A Protocol for Secure Distributed Spatial Searching Using Homomorphic Encryption

by

Jimmy Katambo

Dissertation submitted to the University of Zambia in partial fulfillment of the requirements of the degree of

Master of Science in Computer Science

Declaration

I declare that this dissertation is my own, original work, except to the extent that has been acknowledged. This dissertation is being submitted for the degree of Master of Science, in Computer Science at the University of Zambia. The dissertation has not been submitted before for any degree or examination to any other University.

Sections of this work have been published as part of the conference proceedings and abstracts:

J. Katambo, M. Nyirenda and D. Zulu, "A Protocol for Secure Distributed Spatial Searching Using Homomorphic Encryption," Proceedings of The International Conference in ICT (ICICT2019) - Lusaka, Zambia (20th - 21st November 2019), pp. 178-183, Dec. 2019.

Signature: .....

Date: .....

Certificate of Approval

This dissertation of Jimmy Katambo has been approved as fulfilling the requirements or partial fulfilment of the requirements for the award of Master of Science in Computer Science by the University of Zambia.

Examiner's Signature: ..... Date: .....

Examiner's Signature: ..... Date: .....

Examiner's Signature: ..... Date: .....

Chairperson: .....Signature: .....Date: .....
Board of Examiners

Supervisor: .....Signature: .....Date: .....

Dedication

This dissertation is dedicated to

God Almighty

and

my beloved family and friends.

Acknowledgments

I am highly indebted to Dr. Mayumbo Nyirenda (lecturer at the Computer Science Department of UNZA) who unreservedly supervised this research. Without his guidance this research would have been less comprehensive.

The work of Dr. Jackson Phiri, whose motivation to us as part of his students in the class of 2017/2018 intake, is greatly appreciated.

I am also indebted to Twashuka Investments Limited who allowed me time to study.

I value the precious time I shared with my fellow postgraduate students.

May God bless all the aforesaid contributors to the research.

Abstract

Privacy of a person doing a spatial search as well as security of data being provided by a data provider can be of great concern in collecting statistical data. This has led to those who engage in searching for data to ask questions on whether they are anonymous to the one who is providing the data and also whether the one who is providing the data is anonymous to those who are doing the searching. However, such questions have provided an opportunity to the computer science community (researchers) to seek for solutions that can be used to deal with this problem of lack of anonymity of data and confidentiality. This study therefore aimed to propose a protocol by putting into application Homomorphic Encryption and a Distributed Ring Algorithm, to ensure anonymity of data of both parties involved in a spatial search, that is, a data provider and a searcher. To achieve this aim, three objectives were set. The first objective was to identify a Homomorphic Encryption technique that can support a spatial search. This was achieved by reviewing literature on Homomorphic Encryption techniques. Paillier Homomorphic Encryption technique was identified as the best approach that can support a spatial search. The second objective was to develop a protocol for distributed spatial searching based on the best Homomorphic Encryption technique which was identified. After analysing how Pallier Encryption works, a protocol was then designed based on distributed system principles. The third objective was to develop a proof of concept using the proposed protocol. A prototype implemented as distributed application was written in Java using the proposed protocol. The study implemented an application from the protocol developed which proved useful for collecting statistical data with guaranteed confidentiality. It also proved that, by putting into application Homomorphic Encryption, the person who was doing the search became anonymous to the providers of the data and the data provided by the providers became anonymous to the person who was doing the searching.

Keywords: Homomorphic Encryption, Paillier Cryptosystem, Confidentiality, Anonymity, Spatial search, Privacy, Distributed Systems, Ring Algorithm.

Table of Contents

List of Tables

List of Figures

Acronyms

CC	Cloud Computing
HE	Homomorphic Encryption
SLR	Systematic Literature Review
RSA	Rivest, Shamir and Adleman
ACM	Association for Computing Machinery
LBS	Location Based Services
DCRA	Decisional Composite Residuosity Assumption
GIF	Graphical Interchange Format
GIS	Geographic Information System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
RAM	Random Access Memory
W3W	World Wide Web Consortium
IoT	Internet of Things
RBAC	Role-Based Access Control
POIs	Point of Interests
EHR	Electronic Health Records
PHE	Partially Homomorphic Encryption

FHE	Fully Homomorphic Encryption
MPC	Model Predictive Control
SIP	Spatial Information Systems
CS	Critical Section

CHAPTER 1 RESEARCH INTRODUCTION

This chapter is an introduction of the research study. The subtopics contained in this chapter are: Background, Statement of the problem, Aim of the study, Objectives, Research Questions and Significance of the study prior to the Summary of the entire chapter.

1.1 Background

Spatial searching permits us to search for data based on location preferences. The result is limited to the physical area of the location. The question that arises is whether the person who is doing the search is anonymous to the providers of the data and the data provided by the providers is anonymous to the person who is doing the searching. This introduces a security flaw. Some common examples of spatial searches are finding all highways within a distance of a city, finding the cities above a specified population nearest to a highway, mapping applications to search library, museum and archive collections, etc. However, our research will emphasize more on the following key examples. In secure geocoding, a geocoder can be made available to the cancer surveillance community where members can submit the addresses of their cases to this geocoder to convert addresses into latitude/longitude coordinates suitable for data aggregation (e.g., calculation of county-specific incidence rates) and analysis. Sharing of confidential data across cancer registries to support case aggregation is another key example in our research. To study rare cancers, researchers may wish to aggregate all cases of a specific cancer across geographic boundaries to pool data from a sufficient number of cases for statistical analyses, for

example, to find an average number of cases per geographic boundary. Figure 1-1 shows a representation of a spatial search with the searcher on one end and the data provider on the other.

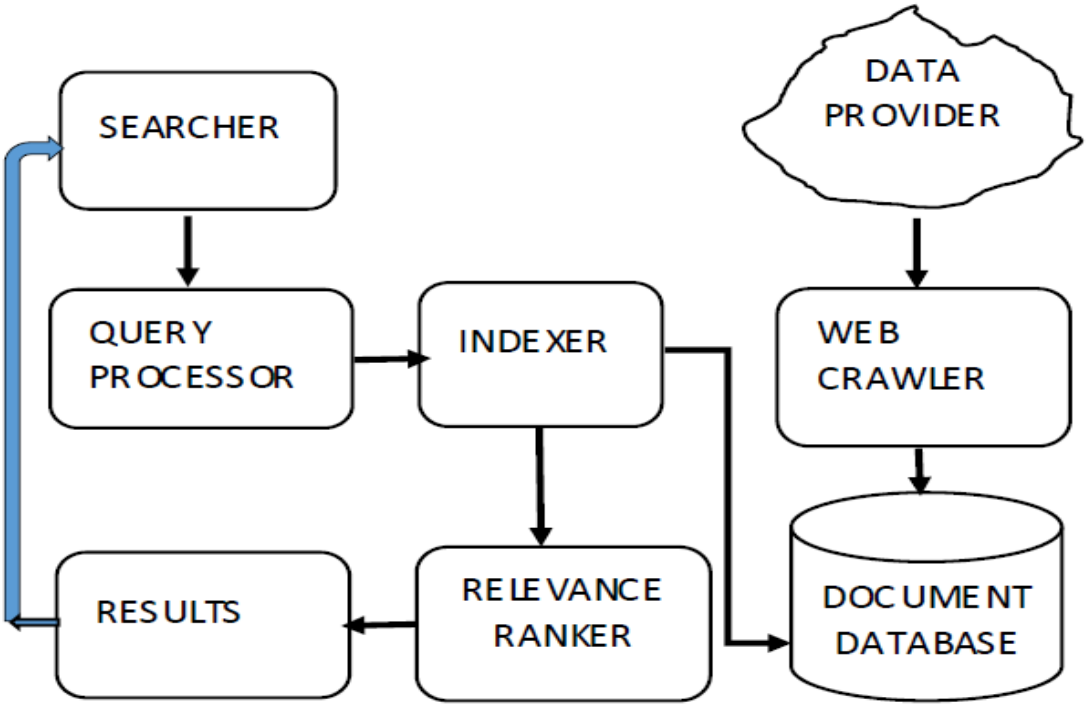

Figure 1-1: Shows a representation of a spatial search with the searcher and the data provider.

This research aims to design a protocol by putting into application Homomorphic Encryption and a Distributed Ring Algorithm, to ensure anonymity of data of both parties involved in a spatial search, that is, a searcher and a data provider.

1.2 Problem Statement

There is no anonymity of data during a spatial search. Hence, there is lack of confidentiality.

The problem of lack of anonymity and confidentiality can be experienced by those who collect statistical data online as well as those who provide the data. One end may be secure, for example, the one providing data and yet the other end, for example, the one collecting data, may not be secure. In another scenario both the data provider and collector may seek anonymity.

Anonymity, the basic definition of this term is "being without a name." Simply understood someone is anonymous if his/her identity is not known. Anonymity is also associated with privacy as sometimes it is desirable not to have a direct link with a specific entity, though sometimes it is required by law to present an identity before and/or during the period an action is performed [1]. Anonymity and confidentiality are also terms that people often mix. To keep a participant's participation in your study completely anonymous means not having any Personally Identifying Information (PII) about them. Since we typically conduct screeners to qualify participants for our studies, we know their names, e-mail addresses, etc. Instead, we typically keep their participation confidential. We do not associate a participant's name or other personally identifiable information with his or her data (e.g., notes, surveys, videos) unless the participant provides such consent in writing. Instead, we use a participant ID (e.g., P1, participant 1). It is important to protect a participant's privacy, so if you are unable to provide anonymity, at a minimum, you must keep his or her participation confidential. If the participants are employees of your company, you have a special obligation to protect their anonymity from their managers [2]. Similarly, our research identified the lack of anonymity and confidentiality during the period a person is conducting a spatial search online.

