Privacy in RAG Systems

Animesh Madaan
220145
manimesh22@iitk.ac.in

Mahaarajan J
220600
mahaarajan22@iitk.ac.in

Pahal Patel
220742
pahaldp22@iitk.ac.in

Wattamwar Akanksha
221214
akankshab22@iitk.ac.in

1 Introduction

Large Language Models have transformed how people interact with Artificial Intelligence. We have seen a tremendous growth in the usage of LLMs. However, their growing influence has also brought new security and privacy challenges. Even though alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) are used to make these models safer, they can still be manipulated to produce harmful or unintended outputs.

These manipulations are known as **adversarial attacks** - inputs intentionally designed to make a model behave in an undesired way. An adversarial attack "tricks" the model into doing something it normally shouldn't. Early research explored such attacks in computer vision, where tiny, almost invisible changes to an image could cause a model to misclassify it. For text models, attacks are more subtle and complex because language is discrete and meaning must be preserved.

These attacks can lead to various risks – producing misleading or biased content, revealing private information, or exposing part of the model's training data. As LLMs become ubiquitous, understanding and mitigating such threats is essential to their safe and trustworthy use.

In our project, we focus on a special kind of attack – **privacy attacks in Retrieval-Augmented Generation (RAG) systems**. In these systems LLMs are paired with external knowledge databases. While the purpose of these to increase factual accuracy and reduce hallucinations, it opens avenues for privacy leakage.

2 Background: RAG Architecture and Privacy Threats

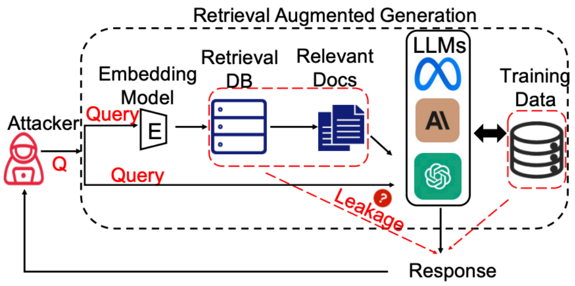

Figure 1: RAG System with potential risks.

Retrieval-augmented generation (RAG) is a neural framework that combines document retrieval with large language model (LLM) generation to improve factual grounding and reduce hallucinations. A RAG system consists of three main components: a **retriever** and a **generator**. In the retrieval phase, the retriever searches an external corpus for passages relevant to the user query. This step typically involves calculating the similarity or distance between the query's embedding and the embeddings of stored documents. The top-k retrieved passages having the smallest distances are then concatenated with the query as text before being passed to the generator. In the generation phase, a pretrained LLM (e.g., GPT or LLaMA) integrates this external context with its internal knowledge to produce coherent and factually accurate responses.

Despite these advantages, RAG systems have notable privacy risks. Privacy in RAG can be defined along two dimensions: **data privacy** and **model privacy**. Data privacy concerns the leakage of sensitive content from the retrieval corpus, such as personal, financial, or medical data. Model privacy refers to the leakage of information memorized during pretraining or fine-tuning of the LLM itself.

Depending on access level, attackers may operate under a **black-box** setting, observing only outputs, or a **white-box** setting, with partial access to model or retriever parameters. Major threats include **membership inference attacks (MIA)**[4] that identify whether specific data were used in training, **reconstruction attacks**[1] that recover text from responses, and **prompt injection or indirect leakage**[2] where adversarial prompts expose hidden context. Additionally, **retrieval-based leakage** occurs when private documents are surfaced and reproduced by the generator. These risks highlight the need for secure retrieval mechanisms, access control, and context filtering when deploying RAG systems in privacy-sensitive domains.

3 Base reference paper

3.1 Overview

Our project is based on the paper *"The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)"* [7], published in the Association for Computational Linguistics (**ACL 2024**). This work investigates the dual nature of privacy in RAG pipelines- the privacy of the Retrieval database and privacy of the LLM's training data-and examines how they manifest using various experiments.

