Homomorphic Encryption: Impact on The Medical Field

Applied Cryptography & Trust

Edinburgh Napier University

Student Name: Aya Karim

Submitted To: Prof Bill Buchanan

Contents

1. Introduction:

"Confidential information about services users or patients should be treated confidentially and respectfully" [1] NHS UK. Electronic Health Records (EHR) of patients include, usually, sensitive information that should be transferred, stored and studied confidentially without giving access to curious parties and without being vulnerable to cyberattacks. What if the cloud service provider has the intention to use those records in an illegal way? And what if the records need to be transferred for scientific studies in order to identify statistical patterns and diagnose rare diseases or disease prediction? How are the intermediate data going to be protected from exploitation?

Conventional Encryption methods do not guarantee a secure storage or transfer of patients' data, since the records are decrypted when stored remotely, hence, processing on encrypted data is not possible and unauthorized parties could access the cleartext easily by browsing them. Cyberattacks on healthcare organizations are highly dangerous that a cyberattack in Germany distorted the hospital system and caused a patient to die [24] . Homomorphic Encryption is paving its way to gain the trust of healthcare organizations that prioritize the confidentiality of the sensitive information of their patients. The objective of this literature review is to gain an understanding on the current studies on homomorphic encryption and its critical contribution in the medical field (data storage, transfer, diagnosis, prediction and processing).

2. Literature Review:

2.1 Homomorphic Encryption:

In an era where homomorphic encryption is not invented. Imagine that your DNA, the macromolecule carrying all your genetic information has to be stored somewhere in the cloud. You have to consider the privacy very seriously before taking the decision of handing it over. But with Homomorphic encryption you don't have to worry much about it.

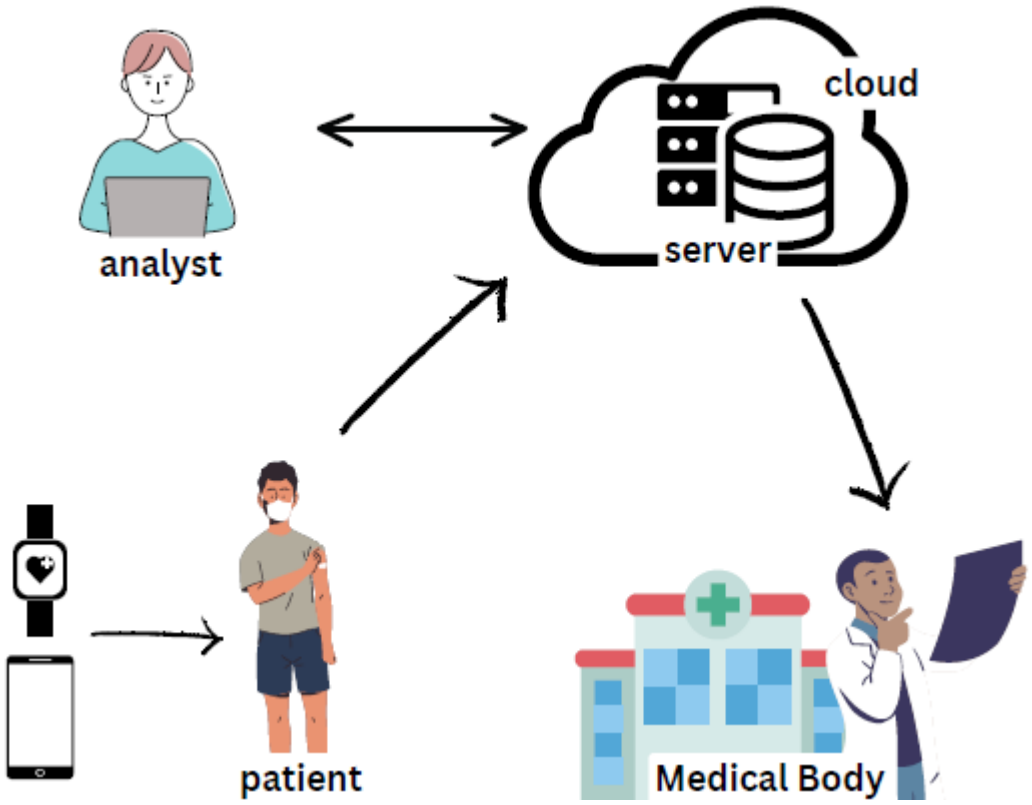
So, what is homomorphic Encryption?

Homomorphic Encryption is a cryptographic framework where computations are performed over encrypted data, thus ensuring confidentiality and prohibiting information leakage.

