Homomorphic Encryption: What Is It and How Can It Help Secure Healthcare Systems

Carlos Eugenio Lopes Pires Xavier Torres
Department of Computer Science
University of Colorado, Colorado Springs
Colorado Springs, CO, USA
carlos.torres@uccs.edu

***Abstract*—An introductory discussion about homomorphic encryption, what it is, how it works, and how it can help secure healthcare systems and protect patient data privacy.**

***Index Terms*—homomorphic encryption, healthcare systems, patient data privacy**

I. Introduction

In this paper, we discuss a relatively novel cryptographic scheme called *homomorphic encryption* and its application in securing healthcare systems. We define what it is, how it has been used, and how it can help secure healthcare systems and protect patient data privacy. We present its formal security definition, threat model, main applications, and limitations. In addition, a use case is studied on how homomorphic encryption is aiding healthcare systems, the current and future states, and a working demo is implemented and presented to illustrate this mechanism.

Homomorphic encryption (HE) [1] is a form of encryption which allows specific types of computations to be carried out on ciphertexts and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintexts. It [2] provides the ways for securely transmitting and storing confidential information across computer networks and in a computer system, which is a desirable feature in modern communication system architectures. Homomorphic encryption can be divided in three main types: partially, somewhat, and fully homomorphic encryption. This division is according to the types and frequency of mathematical operations that can be performed in the ciphertext. We discuss more in depth the definition, characteristics, and applications of homomorphic encryption in Section III.

Our main contributions with this work are: introduce homomorphic encryption to the reader, along with its main applications, and finally present its relation with healthcare system, providing a solution to use this kind of encryption to send, receive, process, and store patient data securely and in a privacy-preserving way.

This paper is the result of the class project of the course CS 5920 - Intro to Applied Cryptography for the PhD in Computer Science program at UCCS, Spring 2023. "I have neither given nor received unauthorized assistance on this work." Signature: Carlos Eugenio Lopes Pires Xavier Torres

II. Problem Statement

In the healthcare industry, patient health records are known as electronic health records (EHR), a digital record of patient's medical history usually kept by hospital or other healthcare providers [3]. It may include clinical data, patient demographics, medical history, and medical reports. These records are subject to data security protocols that must ensure: confidentiality (only authorized people can have access to the information), integrity (information should be correct and unauthorized people should not be able to modify), and availability (information should accessible and available at any time but only by authorized personnel).

This work focuses on patient clinical data and demographics (e.g. blood test exams results, age, sex, history of illnesses etc) that is transmitted over a communication channel to a healthcare system (hosted on an internet server) that performs a computation on the data and returns the result back to the patient. The data should be transmitted encrypted from the patient computer to the online healthcare system, then it must perform the computation on the ciphertext, never decrypting it to have access to the plaintext, and return the result in ciphertext, to be finally decrypted and showed to the patient on their end.

This application can be, for example, a machine learning model, operated by the healthcare system, that requires the patient clinical data input to process and return some useful result back to the patient. This model can be, for example, a statistical prediction if the patient can develop a specific kind of cancer, based on the clinical data informed, with a certain level of confidence.

This work does not focus on the data storage, or the electronic health records management once it is on the healthcare system. But only on the application described above.

The problem is securing the patient's clinical data during the communication between the patient's computer and the online healthcare system, and preserving its privacy during the computation process, so the healthcare provider can never read the plaintext of the patient's clinical data. Traditional encryption schemes need to decrypt the data to its original plaintext before allowing any computation to be performed on it. Homomorphic encryption can protect sensitive information by allowing data to be processed in the ciphertext such that