1.3 Aim of the Study

To design a protocol by putting into application Homomorphic Encryption and a Distributed Ring Algorithm, to ensure anonymity of data of both parties involved in a spatial search.

1.4 Research Objectives

The research was guided by the following three objectives, which when attained would help achieve the aim.

1) To identify a Homomorphic Encryption technique that can support a spatial search.
2) To design a protocol based on the identified Homomorphic Encryption technique in "1."
3) To develop a proof of concept prototype using the proposed protocol in "2."

1.5 Research Questions

Through the research, there was need to answer the following questions based on the research objectives in order to achieve the aim of proposing a protocol by putting into application Homomorphic Encryption and a Distributed Ring Algorithm, to ensure anonymity of data of both parties involved in a spatial search.

1) Which one among the preexisting Homomorphic Encryption techniques can support a spatial search.?
2) How can a protocol be designed that uses the cryptographic primitives of the identified Homomorphic Encryption technique in "1."?
3) What proof of concept prototype can be developed using the proposed protocol in "2."?

1.6 Significance of the Study

Developing a protocol for spatial searches using Homomorphic Encryption will help achieve anonymity of the person doing a search and providers of the data and should improve the security of both parties. This ensures privacy and security of the data of the person doing the search and the providers of the data. This is very useful for collecting statistical data with guaranteed confidentiality. The study achieves not only the anonymity of the person doing the search and the anonymity of the data providers but also provides confidentiality to the person gathering statistical data.

Outline of the Dissertation

The study is organized into five chapters. Chapter one is the introduction to the research study. In this chapter, we give the background to the research by providing an overview of spatial searches and what kind of spatial searches this study was focusing on. This chapter has identified the lack of anonymity and confidentiality to be a problem faced by people who collect statistical data and those who provide the data online. The aim of the study and the objectives have been highlighted in this chapter.

Chapter two deals with the literature that adequately address the major thematic areas of the study. Homomorphic Encryption, Homomorphic Encryption Techniques, Spatial Searches and Ring Algorithms in Distributed Systems have been discussed. A section on the knowledge gaps and the various proposed solutions was also added in this chapter.

Chapter three of this research study discusses the methodology of this study. In the methodology, approaches are discussed that help in achieving the research objectives that were set out for this study. The steps that lead to the designing of the protocol are given.

A discussion on how the implementation of the protocol was given. A section on how the validity of the results was done is also given.

Chapter four gives the findings of the research study with chapter five finally giving the comprehensive discussion of these findings. Chapter five also gives the recommendation based on the findings of this research and highlights the future works for consideration.

Summary

The contents of this chapter focused on dissertation introduction. The chapter highlighted a major problem faced when conducting a spatial search. The chapter indicated the lack of security on both ends, that is, the searcher ‘s end and the data provider’s end or just one of the two ends, that is experienced when searching for spatial data online. The premise of proposing a protocol by using Homomorphic Encryption and a Distributed Ring Algorithm, to ensure anonymity of data of both parties involved in a spatial search was explained in the foregoing chapter.

CHAPTER 2 LITERATURE REVIEW

This section contains different literature reviewed from various sources like: journals, conference papers, reports, text books, government documents coupled with selected items from the internet. The subtopics contained in the reviewed literature are: Homomorphic Encryption, Homomorphic Encryption Techniques, Spatial Search and Related Works and Ring Algorithms in Distributed Systems. The chapter begins with literature review on HE, followed by HE techniques, Spatial Search and Related Works, Ring Algorithms in Distributed Systems and ends with a Summary of the whole chapter.

2.1 Homomorphic Encryption

There are two types of homomorphic encryption: Fully Homomorphic Encryption (FHE) and Somewhat Homomorphic Encryption (SHE). Somewhat Homomorphic Encryption (SHE) is also referred to as Partially Homomorphic Encryption (PHE). Each type differs in the number of operations that can be performed on encrypted data. Somewhat Homomorphic Encryption (SHE) cryptosystems support a limited number of operations (i.e., any amount of addition, but only one multiplication) and are faster and more compact than FHE cryptosystems. In short, a scheme is additively homomorphic if it considers addition operators (or multiplication but only limited), and multiplicatively homomorphic if it considers multiplication operators. In a Fully Homomorphic Encryption, both addition and multiplication are performed for an unlimited, arbitrary number of computations [3].

Homomorphic Encryption enables servers to carry out sophisticated mathematical computations on encrypted records without acknowledging the original message [4]. This means without having to use the plaintext, computations can be done. The result will be the same as when you perform the same calculations on plaintext. According to A.A. Izang et al, when it comes to the services provided in the cloud, there are three relevant ethical issues in Cloud Computing which are: (1) privacy issues, (2) integrity of data and; (3) confidentiality [5]. Homomorphic encryption is a promising idea for providing security to many applications [6].

M. Zhao and Y. Geng report that Homomorphic Encryption technology supports the management of ciphertext data under privacy protection [7]. It can directly retrieve, calculate and count ciphertext in the cloud and return the results to users in the form of ciphertext. Compared with traditional encryption algorithms, homomorphic encryption technology does not require frequent encryption and decryption between the cloud and users, thus reducing the cost of communication and computing.

Paillier proposed a probabilistic public-key cryptosystem based on higher-order residual classes, namely Paillier cryptosystem in 1999 [8]. Key generation: set $n=p\cdot q$, where p and q are 2 large prime numbers. g $\in \mathbb{Z}_{n^2}^*$ is randomly selected to make g meet gcd($L(g^{\lambda} \bmod n^2)$,n)=1, where function L is defined as:

$$L(\mathrm{u}) = \frac{u-1}{n}. \tag{1}$$

$$\lambda(n) = lcm\,(p-1, q-1) \tag{2}$$

where the public key is $KP = \{n, g\}$ and $SK = \lambda(n)$. The encryption process is to set an arbitrary plaintext m ∈ $\mathbb{Z}_N^*$ and randomly select number r ∈ $\mathbb{Z}_N^*$, then the ciphertext is:

$$c = E(m) = g^m . r^n \ mod \ n^2, \tag{3}$$

The decryption is:

$$M = D(c) = \frac{L(c^\lambda \ mod \ n^2)}{L(g^\lambda \ mod \ n^2)} . \text{mod } n \tag{4}$$

Note the following representations; c stands for the ciphertext, p and q are large prime numbers, KP is a public key, SK or λ is a private key, m is the plaintext message, L is the auxiliary function used in the decryption method to obtain m(plaintext message), E shows the encryption function, D is the Decryption function. Z_N denote the set of nonnegative integers less than n. $\mathbb{Z}_N^*$ denote the set of integers that are relatively prime to n. g is a random number where it has ordered multiple of n. n is the product of two large primes p and q. gcd is the greatest common divisor which is the same as lcm or lowest common multiple [3]. **mod** denotes the modulo function which returns the remainder or signed remainder of a division, after one number is divided by another (called the modulus of the operation).

How Homomorphic Encryption Works

The following example will demonstrate how these equations are used in many types of Homomorphic Encryption Techniques. The example is based on Paillier Homomorphic Encryption. The example illustrates how Paillier Encryption Algorithm is used in an electronic voting scheme. The sample will contain 20 voters that will choose four different candidates. After every voter finishes deciding on the candidates, their vote's data will be

encrypted through the encryption program. Table 1 indicates that even though the voters select the same candidates, every ciphertext will be unique.

Table 2.1: Vote Messages to be Encrypted [9]

Voter's Name	Candidates				Vote Messages (m)
	C1 (10^0)	C2 (10^1)	C3 (10^2)	C4 (10^3)	
V1			v		m=100
V2	v				m=1
V3				v	m=1000
V4				v	m=1000
V5		v			m=10
V6				v	m=1000
V7		v			m=10
V8	v				m=1
V9	v				m=1
V10	v				m=1
V11				v	m=1000
V12				v	m=1000
V13			v		m=100
V14		v			m=10
V15				v	m=1000
V16		v			m=10

V17		v			m=10
V18		v			m=10
V19			v		m=100
V20				v	m=1000
TOTAL	4	6	3	7	

Now we will encrypt every message m as shown in the table above using keys and the formulation as defined in equation (3) above. Table 2.2 shows the ciphertext from each message that is being encrypted by the Paillier algorithm.

Table 2.2: Encrypted Vote [9]

Voter's Name	Vote Message to be Encrypted	Random r	Encrypted Vote C
V1	100	62437	351538799948203897
V2	1	59119	4590993851485312445
V3	1000	31138	2619624597605475591
V4	1000	37055	2493084312669549831
V5	10	33251	8868564164657005235
V6	1000	48639	2610278780605368606
V7	10	10920	7815835086000532337
V8	1	37782	4173633540368342306
V9	1	11337	6166033743055358318
V10	1	37790	2314588247885282943

V11	1000	40454	5956929699649780388
V12	1000	56727	2236617076652008725
V13	100	1867	10666027394708715038
V14	10	16765	939972971689839058
V15	1000	27529	8423042122639404412
V16	10	17867	9705879308879738285
V17	10	16072	7939254862481225551
V18	10	34348	5125409701887114569
V19	100	49819	5091673028311841742
V20	1000	40942	5014329154833019458

For the tallying process with homomorphic properties of Paillier Algorithm, the server will sum up all encrypted data and mod by n.

