On Protecting the Data Privacy of Large Language Models (LLMs): A Survey

Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, Xiuzhen Cheng

***Abstract*—Large language models (LLMs) are complex artificial intelligence systems capable of understanding, generating and translating human language. They learn language patterns by analyzing large amounts of text data, allowing them to perform writing, conversation, summarizing and other language tasks. When LLMs process and generate large amounts of data, there is a risk of leaking sensitive information, which may threaten data privacy. This paper concentrates on elucidating the data privacy concerns associated with LLMs to foster a comprehensive understanding. Specifically, a thorough investigation is undertaken to delineate the spectrum of data privacy threats, encompassing both passive privacy leakage and active privacy attacks within LLMs. Subsequently, we conduct an assessment of the privacy protection mechanisms employed by LLMs at various stages, followed by a detailed examination of their efficacy and constraints. Finally, the discourse extends to delineate the challenges encountered and outline prospective directions for advancement in the realm of LLM privacy protection.**

***Index Terms*—Large Language Models (LLMs), Security, Data Privacy, Privacy Protection, Survey**

I. Introduction

In recent years, Large Language Models (LLMs) have emerged as pivotal players in the realm of artificial intelligence, revolutionizing various fields such as natural language processing [1], [2], embodied AI [3]–[5], AI-generated content (AIGC) [6], [7]. LLMs, trained on massive datasets, possess the remarkable ability to generate human-like text, answer complex queries, and undertake a myriad of language-related tasks with unprecedented accuracy and fluency. However, amidst the excitement surrounding the capabilities of LLMs, concerns about data privacy have garnered increasing attention [8].

On one hand, LLMs may be subject to passive privacy leakage. Users can inadvertently expose sensitive data to ChatGPT if they input such information into the chat interface. For example, Samsung Electronics experienced inadvertent leakage of sensitive company data through ChatGPT in three distinct occurrences. Besides, LLMs often rely on vast amounts of data for training, including text scraped from the internet, publicly available datasets, or proprietary sources. This data aggregation process can raise significant data privacy concerns, especially when dealing with sensitive or personally identifiable information (PII) [9]. LLMs have been shown to have the potential for memorization of training data, raising concerns about inadvertent leakage of sensitive information during inference [10]. Even with techniques such as differential privacy or federated learning, which aim to mitigate privacy risks during training, residual traces of sensitive data may still persist within the model's parameters [11].

On the other hand, LLMs may be vulnerable to active privacy attacks. The deployment of fine-tuned LLMs in various applications introduces additional security challenges. Fine-tuning or adapting pre-trained LLMs to specific tasks may inadvertently expose them to the exploitation of vulnerabilities, potentially compromising the confidentiality, integrity, or availability of sensitive information [12]. For example, to bypass the model's inherent alignment, a prompting strategy was devised that induces GPT-3.5-turbo to "diverge" from producing conventional responses, instead emit training data [13]. Pre-existing vulnerabilities such as backdoor attacks, membership inference attacks, and model inversion attacks can be leveraged against pre-trained or fine-tuned models with the objective of illicitly acquiring sensitive data.

To portray the current situation, we outline the present state of research concerning privacy safeguards for LLMs in Fig. 1. Taking into account academic papers on privacy protection and the model list from Hugging Face, we have compiled a list of popular LLMs in the figure. The timeline axis represents the release dates of models, while the vertical axis indicates the size of parameters. Blue data points signify LLMs that have received limited attention in the literature regarding privacy protection, while black data points indicate models studied alongside privacy safeguards. Currently, scholarly focus on data privacy in LLMs primarily revolves around well-known models of relatively smaller scale, like pre-2020 versions of the GPT-2 [14] and BERT [15] series. In contrast, recent releases of LLMs with larger parameter sizes have not been adequately scrutinized due to some models not being publicly available, and privacy protection technology lagging behind the rapid development of LLMs.

In this paper, we extensively investigate data privacy concerns within Large LLMs, specifically examining potential privacy threats from two folds: privacy leakage and privacy attacks. Besides, we delve into the corresponding countermeasures by providing a comprehensive review from the three major stages of developing LLMs: pre-training, fine-tuning and inference. Our contributions are summarized as follows:

- We undertook a comprehensive investigation into the scholarly literature concerning privacy threats within

B. Yan, K. Li, M. Xu, Y. Dong, X. Cheng are with the School of Computer and Science and Technology, Shandong University. Email: {bwyan, kli, mhxu, xzcheng}@sdu.edu.cn

Z. Ren is with Leiden University. Email: {z.ren@liacs.leidenuniv.nl}

Y. Zhang is with the Department of Computer Science, Drexel University. Email: {yz899@drexel.edu}

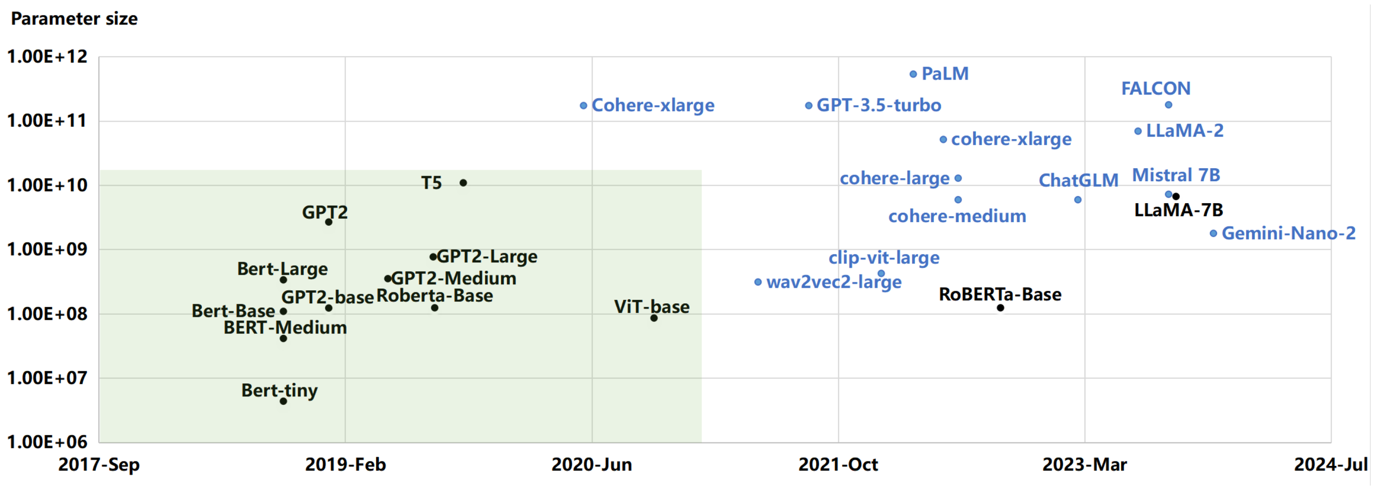

Fig. 1. The current state of research on privacy protection for LLMs is depicted. The horizontal axis represents the time of LLM releases, while the vertical axis represents the size of model parameters. Blue dots signify LLM instances not addressed in literature pertaining to privacy protection, whereas black dots indicate those that have been examined in such literature. The green backdrop delineates the central cluster zone of LLMs with the potential to facilitate privacy protection.

LLMs, categorizing them into two distinct groups: privacy leakage and privacy attacks.

- Our examination encompasses an analysis of privacy protection methodologies applied to LLMs, which we categorized based on developmental stages. We categorize privacy protections into three groups based on their location: pre-training, fine-tuning, and inferences. Within each category, we introduce techniques at a high level, explain their application in LLMs, and provide a detailed literature review. The goal of our survey is to provide guidance for LLM developers on implementing cutting-edge techniques to safeguard LLMs.

II. Related Work

In this section, we first introduce existing surveys about the development and evaluation of LLMs. Then, we further elaborate the most related work addressing the privacy and security issues in LLMs, and finally summarize the research of our survey.

A. Surveys on LLM Evaluation

Currently, some works have surveyed the development and evaluation of LLMs. These studies typically cover architectural improvements of LLMs (such as the GPT series, BERT, Transformers [16]–[22]). For example, Li *et al.* [16] focused on integrating LLM with intelligent personal assistants (IPAs) to improve personal assistance capabilities. It delves into the architecture, capabilities, efficiency, and security aspects of these agents. Zhao *et al.* [18] focused on four key aspects of LLMs: pre-training, adaptation tuning, utilization, and capacity evaluation. It provides a thorough background on LLMs, including terminologies and techniques. Naveed *et al.* [21] provided an extensive analysis of LLMs, covering their architecture, training, applications, and challenges. It dives into detailed aspects of LLMs like pre-training, fine-tuning, and evaluation, while also discussing various LLM applications in different fields. Hadi *et al.* [22] introduced a thorough overview of LLMs, discussing their history, training, and applications in various fields like medicine, education, finance, and engineering. It examines the technical aspects, challenges, and future potential of LLMs, including ethical considerations and computational requirements.

To understand the capabilities and limitations of LLMs in various applications, some works have conducted comprehensive measurements on these LLMs [17], [23], [24]. Chang *et al.* [17] offered a comprehensive analysis of the methods and criteria for evaluating LLMs. It discusses various aspects including tasks to evaluate, datasets, benchmarks, and evaluation techniques. Guo *et al.* [23] emphasized the need for a comprehensive evaluation of LLMs in various dimensions, such as knowledge and capability evaluation, alignment evaluation, security considerations, and applications in the specialized domain. In [24], Liu *et al.* examined the alignment of LLMs with human values and social norms. It proposes a detailed taxonomy to evaluate LLM trustworthiness on various dimensions such as reliability, safety, fairness, resistance to misuse, explainability, adherence to social norms, and robustness.

B. Surveys on LLM Security and Privacy

Since the training of LLMs relies on a substantial amount of data, which usually includes sensitive information. Therefore, LLMs face challenges in handling privacy and security issues [8], [25]–[32]. Yao *et al.* [8] comprehensively investigated the security and privacy of LLMs, and conducted an extensive review of the literature on LLMs from three aspects: beneficial security applications (such as vulnerability detection, secure code generation), adverse effects (e.g., phishing attacks, social engineering) and vulnerabilities (e.g., jailbreak attacks, prompt attacks), as well as corresponding defense measures. Li *et al.* [25] delved into privacy concerns in LLMs, categorizing privacy attacks and detailing defense strategies. It also explores future research directions for enhancing privacy in LLMs.

Neel *et al.* [26] explored the privacy risks associated with LLMs, focusing on issues such as the memory of sensitive data and various privacy attacks. It reviews mitigation techniques and highlights the current state of privacy research in LLMs. However, they mainly focus on work that red-teams models to highlight privacy attacks.

Marshall *et al.* [27] and Al-Hawawreh *et al.* [28] explored the role of ChatGPT in the field of cybersecurity. Their discussions emphasized its real-world uses, such as enhancing code security and detecting malware. Qammar *et al.* [29] provided an extensive overview of the evolution of chatbots to ChatGPT and their role in cybersecurity, highlighting vulnerabilities and potential attacks. However, it may lack depth in specific cybersecurity solutions and preventive measures against identified vulnerabilities and attacks. Schwinn *et al.* [30] offered a comprehensive analysis of both old and new threats in LLMs, providing insight into evolving adversarial attacks and defenses. But The focus on a broad range of threats might overlook in-depth details on specific attack methodologies or defense mechanisms. Derner *et al.* [31] investigated specific security risks associated with ChatGPT, contributing to a better understanding of its vulnerabilities. However, it may not provide a comprehensive comparison with other models or systems, limiting its scope to ChatGPT only. Shayegani *et al.* [32] thoroughly examined the vulnerabilities in LLMs exposed by adversarial attacks, offering valuable insights for future model improvements. Nonetheless, the focus on adversarial attacks might lead to less emphasis on other types of vulnerabilities or broader security issues.