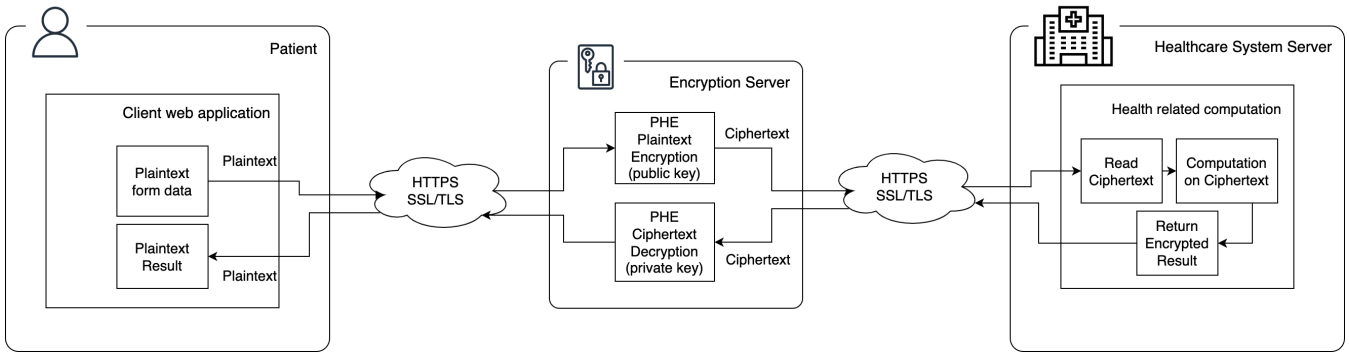

Fig. 1. System model.

only encrypted data is accessible to service providers. This keeps the data private all the time, even during the returning of results. And that is why, to achieve this level of security and privacy, we will use homomorphic encryption on the patient's data and the computation during the whole process.

A. System Model

Figure 1 illustrates the system model we are defining and using on this work. The three main actors are the patient, the ecryption server, and the healthcare system server.

- **Patient:** He/she initiates the process by providing their clinical data in plaintext on the client web application form and sending, via a secure encrypted channel (HTTPS with SSL/TLS) to the Encryption Server.
- **Encryption Server:** Serves as an intermediary between the patient plaintext data and the Healthcare System Server. It will receive the plaintext data from the patient, encrypt it with a public key using Partially Homomorphic Encryption (PHE), using Elgamal scheme [4], that possesses the following homomorphic properties:
  1) Encrypted numbers can be multiplied together.
  2) Encrypted numbers can be divided one by another.

  It also performs the decryption of the ciphertext returned by the Healthcare System Server using the private key.
- **Healthcare System Server:** Receives the ciphertext via a secure channel (HTTPS with SSL/TLS), reads it, and performs the computations on the ciphertext. These computations should follow the scheme used during the encryption process. So, encrypted numbers it will be able to be multiplied together. Then it will return the result obtained by the computation to the encryption server, in ciphertext, to be decrypted and returned to the client application, and presented as plaintext to the patient.

B. Threat Model

The application we are presenting, described on the system model above, requires an assessment of possible threats and attacks that can be performed by an adversary to compromise the integrity, correctness, and privacy of the data during the process of the system. Below we show the possible attacks that can be performed against the system and after that some counter-measures to help avoid them.

- Communication channel attack (man-in-the-middle): An attacker gets to intercept the data been sent via the communication channel and read the plaintext.
- Chosen-plaintext-attack: An attacker can send adaptive chosen plaintexts and receive ciphertexts. In the system's case, the attacker would get access to the Encryption Server and be able to send *patient* data and receive encrypted *results*.
- Chosen-ciphertext-attack: An attacker can send adaptive chosen ciphertexts and receive plaintexts. In the system's case, the attacker would get access to the Healthcare System Server and be able to send *ciphertext* data and receive encrypted *results*.

Some counter-measures to avoid the attacks and are being applied to the system are:

- The communication channel must be secure and encrypted using the HTTPS protocol with SSL (Secure Sockets Layer) or TLS (Transport Layer Security). TLS is preferable because has enhanced security. So an attacker would not be able to read plaintext data on the communication channel.
- The Partially Homomorphic Encryption (PHE) scheme uses a public-private key encryption scheme which increases the level of security in chosen-plaintext and chosen-ciphertext attacks.