$Tally\ (T_c\ ) = \left(\sum_{i=1}^{N_v} C_i\right) mod\ n$ (5).

where $Tally\ (T_c$ is the sum of the encrypted votes above, N_v is the number of votes cast and $C_i$is the value of each encrypted vote as shown in Table 2.2 [9].

$Tally\ (T_c\ ) = \left(\sum_{i=1}^{N_v} C_i\right) mod\ n$ = 106551642554900044578639457146674631799 .

Then the administrator can decrypt T_c to get message m. Using decryption formula in equation (4):

m = L(c^{λ} mod n^2) . μ mod n = 4637.

From the final result from this algorithm, m=4637, this means that candidate C1 has four votes, candidate C2 has six votes, candidate C3 has three votes and candidate C4 has 7

votes. So the candidate C4 is the winner of this election. If we count the number manually, the selections in Table 1 are; four voters select candidate C1, six voters select candidate C2, three voters select candidate C3 and seven voters select candidate C4. The final result will be equal to the final result as calculated earlier.

We can see from Table 2.2 that the size of the ciphertext is a whole lot larger than the plaintext (original message) due to the fact the algorithm is primarily based on big prime numbers of p and q with pretty complex calculations to perform, for instance, factorizing or discrete logarithm with the multiplication of the value of n=pq contribute to these numbers being very big. The function of random number r in this algorithm is to make the result of encryption message that has exactly the same value with different result of ciphertext. Therefore, this algorithm and many more other Homomorphic Encryption Algorithms can guarantee the uniqueness of the data to be stored. The homomorphic properties that we use in the example above proved to produce exactly the same result between tallying done manually and decrypted result after tallying using ciphertext by the computer program [9].

The analysis by K. Balasubramanian and M. Jayanthi shows that Paillier cryptosystem satisfies homomorphism of addition and homomorphism of mixed multiplication. Only one multiplication is allowed [3].

In their paper, A.M. Vengadapurvaja et al, proposes an efficient homomorphic encryption algorithm to encrypt the medical images and to perform useful operations on them without breaking the confidentiality [10]. Healthcare units may require some operations to be performed on the data stored in the cloud as they approach the cloud service providers. In

this case the owners of the medical records have to disclose the original data. Therefore, there must be some approach to perform operations without disclosing the original message. This approach involves Homomorphic Encryption. Homomorphic encryption allows computation to be carried out on cipher text and the results of the operations performed can be decrypted by the data owner. Therefore, the data owner is able to get the same message as if it is performed on two plain texts.

There are some security issues in cloud computing such as data security, third-party control, and privacy. US National Institute of Standards and Technology (NIST) defined Cloud Computing as a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [11]. Applications built on Cloud Architectures are such that the underlying computing infrastructure is used only when it is needed (for example to process a user request), draw the necessary resources on-demand (like compute servers or storage), perform a specific job, then relinquish the unneeded resources and often dispose themselves after the job is done. If all data stored in the cloud were encrypted using traditional cryptosystems, this would effectively solve the three above issues [12]. Traditional Cryptography is a method where communication of confidential information was achieved by encrypting the text which thereby making it incomprehensible. In a traditional cypher, Only the intended receiver has the tools to decrypt this message. The goal of a traditional cipher is to scramble a plaintext in such a way that any interceptor of this cipher text can't make heads or tails of it. A well-known

historic cipher is the cipher Caesar used to communicate. The Caesar's cipher takes the characters from the plaintext, shifting the alphabet by a fixed number, the key, and replacing these characters by their shifted counterparts. For example, a left shift 3 of plaintext "BABE" would become "YXYE" [13]. Similarly, with traditional cryptosystems, to perform a required computation on encrypted data stored in cloud, a user must share the secret key with a cloud provider. After receiving the secret key, the cloud provider decrypts the data to execute necessary operations then sends the result to the user. To solve this issue of having to share the secret key with a cloud provider who has to use this key to decrypt and do necessary operations, it is necessary to use a cryptosystem based on Homomorphic Encryption to encrypt the data since these cryptosystems allow to do computations on encrypted data.

The problem being tackled in our study has to do with lack of anonymity and confidentiality being experienced by those who collect statistical data online as well as those who provide the data. However, the solution lies in not only using Homomorphic Encryption but by also applying a specific kind of Homomorphic Encryption and applying the distributed system principles too. The former and the later applications are part of the discussion in the Literature Review.

Homomorphic encryption is a type of encryption that allows particular computations to be conducted on cipher text and return an encrypted result, the decrypted result is equal the result of conducting the operation on the plain text [12]. The security issues of data stored in the cloud can be solved by using Fully Homomorphic Encryption (FHE) schemes. To secure it, the data should be encrypted with FHE before being sent to the cloud. First, the user logs in and uses the key generation provided by the server to generate

the secret key, the user is the only holder of this secret key. Then, the user encrypts the data that he wants to send to the cloud. During transmitting, the integrity and non-repudiation can be assured by applying other cryptographic technologies such as digital signature. When the user wants the server to execute some computations on these encrypted data (such as search), he can send encrypted request to the cloud server. The server performs the required operations and sends the encrypted result to user. Finally, the user decrypts the data with his secret key to retrieve the correct result. Figure 2-1 illustrates the process of using FHE to cloud computing.

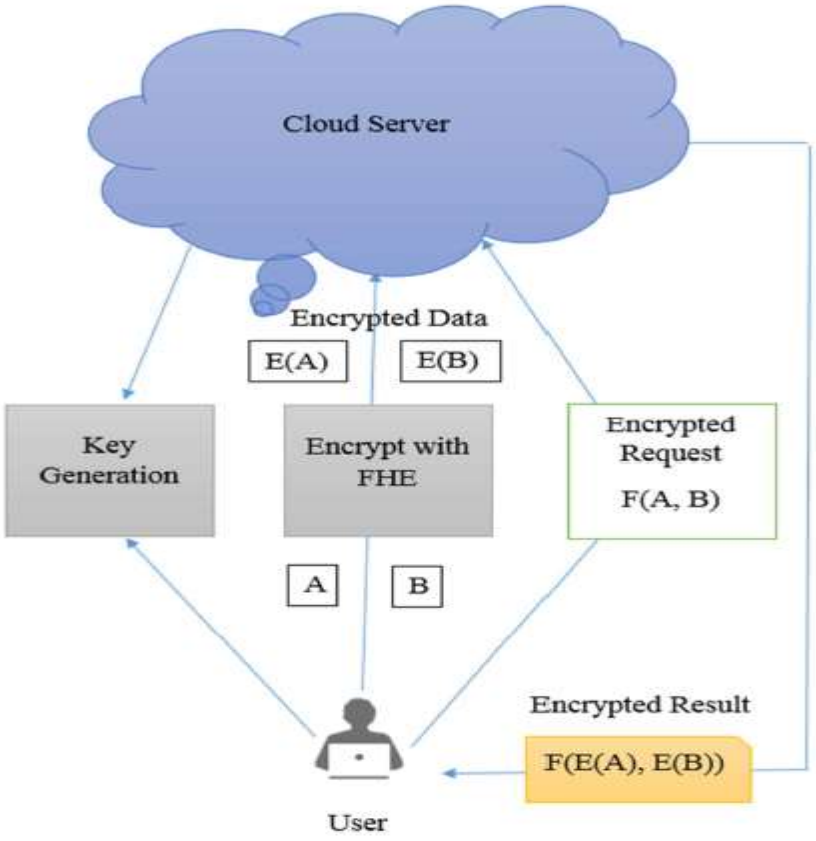

Figure 2-1: Applying FHE to Secure Cloud Data- ‘Adapted From [12]’.

According to S. M. P. C. Souza and R. S. Puttini, the provider will always have privileged access to every part of his service infrastructure, and a demanding service provider could misuse the virtualization and provisioning of basic software stack to capitalize on eventual access to consumer data [14].

There are important concerns when trusting sensitive information to the cloud. Health and financial records, for instance, suffer strict legal restrictions to data escrow. Organizations holding such information need to assure end-users and authorities that a third party will never access restricted data. Client-side encryption is a common solution in literature. Most works fail, however, to reason the impact of security solutions on performance and usability. Both S. M. P. C. Souza and R. S. Puttini suggest Homomorphic Encryption and order preserving encryption systems to mitigate such negative impacts, as they allow the computation of regular searches over encrypted records on the cloud, while preserving information confidentiality and the privacy of end-users [14].

With little control over environment configuration and the resulting security of processes, and little or no access to credible auditing tools, the average consumer can only control what is delivered to the cloud. That makes client-side encryption an important topic. The intuition is that, if every piece of data is encrypted at client-side in a way that even an attacker with enormous computing power cannot break information confidentiality, or end-user's privacy, then the use of a cloud service does not impact information escrow policies [15]. F. Kerschbaum observes that Homomorphic and order preserving encryption systems can mitigate such negative impacts, as they allow the computation of

regular searches over encrypted records on the cloud, while preserving information confidentiality and the privacy of end-users.