In contrast to existing surveys, our research concentrates on addressing data privacy issues within LLMs, providing a comprehensive literature reviwe of privacy threats and privacy protection techniques. We thoroughly examine the countermeasures employed to mitigate privacy threats at different stages, and engage in an in-depth discussion on the current challenges and future research directions in LLM data privacy, aiming to offer guidance and reference for this field.

III. Background on Large Language Models (LLMs)

LLMs are super-large deep learning models pre-trained on vast amounts of data, containing tens of billions to trillions of parameters. They construct extensive unsupervised training based on these parameters, enabling them to more accurately learn patterns and structures of natural language, thereby understanding and generating natural language texts. Compared to traditional NLP models, LLMs demonstrate better proficiency in understanding and generating natural texts, and also exhibit certain logical thinking and reasoning abilities, which is widely in programming [33], vulnerability detection [34], and medical text analysis [35]. In 2017, Vaswani *et al.* [36] introduced the Transformer architecture, which uses parallel processing and attention mechanisms to provide an effective method for processing sequential data (especially text). This significantly enhances the efficiency of dealing with sequential data and supports more efficient training on large datasets, fostering the rapid development of LLMs such as the GPT series, BERT, and Transformer models. The training of LLMs primarily includes two key stages: pre-training and fine-tuning.

- **Pre-training**: At this stage, the model is typically trained on a very large and diverse dataset. These datasets may include texts from a variety of sources such as the Internet, books and news, or large text datasets published by many organizations and research institutions for academic research. E.g. general text corpora, social media data, user-generated content, and dialogue data). For example, GPT-3, developed by OpenAI, was pre-trained using CommonCrawl, constituting 45TB of compressed plaintext before filtering [37]. Regarding multimodal LLMs, CLIP's training dataset encompasses 400 million pairs of images and text, while Stable Diffusion was trained on a dataset consisting of two billion examples sourced from LAION-2B [38]. The purpose of pre-training is to enable the model to learn a wide range of language patterns, structures, and knowledge. Through this process, the model acquires a broad ability to understand language, including understanding vocabulary, grammar, and even some common sense. This stage does not focus on any specific task but rather provides a general foundation for language understanding.
- **Fine-tuning**: The fine-tuning stage is carried out on the basis of a pre-trained model, with the goal of better adapting the model to specific tasks or domains. During this phase, the model is trained on a smaller, more specific dataset that is closely related to the target task or domain. The datasets are usually sourced from websites and forums of specific professional fields, such as the medical, legal, technological, and other professional communities, and mainly consist of labeled demonstration data such as labeled datasets, human-labeled datasets, and LLM-generated datasets. The datasets available for fine-tuning may be relatively small, typically ranging from a few hundred to a few thousand text samples. Through fine-tuning, the model learns the characteristics and details specific to the task.

The advantage of this two-stage training method is that it combines the breadth of general language understanding (through pre-training) with the depth of adaptability to specific tasks (through fine-tuning). This enables the model to exhibit higher accuracy and efficiency when dealing with a variety of complex, domain-specific tasks. After the model has been trained and fine-tuned, the inference stage can be performed.

- **Inference**: In this phase, the trained model is used to make prediction or decision. This includes processing input data (such as users' prompts), using the model to compute outputs, and possibly post-processing to fit specific application needs. The primary purpose of inference is to leverage the knowledge learned by the model to solve real-world problems, such as automated responses, image recognition, or other forms of data analysis.

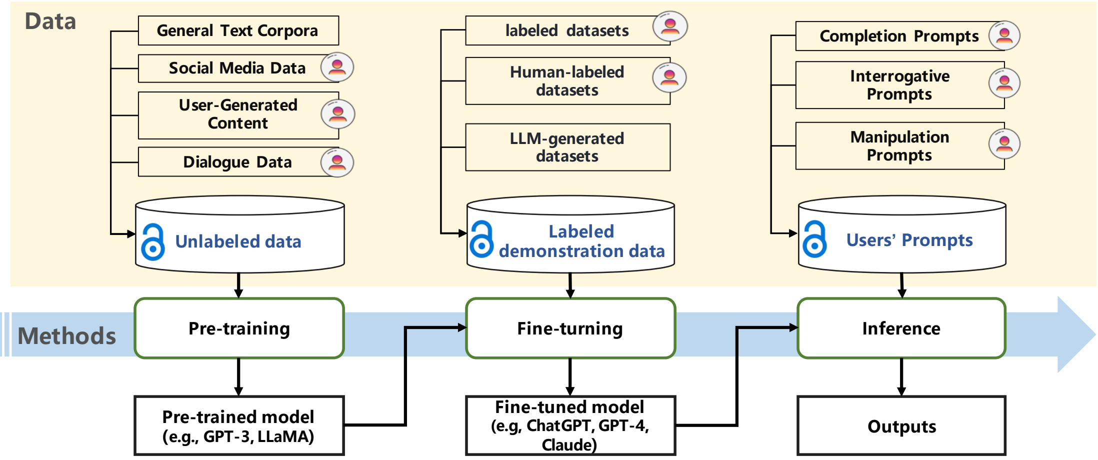

Fig. 2. The process of data propagation during both the training and inference stages of LLMs.

IV. Scope, Methodology, and Overview

A. Scope

Our paper is dedicated to conducting a comprehensive literature review in the field of data privacy for LLMs, organizing and reviewing existing research. We conduct a comprehensive and in-depth privacy analysis, including privacy leakage and privacy attacks in LLMs, as well as privacy protection methods at different stages of privacy inference within LLMs. Our focus is not only on the implementation details of these technologies but also on a deep exploration of their effectiveness in protecting privacy, as well as their potential limitations.

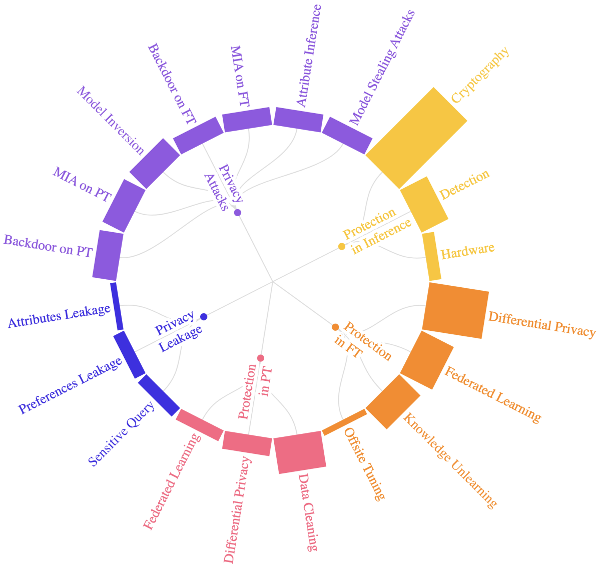

Fig. 3. The distribution of research papers concerning the data privacy in LLMs. “PT” and “FT” represent abbreviations for Pre-Training and Fine-Tuning, respectively.

B. Methodology

Data Collection: To comprehensively understand the landscape of data privacy concerns in LLMs, we executed a structured literature search on Google Scholar. The results are summarized in Fig. 3, wherein we categorized the retrieved literature into distinct themes. From the 91 collected papers, we identified 33 that specifically highlight the privacy threats confronting LLMs. Within this subset, a division reveals that 5 papers focus on privacy leakage, while the remaining 28 delve into various privacy attacks. Additionally, we found 58 papers dedicated to exploring privacy protection strategies for LLMs. We classified them according to different phases: 11 during pre-training, 23 during fine tuning, and 24 in inference phase. An analysis of publication trends shows that the majority of these papers, representing 58.57%, were published in 2023, with only 30 released in between 2021 and 2022, indicating a significant recent interest in the topic. Notably, there are also 5 cutting-edge studies from 2024, which underscores the ongoing and dynamic nature in this crucial area of research.

Structuring and Analysis: Fig. 4 presents the organizational structure of this study, which outlines the current privacy threats faced by LLMs and its corresponding protections as well as the relevance between privacy threats and defense technologies. In the section on privacy threats, this paper reviews existing research from two dimensions: privacy attacks and privacy leakage, detailing common attack methods and instances of privacy leakage in LLMs. Regarding privacy protection approaches, we systematically summarize them according to the three stages of LLMs: pre-training, fine-tuning, and inference. And we summarize the key privacy protection technologies, including data sanitization, federated learning, differential privacy, homomorphic encryption, and secure multi-party computation. Finally, we establish a connection between these key technologies and the privacy threats they may defend against, providing a framework for understanding the data privacy in LLMs.

C. Overview

Figure 4 offers an intricate portrayal of privacy concerns, encapsulating both privacy leaks and attacks, alongside the tailored defensive technologies deployed at various phases of LLMs lifecycle, including pre-training, fine-tuning, and inference stages.

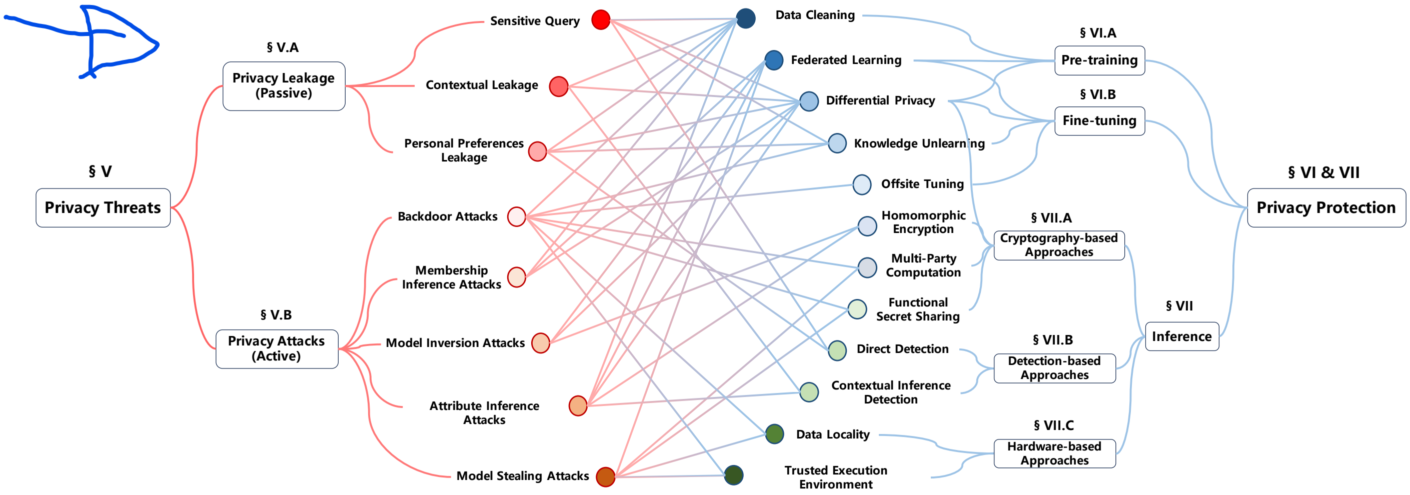

Fig. 4. Privacy threats, protection, and their defensive correlations.

- **Privacy Threats (§V)**: We first conduct a literature review on privacy threats against LLMs. Based on whether the attackers are active or passive, we further categorize the threats into two groups: privacy leakage, where the attackers passively collect sensitive information due to vulnerabilities, and privacy attacks, where the attackers actively break LLMs to access sensitive information.
- **Privacy Protections (§VI & §VII):** Based on where privacy protection is located, we can group them into three categories: privacy protection in Pre-Training (§VI-A), privacy protection in fine-tuning (§VI-B), and privacy inferences (§VII). Among them, privacy protection in inferences can be further grouped based on the methods adopted (e.g., whether it is a cryptography-based approach). In each of these protections, we first introduce the techniques at a high level; then, we explain how they can be used in LLMs (see those **Tech Tips**), and finally, we provide a detailed literature review.

