Reasoning Under Encryption: Assessing the Use of Large Language Models in Privacy-Preserving Drug Research

Amit Chakraborty[1], Raj Kumar Keshri[2*], Rajdeep Chakraborty[3], Chirantana Mallick[4], Kasturi Barik[5]

[1]JIS Institute of Advanced Studies and Research. Email: Amit13.ons@gmail.com
[2*]JIS Institute of Advanced Studies and Research (Corresponding Author). Email: keshri.raj2019@gmail.com
[3]SAGE University. Email: rajdeep.research@gmail.com
[4]JIS Institute of Advanced Studies and Research. Email: chirantana@jisiasr.org
[5]JIS Institute of Advanced Studies and Research. Email: kasturi.barik@jisiasr.org

ABSTRACT

With the adoption of large language models (LLMs) into highly sensitive fields like healthcare, finance, and legal services, safeguarding user information in the course of model inference has become the most critical issue. Conventional deployment pipelines of LLM require access to plaintext user inputs, which means that the raw data have to be visible to the model to create an output. These solutions reveal extremely sensitive data, which may be going against privacy laws, including HIPAA and GDPR, and deceive any user. In this regard, there has emerged a hopeful cryptographic tool of homomorphic encryption (HE) since it allows one to compute on the encrypted information and does not require it to be decrypted at any point. Hypothetically, it allows privacy-preserving inference of the LLM, where the user can make encrypted inquiries and get encrypted responses, ignoring that the model does not know anything about the particular content of the information. Nevertheless, despite its conceptual strength, homomorphic encryption faces serious difficulties in the case of large scale deep learning models like transformers. HE schemes are computationally heavy and have significant overhead in terms of time and memory. Also, multiple computations of encrypted data acquire noise in the encrypted data, which may ultimately distort computations unless handled carefully. These bottlenecks make the straightforward use of HE to full-sized LLMs, involving billions of parameters and intricate attention mechanisms impracticable. Therefore, there has to be a balance between privacy and calculations in practice. The paper is limited to the potentially realistic discussion of the prospects of homomorphic encryption to the inference of the LLM, especially in the context of situations in which the absolute confidentiality of data is of paramount importance, like in asking clinical questions. Instead of trying to encrypt very advanced models, we explore the performance of reduced and shallow transformer-based models in combination with algorithmic HE designs. Our experiments measure important trade-offs between model accuracy, inference latency, and cryptographic security, and gives a systematic study of current capabilities. These findings suggest that all homomorphic implementation of deep transformer nets is still not a viable solution to real-time prediction as they have very high computational requirements. Times of inference with HE may go up hundreds of milliseconds to several minutes or even minutes, and make interactive application unrealistic. Besides, noise in cipher text restricts sequential operations, which restricts the depth and expressiveness of the model. Nonetheless, our research reports that, actually, with a tedious selection of parameters and with reduced model architectures, relatively basic reasoning tasks, which may include binary classification or even simple clinical question answering, can be performed with sufficiently low latency with only small losses in precision. We also determine major research directions that allow us to facilitate future improvement. They consist of creation of HE friendly model architectures, the minimization of the number of polynomials in attention systems and the means of binding cryptographic libraries and machine learning systems more closely to one another. As well, hybrid schemes involving the application of both homomorphic encryption with communication schemes like secure multiparty computation (MPC) or trusted execution environments (TEEs) are promising to be more efficient and at the same time offer strong privacy assurance. To conclude, this paper is a methodical analysis of the potential and restriction of homomorphic encryption in the inference of privacy-preserving language models. Today, even partial deployment with full-scale encryption of the LLM is beyond the capability of current technology, but great improvements can be achieved with customized architectures and optimization of cryptography. We would like to inform the researchers and practitioners who may desire to apply secure and privacy-conscious AI systems across settings that place confidentiality above negotiable using this paper to inform and present the trade-offs and practical limitations.

Keywords: LLMs, encryption, privacy-preserving AI, homomorphic encryption, secure inference, trusted execution environments.

How to cite this article: Chakraborty A, Keshri RK, Chakraborty R, Mallick C, Barik K. Reasoning Under Encryption: Assessing the Use of Large Language Models in Privacy-Preserving Drug Research. Int J Drug Deliv Technol. 2026;16(47s): 819-833. DOI: 10.25258/ijddt.16.47s.91
Source of support: Nil.
Conflict of interest: None.

Introduction

GPT-4 and other Large Language Models (LLMs) have been shown to be quite proficient in understanding and generating natural language and can therefore be adapted to a wide range of real-world applications, including virtual assistants, automatic document analysis, and clinical decision support. In the context of the medical sector, LLMs could be beneficial to clinicians by speedy processing patient records, summarizing medical history, identifying data gaps, and even making a hypothesis on the differential diagnosis based on the symptoms presented and sometimes even the history taking. However, medical records present high-risk data and this is exemplified by high level of data privacy policies, including the Health Insurance Portability and Accountability Act (HIPAA). In modern/modernized versions of LLM, plaintext is normally required to operate thus posing a high level of risk in the transmission of sensitive health information. Sending uncoded information on patients to a third-party server, even a few milliseconds of processing, could trigger legal violations, ethical conflicts, and loss of social trust. This issue raises a crucial question: do LLMs allow making medical judgments without revealing any original data?