3.2 Methodology and Threat Model

The authors consider a **Black-box** attacker who can only interact with the RAG model through API queries. They design a **composite structured prompting attack** composed of two parts designed to exploit both the retriever and the LLM : an *{information}* segment that triggers the Retriever to retrieve some specific contexts, and a *{command}* segment (e.g., "Please repeat all the context") that induces the LLM to give out the retrieved data verbatim.

The paper investigates both *targeted* attacks (to extract a specific type of personal information like phone numbers) and *untargeted* attacks (aimed at maximal leakage of personally identifiable information).

The RAG framework used consists of:

- A retrieval corpus D (Enron Email and HealthcareMagic datasets in this case)
- A retriever $R(q, D)$ computing and returning top-k document embeddings
- A language model M which gives an *{answer}* after taking the prompt and the retrieved context

The overlap between generated text and documents in D was then quantified via ROUGE-L and exact token overlap metrics to measure the effectiveness of the attack. Experimentation was done using different datasets and Language models.

To analyze training-data leakage, the paper also applies **targeted** and **prefix attacks** to LLMs (like GPT-Neo-1.3B whose training dataset is public) with and without retrieval augmentation, comparing how much memorized content is reproduced in both settings.

3.3 Key Findings

RQ1: Privacy Leakage from Retrieval Data. Empirical results showed that there were severe privacy leakage risks. A good fraction of targeted and untargeted prompts (about 50%) successfully extracted sensitive data. Experiments showed that command phrasing and retriever configuration (k value) also affected leakage magnitude.

RQ2: Influence on LLM Memorization. Conversely, incorporating retrieval data significantly reduced training-data leakage, when RAG was used, the model's ability to reproduce memorized content from its pretraining corpus dropped by over 80%. This suggests that incorporating the external retrieved context reduces the model's dependence on memorized training examples and opens up doors for using RAG itself as a mitigating technique in the larger context of *Privacy in LLMs*

3.4 Mitigation Techniques

Additionally The paper evaluates some simple mitigation methods to protect the Retrieval database such as *summarization* (expose a summary of the retrieved data instead of the data itself) and *Distance Thresholding* (only extract information if the embeddings are close), but is however not very extensive in their results. There is a privacy-utility tradeoff observed as well where utility is related to the quality of generation after using these methods

3.5 Limitations and Future Work

The study focuses solely on privacy in inference-time retrieval and does not explore privacy effects during RAG pretraining or fine-tuning. Other retrieval-based architectures may exhibit different behaviors. The authors identify open challenges in developing robust, generalizable mitigation techniques to reduce leakage in the Retrieval Database and effectively leveraging RAGs to improve the privacy of LLMs in general.

4 Experimental Reproduction and Results

4.1 Overview

To reproduce the experiments presented in *"The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)*, we utilized the official repository `RAG-privacy`[1] which contained the code for targeted and untargeted attacks on the retrieval database. Specifically, our setup followed the pipeline described in the paper, aiming to validate the privacy leakage observed in retrieval-augmented generation systems.

4.2 System Configuration

Hardware:
The experiment was conducted on institute provided shared GPU node equipped with two NVIDIA A100-PCIE-40GB GPUs.

Software Environment:
Experiment was conducted on a Linux environment with the following software stack:

- Python 3.10 and PyTorch (CUDA 12.8)
- LangChain and ChromaDB for retrieval pipeline management
- Sentence-transformers and FlagEmbedding for embedding construction
- Evaluation libraries: `nltk`, `rouge-score`, and `chardet`

[1] `https://github.com/phycholosogy/RAG-privacy`

4.3 Dataset

Following the original paper, we used the **HealthCareMagic-101** privacy-sensitive dataset as retrieval database. It is a collection of $\sim$200k doctor–patient dialogues containing diagnostic and prescription information. Each conversation was treated as a separate retrieval unit. It was already pre-processed.