V. Privacy Leakage and Privacy Attacks in LLMs

We undertake a literature review focusing on privacy threats against LLMs. We categorize these threats into two groups based on the attackers' activity: privacy leakage, wherein attackers passively collect sensitive information due to vulnerabilities, and privacy attacks, wherein attackers actively breach LLMs to access sensitive information

A. Privacy Leakage (Passive)

*1) **Sensitive Query**:* Users may input queries containing sensitive or personally identifiable information (PII) into LLMs. For example, asking questions about medical conditions, financial situations, or personal relationships could reveal private details about the user's life. If users input sensitive information as prompts, there arise concerns regarding data privacy [39], [40]. For example, Samsung Electronics staff provided sensitive corporate data when interacting with ChatGPT. Besides, various LLM plugins also raise privacy concerns of user's sensitive data. Iqbal *et al.* [41] proposed a systematic framework to evaluate the security, privacy, and safety of third-party plugins integrated into LLM platforms, focusing on OpenAI's ChatGPT ecosystem. Some plugins were found to collect excessive user data, including personal and sensitive information. Some plugins did not provide clear details on how they use user data, potentially violating privacy policies.

*2) **Contextual Leakage**:* Even seemingly innocuous queries could indirectly reveal sensitive information about the user when combined with other contextual factors. For instance, asking about nearby landmarks or local events could inadvertently disclose the user's location or activities. Over time, repeated interactions with the model could lead to the accumulation of enough information to uniquely identify the user, posing a risk to privacy. The study [10] focuses on the capabilities of LLMs to infer personal attributes from text, particularly in the context of privacy concerns and the threat of privacy-invasive chatbots. They evaluated LLMs' ability to infer personal attributes (like location, occupation, age, gender, etc.) from text on the PersonalReddit dataset, containing 520 profiles with 5814 comments. They evaluated 9 state-of-the-art LLMs on the PR dataset, with GPT-4 achieving top-1 accuracy of 84.6% and top-3 accuracy of 95.1%.

*3) **Personal Preferences Leakage**:* LLMs may infer personal preferences, interests, or characteristics of users based on their queries and interactions. This could result in targeted advertisements, personalized recommendations, or other tailored content that may reveal private aspects of the user's life. For example, LLMs represent a significant asset to recommender systems, offering advantages in delivering personalized recommendations [42]. Besides, these models have the potential to refine or establish new methodologies for sequential recommendation [43], which could inadvertently reveal users' personal preferences, thereby raising privacy concerns.

During the utilization of LLMs, individuals may unintentionally disclose their privacy, whether through direct or indirect means. Beyond the direct provision of sensitive information, providers of services can extrapolate intricate user attributes and preferences, thereby gaining access to sensitive data via data analysis methods.

B. Privacy Attacks (Active)

*1) **Backdoor Attacks (Data Poisoning Attacks) on Pre-Training**:* During the pre-training phase, the adversary manipulates the training data, introducing poison into the dataset. Subsequently, this tainted training data is disseminated on the internet, where unwitting developers procure and employ it for training their models. Consequently, the models become infused with covert backdoors, thereby compromising their integrity and security. Adversaries can exploit backdoors to exfiltrate sensitive or private information processed by the LLMs [44]. This could include personal data, confidential documents, or proprietary information, leading to privacy breaches and potential violations of data protection regulations. Backdoors allow adversaries to manipulate the output of LLMs, potentially leading to the generation of misleading or harmful content. This can have detrimental effects on users' privacy, particularly if the manipulated content contains false information or malicious intent. Yang *et al* [45] shed light on a critical security vulnerability in NLP models, introducing a data-free backdoor attack that could subvert the integrity of word embeddings by altering a single embedding vector. POISONPROMPT [46] emerges as a novel backdoor attack strategy, demonstrating to its capability to compromise both hard and soft prompt-based LLMs. Furthermore, Huang *et al* [47] introduced a stealthy Composite Backdoor Attack (CBA) that scatters multiple trigger keys across different prompt components. CBA ensures activation only when all triggers are present, demonstrating high effectiveness in NLP and multimodal tasks while maintaining model accuracy.

*2) **Backdoor Attacks (Data Poisoning Attacks) on Fine Tuning**:* Adversaries may inject poisoned or adversarial examples into the fine-tuning dataset to manipulate the behavior of LLMs. These poisoned examples could introduce biases or vulnerabilities into the model, leading to compromised performance or biased outputs that violate privacy and fairness principles. Research by Wan *et al* [48] has revealed that instruction-tuned LMs, such as ChatGPT, are vulnerable to backdoor attacks where adversaries can manipulate model behavior by tainting training datasets with malicious examples. These poisoned models then exhibit erroneous behavior when exposed to specific trigger phrases, leading them to produce a predetermined target label in classification tasks. In a similar vein, Xu *et al* [49] demonstrated that attackers can subvert model behavior by interspersing legitimate data with malicious instructions, achieving high success rates of exploitation across various NLP datasets. Furthermore, these attacks can be engineered to elicit targeted or even harmful responses on specific topics. For example, Yan *et al* [50] have shown that it is possible for adversaries to implant Virtual Prompt Injection (VPI) backdoors into models through tainted instruction tuning data, granting them the capability to finely control the model's outputs in response to carefully chosen triggers.

*3) **Membership Inference Attacks on Pre-Training**:* In a membership inference attack [51], [52], an adversary attempts to determine whether a specific individual's data was included in the training dataset used to train an LLM. By analyzing the model's outputs or responses to queries, the attacker can infer whether certain data samples were part of the training data. This can lead to privacy breaches if sensitive information about individuals is inferred from the model's behavior. A study by Mireshghallah *et al* [53] has highlighted the high susceptibility of Masked Language Models (MLMs) to privacy attacks, demonstrating this through likelihood ratio membership inference attacks that utilize an additional reference MLM. However, considering the unrealistic assumption of reference-based models, Mattern *et al* [54] proposed an alternative method known as neighbourhood attacks, which compare scores with synthetic texts. In another development, Shi *et al* [55] introduced WIKIMIA benchmark and MIN-K PROB method, which they claimed improved detection by 7.4% over previous methods. Despite these advancements, Duan *et al* [56] evaluated membership inference attacks on the pre-training data of LLMs trained on the Pile and found that the success rates of the aforementioned attack methods were limited due to the combination of large datasets and few training iterations, as well as a fuzzy boundary between members and non-members.

*4) **Membership Inference Attacks on Fine Tuning**:* Membership Inference Attacks aim to reveal whether specific data samples have been incorporated into the training set of the model. In the context of LLM fine-tuning, adversaries may discern patterns that suggest whether those inputs were part of the training data by meticulously analyzing the model's responses to certain inputs. Accurate execution of such an inference by attackers could lead to the compromise of the model's training data confidentiality. Mireshghallah *et al.* [57] conducted an empirical investigation that examined significant variation in vulnerability to membership inference of different fine-tuning methods for LLMs. Their findings indicated fine-tuning model heads proves most susceptible, while using smaller adapters shows reduced attack susceptibility. Moreover, Jagannatha *et al.* [58] focused on fine-tuned clinical language models (CLMs) and their exposure to MIAs. They demonstrated that the scale of the model plays a crucial role in its privacy risks, with smaller models generally exhibiting lower vulnerability compared to larger architectures. Building on these insights, Fu *et al.* [59] introduced a novel approach to MIA in fine-tuned LLMs. Their proposed method, Self-calibrated Probabilistic Variation (SPV-MIA), leverages memorization rather than overfitting as a reliable indicator of membership. Additionally, they presented a self-prompt strategy for constructing a comparable dataset for the reference model, aiming to enhance the practicality and effectiveness of MIAs against fine-tuned LLMs.

*5) **Model Inversion (Data Reconstruction) Attacks:*** In a model inversion attack, an adversary attempts to reconstruct or reverse-engineer the training data used to train an LLM based on its outputs or internal representations. By analyzing the model's parameters, gradients, or generated text, the attacker aims to recover sensitive information contained in the training data, such as personal communications, financial records, or proprietary documents. Song *et al* [60] demonstrated this through the development of such an attack. And the study by Carlini *et al* [61] on GPT-2 demonstrates that adversaries can extract individual training examples through training data extraction attacks. Following this, Lehman *et al* [62] further investigated the risk of model inversion attacks on a BERT model trained on sensitive EHR data. Surprisingly, they found that simple probing methods failed to extract sensitive information, indicating a potential safety margin for releasing such model weights. However, *Text Revealer* [63] was designed by Zhang *et al*, which is the first model inversion attack specifically designed for reconstructing private texts from transformer-based text classification models. Its attack leverages external datasets and GPT-2 to generate fluent, domain-specific text, optimizing perturbations to the hidden state based on feedback from the target model.

*6) **Attribute Inference Attacks**:* Attribute inference attacks involve inferring sensitive attributes or characteristics of individuals from fine-tuned LLMs. For example, an attacker may attempt to infer demographic information, such as age, gender, or ethnicity, based on the language patterns or topics discussed in the model's generated text [64]. This can lead to privacy violations and discrimination against individuals based on inferred attributes. In a comprehensive study, Pan *et al.* [65] systematically examined the privacy risks associated with 8 state-of-the-art language models. Their examination is anchored on 4 diverse case studies that focus on the threat of attribute inference attacks. The findings are compelling: these state-of-the-art models are indeed susceptible to revealing sensitive details, which include personal identifiers such as identity, genetic information, health data, and geographical locations. This vulnerability stems from the potential for adversaries to reverse-engineer the embeddings within these models. Building on this concern, Staab *et al.* [10] employed Reddit profiles to showcase that LLMs can accurately infer a variety of personal attributes. Remarkably, these models surpass human performance in terms of both efficiency and speed, underscoring the urgent need for effective privacy safeguards in model development.

*7) **Model Stealing Attacks**:* Adversaries may attempt to steal or replicate fine-tuned models trained on proprietary or sensitive datasets. By querying the model and observing its responses, adversaries can extract information about the model's parameters or internal representations, enabling them to reconstruct or replicate the model without access to the original training data. Krishna *et al.* [66] demonstrated the feasibility of model stealing attacks in NLP, showing that adversaries can reconstruct victim models using only random word sequences and task-specific heuristics, without requiring real training data. This exploit is enabled by the widespread use of transfer learning methods in NLP. And then, Truong *et al.* [67] advanced the field with their proposal of data-free model stealing techniques. These methods overcome the need for surrogate datasets, enabling accurate replication of valuable models with limited queries. Besides, Sha *et al.* [68] introduced a novel prompt stealing attack against LLMs, leveraging generated answers to reconstruct well-designed prompts. It involves a two-pronged approach: a parameter extractor dissects prompt types and characteristics, while a prompt reconstructor generates reverse-engineered prompts with notable efficacy.

VI. Privacy Protection in Pre-Training and Fine-Tuning

Privacy protection in pre-training and fine-tuning of LLMs is paramount in safeguarding sensitive data while ensuring model effectiveness. Incorporating techniques such as differential privacy, data cleaning, and federated learning can mitigate privacy risks.