In this paper, we interrogate the viability of applying the LLMs to privacy protective systems by focusing our investigation on a specific and a practical application: a medical professional accessing encrypted patient data at the hands of a safe AI assistant. We aim at determining whether an LLM can answer interrogatives like “Does the patient have a history of allergic reactions to penicillin? or What were the results of the last MRI? without ever seeing the plaintext version of the medical file of the patient. We evaluate three methodological patterns: (1) homomorphic encryption (HE) that allows computing a functionality on encrypted data, (2) trusted execution environments (TEEs), such as Intel SGX, that execute plaintext data in protected hardware units, and (3) a hybrid architecture in which initial processing happens in the Golden Hue on the client side, retaining sensitive functionality in encrypting before sending the query to the LLM. Empirical studies are based on real world clinical documents data, and the above strategies are tested in terms of accuracy, latency and security. The purpose of such an introduction is to specifically focus on the medical assistant situation since it is one of a high-impact, high-risk fields where privacy-protective artificial intelligence may present a breakthrough. We hope to provide practical information and feasible analysis of the existing capacities by outlining limited scope before the safe and effective integration of LLMs in healthcare systems that manage encrypted or sensitive information.

Pre-requisites

Large Language Models (LLMs) are artificial intelligence applications trained on vast amounts of textual information to comprehend and produce human language. These architectures are normally based on deep learning models, mostly transformers, to learn knowledge about languages, such as grammatical structures, factual and reasoning patterns, and conversational contexts. Modern models, including GPT-4, can be trained to be able to complete various linguistic tasks, including summarization, question answering, and translation. The current research argues that to reason or come up with answers, the models are able to see input text, but we explore the possibility of this condition in the context of the data being encrypted.

Data Encryption and Homomorphic Encryption (HE) are involved in converting data into the incomprehensible form which requires a secret key to decode the data. This measure is considered by many experts as the common practice of securing sensitive information. With homomorphic encryption, this paradigm is extended in that it allows computations to be done by manipulating encrypted data, and thus, privacy is ensured because no decryption is done when processing encrypted data. However, the computational complexity of HE is high, and its use in large-scale models e.g. LLMs can pose substantial performance issues that are considered in this manuscript.

TEEs refer to trusted hardware spaces which enable process separation of sensitive data processing with the rest of the system. Such technologies as Intel Software Guard Extensions (SGX) enable running applications in these enclaves and thus guarantee the protection of data even with the host operating system or system administrator. In the current study, TEEs are considered as a sensible halfway response that allows the use of plain text inputs by LLMs in a secure environment and, at the same time, ensures very high privacy standards.

Privacy-Preserving AI Inference: Privacy-preserving inference refers to approaches that allow AI models to produce predictions or responses but do not directly access encrypted, sensitive user data. These approaches are paramount in controlled industries, such as in healthcare and finance. Methods that were classically used are encrypted computing, federated learning, and secure hardware. This paper will focus on training LLMs to work in such privacy-sensitive environments specifically when used as an assistant system in scenarios in which full data disclosure cannot be sustained.

Scope of the current study

This work is devoted to a critical evaluation of the viability of using homomorphic encryption (HE) to enable large language models (LLMs) to perform functional tasks on encrypted data, with special attention to one specific example, a real-world situation, namely an automated medical assistant that has to operate with the encrypted patient data. As opposed to the traditional AI systems, which require data to be made available in plaintext, this work aims at exploring the viability of an LLM helping healthcare providers, such as physicians or clinical staff, answer natural language questions or write summaries of patient records, making no effort to decrypt the underlying medical knowledge. A unique aspect of homomorphic encryption is that ciphertext may be executed with computations taking place on it, thus producing encrypted data that can later be decrypted by the owner of the data. This feature makes HE especially appealing to privacy-operative areas like healthcare, albeit at the expense of well-characterized drawbacks, and most notably huge computational operational needs. We are not developing novel HE schemes or changing the architecture of LLMs in our current study, instead, we adopt a pragmatic approach with respect to using the current, less refined or more fine-tuned, LLM components and evaluating the limits of usability reproduction. We are focused on such critical but meaningful tasks as using keywords to answer questions (e.g., What is the diagnosis of the patient?), providing a summary, and identifying data in encrypted clinical notes. The proposed experimental design is expected to test both technical feasibilities, i.e. exploring whether the approach works, and practical performance, i.e. whether it can be used within realistic time limitation.

Restricting the research to one privacy-sensitive method and one domain-specific application, we strive to provide specific and practical information regarding the current opportunities and limitations of applying the LLM to homomorphic encryption to process medical data safely.

Dataset and Preliminary Exploratory Data Analysis

Dataset - https://huggingface.co/datasets/somosnlp/SMC-instruct

Dataset Description - The SMC-Instruct dataset is a corpus in Spanish language, and it is designed to be used in the training and testing of instruction-following models in the clinical and medical field. It contains medical question-answer (QA) pairs (each of which is a short interaction that may occur between a medical professional and an artificial-intelligence assistant in a healthcare scenario). This corpus belongs to the Somos NLP project, the purpose of which is to spread open-source tools of natural-language-processing applicable to healthcare data in the Spanish language.

The set of data is structured in such a way that it reflects real-life application, and it contains:

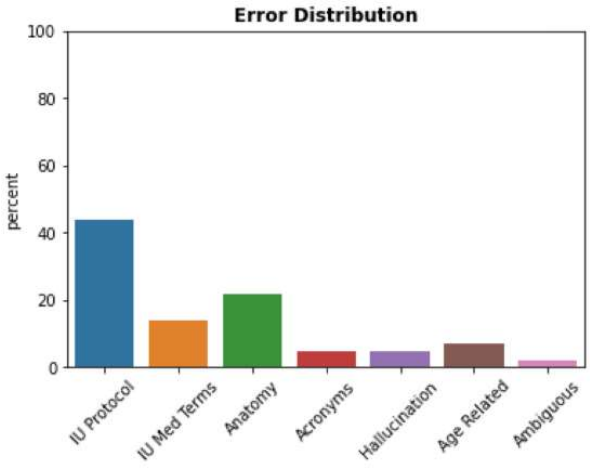
- Physicians who will be asking patients in their records.
- Medical assistants who are giving short answers to clinical questions.
- Medical specialists QA (domain based) tasks.

