Integrating Access Control with Retrieval-Augmented Generation: A Proof of Concept for Managing Sensitive Patient Profiles

Bingxiang Chen
Tampere University
Tampere, Finland
bingxian.chen@tuni.fi

John Tackman
Solita
Finland
john.tackman@solita.fi

Manu Setälä
Solita
Finland
manu.setala@solita.fi

Timo Poranen
Tampere University
Tampere, Finland
timo.poranen@tuni.fi

Zheying Zhang
Tampere University
Tampere, Finland
zheying.zhang@tuni.fi

Abstract

With advancements in Generative AI, particularly large language models (LLMs), there is significant potential for developing domain-specific AI chatbots. However, training on sensitive data, such as healthcare information, poses risks of unauthorized data leakage. Access control is essential to ensure that only authorized personnel can access sensitive training documents. This study proposes integrating fine-grained access control with Retrieval-Augmented Generation (RAG), a promising architecture that enables models to retrieve external data and generate contextually accurate responses. By combining RAG with access control, Generative AI can produce answers strictly based on documents permitted by user rights. This is particularly critical in healthcare, where only authorized personnel, such as doctors and nurses involved in treatment, should access patient-specific information. Using the design science research methodology, we developed a proof-of-concept system and evaluated it with patient profiles and varying access permissions. While not solving all data management challenges, this approach offers a promising solution for secure, domain-specific knowledge applications within LLMs.

CCS Concepts

• **Security and privacy → Software and application security**; **Domain-specific security and privacy architectures**;

Keywords

Large Language Models, Retrieval-Augmented Generation, RAG, access control

ACM Reference Format:
Bingxiang Chen, John Tackman, Manu Setälä, Timo Poranen, and Zheying Zhang. 2025. Integrating Access Control with Retrieval-Augmented Generation: A Proof of Concept for Managing Sensitive Patient Profiles. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3672608.3707848

SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707848

1 Introduction

Large language models (LLMs) are transforming industries by enabling the creation of generative AI tools tailored to specific business needs, particularly in conversational AI and healthcare [4]. These technologies enhance efficiency and accuracy but also raise significant concerns regarding data security, especially when handling confidential data. Ensuring that LLMs follow fine-grained access permissions is crucial to prevent unauthorized access and data leaks, which can harm privacy and reduce trust. For example, in healthcare, it is critical to ensure that only authorized personnel, such as doctors and nurses directly involved in a patient's care, have access to patient-specific information. This access control is vital for maintaining regulatory compliance, such as adhering to the General Data Protection Regulation (GDPR) [14]. Despite the importance of such mechanisms, many current LLMs lack built-in access control, which leaves sensitive data vulnerable to misuse.

The integration of access control into LLMs is a new and growing area of research, particularly in fields like healthcare. Traditional methods, such as Role-Based Access Control (RBAC), work well to protect sensitive patient data [12]. However, they are not designed to make sure that LLM-generated content follows specific user permissions or access rules. Most current research focuses on general security issues with LLMs, like preventing data leaks [4]. While these efforts address overall risks, they do not solve the specific problem of ensuring that LLMs generate content that matches a user's access rights. Approaches like Retrieval-Augmented Generation (RAG) enable models to retrieve external data for generating contextually accurate responses [9]. Although RAG lacks inherent access control mechanisms, its architecture offers a potential foundation for integrating permission-aware content generation by dynamically filtering retrieved data based on user-specific permissions. This represents a promising direction for addressing compliance and trust in sensitive domains like healthcare.

This study investigates how access control mechanisms can be effectively integrated into LLMs to secure domain-specific data. Building on our earlier findings [2], we present a novel approach to embedding access control within RAG systems. To demonstrate the feasibility of this approach, we have developed a Proof of Concept (PoC) tailored for healthcare settings.

2 Methodology

This study follows a structured approach to guide the development of a proof-of-concept (PoC) system for integrating access control into generative AI. The methodology is organized as follows:

Problem Identification, Motivation, and Objective: Generative AI systems, particularly in sensitive domains like healthcare, often lack integrated access control, exposing sensitive data to risks of unauthorized access, privacy breaches, and misuse. Implementing effective access control mechanisms is essential to ensure that users can access only the information relevant to their roles, safeguarding sensitive knowledge. This study aims to develop a prototype that integrates access control mechanisms into generative AI systems, ensuring dynamically filtered content based on user permissions while adhering to data protection standards.