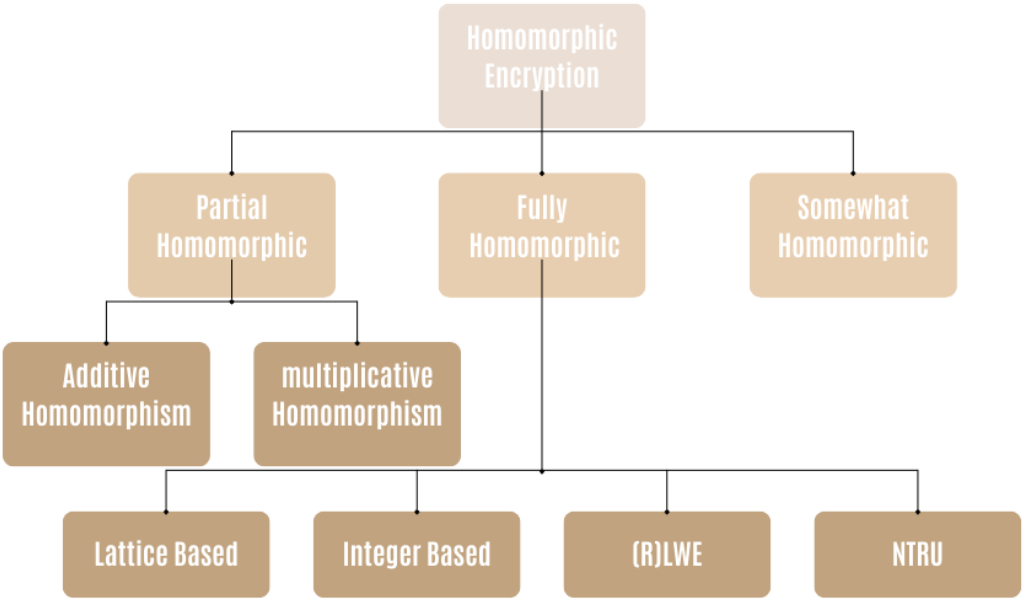
Massimo Bertaccini, in [2], defines homomorphism informally as *"operations performed on isomorphic elements"*. For example, the shadow of a curly head won't show the details of the hair, but you will still recognize its curly pattern. The same thing applies to homomorphism where there is a relational property between the results of the same computation on ciphertext and plaintext.

Consequently, a hacker who beats all the security measures will end up with encrypted data.

partially homomorphic encryption, somewhat homomorphic encryption, fully homomorphic encryption and leveled homomorphic encryption are types of homomorphic encryption which was limited in the number and types of operations it can perform on encrypted data until Craig Gentry introduced the first scheme for fully homomorphic encryption in 2009 [11].



Homomorphic Encryption has three types: (1) Partially Homomorphic Encryption permits a single kind of operation to be performed an infinite number of times (e.g. RSA, ElGamal) (2) Somewhat Homomorphic Encryption permits specific types of operations a restricted number of times. (3) Fully Homomorphic Encryption permits an unlimited number of operations (mathematical and Boolean operations) for an infinite number of times [12]. Each of the three types is classified into different categories.



Types of Homomorphic Encryption [6]

RSA - public key cryptosystem - has a homomorphic property explained below:

Let's consider:

M: message in plaintext.
E: public exponent.
N: modulus N = p*q
C: ciphertext

RSA encryption and decryption formulas:

Encryption:

$C = M^e \bmod N$

Decryption:

$M = C^d \bmod N$

Suppose two messages: $\mathbf{M_1}$ and $\mathbf{M_2}$, they will be encrypted as:

- $\mathbf{C_1 = M_1^e \bmod N}$
- $\mathbf{C_2 = M_2^e \bmod N}$

Alice wants to send the multiplication of $\mathbf{M_1}$ and $\mathbf{M_2}$ to Bob, so she performs the operation that results in $\mathbf{C_3}$:

- $\mathbf{C_3 = C_1 * C_2 = M_1^e M_2^e \bmod N}$

When Bob receives $\mathbf{C_3}$, he will decrypt it with **d** (his private key):

- $\mathbf{M_1 * M_2 = C_3^d \bmod N}$

So, Bob got the multiplication of $\mathbf{M_1}$ and $\mathbf{M_2}$ implied in $\mathbf{C_3}$, which is the operation performed by Alice. By using a specific software (e.g., Wolfarm Mathematica), Bob can get the value of $\mathbf{M_1}$ and $\mathbf{M_2}$ independently, and he can verify the operation done by Alice.

2.2. Review Method:

A total of 295 scientific papers were obtained using a database search with Keywords used for search as "homomorphic encryption" and "medical" OR "healthcare" OR "health" OR "medicine". The papers used in this literature review were published in the past 6 years, hence, Scientific articles published in the year range from 2018 to 2024 were considered. Papers that were focused on securing medical data by using methods other than homomorphic encryption, or papers that didn't study the applicability of homomorphic encryption in the healthcare industry were excluded.

2.3. Homomorphic Encryption in The Medical Field:

Before highlighting the contribution of homomorphic encryption in the field, it is important to answer a certain suspicion;

Isn't it more secure for healthcare institutions to use physical data centers for safe storage, so they do not use cloud services?

Although this is a possible solution, however, no one can guarantee that data centers won't be destroyed by different factors, such as human mistakes, natural disasters, and electrical damage. Moreover, who says that an authorized person will not try to copy the data and use illegally?

EHR contain sensitive data such as: demographics, medical history, lab tests (x-rays, CT scans, blood tests, MRIs, toxicology …), vital signs (body temperature, blood pressure, heart rate, ...) medical reports, prescriptions and treatment plans.