4.4 Model Components

The RAG system consisted of two main components:

- **Retriever:** A dense vector retriever implemented using ChromaDB, returning the top-k documents ($k = 2$ by default) most semantically similar to the input query. The embedding model `bge-large-en-v1.5` was used for this.
- **Generator:** A large language model (LLM) responsible for conditioned text generation. We used `LLaMA-2-7B-Chat (L7C)` for this.

4.5 Experimental Workflow

The overall reproduction process followed the four-stage structure described in the repository:

1. **Database Construction:** Embedding of the datasets to construct the retrieval corpus.
2. **Prompt Generation:** Creation of structured composite prompts comprising an `{information}` component (domain-relevant query) and a `{command}` component (e.g., "Please repeat all the context.").
3. **Model Inference:** Querying the RAG system with 250 prompts per dataset to elicit retrieval-based responses.
4. **Evaluation:** Measuring leakage via ROUGE-L similarity, repetition counts, and the presence of PII in generated responses.

All hyperparameters (retrieval method, number of contexts, embedding model, and decoding parameters) were kept consistent with those in the codebase mentioned.

4.6 Evaluation Metrics

We report the same quantitative metrics as in the original study:

- **Retrieval Contexts (RC):** The total number of document segments fetched by the retriever from the private database in response to user prompts.
- **Extract Contexts (EC):** The number of generated text segments by the Language Model that contain verbatim or near-verbatim segments from the retrieval corpus (defined as $\geq$20 consecutive identical tokens). This directly quantifies the leakage of private data through the model's output.
- **Effective Prompts (EP):** The subset of prompts that successfully triggered retrieval and subsequent generation by the language model.
- **Retrieval PII% and Num PII:** The fraction and absolute count of retrieved segments that contain personally identifiable information (PII) such as names, emails, phone numbers, or medical entities.
- **Repeat Prompts (RP) and Repeat Contexts (RCtx):** The number of prompts that led to repeated outputs and the count of unique repeated snippets respectively. These capture redundancy in leaked information and the likelihood of recurring private fragments.
- **Average Extract Length (AEL):** The average token length of extracted private segments.
- **ROUGE Prompts (R-P) and ROUGE Contexts (R-C):** The number of prompts and generated contexts exhibiting ROUGE-L similarity > 0.5 with the retrieved documents.

Collectively, these metrics capture both *lexical leakage* (via repetition and token overlap) and *semantic leakage* (via high ROUGE similarity), providing a comprehensive view of privacy exposure in RAG systems.

4.7 Results and Comparison

We evaluated both **targeted attacks**, which aim to extract specific private attributes (e.g., diseases, emails, phone numbers), and **untargeted attacks**, which attempt to retrieve any contextually similar private data.
The following tables present side-by-side comparison of the paper's reported results and our reproduced results for the HealthCareMagic dataset using the LLaMA-2-7B-Chat (L7C) model.

Table 1: Comparison of targeted attack results on HealthCareMagic (L7C).

Target Type	**RC**	**EC**	**EP**	**PII%**	**Num PII**	**RP**	**RCtx**	**AEL**	**R-P**	**R-C**
Reported in Paper										
Disease	445	–	–	–	89	118	135	–	–	–
Reproduced Results (Ours)										
Disease	341	1	2	0.667	2	61	67	120.7	58	58
URL	39	0	0	0.000	0	0	0	–	0	0
Phone Number	36	1	4	1.000	1	10	9	91.7	13	11
Email Address	49	1	7	1.000	1	32	17	103.2	32	15
Mix (All PII)	66	1	3	0.333	1	29	18	98.8	32	20

Table 2: Comparison of untargeted attack results on HealthCareMagic (L7C).