According to K. E. Makkaouia et al, concerns over the confidentiality of sensitive data are still the main obstacles limiting the wide-spread adoption of cloud services [16]. Actually, scientists have suggested a new encryption form, called Homomorphic Encryption (HE), which can offer a third-party having the ability to perform operations on encrypted data. The HE property can be considered as a useful method to get over these concerns.

K. E. Makkaoui et al write that confidentiality is among the most important obstacles for widespread adoption of cloud services. In fact, researchers have invented a new promising form of encryption, Homomorphic Encryption, that can be considered as an effective way to overcome these obstacles. Since cloud computing environment is more threatened by attacks and since cloud consumers often use resource constrained devices to get access to cloud computing services, the homomorphic encryption schemes must be promoted, in terms of security level and running time, to work efficiently.

K. E. Makkaoui et al propose a new fast variant of the Cloud-RSA scheme to speed up its algorithms. Simulation results show that the proposed variant gives a large speed up over the Cloud-RSA scheme while preserving a prescribed security level [16].

S. S. Mathew and C. A. Hafsath observe that the key elements in a cloud computing scenario include the data user, data owner and cloud server [17]. They don’t reside in the same trustworthy area and hence maintaining confidentially of the data is a challenge.

There are possibilities of data leakage from the server knowingly or unknowingly. Homomorphic Encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. In the setup phase of Homomorphic Encryption the data owner generated a secret key SK' and a set of public keys PK'.

Homomorphic Encryption techniques enable computing with encrypted data. It means, one is able to perform the operations on this data without converting into the plaintext [18]. Data is in encrypted state in most of the stages on the cloud. Fully Homomorphic Encryption (FHE) technique allows user to perform multiple types of operations on encrypted data. Only one kind of operation is allowed in a Partially Homomorphic Encryption (PHE) technique. Figure 2-2 shows a proposed Fully Homomorphic Encryption (FHE) system [19].

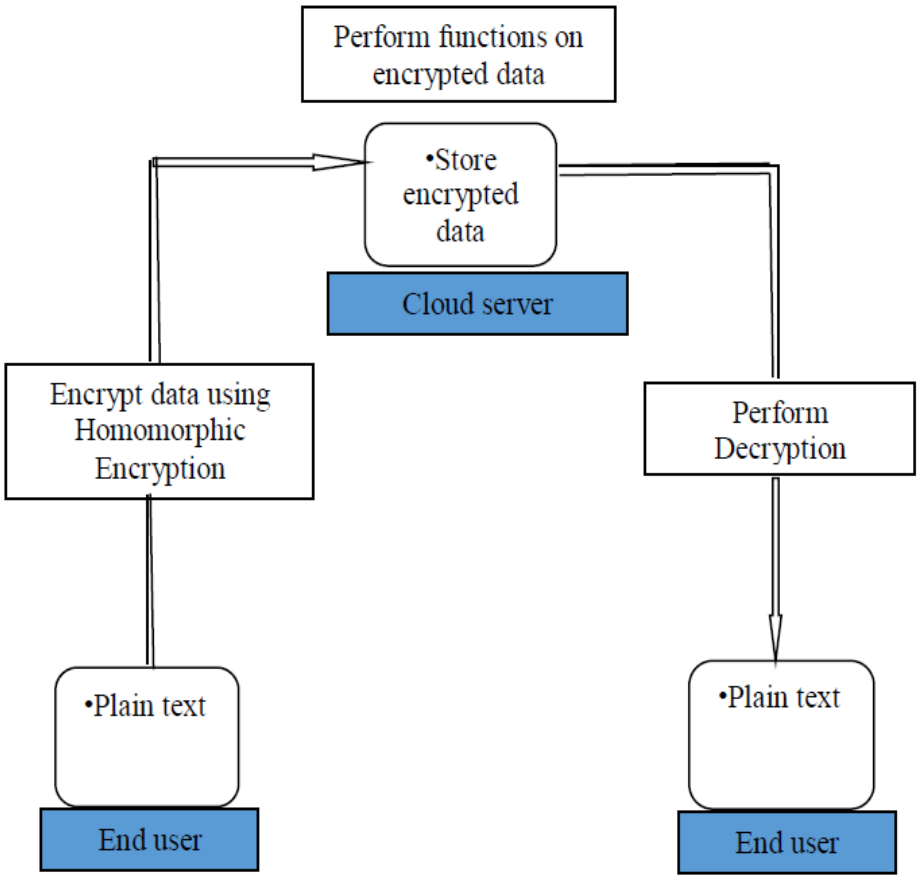

Figure 2-2: FHE Proposed System-'Adapted From [19]'

In their paper, A.M. Vengadapurvaja et al observe that the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 encourages the use of Electronic Health Records (EHR) [20]. EHR improve the data accessibility, ease the updating of computerized data and allow e-messaging between providers. Cloud computing offers various services to health care units, the data are stored in the cloud. Three important aspects of cloud are performance, availability and security. Hence, there is the need for

efficient homomorphic crypto algorithms. Their paper proposes the efficient homomorphic encryption algorithm by using Fully Homomorphic Encryption (FHE) to encrypt the medical images and to perform useful operations on them without breaking the confidentiality.

According to S. Dasgupta, security of data, especially in clouds, has become immensely essential for present day applications [21]. Fully homomorphic encryption (FHE) is a great way to secure data which is used and manipulated by untrusted applications or systems. In his paper, S. Dasgupta proposes a symmetric FHE scheme based on a polynomial over ring of integers. This scheme is somewhat homomorphic due to accumulation of noise after few operations, which is made fully homomorphic using a refresh procedure. After certain amount of homomorphic computations, large ciphertexts are refreshed for proper decryption. The hardness of the scheme is based on the difficulty of factorizing large integers. Also, it requires polynomial addition which is computationally cost effective according to their experiments.

In their paper, F. Farokhi, I. Shames and N. Batterham considered networked control systems with encrypted sensor measurements [22]. They employed a semi-homomorphic encryption technique using Paillier Encryption since it was assumed that the sensors use the Paillier encryption, which is a semi-homomorphic encryption, so that the controller can perform the required computation on the encrypted data. The parameters of the encryption technique were constructed to guarantee the stability of the closed-loop system and to ensure certain bounds on the closed-loop performance. This is shown in Figure 2-3 below.

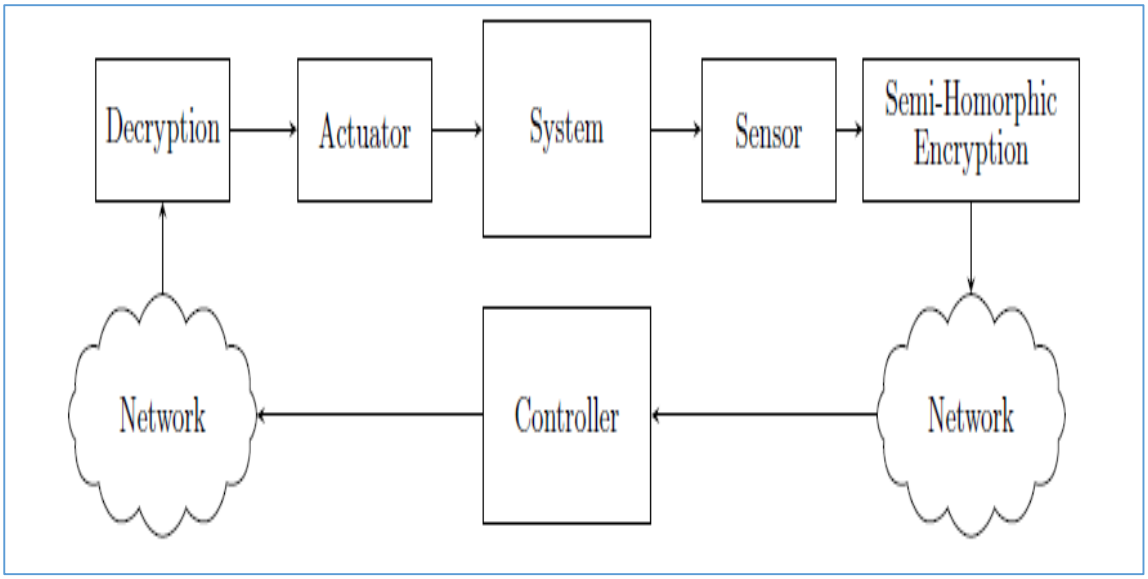

Figure 2-3: The schematic diagram of a networked control system with semi-homomorphic encryption-decryption units -'Adapted From [22]'.

M. Ogburn et al, in their paper, illustrate that the study of homomorphic encryption techniques has led to significant advancements in the computing domain, particularly in the sphere of cloud computing [23]. Homomorphic encryption provides a means for securely transmitting and storing confidential information across and in a computer system. The aim of their paper was to discuss the concepts and significance of Homomorphic Encryption along with the subdivisions and limitations associated with this type of encryption scheme. Recent studies conducted on the topic of Homomorphic Encryption were highlighted and some customary models of homomorphism were demonstrated. They also developed a proof of concept algorithm that demonstrates a practical use for a Homomorphic Encryption technique, the results of which were also provided. The applications of Homomorphic Encryption methods are vast outside of the computational realm, and its purpose in other fields were explored in their article.