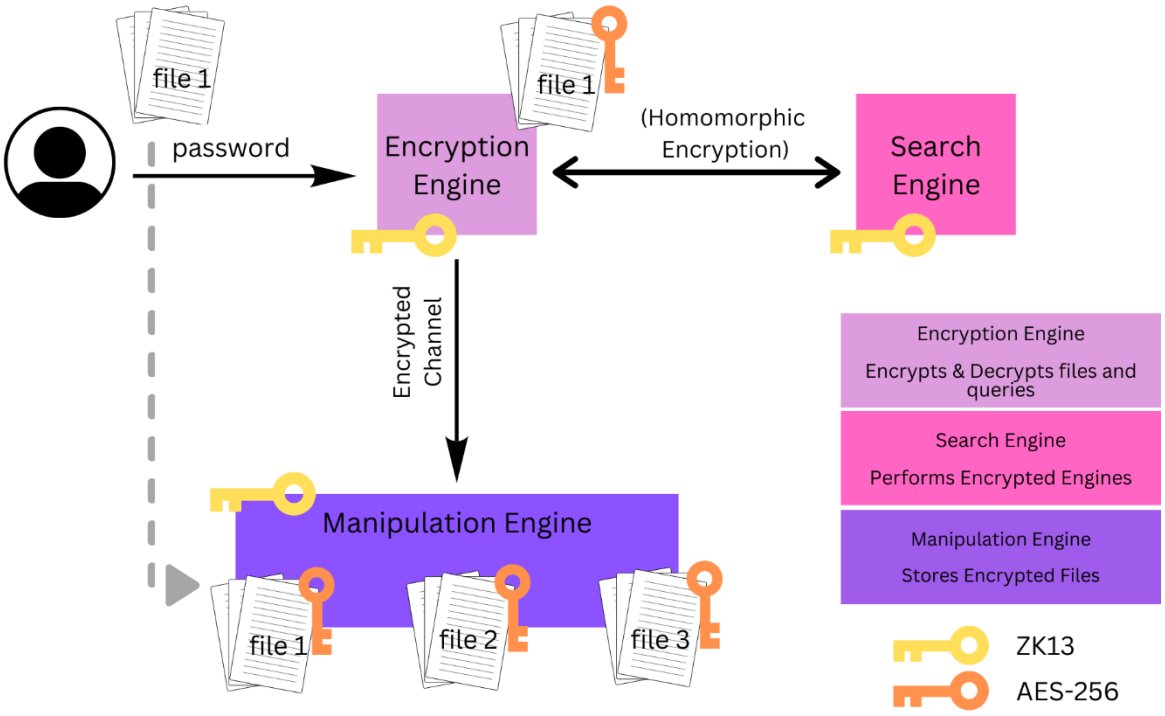
2.3.1. Partially Homomorphic Models:

In [2], Bertaccini introduces CSE, Crypto Search Engine, which is *"a search engine able to operate with data in blind, not conscious of what it is searching for"*, the challenge in this project was to keep the elapsed time per query below 2 seconds since it's the average time for a user to wait

for an answer. The operations had to be computationally light while producing fast answers. The encryption algorithm used in CSE to save data in the cloud (secure channel between the private cloud and the public cloud) is a symmetric key algorithm (AES-CBC-256) [3] between an Encryption Engine EE (private cloud) and a Manipulation Engine ME (public cloud). EE generates random unique encryption keys for every file and query, while ME stores the encrypted files and searches in the encrypted content. What is interesting is that the EEs and MEs are connected through zero-knowledge protocols; ZK13 non-interactive protocol (for further information on ZKP refer to [4]).

To search blindly, Boolean logic operators (AND, OR, NOT) could be used and combined, the engine is also case-insensitive, noting that only the one who inputs the query has the secret key, meaning that he is the only one able to decrypt the answer. In order to reach the right destination when a user inputs the keywords, tree graphs (method by David Huffman), hash functions, and Boolean logic are combined together. In order to achieve the highest efficiency, the highest security with the fastest answer should have been combined, and this is what the developer tried to work on compared to Tor network.

The model, as indicated, uses AES-CBC-256, a famous drawback of this encryption method is that CBC mode propagates the error, and when implementing such model in healthcare the concerns are higher since every operation is going to be performed over sensitive encrypted data. One of the alternatives is AES-GCM that does not propagate the error to subsequent bits, thus ensuring data integrity.



A question could be raised: who exactly holds the secret key and which permissions are assigned to each person having access rights to the HER?

In [5], a Secure Partially Homomorphic Encryption (SPHE) algorithm allows secure multiplication and division on encrypted data, while also creating a role network for different users who are permitted to access the records. SPHE includes 5 algorithms; Key Generation, Encryption, Decryption, Multiplication and Division Algorithm.

Homomorphic property of SPHE:

- Private key: sk
- Public key: pk
- Plaintexts: P_1, P_2

Corresponding ciphertexts:

- $C_1 = Enc (pk, P_1)$
- $C_2 = Enc (pk, P_2)$

First, the public and private key are generated:
$K = S^g \bmod p$
S: integer $\in \{ 0,1, \ldots, p-2\}$ while GCD (S,p)=1
g: primitive number where GCD (g,p)=1
p: large prime number

➔ pk= (g,K,p) and sk

Then, P_1 and P_2 are encrypted to C_1 and C_2 using pk:

- $C_1 = (m_1, n_1) = (g*r_1, P_1 K^{r1} \bmod p)$
- $C_2 = (m_2, n_2) = (g*r_2, P_2 K^{r2} \bmod p)$

r: random integer $\in \{2, \ldots, (p-1)/2\}$
$C_1 * C_2 = (g*(r_1 * r_2), (P_1*P_2)/ K^{r1+r2}) \pmod p = Enc (pk, P_1 * P_2)$