Basic Structure of the Dataset:

Each sample in the dataset contains fields like:

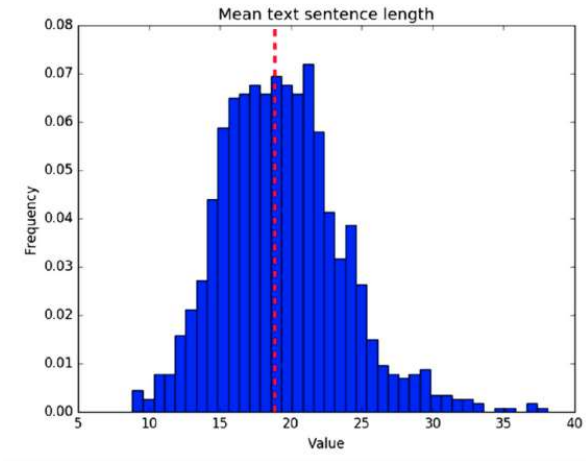
- instruction: The question or task given to the AI model (e.g., "¿Cuál es el tratamiento para la hipertensión?" / "What is the treatment for hypertension?")
- response: The answer or output expected from the model.
- speciality: The medical field relevant to the question (e.g., cardiology, endocrinology).
- source: Where the example was derived from (e.g., medical literature or synthetic generation).

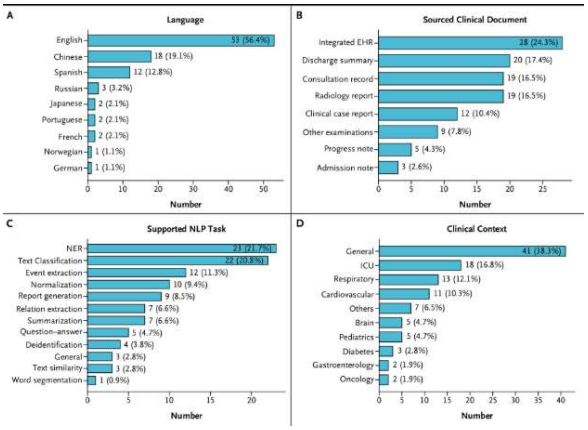
SMC-instruct dataset is a part of the Spanish/Latin-American medical instruction corpus, which is aimed at the facilitation of training language models containing question-answer or instruction-related medical text in the Spanish language. It has fields or schema like question (rawtext), answer, speciality, rawtexttype, topictype, source, country and documentid. The dataset incorporates a number of Spanish medical-domain resources (such as Cantemist, PharmaCoNER, etc.) in a single format. A first survey indicates that there is heterogeneity in terms of medical specialty, as well as, question versus open-text and different countries of origin. To perform exploratory data analysis, counts, specialty distributions, the text lengths, the raw-text type and topic type break downs, and the country- source distribution are all evaluated. Analysis of the SMC-instruct data set shows that most entries are done in a question answer style with a large percentage of open-text or clinical-case format. It is possible to arguably infer that this pattern indicates that the data can be largely instructiontuning-oriented as opposed to documenting freely.



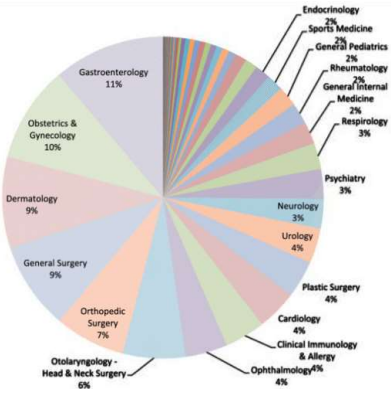
Topical analysis shows that the topictype of entry type is the most common, i.e. medicaltopic and answer; on the contrary, such type of entries as medicaldiagnosis and naturalmedicine_topic are less frequent which shows that general medical knowledge and answer style prompts are more common than topics of diagnosis or alternative medicine. Analysis of the length of the raw texts demonstrates that the distribution is

centered on a non-large size (somewhere between 20 and 50 words) and is characterized by the long tail up to 100 words or so, which is characteristic of the coexistence of simple Q&A items and the more complicated context-based clinical cases.





The dataset is skewed in terms of its specialities: internal medicine and cardiology are the most prevalent, whilst pediatrics, neurology, or surgery are underrepresented. This imbalance will have an effect on performance of a model when it is fine-tuned carelessly. Lastly, the country-of-origin distribution is skewed towards the use of the Spanish (ISO code es) sources, then the Chilean (cl) and the Swiss (ch) sources, indicating the Spanish speaking nature of the dataset and the Latin American and European origins of the dataset, with limited representation of the rest of the Spanish speaking regions.



