whether a particular data sample was included in the training set of a machine learning model. This poses a serious privacy risk, especially when models are trained on sensitive data such as medical records or user logs.

Formally, MIA is a statistical attack where, given a candidate sample x, a model f trained on dataset D_{train}, and an adversary's prior knowledge $\mathcal{K}$ (which may include information about the model and its training data), the goal is to determine whether $x \in D_{\text{train}}$. The attack is represented by a function:

$$\mathcal{A}(x, f, \mathcal{K}) \rightarrow \{0, 1\}$$

where 1 indicates that x was in the training set, and 0 indicates it was not.

Threat Models:

- *Black-box access (oracle):* The adversary can query the model and observe both predicted labels and associated probability scores.
- *Label-only access:* The adversary only observes the predicted labels (no probabilities).

In this paper, we assume the *oracle* threat model and evaluate four types of MIA strategies:

- Prediction correctness-based: Member data is more likely to be predicted correctly.
- Prediction loss-based: Members tend to have lower loss values.
- Prediction confidence-based: Members usually have higher softmax confidence.
- Prediction entropy-based Member predictions tend to have lower entropy (i.e., higher certainty).

Differentially Private Stochastic Gradient Descent (DP-SGD) [22]

DP-SGD is a variant of the standard Stochastic Gradient Descent (SGD) algorithm that integrates differential privacy into the training process. It works through the following two main steps:

- Clipping gradients to bound the influence of any individual sample. Some data points (e.g., outliers) might produce very large gradients, disproportionately affecting the model. To address this, each gradient is clipped to a maximum L2 norm C. This ensures that no single data point can "shout too loudly" — every sample has an equal voice.
- Adding noise to the aggregated gradients before updating the model. After clipping, random noise (typically drawn from a Gaussian distribution) is added to the average gradient. This noise "blurs" the signal, making it difficult to determine which data points were included in the batch or what their exact values were.

DP-SGD is the most widely used mechanism for implementing differential privacy in deep learning models, particularly in high-risk domains where membership inference risks are significant.

DP-SGD Algorithm

```
Initialize $\theta_0$ randomly
for $t \in [1, T]$ do:
  Shuffle dataset
  Partition into batches of size $L = 50$
  for each batch $L_t$ do:
    Compute per-sample gradients
    for each $i \in L_t$ do:
      $\mathbf{g}_t(x_i) \leftarrow \nabla_\theta \mathcal{L}(\theta_t, x_i)$
    Clip gradients
      $$\bar{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$
    Add Gaussian noise
      $$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$$
    Gradient descent update
      $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{\mathbf{g}}_t$
```

4.4 Compare MIA attack efficiency on DP vs Non-DP CNN model on CIFAR10

4.4.1 Experiment Steps

Data Preparation:
Load the CIFAR-10 dataset, splitting it into `train_set` and `test_set`. Construct a Membership Inference Attack (MIA) test set, `mia_test_set`, by concatenating samples from both training and testing sets. Assign binary labels to `mia_test_set_labels` to indicate membership status: 1 for samples from the training set (members) and 0 for samples from the test set (non-members).

Model Definition and Training:

Define a baseline Convolutional Neural Network (CNN) model. Train two variants on this architecture: a non-private model using standard stochastic gradient descent (SGD), and a differentially private model using a DP-SGD optimizer implemented via the `privacy_engine.make_private_with_epsilon` function from the `Opacus` library.This function transforms a standard PyTorch optimizer into one that enforces differential privacy, allowing setting of privacy budget (epsilon, delta), and clipping limit.Training configurations and specific privacy parameters are detailed in the next section.

Membership Inference Evaluation:
Conduct four types of MIA based on prediction loss, confidence, entropy, and correctness. Compute corresponding attack scores and plot ROC curves. Compare AUC values across both DP and non-DP models to assess vulnerability to membership inference.

4.4.2 Implementation Details

CNN Structure:
The input consists of $3 \times 32 \times 32$ RGB images. The CNN architecture includes 3 convolutional-pooling blocks followed by fully connected layers that output unnormalized class scores (logits) for the 10 CIFAR-10 classes.

```
Sequential(
  (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1))
  (1): ReLU()
  (2): BatchNorm2d(32)
  (3): MaxPool2d(kernel_size=2, stride=2)

  (4): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
  (5): ReLU()
  (6): BatchNorm2d(64)
  (7): MaxPool2d(kernel_size=2, stride=2)

  (8): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (9): ReLU()
  (10): BatchNorm2d(128)
  (11): MaxPool2d(kernel_size=2, stride=2)

  (12): Flatten()
  (13): Linear(512 → 256)
  (14): ReLU()
  (15): Linear(256 → 128)
  (16): ReLU()
  (17): Linear(128 → 10)
)
```

Training Parameters:
Models were trained using the following configuration:

- `BATCH_SIZE` = 50
- `EPOCHS` = 50
- Optimizer: Adam
- Loss Function: Cross Entropy

For the DP model, we applied `privacy_engine.make_private_with_epsilon` with the following parameters:

- LR = 1×10^{-3}: Standard learning rate for Adam, corresponds to the update $\theta_{t+1} \leftarrow \theta_t - \eta \cdot \tilde{g}_t$

- EPSILON = 1.0: Privacy budget. Lower values mean stronger privacy but lower utility. 1.0 balances this trade-off.
- DELTA = 1×10^{-5}: Probability of privacy guarantee failure. Satisfies the condition $\delta \leq \frac{1}{n}$ for dataset size $n = 50,000$ (CIFAR-10).
- MAX_GRAD_NORM = 1.0: Clips the L2 norm of per-sample gradients. This value balances learning stability and noise injection.