A. Privacy Protection in Pre-Training

*1) **Data Cleaning**:* Data cleaning enhances data quality by rectifying errors and inconsistencies, serving as a foundational step that also plays a critical role in privacy protection by implementing anonymization, data minimization, and secure practices to safeguard sensitive information. To be more specific, we can remove or generalize personally identifiable information (PII) such as names, addresses, social security numbers, etc., to make it harder to identify individuals in the dataset (e.g., we can also mask sensitive information by replacing it with non-sensitive placeholders or pseudonyms while still preserving the structure and relationships within the dataset); we can aggregate data at a higher level to reduce the risk of re-identification. For example, instead of storing individual inference query details, and aggregate queries by day or week.

Tech Tips: When utilizing data cleaning techniques for privacy protection in LLMs, it's essential to prioritize thorough data sanitization before fine-tuning the model for specific tasks. Anonymizing or pseudonymizing sensitive information, and aggregating data to reduce granularity are key strategies to safeguard individual privacy.

OpenAI [37] underscores the thorough measures implemented to elevate the quality and security of their training data. They utilize filtering and fuzzy deduplication techniques to remove personally identifiable information from the corpora utilized for model training. This methodology not only purifies the data but also secures a heightened level of privacy protection. These measures are also employed in [69]. Anthropic [70] adopts a strategic approach in their training methodologies, focusing on the exclusive use of beneficial human feedback data to develop AI assistants. This selective data utilization guarantees the creation of assistants that are intrinsically helpful and non-harmful, founded on a foundation of entirely positive interactions. Additionally, their commitment to fostering AI behavior that aligns with constitutional and ethical standards is further highlighted in [71]. Kandpal *et al.* [72] demonstrated that removing duplicated sequences from training data significantly reduces the vulnerability of language models to privacy attacks, such as those allowing adversaries to recover memorized information [61]. Through empirical analysis, the authors show that duplication in training data is a key factor contributing to these privacy risks. By deduplicating the training sets, the models become less likely to regenerate sensitive or specific information, hence improving their security against such attacks without compromising the model's performance.

*2) **Federated Learning**:* Federated learning revolutionizes machine learning by decentralizing the training process, enabling model training across multiple edge devices or servers while preserving data privacy. Initially, a global model is distributed to participating devices, which independently train the model using their local data. Instead of sending raw data to a central server, only model updates are transmitted, ensuring user privacy as data remains localized. These updates are aggregated at the central server to refine the global model iteratively, leading to continuous improvement without compromising privacy. Federated learning thus offers a paradigm shift, promoting collaborative machine learning in privacy-sensitive environments by leveraging distributed data processing and maintaining data locality.

Tech Tips: In the pre-training of LLMs, federated learning offers a privacy-centric approach by eliminating the need for centralized data storage. Training occurs on local devices, with only model parameters or updates sent to a central server for aggregation. This method keeps personal data on its original device, drastically reducing data breach risks and addressing privacy and security concerns associated with centralized storage.

Chen *et al.* [73] introduced a federated learning framework for LLMs that focuses on privacy without sacrificing performance, incorporating federated pre-training to securely utilize decentralized data for improved privacy, security, and model generalization. Yu *et al.* [74] developed Federated Foundation Models to enhance privacy in collaborative learning, focusing on the entire lifecycle of foundation models with federated learning. They tackle privacy, performance, and scalability, paving the way for future research on privacy-preserving, personalized models.

Finding: Federated learning is not enough

Federated learning protects data privacy across various participants by decentralizing the training process, where data remains on users' devices and only model updates are shared. However, it's not entirely secure against privacy breaches; malicious servers could potentially extract private user data from shared gradients. To bolster security, federated learning often integrates additional privacy-preserving techniques such as differential privacy, secure multi-party computation, homomorphic encryption, and adversarial training. These methods collectively enhance the robustness of privacy protection in federated learning frameworks.

*3) **Differential Privacy**:* Differential privacy is a technique for protecting data privacy, particularly in the fields of statistical release and data analysis. Its purpose is to allow researchers to extract useful statistical information from an entire dataset without revealing any individual data. Differential privacy achieves this by adding a certain amount of random noise to the data, ensuring that even if attackers have complete background knowledge except for the target dataset, they cannot determine whether the dataset contains information about a specific individual. We can define differential privacy as follows:

Definition 6.1: Given two datasets D_1 and D_2, that differ by only one element (i.e., they are "adjacent datasets"), a randomized algorithm A satisfies ϵ-differential privacy if and only if for all output sets S from the algorithms on D_1 and D_2, the following holds:

$$\frac{\Pr\left[\mathcal{A}\left(D_1\right) \in S\right]}{\Pr\left[\mathcal{A}\left(D_2\right) \in S\right]} \leq e^{\epsilon} \tag{1}$$

where $\Pr\left[\mathcal{A}\left(D_1\right) \in S\right]$ represents the probability that the result of running algorithm $\mathcal{A}$ on dataset D_1 falls within the set S. ϵ is a non-negative parameter known as the privacy budget. The smaller the ϵ, the higher the level of privacy protection, but this may reduce the utility of the data. e is the base of the natural logarithm, approximately equal to 2.71828.

Since the algorithm $\mathcal{A}$ is random, differential privacy can ensure that for adjacent datasets (i.e., datasets that differ by only one element), the output of an algorithm is "almost identical." This means that it is nearly impossible to infer any specific information about an individual from the output. By adjusting the value of ϵ, a trade off can be realized between data privacy protection and data utility.

Tech Tips: Integrating differential privacy into the pre-training process of LLMs involves adding noise to the training data or model updates to safeguard individual privacy while maintaining effective model training. This can be achieved by injecting random noise into training data or perturbing gradients during backpropagation. Adaptive noise mechanisms dynamically adjust noise levels based on data sensitivity and privacy budgets. Careful management of the privacy budget ensures desired privacy levels are maintained.

Hoory *et al.* [75] examined the application of differential privacy to pre-trained language models. It focuses on evaluating and enhancing the performance of these models under privacy constraints. Du *et al.* [76] focused on providing differential privacy in forward propagation for large-scale models. It addresses the challenge of protecting data privacy while performing forward propagation in large models. Li *et al.* [77] argued that LLMs can be effective learners under differential privacy constraints. It explores techniques to optimize model performance while adhering to privacy standards.

B. Privacy Protection in Fine Tuning

*1) **Federated Learning**:* Federated learning transcends its initial application in pre-training, proving equally effective in the fine-tuning phase. This expanded application not only extends its utility but also underscores its versatility in bolstering privacy protection. By operating across data, models, and commands, federated learning presents a holistic solution, showcasing its comprehensive applicability and potential for addressing privacy concerns in diverse contexts.

Tech Tips: Similarly, in the fine-tuning phase, federated learning is employed by initially distributing the pre-trained global model to edge devices or local servers where fine-tuning tasks are performed. On each device or server, the global model is fine-tuned using locally held data pertinent to the specific task.

Xu *et al.* [78] and Zhang *et al.* [79] integrated federated learning into the fine-tuning of LLMs to significantly enhance privacy protection. Their approaches focus on keeping sensitive data on the user's device, eliminating the need for direct data transmission and sharing. By employing advanced privacy-preserving techniques such as differential privacy, secure aggregation, and homomorphic encryption, they ensure that user privacy is safeguarded during the fine-tuning process. Sun *et al.* [80] introduced FedBPT, a federated learning framework for privacy-preserving prompt tuning in language models, optimizing prompts locally and sharing only updates to minimize communication overhead and ensure data privacy. This method facilitates secure, collaborative model enhancement without exposing sensitive data. Zhao *et al.* [11] enhanced privacy in model fine tuning across decentralized nodes by aggregating local updates into a central model without centralizing data, effectively keeping sensitive information local and mitigating data breach risks while leveraging collaborative learning benefits. Fan *et al.* [81] presented an approach that combines federated learning with knowledge distillation and parameter-efficient fine-tuning in LLMs to ensure privacy. They also introduce secure aggregation for safely merging model updates, enabling collaborative, privacy-preserving learning across different organizations.

Finding: Federated Learning in Pre-Training V.S. in Fine-Tuning

In federated learning, pre-training employs extensive, general datasets for foundational language comprehension through distributed learning, emphasizing data privacy. Fine-tuning, however, focuses on specialized tasks using targeted datasets, prioritizing personalized optimization and stricter privacy on local devices. The technical needs for privacy protection distinctly vary between these stages. However, most research on addressing privacy issues in LLMs through federated learning focuses on optimizing the computational and communication overhead. These studies either claim applicability to both pre-training and fine-tuning phases or claim relevance to a specific phase without making targeted adjustments or designs for that stage. This highlights a gap: the need for precise, stage-specific optimization and design in federated learning for LLMs, essential for improving privacy protection's effectiveness and efficiency at different stages.

*2) **Differential Privacy**:* The approaches primarily employ differential privacy techniques to handle privacy-sensitive tuning data, thereby enabling secure and private inference. These approaches focus on balancing the data utility in model tuning with the data privacy [64], [75]–[77], [82]–[87]. Behnia *et al.* [82] introduced EW-Tune, a framework for fine-tuning LLMs with differential privacy guarantees. EW-Tune employed the Edgeworth accountant method, offering finite-sample privacy guarantees suitable for the fine-tuning context. It solves the problem of how to fine-tune LLMs on private data without compromising privacy. Shi *et al.* [83] presented a framework for enhancing the privacy of LLMs without significantly compromising their utility. The proposed approach, Just Fine-tune Twice (JFT), focuses on selectively applying differential privacy (SDP) to only the sensitive parts of data, based on a policy function. This is achieved through a two-phase fine-tuning process: first with redacted data and then with original data using a privacy-preserving mechanism. This method is shown to be effective for transformer-based models and addresses limitations of prior SDP applications. Wu *et al.* [84] designed an Adaptive Differential Privacy (ADP) framework for language model training. It estimates the privacy probability of linguistic items without resorting to the prior privacy information and designs a novel Adam algorithm to adaptively adjust the degree of differential privacy noise, potentially improving model utility while maintaining privacy. Li *et al.* [64] explored a method for prompt tuning LLMs in a privacy-preserving manner. This approach seeks to leverage the power of large models while safeguarding user privacy.

*3) **Knowledge Unlearning**:* Knowledge unlearning, also known as machine unlearning, is a strategy aimed at bolstering privacy within machine learning models, especially LLMs [88]. When a machine learning model is trained on data, it learns patterns and correlations present in that data. However, sometimes these patterns may inadvertently encode sensitive information about individuals. If the model retains this information, it can pose privacy risks when the model is deployed in real-world applications, especially in scenarios where the model may be exposed to sensitive data. Knowledge unlearning techniques aim to mitigate these risks by selectively forgetting or removing sensitive information from the model.

Tech Tips: In the fine-tuning stage, it functions by ensuring that the model does not hold onto or disclose sensitive details learned during its initial training phases. This process involves retraining the model to eliminate its memory of certain information, effectively reducing the risk of privacy breaches while maintaining or enhancing the model's performance.

Zhang *et al.* [89] analyzed the Right to be Forgotten in LLMs, identifying the unique legal and technological hurdles and proposing solutions like differential privacy and machine unlearning to balance privacy with technological progress. Chen *et al.* [90] introduced an efficient unlearning technique for LLMs using unlearning layers within transformers, enabling precise data removal without retraining and effectively managing sequential deletion requests with minimal performance loss. Jang *et al.* [91] proposed a targeted unlearning method for LMs through gradient ascent on specific sequences, offering an efficient way to erase sensitive information while preserving overall model performance. Eldan *et al.* [92] detailed a novel unlearning approach for LLMs by fine-tuning on datasets modified to omit targeted knowledge, employing reinforcement bootstrapping to forget information without compromising model integrity.

*4) **Offsite Tuning**:* Offsite tuning, detailed by Xiao *et al.* [93], refines the adaptability of models to specific tasks, prioritizing data privacy through the deployment of lightweight adapters and compressed emulators for localized adjustments.