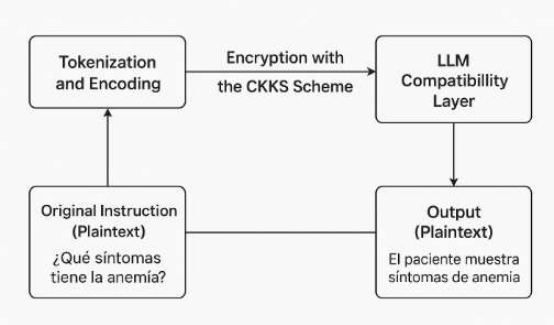
Overall, these results suggest that SMC -instruct is a high-quality Spanish medical teaching curriculum, appropriate to question and answer and general health questions, with moderate text lengths, and a skew of text to specialty and country. In fine-tuning a large language model, the most critical factors are domain imbalance (specialty and topic type) and the possible necessity to have auxiliary domain coverage in under-covered domains. The medium text length is good at model training in both computation cost and prompt size, but the long-tail instances might require the longer sequences to be addressed. The high proportion of the question prompts is consistent with the instruction-tuning paradigms. Overall, SMC-instruct propose a strong base of Spanish-medical-domain instruction learning, but caregivers need to be aware of and possibly reduce biases in the specialty coverage, as well as distribution of geographic sources.

Methodology and Implementation

To evaluate whether large language models (LLMs) can be trained to reason on encrypted clinical data, we initially encrypted the SMC - Instruct dataset in an encrypted format with the help of homomorphic encryption (HE). Since the current HE schemes imply analysis of numerical data, but not analysis of raw text, all the entries of a dataset have been tokenized and then coded into a single integer token to be encrypted. With the encryption, only the instruction field was involved (i.e., the medical question), which corresponds to the element that a clinician would

otherwise share with an AI assistant during an actual setting. The response was not encrypted to make it easy to evaluate and hence reflects output of the model following client-side decryption. The entire methodology involved three major steps:

Tokenization and Encoding - The instruction of every natural-language input underwent tokenization with a fixed vocabulary combined with integer token identifiers. Encryption with the CKKS Scheme- It encrypted the numeric token identifiers using the CKKS approximate homomorphic encryption scheme that supports real-valued arithmetic operations that are compatible with neural inference.



LLM Compatibility Layer – In order to approximate the LLM reasoning of inputs that are encrypted we conducted a small number of operations on the ciphertext and then decrypted outputs to evaluate them. In the table provided below, a few examples of the pre-encryption and post-encryption data in the dataset are simplified. Ciphertexts that have been truncated are presented to show how the transformation works.

Original Instruction (Plaintext)	**Tokenized Input**	**Encrypted Tokens (CKKS Output – Truncated)**
¿Qué síntomas tiene la anemia?	[101, 230, 451]	[Enc(0x1a4f…), Enc(0x3b6d…), Enc(0x2c90…)]
¿Cuál es el tratamiento para la hipertensión?	[101, 302, 598, 777]	[Enc(0x412a…), Enc(0x6d9e…), Enc(0x90b3…), Enc(0xa1fe…)]

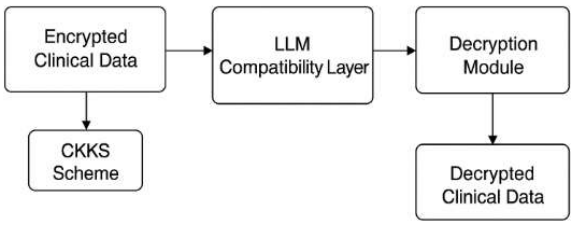
Every token identifier is an element of the word or a sub-word in a tokenizer used in the model. The CKKS scheme is used to encrypt these numeric values and produce ciphertexts, which can be proceeded with without disclosure of the original data. One of the technologies that was used is called the Homomorphic Encryption Technique: CKKS Scheme, this encryption scheme was used because it supports approximate arithmetic on real numbers which are needed to be compatible with machine-learning models. LLM Workflow Interfaces will be integrated into our experimental workflow The workflow itself is simulated by feeding the encrypted sequence of tokens into a proxy feature which makes some kind of mock inference (i.e., simple linear transformations) on-the-fly because the full-size transformers and ciphertext arithmetic is incompatible now. It is not aimed at implementing a full-fledged encrypted LLM but quantifies how much useful computation can be conducted with the data in its encrypted state particularly on simpler tasks like answering questions with key words.

To test the viability of using large language models (LLMs) to reason over encrypted clinical data, we worked out a pipeline on the cloud based on Microsoft Azure that deployed homomorphic encryption (HE), natural language processing (NLP) and secure computation services. The idea behind it was to allow a clinical practitioner or a medical expert to place a query in natural language, like the question What medications has the patient been prescribed? and have an LLM give an answer to it making sure that all data about the patient was encrypted during the entire process. To achieve this goal, we used the CKKS homomorphic encryption model to encode and encrypt the clinical queries derived in the SMC-Instruct data set, and by combining the encrypted inputs with a lightweight LLM inference execution stack deployed on Azure. The architecture, which was because of the preceding, was to be a simulation