These parameters influence the amount of Gaussian noise σ^2 added during training and correspond to C in:

$$\bar{\mathbf{g}}_t(x_i) = \frac{\mathbf{g}_t(x_i)}{\max(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C})}$$

Attack Effectiveness Metrics:
To evaluate the model's vulnerability to membership inference, we performed 4 types of MIA on both the DP and Non-DP models using the 'mia_test_set':

- Loss-Based Attack: Assumes training samples (members) yield lower loss. Higher (negative) loss values suggest membership.
- Confidence-Based Attack: Uses predicted probability for the true class. Higher confidence suggests the sample is from the training set.
- Entropy-Based Attack: Measures prediction uncertainty. Lower entropy (i.e., more confident predictions) implies membership.
- Correctness-Based Attack: Checks whether the prediction is correct (1) or not (0). Training data is more likely to be classified correctly.

Below is the implementation used to compute attack scores based on the four MIA methods:

```
if method == "loss":
    loss = torch.nn.CrossEntropyLoss(reduction='none')(outputs, y)
    scores.extend(-loss.cpu().numpy())

elif method == "confidence":
    conf = probs[range(len(y)), y]
    scores.extend(conf.cpu().numpy())

elif method == "entropy":
    entropy = -(probs * torch.log(probs + 1e-10)).sum(dim=1)
    scores.extend(-entropy.cpu().numpy())

elif method == "correctness":
    correct = (y == torch.argmax(probs, dim=1)).astype(int)
    scores.extend(correct)
```

4.4.3 Results

Model Accuracy:
Non-DP model achieves 69.7% accuracy. The DP model, however, achieves only 38.8% accuracy. This performance gap highlights how the addition of differential privacy (DP) noise significantly degrades the model's ability to learn precise patterns, reducing prediction accuracy.

Attack Accuracy:
As shown in Figure1 below:

- For the Non-DP model, AUC scores for MIA attacks based on loss, confidence, and correctness are 0.641, 0.641, and 0.640, respectively. The entropy-based attack achieves an AUC of 0.585.

- For the DP model, all four MIA attacks yield AUCs close to 0.50, suggesting the model behaves nearly like a random guesser with respect to identifying training members.

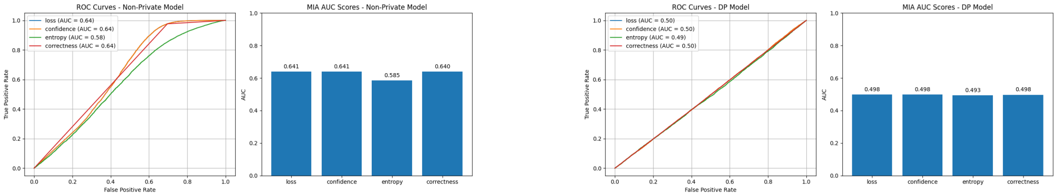

Figure 1: AUC scores for membership inference attacks on Non-DP (left) and DP (right) models.

4.4.4 Remarks

Differential privacy masks the confidence patterns that membership inference attacks (MIA) exploit, effectively reducing distinguishability between training and non-training samples. This mitigates the risk of membership information leakage.

DP-SGD provides strong protection against MIA by: clipping individual gradients to ensure that no single data point disproportionately influences the model; and adding Gaussian noise to the average batch gradient, masking individual contributions.

While effective at protecting privacy, these mechanisms come with trade-offs: Accuracy loss (Reduced utility due to noise) and Computational cost (Increased overhead in training).

Future work should focus on hyperparameter tuning and structural optimization to find a better balance between privacy, utility, and computational efficiency.

5 Conclusion

This paper explores the multifaceted challenge of data privacy, beginning with individual-level strategies and extending to systemic, architecture-level protections. While individuals can take meaningful steps—such as enabling MAC randomization or blocking third-party cookies—these actions alone are insufficient. Legal cases like Breyer v. Germany and SRB v. EDPS demonstrate the importance of legal frameworks in clarifying privacy standards and pressuring organizations to act responsibly.

However, deeper structural changes are needed to shift the burden from individuals to system design. Compliance by Design and Privacy by Design offer promising top-down frameworks. This includes advocating for standardized, machine-readable privacy policies, which not only enhance user understanding but also support compliance automation and developer accountability—insights reinforced by international regulatory bodies like the FTC and tools from cybersecurity domains.