III. Homomorphic Encryption

A. Definition

The term *homomorphic* (from the Greek *homos morphe* meaning *same form*) is used in mathematics to denote a transformation of one set into another that preserves in the second set the relations between elements of the first. This behavior is illustrated in Figure 2.

In computer science, *homomorphic encryption* is a type of encryption, encoding of plaintext to ciphertext, that allows certain types of mathematical computations to be performed on the ciphertext, in a way that preserves the result encrypted, and when decrypted, corresponds to the same operation if it was performed in plaintext. In other words, homomorphic

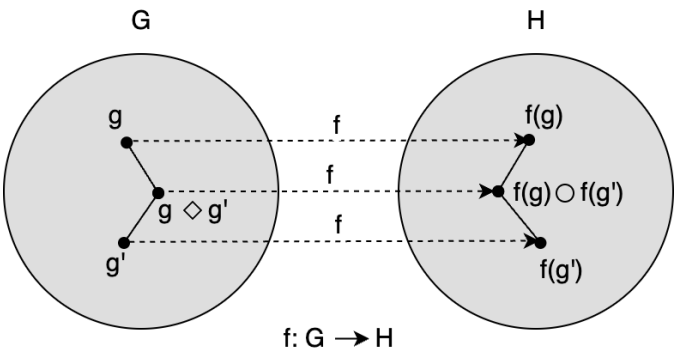

Fig. 2. Homomorphism in sets. Adapted from Yi et al. (2014) [1].

encryption allows computations in encrypted data preserving the privacy in the process [3].

Homomorphic encryption is considered a public-key type of encryption, like RSA-based, but not just limited to it, being able to be lattice-based or elliptic curve cryptography. Besides the public-key scheme, with the properties of correctness and confidentiality being satisfied, it also must have a homomorphic property, meaning that the algebraic structure of the plaintext data is preserved under the operation on the ciphertext.

B. Classification

Homomorphic encryption is classified in three main types, according to the types and frequency of mathematical operations that can be performed in the ciphertext. Figure 3 shows the classification and the best known schemes.

- Partially Homomorphic Encryption (PHE). Allows one operation to be performed on the ciphertext unlimited times.
- Somewhat Homomorphic Encryption (SWHE). Supports an arbitrary number of operations, but with each operation, a noise is generated and after a limit the underlying encrypted value is lost [3].
- Fully Homomorphic Encryption (FHE). Allows any number of operations to be performed on the ciphertext unlimited times.

C. Schemes

There are several schemes known for the different types of homomorphic encryption. Figure 3 shows the most known ones. An encryption scheme is an approach or technique developed to make HE support one more operations, and with a specific number of times (PHE), or unlimited (FHE). In this work we want to focus on a few that are most used.

- Lattice-based scheme for FHE [5]. It is an advanced cryptosystem that is safe against quantum computers based attacks.
- Elgamal [4] method for PHE. It is a scheme that allows multiplication and division HE. As this scheme is being used in our prototype system, it will be more in depth described and have its mathematical properties explained below.
- Paillier [6] approach for PHE. It is a method for additive and multiplicative (with a constant) HE.

D. Elgamal cryptosystem

The Elgamal cryptosystem [4] is a public-private key PHE scheme that allows multiplication and division of ciphertexts. It was proposed by Taher Elgamal in 1985 and is based on the discrete logarithm problem.

Below we describe the system in terms of how it computes the public-private keys, the encryption, and decryption.