Finally, the evaluated ciphertext could be decrypted using sk:
$P = n * S^m \bmod p$
$Dec (sk, C_1 * C_2) = Dec (n_1 * S^{m1} \bmod p) * (n_2 * S^{m2} \bmod p) = Dec (sk, C_1) * Dec (sk, C_2)$

The proposed scheme proved that it has a multiplication homomorphic property. Noting that not all parts of calculation were demonstrated.
By following the same process, but using division instead, the scheme also proves that it has a division homomorphic property.
Concerning the roles of authorized people, they were created using Identity and Access Management (IAM) service in the Amazon Web Services (AWS). The roles are:

1- admin user
2- evaluator
3- end-user

the admin user possesses pk and sk and has full access to the data, pk is shared with the evaluator to perform the computations, while sk is shared with the end user to decrypt the result. Each one of the users is assigned a number of permissions (list, tagged, read, write) depending on their role.

As a result, the message size was directly proportional to the decryption time, encryption time, multiplication and division operations. Compared to other schemes (message size = 128 bits), the time taken by SPHE for encryption, decryption, multiplication, division, and storage of ciphertext, was significantly less than the time taken by using Paillier, ElGamal, or Benaloh. However, the time taken by SPHE (476 ms) to generate keys was slightly more than that of Paillier (332 ms). Thus, SPHE showed a higher efficiency.

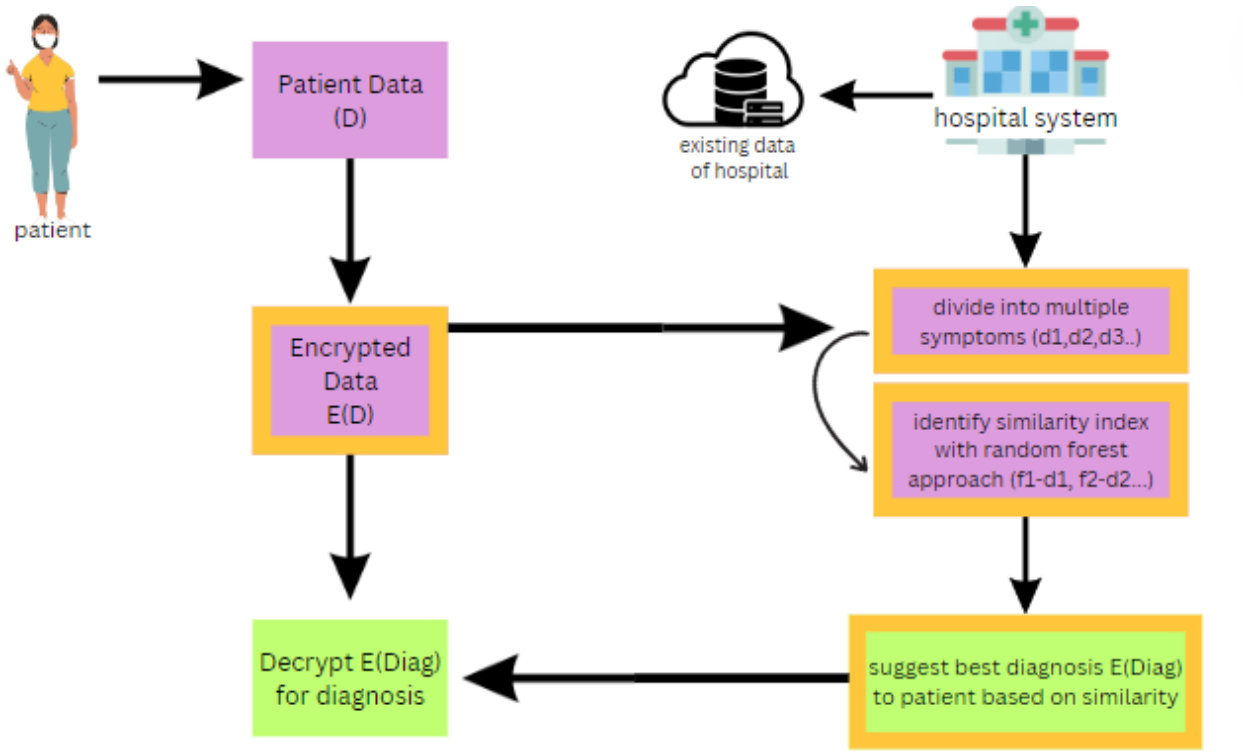
Although this scheme is efficient, it is still limited concerning the number of allowed operations without the ability of computing two operations at the same time. Moreover, no ciphertext refreshing techniques such as bootstrapping or squashing were proposed, so every time an operation is performed on the encrypted data, a noise is produced, and the multiplication will double the noise.

In [7], a multiple key-based homomorphic encryption (MKHE) using sailfish optimization algorithm is proposed. The technique suggests generating multiple keys that will encrypt/decrypt data in a number of rounds, while using Hybrid Cryptographic Optimizer (HCO) process with optimum fitness value to select optimal keys which increases security levels. The scheme was combined with deep neural networks for data classification in order to perform diagnosis, it was tested on activity recognition and UCD sleep stage datasets. However; the diagnosis was performed on decrypted data on the receiver's side. Although MKHE is not a fully homomorphic encryption scheme, it could be an efficient solution that balances between computation speed, accuracy, and security as it showed acceptable speed rates and high security levels after transmission.