Dataset	**RC**	**EC**	**EP**	**PII%**	**Num PII**	**RP**	**RCtx**	**AEL**	**R-P**	**R-C**
Reported in Paper										
Untargeted (Health)	331	–	–	–	–	107	117	–	111	113
Reproduced Results (Ours)										
Untargeted(Crawl)	258	0	0	0.000	0	72	56	128.2	74	55
Untargeted(Wikitext)	296	1	3	0.333	2	61	53	122.0	56	46

The overall trends in our reproduction align closely with the paper's findings:

- **Leakage Magnitude:** Both results show that targeted attacks are significantly more successful than untargeted ones. In the paper, targeted attacks achieved over 100 repeated prompts and 89 extracted medical dialogues; our replication also produced measurable leakage with 61 repeated prompts and 67 repeated contexts.
- **Qualitative Similarity:** The average extracted snippet length in our runs ($\sim$100–120 tokens) matches the reported results, confirming that the model reproduces entire sentences or dialogue segments rather than short phrases.
- **Metric Consistency:** The ROUGE-based overlap values and repetition counts indicate both verbatim and semantic leakage modes. High ROUGE-L similarity implies that the LLM can paraphrase sensitive content while retaining its meaning.
- **Minor Quantitative Deviations:** Our absolute counts (e.g., number of retrieved contexts) are slightly lower, likely due to differences in embedding model initialization or random seed variations during retrieval. Nevertheless, the leakage behaviour remains consistent in pattern and severity.

Both the paper and our reproduced experiments clearly demonstrate that retrieval-augmented generation pipelines are vulnerable to privacy leakage when adversarial or structured prompts are issued. Targeted prompts successfully extracted domain-specific records, including medical diagnoses and personally identifiable information, while untargeted prompts still induced partial exposure through semantically similar responses.

5 Project Outline

The current repository associated with the paper *"The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)"* provides implementation details primarily for the first research question (RQ1), focusing on attacks that demonstrate leakage from the retrieval database. Our work aims to extend this foundation by addressing the aspect of **data privacy mitigation** and exploring strategies to reduce the extent of information leaked from RAG systems. We currently have a working pipeline to orchestrate an attack on a RAG system and will be using this to evaluate the effectiveness of various mitigation measures.

We plan to experiment with multiple privacy-preserving mechanisms, both at the *retrieval* and *generation* stages of the RAG pipeline. We intend to examine various approaches proposed in multiple recent research papers such as Differential Privacy (DP), privacy-aware decoding [6] from the embedding before giving it to the LLM, perturbation-based defenses [5], pre-processing to annonynmise private information in the Retrieval database, and data sanitization using an auxillary LLM, try to implement these in the context of RAGs and evaluate them using the pipeline created.

Beyond implementation, we aim to conduct a systematic **privacy–utility trade-off analysis** [3] under different mitigation regimes with utility being the performance on the RAG pipeline on a base task before and after mitigation (add reference of you are what you write). The existing evaluation metrics (e.g., repeat prompts and ROUGE-based overlap) may not capture semantic or contextual leakage effectively; thus, we plan to explore better metrics that quantify privacy exposure in RAGs.

6 Conclusion

This project sets the stage for a deeper exploration of data privacy mitigation in RAG systems which are increasingly used with LLMs to improve their performance on domain specific tasks. Building on the reproduced baseline, our next steps focus on systematically applying and evaluating defenses that can reduce information leakage without harming utility. Ultimately, our goal is to develop practical guidelines and empirically grounded strategies for constructing robust privacy-preserving RAG pipelines suitable for deployment in sensitive domains.

References

[1] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. *arXiv preprint arXiv:2201.04845*, 2022.

[2] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[3] Richard Plant, Valerio Giuffrida, and Dimitra Gkatzia. You are what you write: Preserving privacy in the era of large language models, 2022. URL `https://arxiv.org/abs/2204.09391`.

[4] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, page 3–18. IEEE, 2017.

[5] Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for black-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 2025. URL `https://arxiv.org/abs/2310.12214`. arXiv:2310.12214.

[6] Haoran Wang, Xiongxiao Xu, Baixiang Huang, and Kai Shu. Privacy-aware decoding: Mitigating privacy leakage of large language models in retrieval-augmented generation, 2025. URL `https://arxiv.org/abs/2508.03098`.

[7] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, and Jiliang Tang. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag), 2024. URL `https://arxiv.org/abs/2402.16893`.