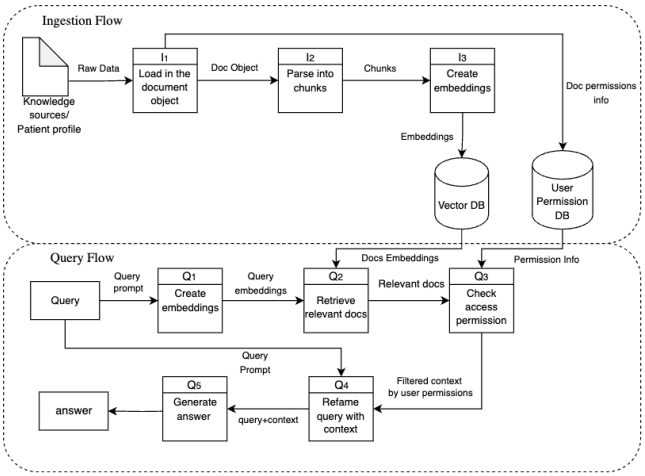

Figure 1: Overview of the RAG System with Access Control.

System Design: Figure 1 illustrates the system's architecture, which consists of two main flows: ingestion and query. The ingestion flow processes data from various sources by segmenting it into chunks, embedding the chunks into vectors, and storing these embeddings in a vector database. Simultaneously, a permissions database is updated to link document identifiers with their corresponding access controls, defining which users or roles can access specific documents.

In the query flow, user queries are embedded into vectors for semantic search within the vector database. The retrieved documents are cross-referenced with the permissions database to verify access rights. After filtering the documents, the authorized ones are combined with the user query and submitted to the LLM to generate a response.

Application Development: The system is built using FastAPI [7] for the backend and React [11] for the frontend, ensuring an efficient user experience. LlamaIndex [10] coordinates the RAG pipeline, integrating both private and public data for optimal LLM performance. SQLite manages user data and permissions, while ChromaDB [5] handles the vector database for document embeddings. We chose Mistral-7B [8], an open-source 7-billion-parameter model, for its language generation capabilities and the ability to deploy locally for enhanced data privacy. For embeddings, we use the BAAI/bge-small-en-v1.5 model [1], optimized for English.

Figures 2, 3, and 4 illustrate the User Management interface, Knowledge Management page, and the query-response system based on access permissions, respectively.

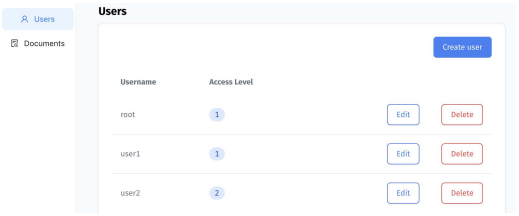

Figure 2: User Management.

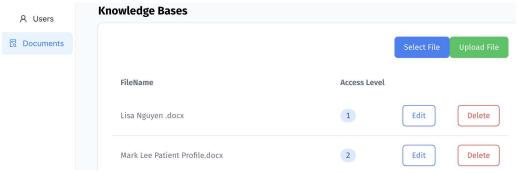

Figure 3: Docs Management.

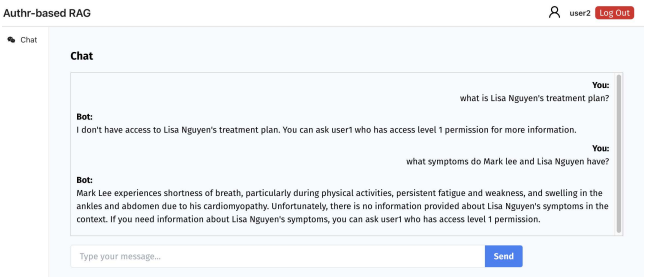

Figure 4: Chat.

Evaluation: We evaluated a demo application handling medical data, focusing on privacy and access control. GPT-4 [13] generated 40 patient profiles (700–800 words each), including personal details, medical history, and treatment plans. Access control had two levels: AL1 (full access) and AL2 (restricted to three profiles). The tests ran on a Google Compute Engine instance with 16 vCPUs, an NVIDIA L4 GPU, 64GB RAM, and 128GB storage.

The primary evaluation tested the system's adherence to access permissions, ensuring AL2 users could access only relevant profiles, while AL1 users had full access. For example, the query "What symptoms do Mark Lee and Lisa Nguyen have?" allowed AL2 users to access Mark's profile but restricted Lisa's data. Secondary evaluations compared the RAG system with a non-RAG approach. The non-RAG method processed all 40 profiles together, while the RAG system retrieved only query-relevant portions.

We used Retrieval Augmented Generation Assessment (RAGAs) metrics [6], including Answer Relevancy, Context Relevancy, and Context Recall, to evaluate how well the response matches the question and the retrieved context aligns with the ground truth.

Table 1: Examples of Query Responses Varying by User Access Permissions.