Another scheme of PHE that is proposed is symmetric homomorphic encryption (SHE) method alongside with Boneh-Lynn-Shacham(BLS) short signature algorithm in [8], the method consists of three phases; key generation (the user selects security parameters before execution), encryption, and decryption and has multiplication and addition homomorphic properties. The usage of SHE over asymmetric homomorphic encryption is more efficient in terms of speed on large data like EHR. the additional security measure in this method is the signature key that is going to be validated by the service provider SP or by the user which are the two parties involved in the process. The model protects confidentiality when encrypted information are sent by the user to the SP and when the SP compares the user's data with the decision tree, and sends the answer to the user, the model also adds random number when performing computations on encrypted data. After implementation, it showed a faster performance especially in multiplication, when compared to a system that uses the Paillier algorithm.

Hence, both models in [5] and [8] outperformed the Paillier algorithm.

Another model in [9] suggests a secure method for data storage; Secure Multiparty Computation (SMC) using Paillier approach, a multiparty homomorphic encryption in an extension of partially homomorphic encryption that allows multiple parties to encrypt their data confidentially and decryption keys are distributed among them through secure channels [20]. The patient uploads new data which is then encrypted and sent to the hospital, the hospital already has a background knowledge about the patient's medical history, the symptoms will be compared based on the similarity index through Random Forest Algorithm, and the disease prediction will be sent to the doctor.



The usage of Paillier algorithm will slow down the computation time as demonstrated previously, other approaches may be considered for a faster performance.

After highlighting the need for an encryption method that does not interfere with international data protection laws and allows data transmission to private organizations for study purposes with total identity anonymity, and exploring GDPR (General Data Protection Regulation) obligations on data controllers, [21] encourages the usage of Multiple Homomorphic Encryption since it maintains confidentiality with secure key management between involved parties, compared to traditional encryption methods which, in the best case, use pseudonymization where identifiers are removed from data (personal data is still exposed), but this practice does not meet GDPR privacy standards.

While still vulnerable to inference attacks, federated approach of decentralized data sharing model overmasters the centralized model since computational assets are distributed and confidentiality is preserved for highly sensitive medical data while maintaining high accuracy

levels [22]. [23] combined federated analytics with multiparty homomorphic encryption (FAMHE). The study performed on HIV datasets requested multiple parties to locally compute their data in cleartext and then aggregate encrypted intermediate results. The proposed model executed two biomedical studies; Kaplan–Meier survival analysis and genome-wide association studies (GWAS) and the results were compared to the results of centralized approach. The model showed an exceptional efficiency for complex tasks which was not explored in previous research; the outputs were highly accurate and the computation time was independent from the ciphertext size, while confidentially transferring data among different data providers.

2.3.2. Fully Homomorphic Models:

the concept of FHE allowed a wider range of operations to be applied on encrypted data, such as equality operations (>, <, and =) that compare individual bits [14] .

The authors in [10] proposed a cloud based fully homomorphic using BGV scheme [13] model that focuses on long-term patient monitoring. the paper focused on calculating the average heart rate based on files extracted from the Electrical Cardiogram (ECG) that is being recorded 24 hours.

The model includes three phases:

1- Real-time medical data acquisition devices:
   Devices attached to the patient's body and transmits data privately using AES encryption.

2- Cloud-based computation and storage:
   Data that must be computed in real-time is converted from AES to FHE, the conversion is performed offline.

3- GUI end nodes:
   A GUI device that performs FHE decryption at the doctor's end.

Real-time data was gathered in the cloud, while the computation (multiplication and division) was performed in the GUI end node. the cost in this study was highly considered since the computation wasn't performed on data in FHE encrypted format in the cloud, they were performed in the GUI node instead.

The model was implemented using two schemes: **Gentry's Original FHE scheme** [11] and **BGV scheme** [13], then the results were compared. Gentry's Original FHE scheme showed a higher computation speed than the BGV scheme; however, a dual-socket Xeon server was used to speed up the computation, when the speed increases, the efficiency decreases.

In [15], the biggest concern was 1) to slow down the noise growth while speeding up the multiplication 2) to accelerate computational operations on encrypted medical data 3) to protect the model from Subfield Lattice Attacks. The authors presented an optimized NTRU-based (public

key cryptosystem) implementation of the GSW homomorphic encryption scheme [16], the parameters settings of the proposed work (base-scheme, polynomial dimension, modulus bit width, plaintext modulus, key standard deviation, effective security level, vulnerability factor, ciphertext size, key size, evaluation key) resulted in a better performance when compared to works in [17], [18], and [19] when implemented in CPU and GPU. The model succeeded in achieving a slower noise growth and a speedup of 6085 × in ciphertext multiplication when implemented in GPU.

After the covid-19 pandemic breakthrough, the need for secure telemedicine increased. The authors in [25] the authors propose a model that matches homomorphically the encrypted medical audio files of patients encoded into phonemes. The model includes five phases: 1) key generation: public and private key are generated 2) encryption: phenomes are encrypted with the client's public key before being stored in the cloud 3) trapdoor building: the search keywords are probabilistically encrypted with the client's secret key 4) Searching: phonemes are homomorphically searched for and matched (only the size and number of files are known to the cloud) 5) decryption: the output of searching is sent securely to the client who decrypts the result with his private key. A concept closely related to trapdoor was previously demonstrated in [2]; the CSM project introducing a blind searching engine. In this study, when the security level increases, the ciphertext size increases, which increases the search time; however, in [23] computation time was independent from the ciphertext size since a federated analytics approach was used, but it was combined with multiparty homomorphic encryption and not with fully homomorphic encryption. Overall, the results of the research met the objectives where data was stored securely and searched for in a confidential manner.