Tech Tips: This innovative method transmits only essential components to the data owner for offsite tuning, thereby avoiding the exposure of the entire model and ensuring that sensitive data remains under the data owner's control. This significantly lowers the risk of privacy breaches. The adapter, fine-tuned with local data, is updated without direct data exposure and seamlessly reintegrated into the foundation model, effectively safeguarding data privacy throughout the adaptation process.

VII. Privacy Protection in Inference

During the inference process of LLMs, the issue of privacy leakage has garnered widespread attention. To address this issue, researchers have developed numerous strategies to ensure privacy security during the inference phase. In this section, we summarize the privacy protection approaches for the inference stage of LLMs, focusing on various approaches including encryption-based privacy protection approaches, privacy protection approaches through detection, and hardware-based approaches.

A. Cryptography-based Approaches

*1) **Homomorphic Encryption**:* Homomorphic encryption [108] is a cryptographic technique that allows for computations to be performed on ciphertexts, ensuring that the result, when decrypted, is identical to the result of the same operations performed on the plaintext. This encryption method is key in enabling data to be processed while maintaining its encrypted state, adding a new dimension to data privacy and security. Homomorphic encryption is primarily categorized into three types:

- Partial Homomorphic Encryption (PHE): Supports one type of operation (usually addition or multiplication) on ciphertexts.
- Somewhat Homomorphic Encryption (SWHE): Allows a limited number of operations on ciphertexts.
- Fully Homomorphic Encryption (FHE): The most powerful, supporting an unlimited number of both addition and multiplication operations on ciphertexts.

To better understand homomorphic encryption algorithms, we provide the following definition.

Definition 7.1: An encryption scheme is considered homomorphic over an operation $\circ$ if it satisfies a specific mathematical property. Specifically, it supports the following equation:

$$E(m_1) \circ E(m_2) = E(m_1 \circ m_2), \quad \forall m_1, m_2 \in \mathcal{M} \qquad (2)$$

Here, E represents the encryption algorithm, $\mathcal{M}$ denotes the set of all possible messages that can be encrypted, and m_1 and m_2 are any two messages in the scheme. The operation $\star$ can be any binary operation (e.g. addition or multiplication).

Tech Tips: Homomorphic encryption safeguards privacy during the inference stage by encrypting both the model parameters and input data. With HE, computations can be performed directly on encrypted data, allowing the model to make predictions without decrypting sensitive information. This process ensures that neither the raw data nor the model architecture is exposed in their unencrypted form, preserving privacy throughout the inference process. Decryption of the results is only done by trusted parties possessing the decryption key, maintaining the confidentiality of the information. Additionally, HE facilitates secure outsourcing of computations to untrusted servers, enabling organizations to utilize external resources without compromising data privacy.

We now introduce privacy inference approaches based on HE [94]–[98]. The THE-X [94] presented a novel approach for enabling privacy-preserving inference on pre-trained transformer models using homomorphic encryption, in which they utilized ReLU to replace GELU and used approximation methods for SoftMax and LayerNorm to support the fully HE operations. However, THE-X may lead to privacy leakages as it poses intermediate results to the client during the computing of ReLU. Iron [95] focused on enhancing privacy in client-server settings, where clients have private inputs and servers hold proprietary models. It introduces several new homomorphic encryption-based protocols for matrix multiplication and complex non-linear functions (like Softmax, GELU activations, and LayerNorm) which are crucial in Transformer-based models. Bumblebee [96] optimized homomorphic encryption-based protocols for large matrix multiplication and efficient, accurate protocols for non-linear activation functions in transformers, enhancing data privacy during inference. Zimerman *et al.* [97] explored secure transformer models tailored for HE, which converts the operators to their polynomial equivalent. Liu *et al.* [98] proposed a framework to enhance the efficiency of private inference on transformer-based models. It focuses on replacing computation-intensive operators (e.g., ReLU, GELU) in transformers with privacy-computing-friendly alternatives. The framework achieves significant reductions in private inference time and communication overhead while maintaining near-identical model accuracy.

*2) **Multi-Party Computation**:* Multi-Party Computation [109], [110] is a cryptographic protocol that enables allows multiple parties (often mutually distrusting) to collaboratively perform a computation task while keeping their individual data private. This means that even though the parties are working together to compute a result, none of them can see the other parties' private data. The objective of secure multi-party Computation is to construct a secure protocol that allows multiple mistrustful participants to jointly compute a target function on their private inputs, while ensuring the accuracy of the output, and protecting and controlling their private inputs even in the presence of dishonest behavior. SMPC can be

TABLE I
PRIVATE INFERENCE APPROACHES

Schemes	Tools	Improved Components	Matrix Multiplication	Nonlinear to Polynomial	Threat Model	Experiments on
THE-X† [94]	HE	GELU, SoftMax, LayerNorm	●	○	CPA	Bert-tiny
Iron [95]	HE	GELU, SoftMax, LayerNorm	●	○	Honest-but-curious	Bert
Bumblebee [96]	HE	SoftMax, LayerNorm	●	○	Static semi-honest	Bert-base/Large, GPT2-base, LLaMA-7B, ViT-base
Zimerman *et al.*† [97]	HE	GELU, Softmax	○	●	CPA	BerT-like
Liu *et al.* [98]	HE, MPC	GELU, SoftMax, LayerNorm	●	○	Semi-honest	BERT-Tiny, BERT-Medium, RoBERTa-Base
Wang *et al.* [99]	MPC	SoftMax, Embedded Tables	○	○	Semi-honest	XLM, ViT
CipherGPT [100]	MPC	GELU	●	○	Semi-honest	GPT2
East [101]	MPC	SoftMax, LayerNorm	○	●	Semi-honest	BerT-like
Privformer [102]	MPC	Sigmoid	●	○	Honest majority	Transformer
Puma [103]	MPC	GELU, SoftMax	○	●	Semi-honest	Bert-Base/Large, GPT2-Base/Medium/Large, Roberta-Base, LLaMA-7B
Sigma [104]	FSS	GELU, SoftMax	○	●	Semi-honest static	Bert-Tiny/Base/Large, GPT2, GPT2-Neo
Majmuda *et al.* [85]	DP	SoftMax	○	○	Semi-honest	RoBERTa-style
Dp-forward [86]	DP	Embedding	○	○	Semi-honest	Bert
Mai *et al.* [87]	DP	Embedding	○	○	Attribute inference attack	Bert, GPT2, T5
Textfusion [105]	Token fusion	Tokenizer	○	○	Text reconstruction attack	Bert-Base, Bert-Large
Yuan *et al.* [106]	Permutation	RELU, SoftMax, LayerNorm	●	○	-	Transformer, LLaMa

[CPA] Chosen plaintext attacks.
[†] Note that the CKKS homomorphic encryption scheme might be vulnerable to passive attacks. [107]

formally described as follows: Consider n parties, denoted as $P_1, P_2, ..., P_n$. Each party P_i holds a private input X_i. There is a predefined function f that takes n inputs. This function is of the form $f : (X_1, X_2, ..., X_n) \rightarrow Y$, where X_i represents the input for party P_i and Y is the output using the secret data of all parties. Then, the parties compute the result $Y = (Y_1, Y_2, .., Y_n)$ based on the function $f(X_1, X_2, ..., X_n)$ such that each party learns Y (or a portion of Y relevant to them) but learns nothing about the inputs X_i of the other parties, for all $j \neq i$.

Tech Tips: MPC enables secure aggregation of model updates in federated learning setups, allowing parties to collaboratively train a shared model. MPC ensures privacy during model inference by performing computations on encrypted data, shielding sensitive information from central servers. MPC facilitates secure data labeling by allowing multiple parties to label data collaboratively without exposing raw labels, thus maintaining the confidentiality of sensitive information throughout the process.

Similar to HE, MPC is another crucial method that can be used to protect model privacy [99]–[103]. Wang *et al.* [99] focused on the challenges and solutions for private inference in transformer models using MPC. While it advances the field of privacy-preserving inference, the complexity of MPC might affect practicality and efficiency. Hou *et al.* [100] presented a framework CipherGPT for secure GPT model inference in a two-party setting. It introduces optimized cryptographic protocols for operations like matrix multiplication and GELU activation, essential for GPT models. The framework focuses on preserving privacy while ensuring the efficiency of the inference process. However, the specific focus on two-party settings may limit the framework's applicability in more diverse operational environments. Ding *et al.* [101] proposed a communication-efficient protocol called East for activation functions like GELU and tanh, as well as optimized protocols for softmax and Layer Normalization (LN). These protocols are designed to enhance performance by reducing runtime and communication overhead, ensuring the security of the scheme. Akimoto *et al.* [102] presented a MPC-based approach to secure inference of Transformer models in natural language processing using ReLU functions. This method addresses the challenge of computing the Transformer's attention mechanism efficiently and securely in an MPC setting. Dong *et al.* [103] introduced PUMA, a framework for efficient and secure inference on Transformer models using replicated secret sharing. PUMA offers approximations for expensive non-linear functions (e.g., GeLU and softmax), which can also evaluate the large models like LLaMA-7B efficiently under MPC.

*3) **Functional Secret Sharing**:* Function Secret Sharing (FSS) [111] involves dividing an original secret into multiple shares using a mathematical function (such as a polynomial), encoding the secret into each share in such a way that each is independent and insufficient to reveal the entire secret. These parts are then distributed to different participants, who can independently execute predetermined functions, such as arithmetic or logical operations, on their portion of the secret. These computations are carried out on secret shares that are in an encrypted or hidden state, preventing participants from obtaining any information about the original secret from their share alone. The results obtained by each participant are then aggregated, and when a sufficient number of shares are combined and computed, the outcome of executing the function on the entire secret is recovered. The security of this process lies in the fact that each share does not contain enough information to reveal the secret by itself; hence, even if some shares are compromised or participants are dishonest, the secret remains secure. The original secret's information is only revealed when the predetermined threshold is reached, that is, when a certain number of shares are correctly combined.

Tech Tips: In FSS, the LLM or function is partitioned into shares using cryptographic methods, with each party holding a share. During computation, parties perform operations on their shares using their private data, ensuring that individual inputs remain undisclosed. After computation, the parties collaboratively combine their shares to reconstruct the result of the function, maintaining privacy while revealing the final output.

As far as we know, there has been only one secure privacy inference approach based on Function Secret Sharing (FSS), which was proposed by Gupta *et al.* [104]. The approach discussed a system named SIGMA for secure inference of transformer-based models, specifically focusing on Generative Pre-trained Transformers. SIGMA is designed to be efficient in terms of latency and communication overhead while maintaining standard 2-party computation (2PC) security by leveraging FSS. It introduces new FSS-based protocols for complex machine learning functionalities like Softmax and GeLU and optimizes them for GPU acceleration. SIGMA claims significant improvements in latency over state-of-the-art systems and demonstrates scalability to large GPT models. However, the paper does not explicitly outline specific disadvantages, which typically in such systems could include complexity of implementation, computational resource requirements, or potential limitations in the types of models or data that can be securely processed.

*4) **Differential Privacy in Inference**:* Similarly, differential privacy can also be applied in the inference stage of LLM, providing a crucial layer of privacy protection during the generation of model predictions or outputs.

Tech Tips: In the inference stages of LLMs, DP can introduce noise to model outputs to safeguard individual data privacy while preserving prediction accuracy. Parameters are adjusted to manage the privacy budget effectively, with continuous monitoring ensuring a balance between privacy and utility over time.