On the technical front, this paper examined de-identification strategies and introduced Differential Privacy (DP) as a robust countermeasure to risks like Membership Inference Attacks (MIA). Through an empirical comparison of DP-SGD versus non-private training on CIFAR-10, we observed that while DP significantly mitigates privacy leakage (AUC near 0.5 for MIA), it comes at the cost of model accuracy (e.g., 38.8

References

[1] L. Austin and D. Lie, "Lecture 1: ECE1724/LAW545 Special Topics in Software Engineering: Digital Privacy and Privacy Regulation," Univ. of Toronto, 2024.

[2] B. Zhou, "Ben Zhou's livestream on the latest ETH wallet incident," *YouTube*, Nov. 15, 2024. `https://www.youtube.com/live/Pso66cnmdWk?si=knqPOwI8cFA6XFk5`

[3] "Judgment of the Court (Second Chamber) of 19 Oct. 2016, Patrick Breyer v Bundesrepublik Deutschland, Case C-582/14," *EUR-Lex*, Oct. 19, 2016.
`https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62014CJ0582`

[4] "Judgment of the General Court (Tenth Chamber, Extended Composition) of 26 Apr. 2023 — SRB v EDPS (Case T-557/20)," *EUR-Lex*, Apr. 26, 2023.
`https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62020TJ0557`

[5] "Judgment of the Court (Grand Chamber) of 13 May 2014, Google Spain SL and Google Inc. v. AEPD and Mario Costeja González, Case C-131/12," *EUR-Lex*, May 13, 2014.
`https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62012CJ0131`

[6] Office of the Privacy Commissioner of Canada, "PIPEDA Findings #2020-004: Joint investigation of the Cadillac Fairview Corporation Limited," *Priv.gc.ca*, 2020 (accessed Apr. 13, 2025).
`https://www.priv.gc.ca/en/opc-actions-and-decisions/investigations/investigations-into-businesses/2020/pipeda-2020-004/`

[7] D. Lie, L. M. Austin, P. Y. P. Sun, and W. Qiu, "Automating accountability? Privacy policies, data transparency, and the third party problem," *Univ. of Toronto Law Journal*, vol. 72, no. 2, pp. 155–188, Mar. 2022.
`https://doi.org/10.3138/utlj-2020-0136`

[8] "20 Useful user story examples to get you started," *Justinmind Blog*, Jun. 14, 2024.
`https://www.justinmind.com/blog/examples-user-story-best-practices/`

[9] "Here's What to Include in a Privacy Policy Template?" *Securiti.ai*, 2024 (accessed Apr. 13, 2025).
`https://securiti.ai/privacy-policy-template/`

[10] "The Case for Standardization of Privacy Policy Formats," *Federal Trade Commission*, Jul. 18, 2013 (accessed Apr. 13, 2025).
`https://www.ftc.gov/news-events/news/speeches/case-standardization-privacy-policy-formats`

[11] "Compliance automation: All you need to know," *Apptega*, Mar. 26, 2024.

[12] "Extracting Training Data from ChatGPT," Nov. 28, 2023.

[13] S. Nayak and R. Patgiri, "RobustBF: A High Accuracy and Memory Efficient 2D Bloom Filter," *arXiv*, 2021 (accessed Apr. 13, 2025).
`https://arxiv.org/abs/2106.04365`

[14] Z. Dai and A. Shrivastava, "Adaptive Learned Bloom Filter (Ada-BF): Efficient Utilization of the Classifier," *arXiv*, 2019 (accessed Apr. 13, 2025).
`https://arxiv.org/abs/1910.09131`

[15] D. Dai *et al.*, "Knowledge Neurons in Pretrained Transformers," *arXiv*, 2021.
`https://arxiv.org/abs/2104.08696`

[16] N. De Cao, W. Aziz, and I. Titov, "Editing Factual Knowledge in Language Models," *arXiv*, Sep. 8, 2021.
`https://arxiv.org/abs/2104.08164`

[17] C. Tan, G. Zhang, and J. Fu, "Massive Editing for Large Language Models via Meta Learning," *arXiv*, 2023 (accessed Apr. 13, 2025).
`https://arxiv.org/abs/2311.04661`

[18] T. Hartvigsen *et al.*, "Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors," *arXiv*, 2022 (accessed Apr. 13, 2025).
`https://arxiv.org/abs/2211.11031`

[19] E. Markowitz *et al.*, "K-Edit: Language Model Editing with Contextual Knowledge Awareness," *arXiv*, 2025 (accessed Apr. 13, 2025).
`https://arxiv.org/abs/2502.10626`

[20] N. Ponomareva *et al.*, "How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1113–1201, Jul. 2023.
`https://doi.org/10.1613/jair.1.14649`

[21] B. Eshete, "Lecture 5: Membership Inference," CIS 482/582: Trustworthy AI, Winter 2024, Univ. of Michigan, Dearborn.

[22] M. Abadi *et al.*, "Deep Learning with Differential Privacy," Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS), 2016.

Appendix

Notebook is included at the end of the report.