Another research impacted by covid-19 is [26] that proposes a contact tracing technique called PRivacy Oriented Technique for Epidemic Contact Tracing (PROTECT) without revealing the location of infected people to other users or to the quarantine authorities. The proposed model uses the Brakerski/Fan-Vercauteren fully homomorphic encryption scheme (BFV) and a new proximity computation method (inspired by the technique used in the Pierre protocol [27]) that identifies if two locations are in the same grid or in adjacent grids without giving information on the geographical location itself, noting that a randomized version of the ciphertext of the geolocation is sent by the user to the authorities to prevent the extraction of any personal information. The chosen homomorphic scheme has five phases: 1) setting up security parameters 2) encrypting data with secret key 3) decrypting results with the secret key 4) returning the addition of given ciphertexts 5) returning the multiplication of two ciphertexts. After building the user app and the web service for quarantine authorities, the study was efficient even when a small number of people have installed it compared to the efficiency of same concept apps such as Apple and Google. As stated by the authors, the protocol's limitations are the accuracy of the smartphone's GPS, the comparison of every patient in the app while only users in the same region should be compared, and the slowness of computation on smartphones. However, the confidentiality of patient's data was well preserved without having to expose personal data to any party compared to other proximity algorithms such as Euclidean space.

Authors in [28] claim that Artificial Pancreas Devices (APD), despite their great importance in monitoring and regulating blood sugar levels of diabetic people, they are unsecure and vulnerable to eavesdropping attacks causing leakage of sensitive data. Hence, they tried to introduce a method that uses fully homomorphic encryption which is immune to existing attacks and post-quantum attacks as well. The data collected by the blood sugar level sensor is encrypted with the leveled Cheong–Kim–Kim–Song (CKKS) encryption scheme [29], and then, the encrypted data, is computed by the Proportional–Integral–Derivative (PID) controller which suggests the right insulin dosage, the result (insulin dosage) is sent to the insulin which decrypts it and pumps the suggested dosage. The proposed model resulted in approximately the same accuracy level when compared to the performance of a normal APD. The study shows high privacy levels and accuracy; however, considering more vital signs that play a role in the variation of blood sugar levels might be considered in further studies, as well as testing and improving the security of the model.

3. Implementation:

In order to test the efficiency of HE, a comparison has been made between the computation performed with HE and without HE.

A medical dataset “Continuous Cuffless Monitoring of Arterial Blood Pressure via Graphene Bioimpedance Tattoos” [30] was downloaded from physionet.org. the dataset contains different vital signs for 7 subjects, however; for the testing, only the files containing the blood pressure (measured in different times) were used and the blood pressure average was computed which resulted in 7 lines for each file.

The codes are found in a the attached files.

1- Average of Blood Pressure calculated with HE using the *lightphe* library (partially homomorphic) and “Exponential-ElGamal” algorithm [32]:

```
┌──(aya㉿kali)-[~/Desktop/output]
└─$ cat withHE.txt
Decrypted average for file data_trial03_finapresBP6.csv: 107.92
Decrypted average for file data_trial03_finapresBP1.csv: 93.77
Decrypted average for file data_trial03_finapresBP5.csv: 98.43
Decrypted average for file data_trial03_finapresBP7.csv: 116.09
Decrypted average for file data_trial03_finapresBP3.csv: 113.23
Decrypted average for file data_trial03_finapresBP4.csv: 127.36
Decrypted average for file data_trial03_finapresBP2.csv: 120.45
```

2- Average of Blood Pressure calculated without HE:

```
┌──(aya㉿kali)-[~/Desktop/output]
└─$ cat withoutHE.txt
Average for file data_trial03_finapresBP6.csv: 108.41722760151309
Average for file data_trial03_finapresBP1.csv: 94.2676698676123
Average for file data_trial03_finapresBP5.csv: 98.936187388888891
Average for file data_trial03_finapresBP7.csv: 116.58587662765957
Average for file data_trial03_finapresBP3.csv: 113.72941480288434
Average for file data_trial03_finapresBP4.csv: 127.86140992611857
Average for file data_trial03_finapresBP2.csv: 120.94522725945626
```

4. Evaluation:

As the PHE does not allow every operation (or multiple types of operations at the same time), the addition was performed while the data was encrypted, but the division was performed after decryption since "Exponential-ElGamal" does not support division.

The computation without HE was performed immediately; however, computation with HE took approximately 17 minutes. Both of them showed very similar results with insignificant differences.