- Public key generation: The public key consists of a prime number p, a generator g of the multiplicative group of integers modulo p, and an element $Y = g^x \mod p$, where x is a randomly chosen secret integer. The private key is the secret integer x.
- Encryption a message m: the sender chooses a random integer k and computes the ciphertext $(c1, c2)$, where $c1 = g^k \mod p$ and $c2 = Y^k * m \mod p$. The ciphertext $(c1, c2)$ can be sent securely to the receiver.
- Decryption of ciphertext $(c1, c2)$: the receiver uses their private key x to compute $c1^x \mod p$, which equals $Y^k \mod p$. The receiver can then multiply $c2$ by the inverse of this value modulo p, which gives $m = c2 * (c1^x)^{-}1 \mod p$.

E. Applications

- E-Health: Healthcare systems operate in an environment where sensitive information must be protected from leaks, yet available for everyday processing. The chance that attackers are out to steal data is higher than ever, so it is important to secure systems properly. FHE can help address the balance between risk and information usefulness. Billing and report generation are two applications that can benefit from this.
- E-cash: Imagine paying for an item online without the vendor needing to know your credit card number, and hence your identity. This is thanks to a homomorphic property called self-blinding, the ability to change one ciphertext into another without changing the content of its decryption.
- E-voting: Consider a simple binary (”for” or ”against”) vote. Let m voters cast a vote of either 1 (for) or 0 (against). Each voter encrypts their choice before casting their vote. The election official takes the product of the m encrypted votes and then decrypts the result and obtains the value n, which is the sum of all the votes. The election official then knows that n people voted for and $m - n$ people voted against. The role of the random r ensures that two equivalent votes will encrypt to the same value only with negligible likelihood, hence ensuring voter privacy.
- Ad privacy: While ads are often unwanted, they can be useful when tailored to the needs of the user. Many users are concerned about the privacy of their data. Jeckmans

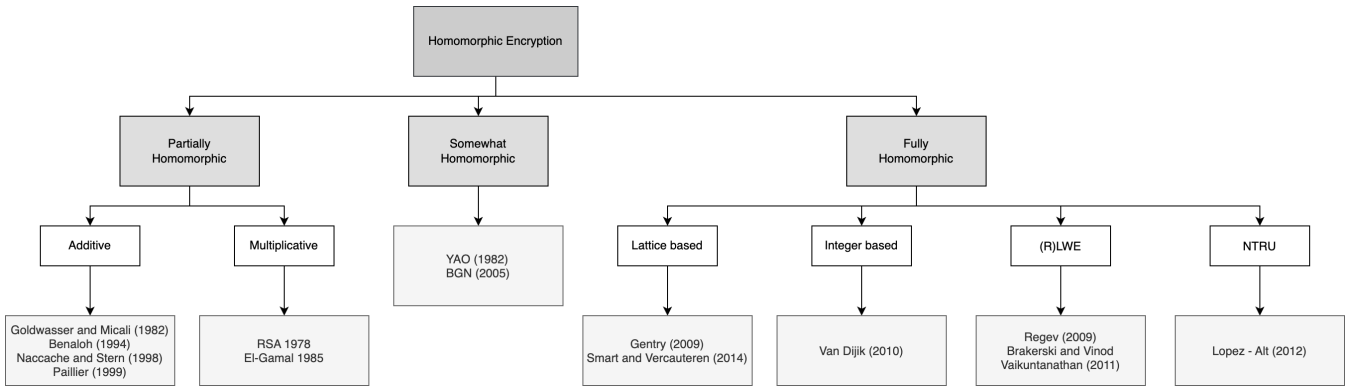

Fig. 3. Homomorphic encryption classification. Adapted from Munjal and Bhatia (2022) [3].

et al. [7] sketch a scenario in which a user on a social platform wants product recommendations. The proposed system applies FHE so that friend recommendations can be completely anonymous.