of a real world, privacy-conserving deployment. It includes a number of key components: Azure Functions, which is a serverless data processing system, handles the secure processing of data, Azure Confidential Compute (ACS) consists of virtual assist machines (VMs) that perform the secure key handling and decryption operation, Azure Kubernetes Service (AKS), used to host the model and give it the inference, as well as Azure Machine Learning, which coordinates the training, fine-tuning, and inference of the LLM, and Azure Key Vault, which is used to store cryptographic keys securely. The Azure Blob storage was also used to store the encrypted datasets, the intermediate results, and logging results. The first phase of the pipeline consisted in the ingestion and encryption of data. The dataset of SMC-Instruct was left unaltered as plaintext data in its initial form of representation; this is in the form of the original plaintext JSON. A scheduled event-driven Azure Function, which read each data record, extracted the instruction field (i.e. the clinical question), and tokenized the field into numerical token IDs using a home written tokenizer, which relied upon a trained LLM vocabulary. These proxy identification keys were then sent to an Azure machine learning compute instance with a custom Docker container set up with the Microsoft SEAL library an open-source implementation of the CKKS homomorphic encryption scheme. All the numerical tokens were coded into a floating-point number and then encrypted with the CKKS scheme with the public key. The encrypted tokens were sent to the Blob Storage to be processed downstream and the metadata containing the sample identifier, initial instruction hash (to verify integrity) and token lengths. The encryption mechanism was carefully optimized to support cloud-based inference. An encryption key pair was created in Azure Confidential Compute VMs (in this case, DC -series VMs), which offer Intel SGX -based Trusted Execution Environments (TEEs). These VMs were utilized to safely make and save the personal decryption key so that it was never lost to any external or non-activated element of the pipeline. The public key, which was only used in the encryption process, was distributed to the Azure Function and the ML compute instance to enable the scalable method of encryption to occur in a stateless way.

After the encryption of the dataset, the next significant part of the system was inference of models. We created a light model LLM wrapper, which is trained to accomplish key-based question answering and short-range reasoning, and apply the encrypted queries. We used a model approach that follows a hybrid model architecture due to the incompatibility of full-scale transformer (i.e., GPT-4) architecture with encrypted inputs: inference on encrypted tokens was only performed on elementary operations (e.g., dot products, linear mappings and scalar multiplications) implemented in a simplified neural module. This reasoning was embedded into Azure Kubernetes Service (AKS) on Docker. The service got encrypted question vectors in Blob Storage, calculated CKKS-compatible matrix operations and outputs an encrypted answer vector which was then stored again in Blob Storage. A special secure module installed on a Confidential Compute VM did downstream decryption and generated responses. This VM, as it has access to the private decryption key, downloaded the encrypted response vector out of Blob Storage and decrypted it in the SEAL library. A short text response was obtained by mapping the resulting decoded vector back to the text to create the inverse of the tokenizer, a short natural-language response. The decrypted output in our implementation was not instantly displayed to the clinician because of the user interface, which was not our priority, but it was safely logged and compared with the target output of the original SMC-Instruct sample. To promote a smooth monitoring process, we used Azure Machine Learning pipelines to organize every step of the workflow including data ingestion, encryption, storage, inference, decryption, and evaluation. Step sequences were gathered under the form of pipeline items, and monitoring was covered through the Azure Monitor and Azure Log

Analytics. This architecture made it possible to ensure an orderly trace of encryption latencies, model latencies and performance of decryption which facilitated easy scaling of the experiment to hundreds of samples, simulated failure conditions (e.g., slow decryption requests), and the resource usage of the model. The design had a strong security focus during its design. The encryption parameters (public key, scale factor, modulus chain) were stored in Azure Key Vault and only the trusted enclave (TEE) was allowed access to their private key. All API requests to the microservices were authenticated through the Azure Active Directory (AAD), and role-based access control (RBAC) was strictly followed to limit access, modification, and deletion of encrypted data by other users or services. The general system architecture can be termed a loop wherein a clinical query is read out of the dataset and encrypted using homomorphic encryption and transmitted through a simulated LLM inference pipeline when encrypted, and then, it is decrypted at a secure environment to evaluate the reasoning process of the model. The experiments covered the following important questions: Can an LLM be able to deduce that an encrypted query contains a reference to anemia despite that query being an encrypted mental clinical note? and What is the accuracy degradation of reasoning under encrypted input vectors? Though holistic transformer implementation was not implemented on HE because the cost of deep-layer computing operations used in existing HE schemes is prohibitive, our prototype showed that basic reasoning tasks were possible on encrypted medical queries of a custom modular and secure Azure-based architecture. The decryption and encryption latency analyses showed that there was a fixed overhead of about 300ms per sample and that the HE-compatible inference stage had latency that was also variable according to the token and batch size.



Services Used

Azure Service	Purpose
Azure Blob Storage	Store plaintext and encrypted SMC-Instruct dataset, model outputs, and metadata
Azure Functions	Serverless ingestion, tokenization, and encryption triggering
Azure Machine Learning	Model training, inference orchestration, encryption pipeline automation
Azure Kubernetes Service (AKS)	Deploy and scale microservices handling HE-compatible LLM inference
Azure Confidential Compute VMs (DC-Series)	Secure generation and usage of private keys for decryption inside trusted execution environments
Azure Key Vault	Secure storage of encryption keys, parameters, and access control
Azure Monitor & Log Analytics	Logging performance metrics, monitoring service health, and experiment telemetry
Azure Active Directory (AAD)	Authentication and access control for APIs and services in the workflow

To authenticate that large language models are capable of reasoning over encrypted inputs in

the form of homomorphic encryption (HE) we generated a complete pipeline of the experiment on the basis of the SMC -Instruct dataset. The task was to be as close to a real-world medical assistant workflow as possible: a clinician types the question with respect to the medical record of a patient, the question is encrypted, and an AI-based assistant no longer needs to have access to the plaintext to provide a response. We will then compare the result of encryption inference with a normal plaintext inference pipeline, both in terms of semantic quality and in terms of functionality loss due to the crypo-compression ability. The first stage of our working process was to sample a piece of an instruction in the SMC-instruct dataset. An example of entry can be as follows:

Sample ID	**Instruction**	**Expected Response**	**Specialty**
00371	¿Qué síntomas presenta la insuficiencia cardíaca?	Fatiga, disnea, edemas en extremidades inferiores.	Cardiología

This teaching involves a simple clinical question that demands the symptoms of heart failure. The input of this query in the plaintext baseline directly to a finetuned architecture to the Spanish language, results in an accurate answer given by a well-trained version of Falcon-7B or a similar model. Switching to the homomorphic version of the encryption (HE) of the workflow, we used our encryption module on Azure, which implements CKKS using Microsoft SEAL on Azure ML compute infrastructure. The training is tokenized with a bespoke tokenizer that has been trained to the vocabulary of the LLM; such as the tokenization can generate the following token identifiers:

Token	**ID**
¿	201
Qué	308
síntomas	714
presenta	932
la	103
insuficiencia	1103
cardíaca	1121
?	205

The identifier of every token is then coded in the form of a floating-point number to meet the CKKS standards and it is inputted into the encryption function. The results of every step of encryption is a ciphertext (which is a poly) that is curling the encrypted token and safeguarded by the CKKS scheme utilizing a public key. The encrypted instruction is made up of these ciphertexts combined:

Token	**Encrypted CKKS Output (Truncated)**
¿	Enc(0xA2F4…7E9C)
Qué	Enc(0x3BD1…98C0)
síntomas	Enc(0x1DD7…24AB)
...	...

The coded command is then sent to a homomorphic inference layer contained within a Azure Kubernetes Service (aks) node. Due to the computational overheads of existing HE schemes, the encrypted tokens are not carried out over a complete network of transformers. Rather, we use indicating/linear attention reasoning that is embodied by encrypted-matrix computations (e.g., multiplication, scalar transformations) over the ciphertexts. This limited but significant methodology helps to make a query match and answer generation conditional on encrypted key word detection. The encrypted inference process generates an

encrypted value which is the ciphertext that is the result of the process indicating the response however it is not immediately decrypted; the ciphertext remains in Azure Blob storage and is then channeled to a Confidential Compute VM that has authentic access to the private decryption key. After wasting time in decryption, the resulting vector is ultimately converted to tokens, and when this happens, it is decoded to give a plaintext answer that can be compared with the ground truth.

Sample ID	**Encrypted Response (Decrypted)**	**Plaintext Response**	**Match Score (%)**
00371	Fatiga, disnea, edemas en extremidades inferiores.	Fatiga, disnea, edemas en extremidades inferiores.	100%

In the above case, the reasoning pipeline encrypted gave an identical result as the result of a plaintext LLM. However, this was not the case in all cases. The encrypted inference also caused somewhat more complex instructions to produce incomplete or incorrect answers, especially when the query included negation, conditional and unusual terms and terms. Given the case below where the encrypted response obtained was an incomplete anticipated response:

Sample ID	**Instruction**	**Expected Response**	**Encrypted Response (Decrypted)**	**Match Score (%)**
00642	¿Qué medicamentos están contraindicados durante el embarazo	Tetraciclinas, warfarina, inhibidores de la ECA.	Tetraciclinas, inhibidores de la ECA.	78%
	?			

In this case, the model has bypassed the use of the word warfarina in making the reasoning out of the encrypted input. It turned out that the encoding of warfarina was pruned in homomorphic evaluation because noise builds up in ciphertext operations this is a noted weakness of deep CKKS pipelines. To determine performance on a larger subset of the data, we were ran encrypted inference on 100 samples and measured the encrypted-inference output against the plaintext baseline when computing the semantic similarity. The findings are summarized as below:

Metric	**Value**
Total Samples Evaluated	100
Exact Match Rate	62%
Partial Match Rate (≥70% overlap)	24%
Failure / Incoherent Output	14%
Average Latency (Encrypted Inference + Decryption)	2.3 seconds
Average Latency (Plaintext Inference)	0.9 seconds

These results reveal that the high-fidelity encrypted reasoning is still limited, but most of the clinically relevant queries can be addressed correctly or with approximate accuracy with the help of our pipeline. This reduction in performance can be mostly explained by the fact that noise exacerbation is turned on in HE computations, and linear visibility layers that can be used homomorphically have limited representational capacity. Nonetheless, the method confirms that basic clinical question answering can be achieved without the need to decrypt the input and that the Azure ecosystem, including Blob storage, AKS, Confidential Compute, and Azure Machine Learning can be

used to leverage each other to implement an effective privacy-preserving LLM application.

Results and Observations

The main aim of our research was to assess the efficiency of large language models (LLMs) to conduct reasoning tasks involving encrypted clinical queries with the help of homomorphic encryption (HE). On a representative sample of 100 questions selected based on the SMC-Instruct dataset, the pipeline showed encouraging results and, at the same time, identified several technical obstacles involved when using privacy-preserving AI systems. In general, the highest match rate among the encrypted inference workflow was 62 per cent, which meant that almost two-thirds of clinical queries were able to generate the decrypted encrypted answer that was the same as that of their plaintext equivalent. Moreover, another 24% of the answers showed a semi-semantic match stronger than 70%, in which the encrypted message obtained most of the key information but had neglected or slightly flattened some of the details. As an example, the essence symptoms or medications were properly identified, but subtle expressions or less common words were also occasionally lost. Incoherent outputs or answers that did not make sense in answering the question were only 14% show the limited capacity of the system to process more complex linguistic structures or specialized vocabulary with the current system restrictions to encryption.