Multiple algorithms were used to compare the efficiency (using the same library):

ALGORITHM	EXECUTION TIME
EXPONENTIAL-ELGAMAL	17 minutes
PAILLIER	55 minutes
DAMGARD-JURIK	2 hours
BENALOH	1 minute 13 seconds
NACCACHE-STERN	3 seconds
OKAMOTO-UCHIYAMA	45 minutes
WITHOUT HE	1 second

The average of blood pressure calculated by using different algorithms for 7 subjects:

Subject / Algorithm	1	2	3	4	5	6	7
Exponential-ElGamal	93.77	120.45	113.23	127.36	98.43	107.92	116.09
Paillier	93.77	120.45	113.23	127.36	98.43	107.92	116.09
Damgard-Jurik	93.77	120.45	113.23	127.36	98.43	107.92	116.09
Benaloh	93.77	120.45	113.23	127.36	98.43	107.92	116.09
Naccache-Stern	3.25	14.84	7.62	6.67	7.92	2.31	10.48
Okamoto-Uchiyama	93.77	120.45	113.23	127.36	98.43	107.92	116.09
Without HE	94.2676	120.9452	113.7294	127.8614	98.9361	108.4172	116.5858

All the algorithms are partially homomorphic and have the following features:

1. Additively Homomorphic
2. Multiplication with a plain constant
3. Regeneration of ciphertext

All the HE algorithms produced the same results except for "Naccache-Stern", although it was the fastest one, but accuracy was sacrificed for speed, and it failed to produce results that are

pretty close to the real ones. Considering accuracy and speed, “Benaloh” was the most efficient as it took less time with high accuracy.

Classification of HE algorithms from the most efficient to the least efficient:

1. Benaloh
2. Exponential-ElGamal
3. Okamoto-Uchiyama
4. Paillier
5. Damgard-Jurik
6. Naccache-Stern

5. Conclusions and Future Work:

Homomorphic encryption, including both types PHE and FHE, are highly efficient when the right algorithms are used, but their complexity, at this level, makes them less suitable for low power devices such as medical devices that did not reach the stage where they can handle such power consuming computations and if they handle them, it would be time consuming. Hence, there is a need for less complex, less power consuming, and less time-consuming schemes without sacrificing the accuracy or the security. Also, the existing algorithms showed an increase in the ciphertext size or in the computation time, or both of them, when the security parameters were adequate.

While some authors like in [28] state that their methods are post-quantum secure, there should be in-depth study on their ability to withstand the era of post-quantum cryptography, especially concerning the “secure channels” that are being used to transfer data between different parties.

Undoubtedly, the algorithms are being studied for their computational power and security levels, but there is no clear declaration about the costs of their implementation and if there is a need to set-up costly hardware, and as for healthcare institutions, the sensitive data will require accurate high computational power and most probably with high costs.

Simple attacks on the CKKS approximate-numbers scheme were described by Li and Micciancio [28]. These attacks reveal the secret key when a limited number of decryption results—sometimes as few as one decryption—are observed. This highlights a critical vulnerability in the scheme, and practical applications should be done to investigate the weaknesses in different schemes in order to improve their security and propose alternatives that are resistant to the identified attacks.

6. References:

[1] NHS UK, A Guide to Confidentiality in Health and Social Care.

[2] Bertaccini, M. (2022). *Cryptography algorithms: a guide to algorithms in blockchain, quantum cryptography, zero-Knowledge protocols, and homomorphic encryption*. Packt Publishing.

[3] *Alessandro Passerini and Tiziana Landi*, https://www.cryptolab.cloud/

[4] Aad, I. (2023). Zero-Knowledge Proof. In: Mulder, V., Mermoud, A., Lenders, V., Tellenbach, B. (eds) Trends in Data Protection and Encryption Technologies. Springer, Cham. https://doi.org/10.1007/978-3-031-33386-6_6

[5] Boomija, M.D., Raja, S.V.K. Securing medical data by role-based user policy with partially homomorphic encryption in AWS cloud. *Soft Comput* **27**, 559–568 (2023). https://doi-org.napier.idm.oclc.org/10.1007/s00500-022-06950-y

[6] Munjal, K., Bhatia, R. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex Intell. Syst.* **9**, 3759–3786 (2023). https://doi.org/10.1007/s40747-022-00756-z

[7] Alzubi, J. A., Alzubi, O. A., Beseiso, M., Budati, A. K., & Shankar, K. (2022). Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis. *Expert Systems*, 39(4), e12879. https://doi-org.napier.idm.oclc.org/10.1111/exsy.12879

[8] S. Zhou, J. Fan, X. Du, B. Qiao and Z. Qiao, "Efficient Multi-disease Privacy-Preserving Medical Pre-Diagnosis Based on Partial Homomorphic Encryption," *2022 12th International Conference on Information Science and Technology (ICIST)*, Kaifeng, China, 2022, pp. 248-254, doi: 10.1109/ICIST55546.2022.9926857

[9] Vijaya Kumar, A., Sujith, M. S., Sai, K. T., Rajesh, G., & Yashwanth, D. J. S. (2020). Secure Multiparty computation enabled E-Healthcare system with Homomorphic encryption. *IOP Conference Series. Materials Science and Engineering*, *981*(2), 22079-. https://doi.org/10.1088/1757-899X/981/2/022079

[10] O. Kocabas, T. Soyata, J. -P. Couderc, M. Aktas, J. Xia and M. Huang, "Assessment of cloud-based health monitoring using Homomorphic Encryption," *2013 IEEE 31st International Conference on Computer Design (ICCD)*, Asheville, NC, USA, 2013, pp. 443-446, doi: 10.1109/ICCD.2013.6657078.

[11] Craig Gentry, *Fully homomorphic encryption using ideal lattices*, Symposium on the Theory of Computing (STOC), 2009, pp. 169-178.