Majmudar *et al.* [85] presented a method for ensuring differential privacy in the decoding process of LLMs. This approach aims to protect privacy during text generation. Du *et al.* [86] proposed a method for fine-tuning and inference in language models while maintaining differential privacy during the forward pass. It tackles the challenge of protecting privacy during both fine-tuning and inference phases. Mai *et al.* [87] introduced the Split-and-Denoise method, combining local differential privacy with a denoising technique to protect privacy in large language model inference. Zhou *et al.* [105] introduced a method for privacy-preserving inference in pre-trained models using token fusion. The advantage is maintaining privacy during inference, but it could impact the inference accuracy or efficiency. Yuan *et al.* [106] detailed a three-party protocol for secure Transformer model inference, safeguarding both model parameters and user data. It applies permutation instead of complex encryption, offering strong security with practical feasibility for global matrix multiplication-based layers.

Finding: Cryptography-based Private Inference

Privacy protection techniques grounded in Homomorphic Encryption (HE), Multi-Party Computation (MPC), and Functional Secret Sharing (FSS) offer demonstrable security assurances within rigorously defined threat models, as indicated in Table I. Nevertheless, limitations in performance and efficiency present obstacles to their near-term adoption by prominent model service providers. Even though these techniques have enhanced the efficiency of critical components, their experimental results demonstrate that deploying HE, MPC, and FSS might lead to degraded performance. Alternative approaches often rely on principles of obfuscation, yet their levels of randomness and security are weaker than cryptography-based solutions, and they typically consider specific attacks.

B. Detection-based Approaches

In existing research on Language Models (LMs), some efforts focuses on detecting privacy leaks [112]–[115]. These studies predominantly examine whether the content generated by LMs directly exposes data privacy or if such privacy can be inferred through contextual associations. This approach is equally applicable to LLMs, suggesting a viable pathway for assessing and mitigating privacy risks in more advanced linguistic computational models.

Tech Tips: Detection-based methods for protecting the privacy of LLM involve identifying and mitigating potential privacy risks in the text generated by these models which two main strategies: (i) Direct detection methods involve directly examining the text generated by LLMs to identify privacy leaks. (ii) Contextual inference detection methods focus on identifying privacy breaches that may not be explicitly evident in the generated text but can be deduced through contextual correlations.

*1) **Direct Detection**:* Kim *et al.* [116] developed a black-box probing method to evaluate privacy risks in LLMs by using crafted prompts to elicit Personally Identifiable Information (PII) from model outputs. This approach assesses the likelihood of LLMs inadvertently revealing PII, offering a targeted strategy for understanding privacy vulnerabilities in generated text. Phute *et al.* [117] unveiled a zero-shot defense strategy for LLMs aimed at curbing harmful content generation. By deploying a harm classifier from the same LLM, this method significantly reduces the efficacy of adversarial attacks, eliminating the need for fine-tuning. Chen *et al.* [118] developed a moving target defense system for LLMs to counter adversarial attacks, using N-Gram models and naive Bayes classification for evaluating responses and BERT for assessing question-answer coherence, effectively distinguishing between beneficial and malicious content.

*2) **Contextual Inference Detection**:* Mireshghallah *et al.* [119] introduced CONFAIDE, a benchmark that evaluates LLMs' privacy reasoning across four complexity levels, revealing notable deficiencies in models like GPT-4 and ChatGPT in terms of privacy preservation and social reasoning. Huang *et al.* [120] proposed a framework to assess PLMs' risk of privacy leakage, focusing on email addresses. Their approach, which analyzes memorization and association, highlights vulnerabilities in how models might unintentionally disclose or link email addresses to individuals.

Finding: Detection-based Approaches

Due to the inherent complexity and variability of text data, scrutinizing the outputs of LLMs in practical applications has its limitations. Attackers can exploit these limitations by crafting impermissible outputs from seemingly permissible ones [121]. This underscores the necessity for advanced and dynamic security measures, beyond simple output filtering or static rules, to effectively counteract sophisticated manipulation techniques and ensure the integrity and safety of LLMs applications.

C. Hardware-based Approaches

Hardware-based approaches for protecting the privacy of LLM focus on leveraging specialized hardware features and technologies to establish secure execution environments and safeguard data during processing.

Tech Tips: Hardware-based Approaches such as Trusted Execution Environments (TEEs), hardware virtualization, secure enclaves, hardware Root of Trust (RoT), and encrypted processing, aim to ensure the confidentiality, integrity, and privacy of both the model parameters and the data being processed.

*1) **Data Locality**:* PrivateLoRA [122] leveraged edge devices' storage for private data and personalized parameters, while utilizing the cloud for computational enhancement. It splits model parameters across the cloud and edge devices and transmits only unreadable activations and gradients to maintain data locality. The method integrates three sequential low-rank matrices for weight adaptation and reduces communication overhead through Low Rank Residual Transmission. It ensures data locality by keeping personalized parameters on edge devices and raw data derivatives on the cloud. The model targets query, key, and value projections in self-attention for adaptation to minimize communication overhead. PrivateLoRA is a paradigm that powers a heterogeneously distributed inference and training cycle, achieving high throughput and performance on smart phones.

*2) **Confidential Computing with Trusted Execution Environment (TEE)**:* Confidential computing aims to address this gap by safeguarding data even while it is being processed. One key technology used in confidential computing is Trusted Execution Environments (TEEs). A TEE is a secure area of a computer's processor that ensures code and data loaded inside it are protected from unauthorized access or modification, even from the operating system or hypervisor. TEEs provide a secure environment where sensitive computations can be performed, ensuring the confidentiality and integrity of the data being processed [123]–[130].

The NVIDIA H100 GPU, featuring support for confidential computing, enhances data privacy by establishing a secure execution environment through hardware virtualization and a TEE [131]. This environment ensures that data and code are processed securely during training or inference, preventing unauthorized access or modification by unauthorized users. By anchoring security measures in an on-die hardware root of trust (RoT), NVIDIA ensures the integrity of the GPU's boot sequence and establishes a chain of trust through cryptographic attestation. Furthermore, NVIDIA continues to enhance the security and integrity by incorporating features such as encrypted firmware, firmware revocation, and fault injection countermeasures. The TEEs applied in [132] protect privacy by securely executing custodial operations, encrypting and controlling access to data, and facilitating encrypted transmission of user queries and prompts. Huang et al. [133] introduced a method deploying TEEs on both client and server sides, implementing secure communication and split fine-tuning of a language model to maintain accuracy.

VIII. Challenges and Future Directions

A. Difficulties in Understanding Black-Box LLMs

1) Challenges: Pre-trained LLMs are often treated as black box models [134], [135], meaning that their internal workings and decision-making processes are not fully transparent or interpretable. This opacity makes it challenging to analyze and understand how these models handle sensitive information and whether they inadvertently leak privacy. In addition, LLMs are trained on vast amounts of diverse data, which may include sensitive or personally identifiable information. Understanding how these models process and retain such data without compromising privacy is inherently complex, especially given the intricate relationships between input data and model outputs. Language is dynamic and context-dependent, leading to challenges in predicting how LLMs will behave in various real-world scenarios. Privacy risks may vary depending on the context in which the model is deployed, making it difficult to generalize findings across different applications or domains.

2) Future Directions: Developing techniques to interpret and explain the decisions of pre-trained LLMs can shed light on their privacy implications. This may involve analyzing model activations, attention mechanisms, or other internal representations to identify potential privacy vulnerabilities. Conducting adversarial testing to evaluate the robustness of pre-trained LLMs against privacy attacks. For example, adversarial examples can be generated to probe the model's behavior and identify weaknesses that may lead to privacy breaches [136]. Besides, we can focus on developing fine-tuning techniques that explicitly consider privacy concerns, such as differential privacy-aware optimization or adversarial training with privacy objectives. These techniques aim to mitigate privacy risks during the fine-tuning process.

B. Privacy in Multimodal LLMs

1) Challenges: The majority of research on LLMs has focused on purely textual models such as GPT and BERT. As a result, there may be a tendency for researchers to prioritize investigating the privacy implications of these models, leaving less attention on Multimodal LLMs. Multimodal LLMs, which integrate both textual and visual information, are a relatively recent development compared to their purely textual counterparts [137], [138]. As such, there has been less time for researchers to explore and investigate their privacy implications thoroughly. Multimodal LLMs process a more diverse range of data types, including text, images, and possibly other modalities such as audio or video. Analyzing the privacy implications of such complex and heterogeneous data poses additional challenges compared to purely textual data, which may deter some researchers from delving into this area.

2) Future Directions: Redefining privacy in Multimodal LLMs is necessary to address the increased data complexity, unique privacy risks, intermodal interactions, user expectations, and regulatory considerations associated with multimodal data processing. Developing techniques to fuse different modalities while preserving user privacy is an important research direction. This could involve exploring encryption methods, differential privacy techniques, or novel privacy-preserving machine learning algorithms tailored to multimodal data. Conducting adversarial analyses to identify potential vulnerabilities and privacy risks in Multimodal LLMs. This could involve exploring adversarial attacks and defenses specific to multimodal data, such as perturbing images or textual inputs to compromise privacy.

C. Privacy in Personalized LLMs

1) Challenges: Personalized LLMs may store and process sensitive user data, such as personal conversations, search queries, or browsing history. If not adequately protected, this data could be vulnerable to unauthorized access or misuse, leading to privacy breaches and potential harm to individuals. Personalized LLMs have the capacity to infer personal information about users based on their interactions with the model. This includes sensitive attributes such as health status, political views, financial situation, or intimate preferences. Such inferences could be unintentionally revealed through model responses or recommendations, compromising user privacy. Numerous small-scale enterprises offer users specialized large-scale model services tailored to vertical domains, encompassing sectors such as judiciary, education, and finance. These expansive models entail a greater incorporation of domain-specific personal data. However, owing to the comparatively limited privacy safeguarding capabilities inherent in small-scale enterprises, the susceptibility to user privacy breaches is heightened, potentially precipitating irreversible ramifications.

2) Future Directions: To safeguard personalized fine-tuning of LLMs from privacy leakage, we need to explore architectures specifically designed with privacy [139]. In addition, we can develop a combination of techniques. This includes implementing differential privacy methods to add noise during training, utilizing federated learning to train models locally on user devices, employing secure multi-party computation to jointly train models without sharing private data directly, introducing data perturbation to prevent memorization of sensitive information. We can also apply regularization methods to prevent overfitting, and exploring privacy-preserving architectures designed specifically for protecting sensitive data during fine-tuning.

D. Privacy Protection Throughout the Entire Creation Process of LLMs

1) Challenges: Given the intricate complexity involved in training LLMs, privacy protection research tends to dissect various phases of LLM development and deployment, including pre-training, prompt tuning, and inference. Nevertheless, each segment within the LLM lifecycle harbors its own set of privacy vulnerabilities, and these stages do not operate in isolation [140]. For instance, privacy breaches detected during the inference phase might originate from potential backdoors introduced during pre-training. Thus, safeguarding privacy comprehensively across large models demands concurrent scrutiny of multiple stages, a task that also introduces complexities and challenges into privacy protection efforts.

2) Future Directions: Protecting the privacy of LLMs throughout their creation process is paramount and requires a multifaceted approach. Firstly, during data collection, minimizing the collection of sensitive information and obtaining informed consent from users are critical steps. Data should be anonymized or pseudonymized to mitigate re-identification risks. Secondly, in data preprocessing and model training, techniques such as federated learning, secure multiparty computation, and differential privacy can be employed to train LLMs on decentralized data sources while preserving individual privacy. Additionally, conducting privacy impact assessments and adversarial testing during model evaluation ensures potential privacy risks are identified and addressed before deployment. In the deployment phase, privacy-preserving APIs and access controls can limit access to LLMs, while transparency and accountability measures foster trust with users by providing insight into data handling practices. Ongoing monitoring and maintenance, including continuous monitoring for privacy breaches and regular privacy audits, are essential to ensure compliance with privacy regulations and the effectiveness of privacy safeguards. By implementing these measures comprehensively throughout the LLM creation process, developers can mitigate privacy risks and build trust with users, thereby leveraging the capabilities of LLMs while safeguarding individual privacy.