F. Libraries

- PythonPaillier [8]: Implements Paillier (1999) additive PHE scheme in Python.
- Elgamal-Python [9]: Implementation of Elgamal multiplicative PHE scheme in Python. This is what we used in our prototype system.
- OpenFHE [10]: Open source FHE implementation in C++, community maintained.
- TFHE [11]: Fast Fully Homomorphic Encryption for C/C++. (not production ready).
- Intel Paillier cryptosystem [12].
- Microsoft SEAL [13]: Microsoft SEAL is an easy-to-use open-source homomorphic encryption library developed by the Cryptography and Privacy Research Group at Microsoft. Microsoft SEAL is written in modern standard C++ and is easy to compile and run in many different environments.
- IBM HELib [14]: An open-source software library that implements homomorphic encryption. It supports the BGV scheme with bootstrapping and the Approximate Number CKKS scheme. HElib also includes optimizations for efficient homomorphic evaluation, focusing on effective use of ciphertext packing techniques and on the Gentry-Halevi-Smart optimizations.

G. Advantages and Limitations

Below we show a list of advantages and limitations of HE in its current state.

Advantages

- Computations in ciphertexts.
- Secure and efficient cloud use.
- Regulatory compliance.
- Data sharing.
- Post quantum cryptography (Lattice based).

Limitations

- FHE not efficient yet.
- Need a lot of computing power.
- Lack of maturity.
- No support for multiple users.
- High data noise.
- PHE can only use addition or multiplication, not both.

H. Future

- Homomorphic encryption will empower machine learning.
- Post quantum cryptography with Lattice based and FHE.
- More practical when computer power gets cheaper.
- More libraries need to appear to make performance better.

IV. Solution Design and Prototype Implementation

By understanding and applying the two main fundamental aspects presented on this work: homomorphic encryption and privacy-preserving data processing in healthcare systems, we are be able to design our solution. We are using the properties provided by Partially Homomorphic Encryption (PHE) to create a system that will provide a secure and privacy-preserving way to share clinical data from a patient with a healthcare system server. Computations are performed on the ciphertext and a result is returned (encrypted) to the patient, via an encryption server, that will decrypt the data before presenting to the patient.

The prototype of the system presented on this work is implemented using a technology stack that includes:

- Web client to obtain patient's data and present results, using HTML, JavaScript, and CSS.
- HTTP server to receive client data and process, using Flask, a Python HTTP server framework.
- Python implementation for the Elgamal [4] method on the Encryption Server and Healthcare System Server for the PHE encryption, decryption, and computations called.

Figure 4 show the prototype user interface. It serves as a proof-on-concept for the solution. It performs a few simple algebraic operations on the peaces of data that arrives encrypted and return the result encrypted with the operations applied.

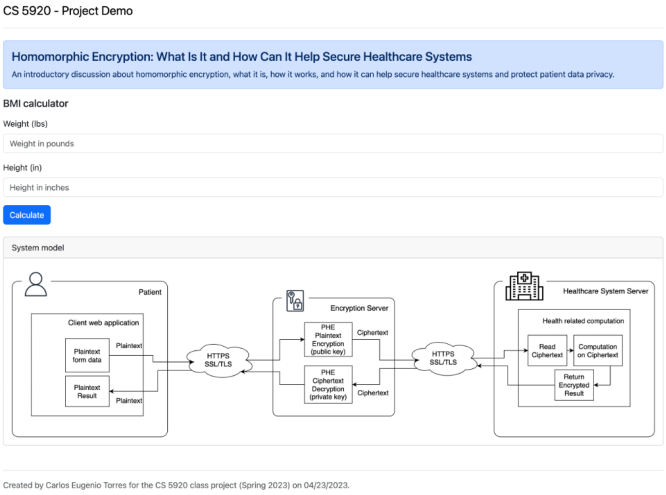

Fig. 4. Prototype user interface.

The proof-of-concept system is a simple computation of the BMI (body mass index) of a patient. It illustrates the use of patient information, like weight and height, to perform a computation on a healthcare system server using HE, so they never know the patient information. The source code is available on GitHub[1]. The formula used to calculate the BMI on the healthcare system server is the following:

$$\text{BMI} = 703 * \frac{\text{weight (lbs)}}{\text{height}^2\text{(in)}}$$

V. Related Work

Munjal and Bhatia (2022) [3] made a review of homomorphic cryptosystem contributions in healthcare and how it can protect sensitive information by allowing data to be processed in an encrypted form such that only encrypted data is accessible to service providers.