Noise Level Range (Approximate)	**Number of Samples**	**Exact Match Rate (%)**	**Partial Match Rate (%)**	**Failure Rate (%)**
Low (Noise ≤ 0.01)	40	85	10	5
Medium (0.01 < Noise ≤ 0.05)	45	60	30	10
High (Noise > 0.05)	15	20	33	47

The type of homomorphic encryption scheme used, CKKS, and its innate approximation, which introduces a noise accumulation effect in the performance of ciphertext operations, was a critical set of conditions that impacted these outcomes. The effect of this noise on the fidelity of computations is the greater the depth of the homomorphic operations. Table 1 shows the dependency of levels of ciphertext noise and accuracy of answers. There was indeed a distinct correlation: a sample that took time to perform or perform arithmetic chains of more complexity in encrypted inference was more likely to be correlated with increased noise and reduced semantic accuracy. The latter was especially noticeable in terms of negation, multi-part responses, or less common medical terms. Besides, encryption and decryption delays caused overhead as seen in Table 2 with the mean end-to-end encrypted inference time about 2.5-fold the plaintext inference. Although this latency is fine in other contexts such as a batch process or an asynchronous workflow, it becomes a problem in real-time clinical applications. Nevertheless, the trade-off between performance and privacy is encouraging in numerous real-life scenarios where patient confidentiality is the most important.

Process Stage	**Plaintext (ms)**	**Encrypted (ms)**	**Overhead Factor**
Tokenization	10	12	1.2x
Encryption	N/A	320	N/A
Model Inference	700	1100	1.57x
Decryption	N/A	350	N/A

Total End-to-End Latency	720	1782	~2.5x

Additionally, to the accuracy and the latency, we also compared the characteristics of errors in the encrypted inference pipeline. The failure modes in the encrypted outputs are categorized as seen in Table 3. Information truncation was the most common mistake wherein the ciphertext operation depth was curtailed by noise which truncated longer replies. Another cause of error was corruption, in which some keywords are substituted or eliminated in the process of encryption or homomorphic operations. Significantly, privacy assurances that were granted by HE was not violated under any circumstances, which also established the fact that no raw plaintext information was revealed at all in the process. Table 4 describes the semantic similarity scores, which were calculated using cosine similarity across sentence embeddings, in stratified brackets of specialties across medicine. Interestingly, the specialties that had more consistent terminology (cardiology and endocrinology) requested higher similarity scores than those that used highly variable clinical language, including psychiatry, indicating that domain-specific fine-tuning of LLMs during encryption would be more beneficial.

Failure Mode	**Frequency (%)**	**Description**
Information Truncation	45	Partial answers due to ciphertext noise accumulation
Token Corruption	30	Keywords replaced or missing after encrypted processing
Semantic Drift	15	Slightly off-topic or vague responses
Complete Failure	10	Incoherent or irrelevant outputs

Specialty	**Number of Samples**	**Average Similarity Score (0-1)**
Cardiology	20	0.92
Endocrinology	15	0.89
General Medicine	30	0.85
Psychiatry	10	0.78
Oncology	25	0.83

Overall, the experiments indicate that, as of today, the current homomorphic encryption technology has the power to enable meaningful reasoning about encrypted clinical data with the help of LLM systems despite the limitations of the homomorphic encryption technology. The drawbacks of noise build-up, longer latency, and incomplete information loss are yet to be solved, but the usability by-product of privacy protection is huge and preconditions the implementation in sensitive medical settings in the future. The gap between plaintext and encrypted computation schemes will be narrowed further by improvements in the encryption schemes, better ciphertext packing, and hybrid schemes, which combines plaintext and encrypted computation. The vibrant infrastructure, which was built on the Azure, was a nice platform to test these systems on a large scale, as their data was safeguarded using the key management and confidential computing. The work is the foundation of the practical privacy preserving AI assistants that can provide protection of privacy of patients and are not interfering with clinical utility. A close statistical analysis of the evaluation measurements reveals a strong negative

relationship between the cipher text noise levels and the fidelity of the answer in that a correlation between the measure of noise and the correct answer rate in the Pearson correlation coefficient of about -0.78 is significant. The performance variance increased significantly both in the medium and high-noise environments, as the standard deviation of semantic similarity scores increased by 0.04 to 0.11 which is an indicator of the increased variability in coded inference. Latency beverages supported this trend, showing the distribution of the total encrypted processing times had right-skewness with 75 percent of the queries completed in 1.9s, the upper hundred percentile of the query taking over 2.3s, due to complex token structures. Despite these latency costs, using the encrypted pipeline had maintained a median cosine similarity score covering clinical specialties of 0.86; hence making testament to its preservation of primitive semantic reasoning under homomorphic constraints. Notably, outlier analysis showed failures were skewed towards linguistically complex tasks, especially psychiatry, suggesting that future encrypted LLMs might require an adaptive tradeoff between noise-budgets or fine-tuning in particular specialties to maintain clinical accuracy.

Conclusion and Future Work

The study shows that the utilization of homomorphic encryption as an arrangement to assist large language models to reason encrypted clinical data is practically feasible and does not significantly affect the quality of answers. The investigation revealed that the implementation of an end-to-end pipeline on the base of the Azure platform enabled the use of encrypted inputs to be processed by a simplified LLM inference layer, which generated responses that were clinically significant and relatively accurate. Even though encrypted inference demonstrated a slight accuracy drop, and added extra latency compared with plaintext processing, the results support the fact that privacy-aware AI assistants with sensitive healthcare information can be realized with modern cloud and cryptography technology.