E. Hardware-assisted Privacy Protection

1) Future Directions: NVIDIA Confidential Computing provides a comprehensive suite of privacy-enhancing features and technologies that safeguard LLM data and operations against unauthorized access, manipulation, and breaches, thereby ensuring the confidentiality and integrity of sensitive information throughout the LLM lifecycle. In the future, we can integrate confidential computing capabilities into LLM workflows, ensuring comprehensive privacy protection across the entire lifecycle, while continued innovation in GPU security features, such as encrypted firmware and fault injection countermeasures, reinforces the company's commitment to advancing data privacy safeguards for sensitive workloads.

IX. Conclusion

In this paper, we thoroughly investigates the data privacy concerns associated with LLMs, focusing on privacy leakage, privacy attacks, and the pivotal technologies for privacy protection during various stages of LLM privacy inference, including federated learning, differential privacy, knowledge unlearning, and hardware-assisted privacy protection. By conducting a detailed analysis of the strengths and weaknesses of existing approaches, this study highlights the challenges and limitations in LLMs and proposes directions for future work. This research is of significant importance for deepening our understanding of data privacy issues in LLMs and promoting further exploration and improvement in LLMs.

References

[1] M. Gao, X. Hu, J. Ruan, X. Pu, and X. Wan, "Llm-based nlg evaluation: Current status and challenges," *arXiv preprint arXiv:2402.01383*, 2024.

[2] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models," *arXiv preprint arXiv:2302.05128*, 2023.

[3] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2998–3009.

[4] Z. Xu, K. Wu, J. Wen, J. Li, N. Liu, Z. Che, and J. Tang, "A survey on robotics with foundation models: toward embodied ai," *arXiv preprint arXiv:2402.02385*, 2024.

[5] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[6] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv preprint arXiv:2303.04226*, 2023.

[7] J. Wu, W. Gan, Z. Chen, S. Wan, and H. Lin, "Ai-generated content (aigc): A survey," *arXiv preprint arXiv:2304.06632*, 2023.

[8] Y. Yao, J. Duan, K. Xu, Y. Cai, E. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *arXiv preprint arXiv:2312.02003*, 2023.

[9] N. Subramani, S. Luccioni, J. Dodge, and M. Mitchell, "Detecting personal information in training corpora: an analysis," in *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, 2023, pp. 208–220.

[10] R. Staab, M. Vero, M. Balunović, and M. Vechev, "Beyond memorization: Violating privacy via inference with large language models," *arXiv preprint arXiv:2310.07298*, 2023.

[11] J. Zhao, "Privacy-preserving fine-tuning of artificial intelligence (ai) foundation models with federated learning, differential privacy, offsite tuning, and parameter-efficient fine-tuning (peft)," *Authorea Preprints*, 2023.

[12] L. Luo, J. Ning, Y. Zhao, Z. Wang, Z. Ding, P. Chen, W. Fu, Q. Han, G. Xu, Y. Qiu *et al.*, "Taiyi: A bilingual fine-tuned large language model for diverse biomedical tasks," *arXiv preprint arXiv:2311.11608*, 2023.

[13] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee, "Scalable extraction of training data from (production) language models," *arXiv preprint arXiv:2311.17035*, 2023.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun *et al.*, "Personal llm agents: Insights and survey about the capability, efficiency and security," *arXiv preprint arXiv:2401.05459*, 2024.

[17] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *arXiv preprint arXiv:2307.03109*, 2023.

[18] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[19] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, and L. S. Chao, "A survey on llm-gernerated text detection: Necessity, methods, and future directions," *arXiv preprint arXiv:2310.14724*, 2023.

[20] S. R. Bowman, "Eight things to know about large language models," *arXiv preprint arXiv:2304.00612*, 2023.

[21] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[22] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage," *Authorea Preprints*, 2023.

[23] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong *et al.*, "Evaluating large language models: A comprehensive survey," *arXiv preprint arXiv:2310.19736*, 2023.

[24] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. G. H. Cheng, Y. Klochkov, M. F. Taufiq, and H. Li, "Trustworthy llms: a survey and guideline for evaluating large language models' alignment," *arXiv preprint arXiv:2308.05374*, 2023.

[25] H. Li, Y. Chen, J. Luo, Y. Kang, X. Zhang, Q. Hu, C. Chan, and Y. Song, "Privacy in large language models: Attacks, defenses and future directions," *arXiv preprint arXiv:2310.10383*, 2023.

[26] S. Neel and P. Chang, "Privacy issues in large language models: A survey," *arXiv preprint arXiv:2312.06717*, 2023.

[27] J. Marshall, "What effects do large language models have on cybersecurity," 2023.

[28] M. Al-Hawawreh, A. Aljuhani, and Y. Jararweh, "Chatgpt for cybersecurity: practical applications, challenges, and future directions," *Cluster Computing*, vol. 26, no. 6, pp. 3421–3436, 2023.

[29] A. Qammar, H. Wang, J. Ding, A. Naouri, M. Daneshmand, and H. Ning, "Chatbots to chatgpt in a cybersecurity space: Evolution, vulnerabilities, attacks, challenges, and future recommendations," *arXiv preprint arXiv:2306.09255*, 2023.

[30] L. Schwinn, D. Dobre, S. Günnemann, and G. Gidel, "Adversarial attacks and defenses in large language models: Old and new threats," *arXiv preprint arXiv:2310.19737*, 2023.

[31] E. Derner and K. Batistič, "Beyond the safeguards: Exploring the security risks of chatgpt," *arXiv preprint arXiv:2305.08005*, 2023.

[32] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. Abu-Ghazaleh, "Survey of vulnerabilities in large language models revealed by adversarial attacks," *arXiv preprint arXiv:2310.10844*, 2023.

[33] Y. Cai, S. Mao, W. Wu, Z. Wang, Y. Liang, T. Ge, C. Wu, W. You, T. Song, Y. Xia *et al.*, "Low-code llm: Visual programming over llms," *arXiv preprint arXiv:2304.08103*, 2023.

[34] M. Karpinska and M. Iyyer, "Large language models effectively leverage document-level context for literary translation, but critical errors persist," *arXiv preprint arXiv:2304.03245*, 2023.

[35] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[38] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang *et al.*, "Datacomp: In search of the next generation of multimodal datasets," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[39] N. Kshetri, "Cybercrime and privacy threats of large language models," *IT Professional*, vol. 25, no. 3, pp. 9–13, 2023.

[40] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, "Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–21.

[41] U. Iqbal, T. Kohno, and F. Roesner, "Llm platform security: Applying a systematic evaluation framework to openai's chatgpt plugins," *arXiv preprint arXiv:2309.10254*, 2023.

[42] H. Lyu, S. Jiang, H. Zeng, Y. Xia, and J. Luo, "Llm-rec: Personalized recommendation via prompting large language models," *arXiv preprint arXiv:2307.15780*, 2023.

[43] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, and M. Fragkoulis, "Leveraging large language models for sequential recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1096–1102.

[44] L. Li, D. Song, X. Li, J. Zeng, R. Ma, and X. Qiu, "Backdoor attacks on pre-trained models by layerwise weight poisoning," *arXiv preprint arXiv:2108.13888*, 2021.

[45] W. Yang, L. Li, Z. Zhang, X. Ren, X. Sun, and B. He, "Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models," *arXiv preprint arXiv:2103.15543*, 2021.

[46] H. Yao, J. Lou, and Z. Qin, "Poisonprompt: Backdoor attack on prompt-based large language models," *arXiv preprint arXiv:2310.12439*, 2023.

[47] H. Huang, Z. Zhao, M. Backes, Y. Shen, and Y. Zhang, "Composite backdoor attacks against large language models," *arXiv preprint arXiv:2310.07676*, 2023.

[48] A. Wan, E. Wallace, S. Shen, and D. Klein, "Poisoning language models during instruction tuning," *arXiv preprint arXiv:2305.00944*, 2023.

[49] J. Xu, M. D. Ma, F. Wang, C. Xiao, and M. Chen, "Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models," *arXiv preprint arXiv:2305.14710*, 2023.

[50] J. Yan, V. Yadav, S. Li, L. Chen, Z. Tang, H. Wang, V. Srinivasan, X. Ren, and H. Jin, "Backdooring instruction-tuned large language models with virtual prompt injection," in *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*, 2023.

[51] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[52] H. Huang, W. Luo, G. Zeng, J. Weng, Y. Zhang, and A. Yang, "Damia: leveraging domain adaptation as a defense against membership inference attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3183–3199, 2021.

[53] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," *arXiv preprint arXiv:2203.03929*, 2022.

[54] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," *arXiv preprint arXiv:2305.18462*, 2023.

[55] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," *arXiv preprint arXiv:2310.16789*, 2023.

[56] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" *arXiv preprint arXiv:2402.07841*, 2024.

[57] F. Mireshghallah, A. Uniyal, T. Wang, D. K. Evans, and T. Berg-Kirkpatrick, "An empirical analysis of memorization in fine-tuned autoregressive language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1816–1826.

[58] A. Jagannatha, B. P. S. Rawat, and H. Yu, "Membership inference attack susceptibility of clinical language models," *arXiv preprint arXiv:2104.08305*, 2021.

[59] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Practical membership inference attacks against fine-tuned large language models via self-prompt calibration," *arXiv preprint arXiv:2311.06062*, 2023.

[60] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020, pp. 377–390.

[61] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[62] E. Lehman, S. Jain, K. Pichotta, Y. Goldberg, and B. C. Wallace, "Does bert pretrained on clinical notes reveal sensitive data?" *arXiv preprint arXiv:2104.07762*, 2021.

[63] R. Zhang, S. Hidano, and F. Koushanfar, "Text revealer: Private text reconstruction via model inversion attacks against transformers," *arXiv preprint arXiv:2209.10505*, 2022.

[64] Y. Li, Z. Tan, and Y. Liu, "Privacy-preserving prompt tuning for large language model services," *arXiv preprint arXiv:2305.06212*, 2023.

[65] X. Pan, M. Zhang, S. Ji, and M. Yang, "Privacy risks of general-purpose language models," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1314–1331.

[66] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer, "Thieves on sesame street! model extraction of bert-based apis," *arXiv preprint arXiv:1910.12366*, 2019.

[67] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-free model extraction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4771–4780.

[68] Z. Sha and Y. Zhang, "Prompt stealing attacks against large language models," *arXiv preprint arXiv:2402.12959*, 2024.

[69] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[70] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan, "Training a helpful and harmless assistant with reinforcement learning from human feedback," 2022.

[71] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec,

S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan, "Constitutional ai: Harmlessness from ai feedback," 2022.

[72] N. Kandpal, E. Wallace, and C. Raffel, "Deduplicating training data mitigates privacy risks in language models," 2022.

[73] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated large language model: A position paper," 2023.

[74] S. Yu, J. P. Muñoz, and A. Jannesari, "Federated foundation models: Privacy-preserving and collaborative learning for large models," 2023.

[75] S. Hoory, A. Feder, A. Tendler, S. Erell, A. Peled-Cohen, I. Laish, H. Nakhost, U. Stemmer, A. Benjamini, A. Hassidim *et al.*, "Learning and evaluating a differentially private pre-trained language model," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 1178–1189.

[76] J. Du and H. Mi, "Dp-fp: Differentially private forward propagation for large models," *arXiv preprint arXiv:2112.14430*, 2021.