Question	User Access Level	Context	Answer
What is Lisa Nguyen's treatment plan?	2	[null]	I don't have access to Lisa Nguyen's treatment plan. You can ask user1 who has access level 1 permission for more information.
What is Lisa Nguyen's treatment plan?	1	["Prescribed triptans for immediate relief during migraine attacks ..."]	Lisa Nguyen's treatment plan includes taking triptans for immediate relief during migraine attacks and a beta-blocker as a preventive measure ...
What symptoms do Mark Lee and Lisa Nguyen have?	2	["Mark experiences shortness of breath, particularly during physical activities..."]	I only have information about Mark Lee's symptoms, which include shortness of breath... For Lisa Nguyen's symptoms, ask User1 with access level 1...
What symptoms do Mark Lee and Lisa Nguyen have?	1	["Mark experiences shortness of breath, particularly during physical activities...", "Lisa Nguyen - Symptoms..."]	Mark Lee experiences shortness of breath, fatigue, and swelling due to cardiomyopathy... Lisa Nguyen has severe pulsating headaches with nausea...

3 Results and Discussion

Table 1 presents examples of the system's responses based on user access levels. AL1 users had full access to all profiles, while AL2 users were limited to Mark Lee's profile. The table shows how access restrictions impact the information retrieved and advises AL2 users to contact an AL1 user for further assistance. We also tested the system with random questions for different profiles and confirmed that the responses were appropriately based on user permissions, demonstrating the system's accuracy in enforcing access control.

Table 2: Performance Comparison of Systems Using RAGAs.

	Q1: What symptoms does [Patient Name] have?	Q2: What is [Patient Name]'s contact information?	Q3: What medications does [Patient Name] currently take?	Average
Answer Rel. (RAG)	0.9936	0.9739	0.9324	0.9772
Answer Rel. (No-RAG)	0.9181	0.9827	0.9608	0.9622
Cont. Rel.	0.0699	0.0749	0.0744	0.0687
Cont. Recall	1	1	1	1

Table 2 presents the performance comparison between the RAG and non-RAG systems using the RAGAs framework. The RAG system outperformed the non-RAG system in Answer Relevancy, indicating better precision in retrieving relevant information. Although the RAG system showed lower Context Relevancy, it achieved perfect Context Recall, ensuring all relevant data retrieval.

The PoC was successful, demonstrating that our local RAG model can match the accuracy levels of GPT-4 in generating answers based on user permissions while effectively respecting access control. When users attempt to access restricted information, the system suggests contacting a user with the necessary permissions. While the PoC shows potential for real-world applications, its scope was limited by a simple numeric access control model and a small patient profile sample size, reducing its external validity.

4 Conclusions

This work demonstrates the potential of integrating access control mechanisms in healthcare systems to protect sensitive patient data while leveraging generative AI. We proposed an approach that combines access control with the RAG system to enhance data security and privacy. The PoC successfully showcased the feasibility of this integration, making it a promising solution for real-world applications. The RAG architecture offers a strong foundation for enabling models to retrieve external data and generate contextually accurate responses, and this capability can be effectively combined with access control to ensure that only authorized users can access relevant information. However, the basic access control model and limited sample size suggest that further testing with larger datasets and more sophisticated models is needed. Future work will focus on refining the access control model to handle more complex scenarios and scaling the system for real-world deployment. Expanding testing to diverse datasets and optimizing the system to securely handle complex queries at scale will be crucial for moving beyond the PoC phase. The source code is available on GitHub [3].

5 Acknowledgement

This work was supported by Solita Oy and Business Finland through the ITEA/Secur-e-Health research project.

References

[1] BAAI Team. 2024. BAAI. https://huggingface.co/BAAI/bge-small-en-v1.5 Accessed on 10 April 2024.
[2] Bingxiang Chen. 2024. An Exploration of Using Retrieval-Augmented Generation With Access Control. (2024). https://trepo.tuni.fi/handle/10024/159017 Master's thesis, Tampere University.
[3] Bingxiang Chen. 2024. Authorization-Based Data Access for RAG-Enabled Generative AI. https://github.com/bingxiangch/thesis_auth_rag/ Accessed on 26 May 2024.
[4] Y. Chen and P. Esmaeilzadeh. 2024. Generative AI in Medical Practice: In-Depth Exploration of Privacy and Security Challenges. *J Med Internet Res* 26, 1 (2024), e53008. https://doi.org/10.2196/53008
[5] Chroma Team. 2024. Chroma. https://www.trychroma.com/ Accessed on 26 May 2024.
[6] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated Evaluation of Retrieval Augmented Generation. arXiv:2309.15217 [cs.CL] https://arxiv.org/abs/2309.15217
[7] FastAPI Team. 2024. FastAPI. https://fastapi.tiangolo.com/ Accessed on 21 April 2024.
[8] Albert Q. Jiang and Alexandre Sablayrolles et al. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL]
[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
[10] LlamaIndex Team. 2024. Llamaindex. https://www.llamaindex.ai/ Accessed on 26 April 2024.
[11] React Team. 2024. React. https://react.dev/ Accessed on 21 May 2024.
[12] Ravi S Sandhu. 1998. Role-based access control. In *Advances in computers*. Vol. 46. Elsevier, 237–286.
[13] OpenAI Team. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774
[14] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.