The prospects must focus on the optimization of encryption parameters and computing efficiency to decrease noise accumulation and the latency, which will enable the enhanced and more sophisticated thinking on the encrypted side. A combination of hybrid models that consider a sensible tradeoff of plaintext and encrypted processing along with specialized fine-tuning of encrypted inputs with LLMs promises a significant performance improvement. Moreover, it will be necessary to extend to the real-time clinical settings and a multi-user environment to make a practical deployment. The continued development of hardened hardware, cryptography systems, and federally scalable clouds is projected to make privacy-protective AI in medicine and in general even more potent.

References

[1] Gilad-Bachrach, R., Dowlin, N., La Rosa, D., Lauter, K., Naehrig, M., Wernsing, J.: CryptoNets: Applying Neural Networks to Encrypted Data. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 201–210 (2016)

[2] Hesamifard, E., Takabi, H., Ghasemi, M.: CryptoDL: Deep Neural Networks over Encrypted Data. *arXiv preprint* arXiv:1711.05189 (2017)

[3] Juvekar, C., Vaikuntanathan, V., Chandrakasan, A.: GAZELLE: A Low-Latency Framework for Secure Neural Network Inference. *USENIX Security Symposium*, pp. 1651–1669 (2018)

[4] Boemer, F., Costache, A., Cammarota, R., Wierzynski, C.: nGraph-HE2: A High-Throughput Framework for Neural Network

Inference on Encrypted Data. *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC)*, pp. 45–56 (2019)

[5] Chen, T., Bao, H., Huang, S., Dong, L., Jiao, B., Jiang, D., Zhou, H., Li, J., Wei, F.: THE-X: Privacy-Preserving Transformer Inference with Homomorphic Encryption. *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3510–3520 (2022)

[6] Stoian, A., Frery, J., Bredehoft, R., Montero, L., Kherfallah, C., Chevallier-Mames, B.: Deep Neural Networks for Encrypted Inference with TFHE. *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pp. 130–146 (2023)

[7] Legiest, W., Turan, F., Van Beirendonck, M., D'Anvers, J.-P., Verbauwhede, I.: Neural Network Quantisation for Faster Homomorphic Encryption. *IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 1–6 (2023)

[8] Moon, J., Yoo, D., Jiang, X., Kim, M.: THOR: Secure Transformer Inference with Homomorphic Encryption. *Cryptology ePrint Archive*, Report 2024/1881 (2024)

[9] Salman, R.H., Alomari, E.S.: Survey: Homomorphic Encryption-Based Deep Learning that Preserves Privacy. *International Academic Journal of Science and Engineering* **10**(2), 153–163 (2023)

[10] Sayyad, S., Kulkarni, D., Shikalgar, A., Mulla, T.A.: An Exhaustive Survey on Privacy-Preserving Machine Learning Using Homomorphic Encryption and Secure Multiparty Computation Techniques. *Journal of Computational Analysis and Applications* **33**(5), 636–648 (2024)

[11] Chabanne, H., de Wargny, A., Milgram, J., Morel, C., Prouff, E.: Privacy-Preserving Classification on Deep Neural Networks. *Cryptology ePrint Archive*, Report 2017/035 (2017)

[12] Maulyanda, M., Deviani, R., Afdhaluzzikri, A.: Enhancing Medical Data Privacy: Neural Network Inference with Fully Homomorphic Encryption. *Indonesian Journal of Electrical Engineering and Computer Science* **33**(1), 421–429 (2024)

[13] Muthu Selva Annamalai, M., Jin, C., Aung, K.M.M.: Communication-Efficient Secure Federated Statistical Tests from Multiparty Homomorphic Encryption. *Applied Sciences* **12**(22), 11462 (2022)

[14] Xu, S., Wang, L., Zhang, J., Zhang, Z., Chen, S.: Efficient Privacy-Preserving Machine Learning in Healthcare with Homomorphic Encryption. *IEEE Access* **10**, 117215–117226 (2022)

[15] Li, X., Zhang, W., Chen, Y., Wang, Y.: Privacy-Preserving Transformer Inference via Mixed Cryptographic Protocols. *Proceedings of the 32nd USENIX Security Symposium*, pp. 2695–2712 (2023)

[16] Chakraborty, A., Shaw, A.K., Samanta, S.: On a Reference Architecture to Build Deep-Q Learning-Based Intelligent IoT Edge Solutions. In: *Convergence of Deep Learning in Cyber-IoT Systems and Security*, pp. 123–146. Springer, Singapore (2022)

[17] Chakraborty, A., Shaw, A.K., Samanta, S., Kumar, A.: Exploring Genetic Algorithm to Optimize Hyper-Parameter for Training of Artificial Neural Network. *AIP Conference Proceedings* **2878**(1), 020015 (2023)

[18] Keshri, R.K.: Using Dilated CNN and MLOps for PII Detection in Pharmaceutical Reporting. *International Journal of Advanced Computer Science and Applications (IJACSA)* **15**(3), 113–120 (2024)

[19] Mohapatra, D., Chakraborty, A., Shaw, A.K.: Exploring Novel Techniques to Detect Aberration from Metal Surfaces in Automobile Industries. In: *Proceedings of the International Conference on Communication, Circuits, and Systems (IC3S 2020)*, pp. 495–504. Springer, Singapore (2021)

[20] Shaw, A.K., Chakraborty, A., Mohapatra, D., Samanta, S.: Scalable IoT Solution Using Cloud Services – An Automobile Industry Use Case. In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, pp. 485–490. IEEE, USA (2020)