[77] X. Li, F. Tramer, P. Liang, and T. Hashimoto, "Large language models can be strong differentially private learners," *arXiv preprint arXiv:2110.05679*, 2021.

[78] M. Xu, D. Cai, Y. Wu, X. Li, and S. Wang, "Fwdllm: Efficient fedllm using forward gradient," 2024.

[79] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, T. Yu, Y. Zhou, G. Wang, and Y. Chen, "Towards building the federated gpt: Federated instruction tuning," 2024.

[80] J. Sun, Z. Xu, H. Yin, D. Yang, D. Xu, Y. Chen, and H. R. Roth, "Fedbpt: Efficient federated black-box prompt tuning for large language models," 2023.

[81] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "Fate-llm: A industrial grade federated learning framework for large language models," 2023.

[82] R. Behnia, M. R. Ebrahimi, J. Pacheco, and B. Padmanabhan, "Ew-tune: A framework for privately fine-tuning large language models with differential privacy," in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2022, pp. 560–566.

[83] W. Shi, R. Shea, S. Chen, C. Zhang, R. Jia, and Z. Yu, "Just fine-tune twice: Selective differential privacy for large language models," *arXiv preprint arXiv:2204.07667*, 2022.

[84] X. Wu, L. Gong, and D. Xiong, "Adaptive differential privacy for language model training," in *Proceedings of the First Workshop on Federated Learning for Natural Language Processing (FL4NLP 2022)*, 2022, pp. 21–26.

[85] J. Majmudar, C. Dupuy, C. Peris, S. Smaili, R. Gupta, and R. Zemel, "Differentially private decoding in large language models," *arXiv preprint arXiv:2205.13621*, 2022.

[86] M. Du, X. Yue, S. S. Chow, T. Wang, C. Huang, and H. Sun, "Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 2665–2679.

[87] P. Mai, R. Yan, Z. Huang, Y. Yang, and Y. Pang, "Split-and-denoise: Protect large language model inference with local differential privacy," *arXiv preprint arXiv:2310.09130*, 2023.

[88] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, X. Xu, Y. Yao, H. Li, K. R. Varshney, M. Bansal, S. Koyejo, and Y. Liu, "Rethinking machine unlearning for large language models," 2024.

[89] D. Zhang, P. Finckenberg-Broman, T. Hoang, S. Pan, Z. Xing, M. Staples, and X. Xu, "Right to be forgotten in the era of large language models: Implications, challenges, and solutions," 2023.

[90] J. Chen and D. Yang, "Unlearn what you want to forget: Efficient unlearning for llms," 2023.

[91] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo, "Knowledge unlearning for mitigating privacy risks in language models," 2022.

[92] R. Eldan and M. Russinovich, "Who's harry potter? approximate unlearning in llms," 2023.

[93] G. Xiao, J. Lin, and S. Han, "Offsite-tuning: Transfer learning without full model," 2023.

[94] T. Chen, H. Bao, S. Huang, L. Dong, B. Jiao, D. Jiang, H. Zhou, J. Li, and F. Wei, "The-x: Privacy-preserving transformer inference with homomorphic encryption," *arXiv preprint arXiv:2206.00216*, 2022.

[95] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, and T. Zhang, "Iron: Private inference on transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 718–15 731, 2022.

[96] W.-j. Lu, Z. Huang, Z. Gu, J. Li, J. Liu, K. Ren, C. Hong, T. Wei, and W. Chen, "Bumblebee: Secure two-party inference framework for large transformers," *Cryptology ePrint Archive*, 2023.

[97] I. Zimerman, M. Baruch, N. Drucker, G. Ezov, O. Soceanu, and L. Wolf, "Converting transformers to polynomial form for secure inference over homomorphic encryption," *arXiv preprint arXiv:2311.08610*, 2023.

[98] X. Liu and Z. Liu, "Llms can understand encrypted prompt: Towards privacy-computing friendly transformers," *arXiv preprint arXiv:2305.18396*, 2023.

[99] Y. Wang, G. E. Suh, W. Xiong, B. Lefaudeux, B. Knott, M. Annavaram, and H.-H. S. Lee, "Characterization of mpc-based private inference for transformer-based models," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2022, pp. 187–197.

[100] X. Hou, J. Liu, J. Li, Y. Li, W.-j. Lu, C. Hong, and K. Ren, "Ciphergpt: Secure two-party gpt inference," *Cryptology ePrint Archive*, 2023.

[101] Y. Ding, H. Guo, Y. Guan, W. Liu, J. Huo, Z. Guan, and X. Zhang, "East: Efficient and accurate secure transformer framework for inference," *arXiv preprint arXiv:2308.09923*, 2023.

[102] Y. Akimoto, K. Fukuchi, Y. Akimoto, and J. Sakuma, "Privformer: Privacy-preserving transformer with mpc," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023, pp. 392–410.

[103] Y. Dong, W.-j. Lu, Y. Zheng, H. Wu, D. Zhao, J. Tan, Z. Huang, C. Hong, T. Wei, and W. Cheng, "Puma: Secure inference of llama-7b in five minutes," *arXiv preprint arXiv:2307.12533*, 2023.

[104] K. Gupta, N. Jawalkar, A. Mukherjee, N. Chandran, D. Gupta, A. Panwar, and R. Sharma, "Sigma: secure gpt inference with function secret sharing," *Cryptology ePrint Archive*, 2023.

[105] X. Zhou, J. Lu, T. Gui, R. Ma, Z. Fei, Y. Wang, Y. Ding, Y. Cheung, Q. Zhang, and X.-J. Huang, "Textfusion: Privacy-preserving pre-trained model inference via token fusion," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 8360–8371.

[106] M. Yuan, L. Zhang, and X.-Y. Li, "Secure transformer inference," *arXiv preprint arXiv:2312.00025*, 2023.

[107] B. Li and D. Micciancio, "On the security of homomorphic encryption on approximate numbers," in *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40*. Springer, 2021, pp. 648–677.

[108] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[109] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, no. 110, pp. 1–108, 1998.

[110] C. Dong, J. Weng, J. Liu, Y. Zhang, Y. Tong, A. Yang, Y. Cheng, and S. Hu, "Fusion: Efficient and secure inference resilient to malicious servers," in *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023.

[111] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2015, pp. 337–367.

[112] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," 2023.

[113] C. Brown and C. Morisset, "Simple and efficient identification of personally identifiable information on a public website," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 4246–4255.

[114] X. Wu, J. Li, M. Xu, W. Dong, S. Wu, C. Bian, and D. Xiong, "Depn: Detecting and editing privacy neurons in pretrained language models," *arXiv preprint arXiv:2310.20138*, 2023.

[115] Y. Shvartzshnaider, Z. Pavlinovic, A. Balashankar, T. Wies, L. Subramanian, H. Nissenbaum, and P. Mittal, "Vaccine: Using contextual integrity for data leakage detection," in *The World Wide Web Conference*, 2019, pp. 1702–1712.

[116] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, "Propile: Probing privacy leakage in large language models," 2023.

[117] M. Phute, A. Helbling, M. Hull, S. Peng, S. Szyller, C. Cornelius, and D. H. Chau, "Llm self defense: By self examination, llms know they are being tricked," 2023.

[118] B. Chen, A. Paliwal, and Q. Yan, "Jailbreaker in jail: Moving target defense for large language models," 2023.

[119] N. Mireshghallah, H. Kim, X. Zhou, Y. Tsvetkov, M. Sap, R. Shokri, and Y. Choi, "Can llms keep a secret? testing privacy implications of language models via contextual integrity theory," 2023.

[120] J. Huang, H. Shao, and K. C.-C. Chang, "Are large pre-trained language models leaking your personal information?" 2022.

[121] D. Glukhov, I. Shumailov, Y. Gal, N. Papernot, and V. Papyan, "Llm censorship: A machine learning challenge or a computer security problem?" 2023.

[122] Y. Wang, Y. Lin, X. Zeng, and G. Zhang, "Privatelora for efficient privacy preserving llm," *arXiv preprint arXiv:2311.14030*, 2023.

[123] H. Chen, H. H. Chen, M. Sun, K. Li, Z. Chen, and X. Wang, "A verified confidential computing as a service framework for privacy preservation," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 4733–4750.

[124] J. Zhu, R. Hou, X. Wang, W. Wang, J. Cao, B. Zhao, Z. Wang, Y. Zhang, J. Ying, L. Zhang *et al.*, "Enabling rack-scale confidential computing using heterogeneous trusted execution environment," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1450–1465.

[125] C. Liu, H. Guo, M. Xu, S. Wang, D. Yu, J. Yu, and X. Cheng, "Extending on-chain trust to off-chain – trustworthy blockchain data collection using trusted execution environment (tee)," *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3268–3280, 2022.

[126] R. Li, Q. Wang, Q. Wang, D. Galindo, and M. Ryan, "Sok: Tee-assisted confidential smart contract," *arXiv preprint arXiv:2203.08548*, 2022.

[127] L. Luo, Y. Zhang, C. White, B. Keating, B. Pearson, X. Shao, Z. Ling, H. Yu, C. Zou, and X. Fu, "On security of trustzone-m-based iot systems," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9683–9699, 2022.

[128] J. Weng, S. Zhijian, Y. Zhang, M. Li, W. Jiasi, Y. Wu, and L. Weiqi, "Peripheral-free secure pairing protocol by randomly switching power," Mar. 1 2022, uS Patent 11,265,722.

[129] K. Liu, M. Yang, Z. Ling, H. Yan, Y. Zhang, X. Fu, and W. Zhao, "On manually reverse engineering communication protocols of linux-based iot systems," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6815–6827, 2020.

[130] B. Pearson, C. Zou, Y. Zhang, Z. Ling, and X. Fu, "Sic 2: Securing microcontroller based iot devices with low-cost crypto coprocessors," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2020, pp. 372–381.

[131] G. Dhanuskodi, S. Guha, V. Krishnan, A. Manjunatha, M. O'Connor, R. Nertney, and P. Rogers, "Creating the first confidential gpus: The team at nvidia brings confidentiality and integrity to user code and data for accelerated computing." *Queue*, vol. 21, no. 4, pp. 68–93, 2023.

[132] T. South, G. Zuskind, R. Mahari, and T. Hardjono, "Secure community transformers: Private pooled data for llms."

[133] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, and L. Wang, "A fast, performant, secure distributed training framework for large language model," *arXiv preprint arXiv:2401.09796*, 2024.

[134] J. Hong, Q. Tu, C. Chen, X. Gao, J. Zhang, and R. Yan, "Cyclealign: Iterative distillation from black-box llm to white-box models for better human alignment," *arXiv preprint arXiv:2310.16271*, 2023.

[135] Y. Wang, X. Ma, and W. Chen, "Augmenting black-box llms with medical textbooks for clinical question answering," *arXiv preprint arXiv:2309.02233*, 2023.

[136] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," *arXiv preprint arXiv:2310.08419*, 2023.

[137] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, "Seed-bench: Benchmarking multimodal llms with generative comprehension," *arXiv preprint arXiv:2307.16125*, 2023.

[138] B. Meskó, "The impact of multimodal large language models on health care's future," *Journal of Medical Internet Research*, vol. 25, p. e52865, 2023.

[139] B. Huang, S. Yu, J. Li, Y. Chen, S. Huang, S. Zeng, and S. Wang, "Firewallm: A portable data protection and recovery framework for llm services," in *International Conference on Data Mining and Big Data*. Springer, 2023, pp. 16–30.

[140] J. Evertz, M. Chlosta, L. Schönherr, and T. Eisenhofer, "Whispers in the machine: Confidentiality in llm-integrated systems," *arXiv preprint arXiv:2402.06922*, 2024.