A secure and private framework for inter-agent communication and coordination is developed by F. Farokhi et al by employing HE. In their research, they particularly use Paillier Homomorphic Encryption [24]. This allows an agent, in their case a fleet owner, to ask questions or submit queries in an encrypted fashion using semi-homomorphic encryption. The submitted query can be about the interest of the other fleet owners for using a road at a specific time of the day, for instance, for the purpose of collaborative vehicle platooning. The other agents can then provide appropriate responses without knowing the content of the questions or the queries. Strong privacy and security guarantees are provided for the agent who is submitting the queries. It is also shown that the amount of the information that this agent can extract from the other agent is bounded. In fact, with submitting one query, a sophisticated agent can at most extract the answer to two queries. This secure communication platform is used subsequently to develop a distributed coordination mechanism among fleet owners.

According to F. Farokhi et al, Paillier encryption method (or rather its security) relies on the Decisional Composite Residuosity Assumption, i.e., for given integers $N \in Z$ and $x \in Z^*_{n^2}$, it is "hard" to decide whether there exists $y \in Z^*_{n^2}$ such that $x \equiv y^n \bmod N$. Here, the notation Z^*_n denotes the set of integers modulo N for all $N \in N$ [24]. The state of the art in factorization dictates that in order to have a secure cryptosystem, the value of N should be at least 2048 bits.

In another paper, S. Ramezanian et al propose a practical protocol for privacy-preserving database queries [25]. They introduce a practical method to perform private membership tests. In this method, clients are able to test whether an item is in a set controlled by the

server without revealing their query item to the server. After executing the queries, the content of the server's set remains secret. One use case for a private membership test is to check whether a file contains any malware by checking its signature against a database of malware samples in a privacy preserving way. They apply the Bloom filter and the Cuckoo filter in the membership test procedure. In order to achieve privacy properties, they present a novel protocol based on some homomorphic encryption schemes. They use the Paillier cryptosystem for homomorphic encryption. Their protocol has a much lower communication complexity than prior schemes and its computation complexity is also low enough for practical use cases.

According to J. W. Bos, K. Lauter and M. Naehrig, increasingly, confidential medical records are being stored in data centers hosted by hospitals or large companies [26]. As sophisticated algorithms for predictive analysis on medical data continue to be developed, it is likely that, in the future, more and more computation will be done on private patient data. While encryption provides a tool for assuring the privacy of medical information, it limits the functionality for operating on such data. Conventional encryption methods used today provide only very restricted possibilities or none at all to operate on encrypted data without decrypting it first. Homomorphic Encryption provides a tool for handling such computations on encrypted data, without decrypting the data, and without even requiring the decryption key. In their paper, they discuss possible application scenarios for Homomorphic Encryption in order to ensure privacy of sensitive medical data. They describe how to privately conduct predictive analysis tasks on encrypted data using Homomorphic Encryption. As a proof of concept, they present a working implementation

of a prediction service running in the cloud, which takes as input private encrypted health data, and returns the probability for suffering cardiovascular disease in encrypted form. Since the cloud service uses Homomorphic Encryption, it makes this prediction while handling only encrypted data, learning nothing about the submitted confidential medical data.

R. Hayward and C. C. Chiang, in their paper, survey several efficient, partially homomorphic schemes, and a number of fully homomorphic, but less efficient schemes [27]. The Gentry's algorithm for fully encryption algorithm was examined in detail for the performance issue. They then present a parallel processing method which can be applied to improve the performance of the Gentry's fully homomorphic encryption algorithm by taking advantage of ample computing power of the cloud. They further observe that Fully homomorphic encryption (FHE) offers the means by which the cloud computing can be performed on encrypted data, while avoiding the data security concerns. FHE is not without its cost, as FHE operations take orders of magnitude more processing time and memory than the same operations on unencrypted data. Cloud computing can be leveraged to reduce the time taken by taking into consideration parallel processing. Hence, in their study, a private cloud was built to support parallel processing of Homomorphic Encryption in the cloud.

M. S. Darup et al, present a novel cloud-based Model Predictive Control (MPC) scheme that is based on a recently proposed real-time proximal gradient method [28]. The cloud-based implementation requires sensitive data (e.g., system states) to be transmitted via public networks and to be processed in the cloud. They guarantee privacy of the data

throughout the control-loop by encrypting the control scheme using (Partial) Homomorphic Encryption. In their case, they employ Paillier Homomorphic Encryption. The resulting encrypted MPC computes encrypted predictive control actions based on encrypted system states (without intermediate decryptions).

To prepare the encryption, it was necessary to quantize the controller and state measurements. In this context, they derived conditions that ensure a reliable operation of the encrypted MPC. More precisely, the identified conditions guarantee global stability of the controlled system and robustness against quantization errors. The functioning of the novel control scheme was illustrated with an example in their paper.

In their paper, K. Xu eta al propose two privacy-preserving mobile application recommendation schemes based on trust evaluation [29]. Recommendations on mobile application are generated based on user trust behaviors of mobile application usage. In these two schemes, user private data can be preserved by applying their proposed security protocols and utilizing Homomorphic Encryption. They further implement two schemes and develop two mobile Apps that can be applied in different scenarios, i.e., a centralized cloud service and distributed social networking. Security analysis, performance evaluation and simulation results show that their schemes have sound security, efficiency, accuracy, and robustness.

Due to the heavy computational complexity of Fully Homomorphic Encryption, they utilize additive homomorphic encryption, concretely Paillier's cryptosystem, to process and secure user data [29].

According to C. Ma and C. W. Chen, a Location Based Service (LBS), allows you to discover nearby like-minded people, and make friends with them [30]. The main privacy threat of this service is that the disclosure of locations leaves opportunities for stalkers. Although location privacy preservation in LBS has recently received much attention, few works have been done on privacy-awareness as it relates to nearby friend discovery, where people want to discover nearby friends without exposing their private locations to arbitrary strangers. Unlike most of other LBS, nearby friend discovery needs to consider the privacy of both of the two communicating entities, i.e., the user who is searching for nearby people, and the user who is being discovered by others. This unique property makes it a great challenge to protect users' location privacy while providing satisfactory service quality. Their paper presents a research addressing this issue by combining the location approximation technique and the homomorphic cryptography. In their case, they also make use of Paillier Encryption Scheme. They show that the proposed scheme provides formal privacy guarantees for the LBS users, and still achieves satisfactory quality of the LBS.

In their article, M. Z. Hasan et al propose a method for secure sharing and computation on genomic data in a semi honest cloud server [31]. In particular, there are two main contributions. Firstly, the proposed method can handle biomedical data containing both genotype and phenotype. Secondly, their proposed index tree scheme reduces the computational overhead significantly for executing secure count query operation. In their proposed method, the confidentiality of shared data is ensured through encryption, while making the entire computation process efficient and scalable for cutting-edge biomedical

applications. They employ the Paillier Encryption technique.

Paillier cryptosystem is a member of homomorphic cryptosystem family. Homomorphic Encryption allows to perform computation on the encrypted data without decrypting it, and if the result is decrypted, it would be the same if computations were performed on the plaintext. Paillier cryptosystem supports addition and it is semantically secure: an adversary with the finite computational power and with the possession of the ciphertext would not be able to extract any information about the plaintext. To guarantee this security, this cryptosystem produces different ciphertexts when a same message is encrypted multiple times. This randomness implies that this cryptosystem is a probabilistic encryption scheme. M. Z. Hasan et al use Paillier Cryptosystem to encrypt the data and utilize its homomorphic properties to execute count query [31].

According to K. E. Makkaoui et al, with Homomorphic Encryption (HE), data can be processed in its encrypted form in Cloud Computing (CC) [32]. This HE property can be considered as a useful solution to get over some concerns limiting the widespread adoption of CC services. Nevertheless, since CC environments are threatened by outsider and insider security attacks and since cloud consumers oftentimes have access to CC services using resource-limited devices, the HE schemes need to be promoted in terms of security level and running time to work effectively. In their study, they boosted the main Paillier's scheme at security level by proposing a variant of the scheme called Cloud–Paillier. The proposed scheme addresses an exception of the Paillier's scheme, supports the additive homomorphism over the integers and withstands more confidentiality attacks. For fast decryption, herein, they proposed two fast variants of the Cloud–Paillier scheme.

The proposed variants use moduli formed of $k \geq 2$ distinct primes. The first variant utilizes the Chinese remainder theorem to decrypt. Whereas, the second variant slightly modifies the form of the Cloud–Paillier's encryption algorithm and decrypts as in the Cloud–Paillier. Theoretical and simulation outcomes show that the suggested variants give a large decryption speed-up over the Cloud–Paillier while preserving a recommended security level [32].

According to K. Muhammad, K. A. Sugeng and H. Murfi, machine learning had been widely used to analyze various kinds of data, including sensitive data such as medical and financial data [33]. A trained machine learning model can be wrapped in a web application so that people can access it easily via internet. However, if the data to be analyzed is private or confidential, this will cause a problem; the application administrator may read the input. According to their paper, this kind of problem can be solved with Homomorphic Encryption scheme. Paillier encryption scheme is one kind of encryption scheme that has homomorphic property. In their research, they show that one type of machine learning model can take an input encrypted by Paillier encryption scheme and produce an encrypted output that shares the same key. A machine learning model was trained with a database of hand-written digits. This model was then tested with the test data encrypted with Paillier encryption scheme. Their experiment showed that the model achieved 92.92% accuracy on the test set.