Another approach on the field is presented by Sinha et al. (2020) [15] where they proposed a Fully Homomorphic Encryption based Privacy-Preserving Data Acquisition and Computation for Contact Tracing. The authors created a secure system for responding in a timely and effective manner to a crisis scenario for the COVID-19 pandemic where it deals with sensitive health data, like infection expose. Their solution protects patient's privacy using FHE, performing computations on the ciphertext. The use of such a data encryption scheme to store and transmit sensitive healthcare data over a network can not only allay the fear of compromising sensitive information but also ensure HIPAA-compliance.

Ghadamyari and Samet (2019) [16] proposed a novel privacy-preserving method to perform statistical analysis on health data in a distributed blockchain network. They aim to increase the security level of the data and, at the same time, to minimize the amount of unnecessary information being exposed to third-parties by using Paillier Homomorphic Encryption and permissioned blockchain. On this work, we are also using the Elgamal [4] method with PHE.

[1]https://github.com/cetorres/cs5920-project-demo

VI. Conclusion

In this work, we discussed *homomorphic encryption*, a cryptographic scheme that allows performing computations on the ciphertext, and have many potential applications in areas such as secure data processing, secure cloud computing, and privacy-preserving data analysis. We focused on the application of securing healthcare systems, more specifically on secure and privacy-preserving patient's data for computing and processing ciphertext data and returning it to the patient. In addition, we presented a use case on how homomorphic encryption is aiding healthcare systems, with its system design and implementation. Finally, we showed the current state of homomorphic encryption, its limitations, and the future of this type of encryption.

References

[1] X. Yi, R. Paulet, and E. Bertino, *Homomorphic Encryption*. Cham: Springer International Publishing, 2014. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-12229-8

[2] M. Ogburn, C. Turner, and P. Dahal, "Homomorphic encryption," *Procedia Computer Science*, vol. 20, pp. 502–509, 2013, complex Adaptive Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050913011101

[3] K. Munjal and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex & Intelligent Systems*, 2022. [Online]. Available: https://doi.org/10.1007/s40747-022-00756-z

[4] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[5] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, crypto.stanford.edu/craig.

[6] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238.

[7] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang, "Privacy in recommender systems," *Social media retrieval*, pp. 263–281, 2013.

[8] C. Data61, "Python paillier library," https://github.com/data61/python-paillier, 2013.

[9] W. J. Buchanan, "Elgamal homomorphic multiplication with python," Asecuritysite.com, 2023, accessed: April 22, 2023. [Online]. Available: https://asecuritysite.com/elgamal/el_homomorphic01

[10] OpenFHE, "OpenFHE: Open source fhe implementation in c++, community maintained." March 2023. [Online]. Available: https://www.openfhe.org

[11] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption library," August 2016, https://tfhe.github.io/tfhe/.

[12] Intel, "Intel paillier cryptosystem." March 2023. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/homomorphic-encryption/iso-compliant-paillier-cryptosystem-library.html

[13] "Microsoft SEAL (release 4.1)," https://github.com/Microsoft/SEAL, Jan. 2023, microsoft Research, Redmond, WA.

[14] IBM, "Ibm helib - an open-source software library that implements homomorphic encryption," March 2023. [Online]. Available: https://github.com/homenc/HElib

[15] K. Sinha, P. Majumder, and S. K. Ghosh, "Fully homomorphic encryption based privacy-preserving data acquisition and computation for contact tracing," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2020, pp. 1–6.

[16] M. Ghadamyari and S. Samet, "Privacy-preserving statistical analysis of health data using paillier homomorphic encryption and permissioned blockchain," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 5474–5479.