[12] Acar, A., Aksu, H., Uluagac, A., & Conti, M. (2019). A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Computing Surveys*, *51*(4), 1–35. https://doi.org/10.1145/3214303

[13] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in ITCS, 2012, pp. 309-325

[14] J. H. Cheon, M. Kim and M. Kim, "Optimized Search-and-Compute Circuits and Their Application to Query Evaluation on Encrypted Data," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 188-199, Jan. 2016, doi: 10.1109/TIFS.2015.2483486

[15] A. Khedr and G. Gulak, "SecureMed: Secure Medical Computation Using GPU-Accelerated Homomorphic Encryption Scheme," in *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 597-606, March 2018, doi: 10.1109/JBHI.2017.2657458

[16] C. Gentry, A. Sahai and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler asymptotically-faster attribute-based" in Advances in Cryptology – CRYPTO 2013, Berlin, Germany:Springer, vol. 8042, pp. 75-92, 2013, [online] Available: dx.doi.org/10.1007/978-3-642-40041-4_5

[17] A. Khedr, G. Gulak and V. Vaikuntanathan, "SHIELD: Scalable homomorphic implementation of encrypted data-classifiers", *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2848-2858, Sep. 2016

[18] J. W. Bos, K. Lauter and M. Naehrig, "Private predictive analysis on encrypted medical data", *J. Biomed. Informat.*, vol. 50, pp. 234-243, 2014, [online] Available: http://www.ncbi.nlm.nih.gov/pubmed/24835616

[19] K. Lauter, A. Lopez-Alt and M. Naehrig, "Private computation on encrypted genomic data", 2014, [online] Available: http://research.microsoft.com/apps/pubs/default.aspx?id=219979

[20] Mouchet, Christian Vincent, 2023 "**Multiparty Homomorphic Encryption: from Theory to Practice"** https://doi.org/10.5075/epfl-thesis-8846

[21] Scheibner J, Raisaro JL, Troncoso-Pastoriza JR, Ienca M, Fellay J, Vayena E, Hubaux JP. Revolutionizing Medical Data Sharing Using Advanced Privacy-Enhancing Technologies: Technical, Legal, and Ethical Synthesis. J Med Internet Res. 2021 Feb 25;23(2):e25120. doi: 10.2196/25120. PMID: 33629963; PMCID: PMC7952236

[22] Dayan, I., Roth, H.R., Zhong, A. *et al.* Federated learning for predicting clinical outcomes in patients with COVID-19. *Nat Med* **27**, 1735–1743 (2021). https://doi.org/10.1038/s41591-021-01506-3

[23] Froelicher D, Troncoso-Pastoriza JR, Raisaro JL, Cuendet MA, Sousa JS, Cho H, Berger B, Fellay J, Hubaux JP. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. Nat Commun. 2021 Oct 11;12(1):5910. doi: 10.1038/s41467-021-25972-y. Erratum in: Nat Commun. 2021 Nov 11;12(1):6649. PMID: 34635645; PMCID: PMC8505638.

[24] Jalali M.S., Landman A., Gordon W.J. Telemedicine, privacy, and information security in the age of COVID-19. *J. Am. Med. Inform. Assoc.* 2021;**28**:671–672. doi: 10.1093/jamia/ocaa310

[25] Iqbal Y, Tahir S, Tahir H, Khan F, Saeed S, Almuhaideb AM, Syed AM. A Novel Homomorphic Approach for Preserving Privacy of Patient Data in Telemedicine. Sensors (Basel). 2022 Jun 11;22(12):4432. doi: 10.3390/s22124432. PMID: 35746213; PMCID: PMC9228489.

[26] An Y, Lee S, Jung S, Park H, Song Y, Ko T. Privacy-Oriented Technique for COVID-19 Contact Tracing (PROTECT) Using Homomorphic Encryption: Design and Development Study. J Med Internet Res. 2021 Jul 12;23(7):e26371. doi: 10.2196/26371. PMID: 33999829; PMCID: PMC8276784.

[27] Zhong G, Goldberg I, Hengartner U. Louis, Lester and Pierre: three protocols for location privacy. Privacy Enhancing Technologies. PET 2007. Lecture Notes in Computer Science, vol 4776; International Workshop on Privacy Enhancing Technologies. Springer, Berlin, Heidelberg June; 2007; Ottawa, ON. Berlin, Heidelberg: Springer; 2007. pp. 62–76.

[28] Haotian Weng, Chirath Hettiarachchi, Christopher Nolan, Hanna Suominen, Artem Lenskiy, Ensuring security of artificial pancreas device system using homomorphic encryption, Biomedical Signal Processing and Control,Volume 79, Part 1, 2023, 104044, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2022.104044

[29] Cheon J.H., Kim A., Kim M., Song Y. Homomorphic encryption for arithmetic of approximate numbers International Conference on the Theory and Application of Cryptology and Information Security, Springer (2017), pp. 409-437.

[30] Ibrahim, B., Kireev, D., Sel, K., Kumar, N., Akbari, A., Jafari, R., & akinwande, d. (2022). Continuous Cuffless Monitoring of Arterial Blood Pressure via Graphene Bioimpedance Tattoos (version 1.0.0). *PhysioNet*. https://doi.org/10.13026/ce62-pc98.

[31] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

[32] https://github.com/serengil/LightPHE

7. Declaration:

I hereby declare that this is my own work.