In their paper, T. Oladunni and S. Sharma, observe that the use of the cloud has brought a drastic improvement into the storage of information in the cloud. Individuals and organization find the technology relevant to their daily use [34]. However, insecurity of

information during computation of data has been a concern to everyone. Therefore, the emergence of Homomorphic Encryption has been an important component in a secured cloud computing. According to T. Oladunni and S. Sharma, the scheme is categorized into full, somewhat or partial homomorphic encryptions. Each of these algorithms has its own challenge. For example, Partial encryption offers encryption mechanism to secure data during computation, however it can only perform either addition or multiplication. It lacks the ability to perform both. Somewhat homomorphic is an improvement over partial homomorphic encryption. It incorporated the evaluate function as part of the cryptosystem. This has the challenge of arbitrary depth. Full homomorphism has the same capability as the somewhat. The algorithm provides a refresh solution to the arbitrary depth problem of somewhat, however, memory usage is still a setback [34].

In their paper, A. Malika et al, present a reversible data hiding scheme based on interpolation technique for encrypted images by using homomorphic and probabilistic properties of Paillier cryptosystem [35]. At first, the image owner generates a location map by using an interpolation technique to estimate the Most Significant Bits (MSBs) of pixel to find whether a pixel can be used for embedding or not. Next, with the help of the location map, the original image is preprocessed to create some spare space for data embedding. Meanwhile, the location map is compressed losslessly and the information of compressed location map is substituted with Least Significant Bits (LSBs) of border pixels. Furthermore, the preprocessed image is encrypted by Paillier cryptosystem and sent to the data hider along with location map and original LSBs of border pixels. At the data hiding phase (note that the original image cannot be accessed), the additional data

and LSBs of border pixels are embedded into homomorphic encrypted image using location map.

Homomorphic Encryption (HE) can help to address the balance of risk and utility in information sharing for some applications in the healthcare industry [36]. Billing and report generation are two such applications. In both cases, analysts need access to individual medical records to compute over some part of their content. By allowing such computation without revealing those records "in the clear", breaches might be avoided without disrupting such applications that are key to daily operations. In Figure 2-4, they illustrate how HE enables a "breach-proof" workflow for such applications in a clinic setting. An analyst (at left in Figure 2-4) queries current medical records to gather information such as statistics on prescriptions issued or medical encounters provided by the clinic. A potentially untrusted server (in the figure right) holds an encrypted corpus of relevant data, potentially including individual medical records subject to Health Insurance Portability and Accountability (HIPAA) protections or other relevant privacy statutes and policies [36]. Homomorphic encryption allows the queries to be computed over that data while it remains encrypted, and returns an encrypted answer to the analyst. The analyst then decrypts the answer on a trusted platform and can include query results in relevant reports or invoices. Because the data corpus remains encrypted both while at rest and while used in computation, any adversaries (upper right in Figure 2-4) learn nothing about the data or the results of such queries.

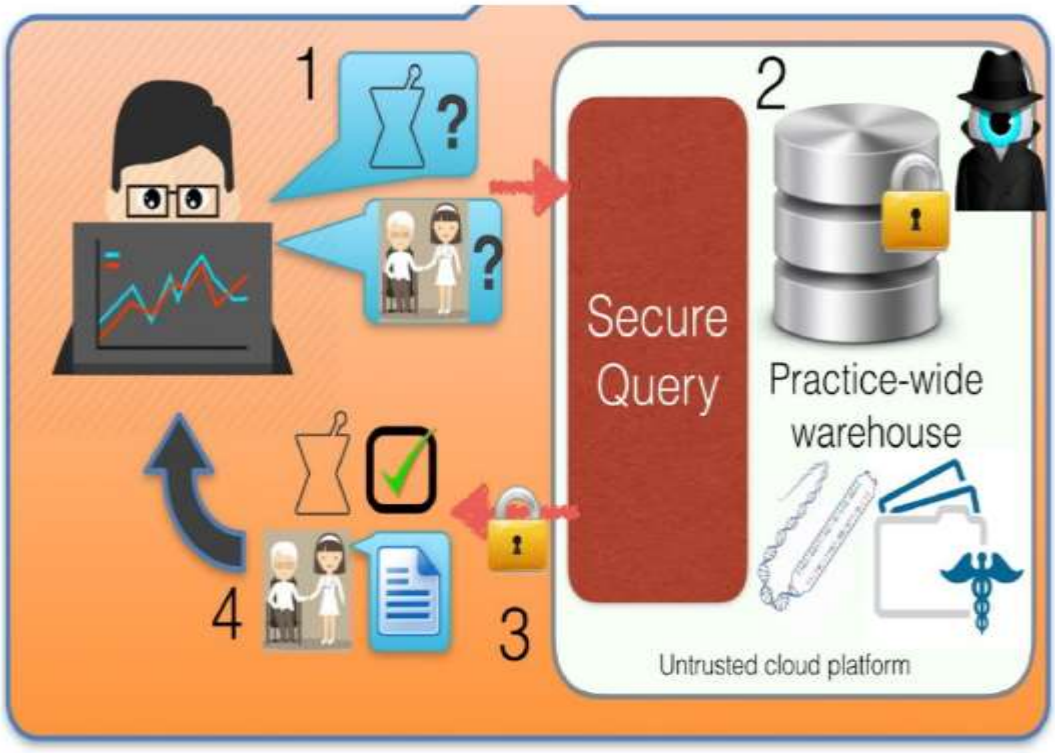

Figure 2-4: Homomorphic Encryption enables clinic analytic workflows over sensitive data-'Adapted From [36]'.

In their paper, W. T. Al-Sit et al write that Cloud Computing provides services rather than products; where it offers many benefits to clients who pay to use hardware and software resources [37]. There are many advantages of using cloud computing such as low cost, easy to maintain, and available resources. The main challenge in the Cloud system is how to obtain a highly secured system against attackers. For this reason, methods were developed to increase the security level in different techniques. Their paper aimed to review these techniques with their security challenges by presenting the most popular cloud techniques and applications. Homomorphic Encryption in cloud computing is presented in their paper as a solution to increase the security of the data. By using this

method, a client can perform an operation on encrypted data without being decrypted which is the same result as the computation applied to decrypted data. In their review paper, some techniques and schemes for Homomorphic Encryption are discussed. Finally, the reviewed security techniques were discussed with some recommendations that might be used to raise the required security level in such a system [37].

An interesting thought from a paper by Z. Cao and L. Liu who observe that cryptography uses modular arithmetic a lot, because it can obscure the relationship between the plaintext and the ciphertext, and drive away the redundancy of the plaintext by spreading it out over the ciphertext [38]. The authors go on to discuss that it is well known that confusion and diffusion are the two basic techniques for obscuring the redundancies in a plaintext message. They could frustrate attempts to study the ciphertext looking for redundancies and statistical patterns. Practically speaking, the real goal of using modular arithmetic in cryptography is to obscure and dissipate the redundancies in a plaintext message, not to perform any numerical calculations.

In their paper, Z. Cao and L. Liu stress that any computations performed on encrypted data are constrained to the encrypted domain (finite fields or rings). This restriction makes the primitive useless for most computations involving common arithmetic expressions and relational expressions. It is only applicable to the computations related to modular arithmetic.

A parallel homomorphic encryption scheme is proposed based on the additive homomorphism of the Paillier encryption algorithm in order to protect data privacy whilst

allowing efficient access to data in multi-nodes cloud environments [39]. In their paper, Z. Min et al propose a parallel homomorphic encryption algorithm, in which plaintext is divided into several blocks and blocks are encrypted with a parallel mode. Their proposed novel privacy-preserving parallel homomorphic encryption scheme uses the additive homomorphism of the Paillier algorithm. The proposed scheme is suitable for cloud computing environments due to its parallelism.

In their survey paper, P. V. Parmar et al mainly focus on public key cryptographic algorithms based on Homomorphic Encryption scheme for preserving security [40]. The case study on various principles and properties of Homomorphic Encryption is given and then various homomorphic algorithms using asymmetric key systems such as RSA, ElGamal, Paillier algorithms as well as various Homomorphic Encryption schemes such as Brakerski Gentry-Vaikuntanathan (BGV), Enhanced Homomorphic Cryptosystem (EHC), Algebra Homomorphic Encryption scheme based on updated ElGamal (AHEE), Non-interactive Exponential Homomorphic Encryption scheme (NEHE) are investigated. Their survey can be helpful to know which and how various cryptographic algorithms are being used for applying Homomorphic Encryption for privacy preservation. There are various Homomorphic Encryption schemes described in their paper which can be used for various mixed Homomorphic Encryption properties.

According to S. Leela and P.Nithyanandam, satellite images are used in determining the dynamic state of the earth surface (viz., land, water, forest etc.,) for each and every moment [41]. Detecting geographical changes on the earth surface using satellite images is crucial in several applications. With the increasing research in geographical sciences

and technologies, there is a huge demand for the privacy and security of these satellite images when they are stored and processed at a remote server. There is a necessity to ensure privacy and enable computations on private data and images located and processed by an untrusted remote server. The goal is to delegate the processing of data without giving away access to it. Homomorphic encryption algorithms are used to attain such special requirements. Their paper proposes a research work on practical implementation of change detection from one image with the other using image differencing technique on encrypted images which are processed by an unsecured remote server. Paillier encryption scheme is proved to be secure and applicable for Homomorphic operations. Paillier algorithm is implemented to examine the Homomorphic operations on encrypted grayscale images. The correctness of the algorithm is verified by encryption, decryption and evaluation function.

According to R. A. Hallman et al, with the ubiquity of mobile devices and the emergence of Internet of Things (IoT) technologies, most of human activities contribute to ever-growing data sets which are used for big data analytics for a variety of uses, from targeted advertising to making medical and financial judgments and beyond [42]. Many individuals and organizations adopt this new big data paradigm without giving any consideration to privacy and security when they create this data and voluntarily give it up for aggregation. Data breaches have become such a common occurrence that it is easy to despair that concepts like privacy and security are outdated and humans should simply accept data leakage as a new normal. Homomorphic Encryption (HE) is a method of secure computation which allows for calculations to be made on encrypted data without

decrypting it and without giving away information about the operations being done. While HE has historically been plagued by computational inefficiencies, the field is rapidly advancing to a point where it is efficient enough for practical use in limited settings. In their paper, they argue that, with sufficient investment, HE will become a practical tool for secure processing of big data sets [42].

A paper by M. K. Ibrahem describes a complete design scheme, and evaluation of a provable secure remote e-voting system [43]. Electronic voting is the voting process held over electronic media. For such a sensitive issue like elections, security is one of the main concerns, such as authentication, confidentiality and integrity. Simplicity is also necessary to ensure the participation of common people. Besides security and simplicity, other issues need to be considered such as reliability, convenience, flexibility, mobility and cost. In the paper by M. K. Ibrahem, a prototype of a web-based robust electronic voting system is presented with two new techniques; the first technique is a new zero knowledge authentication protocol based on Diffie-Hellman (D-H) key exchange algorithm, to ensure a mutual authentication between the election authority server and the voters. The second technique is Homomorphic Encryption scheme based on Paillier Homomorphic Encryption technique to encrypt all the votes and perform the calculation of the votes without revealing any information about it to ensure the security of the votes and maintain the confidentiality. The proposed system provides secure voting over the Internet and maintains the requirements of the voting process.

In their paper, M. Ibtihal, E. O. Driss and N. Hassan describe how they use Paillier Cryptosystem to secure outsourced images in a mobile cloud computing environment

[44]. For this purpose, they propose a secure architecture composed of two clouds a private cloud dedicated for encryption/decryption and a second public cloud dedicated for storage. They have implemented the first cloud using openstack which is an open source project used to build a private/public cloud infrastructure while respecting the encryption as a service concept.

G. M. Penn et al introduce a more efficient matching process for generic binary template vectors by exploiting Paillier's capability of encrypting messages larger than one bit at a time [45]. They experimentally evaluate and compare the corresponding key generation, encryption, and matching mechanisms of the proposed scheme to those of Paillier and Goldwasser-Micali with respect to computational efficiency. Finally, an iris-based identification approach is assessed using the proposed technique. A biometric matching technique more efficient than the Goldwasser-Micali approach is proposed based on exploiting Paillier's capability of encrypting messages larger than one bit at a time.

2.2 Homomorphic Encryption Techniques

2.2.1 Rivest, Shamir and Adleman (RSA):

RSA is a commonly adopted public key cryptosystem which was proposed by Rivest, Shamir and Adleman in 1977 at Massachusetts Institute of Technology (MIT) [46]. The security of RSA rests on the effort to factorize the big numbers of modulus. The size of modulus value is 1024 bits while the recommended length is 2048 bits as 640 bits key is not secure [47]. RSA uses two pairs of related keys *(public key) ku = {e, n }* for encryption and *(private key) kr ={d, p, q}* for decryption. A description of variables and operators in the stated equations is illustrated in full in the following steps below:

Step 1: p and q are two relatively prime and large random numbers.

Step 2: A positive integer n is defined as a product of p and q.

Step 3: Eulers value of

$$\varphi(n) = (p-1)(q-1). \quad (6)$$

Note that in number theory, Euler's totient function counts the positive integers up to a given integer n that are relatively prime to n. It is written using the Greek letter phi as $\varphi(n)$ or $\phi(n)$, and may also be called Euler's phi function. In other words, it is the number of integers k in the range $1 \leq k \leq n$ for which the greatest common divisor gcd(n, k) is equal to 1. The integers k of this form are sometimes referred to as totatives of n.

For example, the totatives of n = 9 are the six numbers 1, 2, 4, 5, 7 and 8. They are all relatively prime to 9, but the other three numbers in this range, 3, 6, and 9 are not, since gcd(9, 3) = gcd(9, 6) = 3 and gcd(9, 9) = 9. Therefore, $\varphi(9) = 6$. As another example, $\varphi(1) = 1$ since for n = 1 the only integer in the range from 1 to n is 1 itself, and gcd(1, 1) = 1.

Therefore, step 2 involves computing their system modulus N= p.q

and ø(N)=(p-1)(q-1). To calculate ø(N)=(p-1)(q-1), we use the following;

$\lambda = \text{lcm}(p-1, q-1)$ or $\lambda = \gcd(p-1, q-1)$, where λ is a private key and gcd is the greatest common divisor which is the same as lcm or lowest common multiple.

Step 4: Choose e such that $1 < e < M < n$ and

$$C = Me \bmod n. \quad (7)$$

Note that M denotes the plaintext, C is the encrypted message or cyphertext while e and n are public keys.

Step 5: In RSA e and n are public keys and d and (p, q) are private keys so the plaintext M is encrypted by:$1 < M < n$ and $C = Me \bmod n$.

Step 6: The cipher text C is decrypted by

$$M = Cd \bmod n \quad (8)$$

Similarly, M denotes the plaintext while C is the encrypted message or cyphertext while **mod** refers to the modulo function which returns the remainder or signed remainder of a division, after one number is divided by another (called the modulus of the operation).

RSA is one of the earliest multiplicative homomorphic encryption schemes [48]. RSA has a multiplicative homomorphic property which can be used in many applications where there is a requirement for those applications to perform multiplicative operations on sensitive user data. There are several techniques that improve the speed of the encryption procedure of RSA, however they also cause more computational complexity on decryption side [49]. Consequently, RSA is mathematically unstable like other cryptographic techniques. Different restrictions could be observed and several successful attacks are developed to break this algorithm [50]. The major problem is related to its factorizing. The whole algorithm is broken once the process of factorization is done. When this happens, RSA has no more guarantees that the secrets it guards will remain secure.

In their discussion on RSA, M. Tebaa and S. EL Hajii [51], write the following;

Let n=pq where p and q are primes. Pick a and b such that ab≡ $1(mod\emptyset(n))$. n and b are public while p, q and a are private.

$$e_k(x) = x^b mod\ n \quad (9)$$

$$d_k(y) = y^a mod\ n \quad (10)$$

The Homomorphism: Suppose x1 and x2 are plaintexts.

Then,

$$e_k(x1)\ e_k(x2) = x1^b x2^b \text{mod n}$$

$$= (x1x2)^b \text{mod n}$$

$$= e_k(x1\ x2) \quad (11)$$

Figure 2-5 shows that Amos encrypts his message with a public key and sends a computation requests to the cloud. The cloud performs calculations on encrypted data that are based on Multiplicative Homomorphic Encryption or RSA and sends the encrypted message back to the same client company where Amos and Sarah are found. Sarah who has the privilege of having the secret or private key is able to decrypt the encrypted data to find out what the original message is.

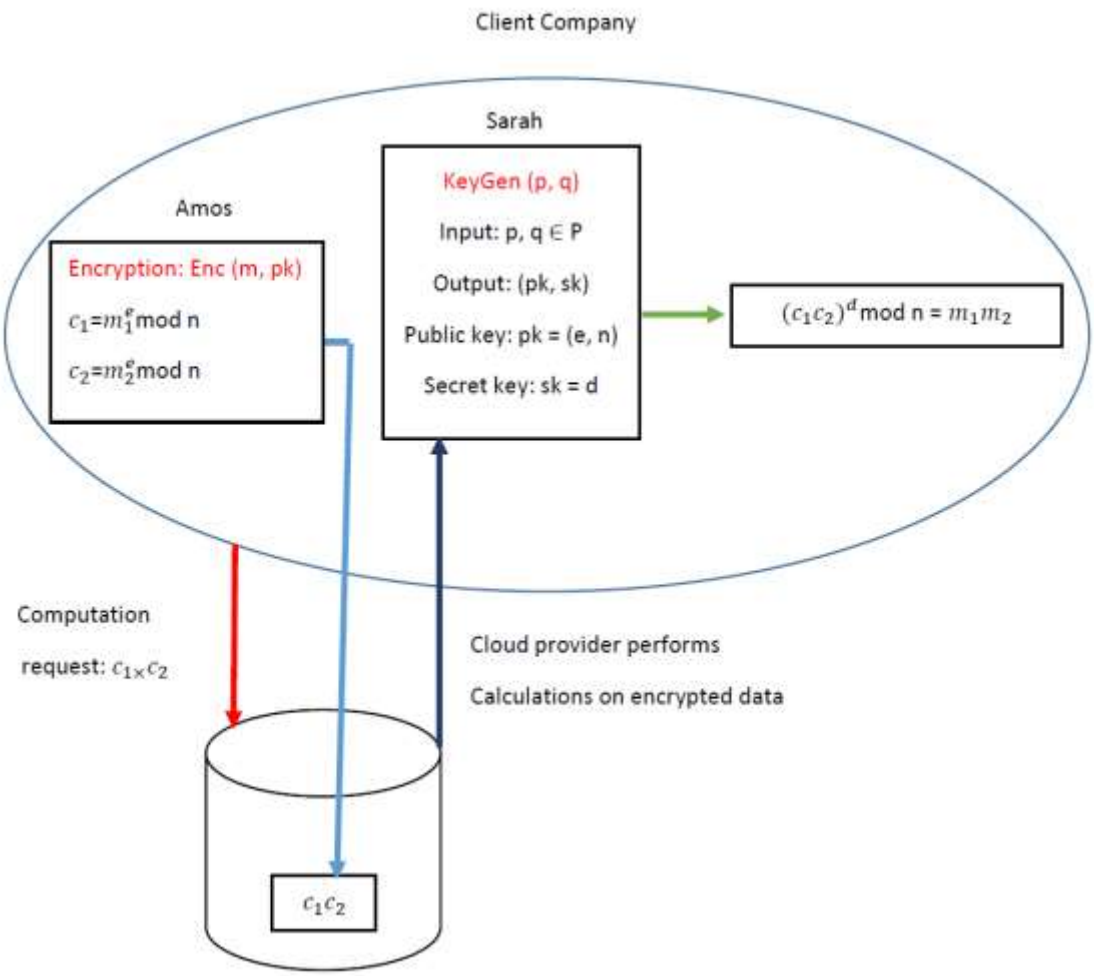

Figure 2-5: Multiplicative Homomorphic Encryption Applied to Cloud Computing- 'Adapted From [51]'.

2.2.2 El Gamal Cryptosystem:

According to M. Alkharji and H. Liu, similar to RSA, the public key encryption scheme given by El-Gamal is a multiplicative homomorphic encryption cryptosystem [52]. It was proposed by Taher El-Gamal in 1984, and its security relies on the hardness of the Diffi-Hellman problem. El Gamal Cryptosystem portrays a multiplicative homomorphic encryption propriety [51].

Let p be a prime and pick $\alpha \in \mathbb{Z}_p^*$ such that α is a generator of $\mathbb{Z}_p^*$. Pick a and β such that $\beta \equiv \alpha^a \pmod{p}$. p, α and β are public; a is private. Let $r \in \mathbb{Z}_{p-1}$ be a secret random number. Then,

$$e_k(x, r) = (\alpha^r \bmod p, x\beta^r \bmod p) \quad (12)$$

Let x1 and x2 be plaintexts. Then,

$$\begin{aligned} e_k(x1,r1)\, e_k(x2, r2) &= (\alpha^{r1} \bmod p, x1\beta^{r1} \bmod p) * \\ &\quad (\alpha^{r2} \bmod p, x2\beta^{r2} \bmod p) \\ &= (\alpha^{r1+r2} \bmod p, (x1, x2)\beta^{r1+r2} \bmod p) \\ &= e_k(x1x2, r1r2) \end{aligned} \quad (13)$$

If we put the plaintext in the exponent, we get:

$$e_k(x, r) = (\alpha^r \bmod p, \alpha^x \beta^r \bmod p) \quad (14)$$

Then the homomorphism is additive:

$$\begin{aligned} e_k(x1,r1)\, e_k(x2, r2) &= (\alpha^{r1} \bmod p, \alpha^{x1} \beta^{r1} \bmod p) * \\ &\quad (\alpha^{r2} \bmod p, \alpha^{x2} \beta^{r2} \bmod p) \\ &= (\alpha^{r1+r2} \bmod p, \alpha^{x1+x2} \beta^{r1+r2}) \bmod p \\ &= e_k(x1+x2, r1+r2) \end{aligned} \quad (15)$$

2.2.3 Goldwasser-Micali Scheme:

According to R. Shruthi, P. Sumana and A. K. Koundinya, the Goldwasser–Micali cryptosystem is an asymmetric key encryption algorithm developed by Shafi Goldwasser and Silvio Micali in 1982 [53]. Goldwasser-Micali has the distinction of being the first probabilistic public-key encryption scheme which is provably secure under standard cryptographic assumptions. However, it is not an efficient cryptosystem, as ciphertexts may be several hundred times larger than the initial plaintext.

One of the drawbacks to the RSA encryption algorithm as originally defined is that it leaks a single plaintext bit in every ciphertext [54], [55]. The GM cryptosystem was the first cryptosystem to provably solve this problem. It was presented by Goldwasser and Micali along with a rigorous definition of security known as semantic security and a proof that the GM cryptosystem is semantically secure against plaintext attacks.

In their research, R. Shruthi, P. Sumana and A. K. Koundinya show that Goldwasser-Micali takes slightly more time for encryption than RSA but it is also more secure against attacks. Secondly, they also show that the Decryption time of RSA remained almost constant between plaintext sizes 2 to 26 bytes while the Goldwasser-Micali had greater varying encryption times reaching a maximum of 27.5milli second plaintext of 6 bytes and minimum of 3.6milli seconds for plaintext of 20 bytes. Thirdly, they found out that RSA and Goldwasser Micali show a linear increase in the number of blocks of ciphertext generated for increasing plain text sizes. However, the increase is more pronounced in the case of Goldwasser Micali which can be attributed to the generation of a random value for every bit of plain text in Goldwasser micali [53].

In their conclusion of their paper, they demonstrate that Goldwasser and Micali develop a bit encryption function based on the hardness of quadratic residuosity problem. The method has many useful properties, but there is one major drawback: for a given security parameter N, the probabilistic encryption of each bit is N bits long, requires N random bits, and uses several operations on N bit integers. A major disadvantage of the Goldwasser-Micali scheme is the message expansion by a factor of lg n bits. Some message expansion is unavoidable in a probabilistic encryption scheme because there are

many ciphertexts corresponding to each plaintext. The homomorphic property of Goldwasser-Micali public-key encryption scheme, has the ability to treat plaintext bit after bit, and the security is based on its semantic security, namely the quadratic residuosity assumption [56].

The algorithms (K, E, D) of Goldwasser-Micali scheme are defined as follows:

1. The key generation algorithm K takes a security parameter 1^las input, and generates two large prime numbers p and q, n = pq and a non-residue x for which the Jacobi symbol is 1. The public key, PK, is (x, n), and the secret key, SK, is (p, q).

2. The encryption algorithm E takes a message m ∈ {0, 1} and the public key (x, n) as input, and outputs the ciphertext c, where c = $y^2.x^m$ mod n and y is randomly chosen from Z_n^*.

3. The decryption algorithm D takes a ciphertext c and the private key (p, q) as input, and outputs the message m, where m = 0 if c is a quadratic residue, m = 1 otherwise.

It is well-known that, if the quadratic residuosity problem is intractable, then the Goldwasser-Micali scheme is semantically secure [56].

2.2.4 Benaloh Cryptosystem:

Benaloh proposed an extension of the Goldwasser-Micali (GM) Cryptosystem by improving it to encrypt the message as a block instead of bit by bit [57]. Benaloh's proposal was based on the higher residuosity problem. Higher residuosity problem (x^{n}) is the generalization of quadratic residuosity problems (x^2) that is used for the GM cryptosystem. Homomorphic property of Benaloh shows that any multiplication operation

on encrypted data corresponds to the addition on plaintext. As the encryption of the addition of the messages can directly be calculated from encrypted messages E(m1) and E(m2), the Benaloh cryptosystem is additively homomorphic.

2.2.4.1 KeyGen Algorithm:

KeyGen is the operation, which generates a secret and public key pair for the asymmetric version of HE or a single key for the symmetric version.

Block size r and large primes p and q are chosen such that r divides p − 1 and r is relatively prime to (p − 1)/r and q − 1 (i.e., gcd (r,(p − 1)/r) = 1 and gcd (r,(q − 1)) = 1). Then, n = pq and φ = (p − 1) (q − 1) are computed. Lastly, y ∈ Z_n^* is chosen such that $y^{\Phi} \not\equiv 1 \bmod n$, where Z_n^* is the multiplicative subgroup of integers modulo n which includes all the numbers smaller than r and relatively prime to r. Finally, (y, n) is published as the public key, and (p, q) is kept as the secret key.

2.2.4.2 Encryption Algorithm:

For the message m ∈ Z_r , where Z_r = {0, 1, ..., r − 1}, choose a random u such that u ∈ Z_n^* . Then, to encrypt the message m:

$$c = E(m)$$

$$= y^m u^r \pmod n \tag{16}$$

where the public key is the modulus n and base y with the block size of r.

2.2.4.3 Decryption Algorithm:

The message m is recovered by an exhaustive search for i ∈ $\mathbb{Z}_r$ such that

$$(y^{-1}c)^{\frac{\emptyset}{r}} \equiv 1 \tag{17}$$

where the message m is returned as the value of i, i.e., m = i.

2.2.4.4 Homomorphic Property:

$$\begin{aligned} \text{E(m1)} * \text{E(m2)} &= (y^{m1}.u1^r \ (mod\ \text{n})) * (y^{m2}.u2^r \ (mod\ \text{n})) \\ &= y^{m1+m2}.(u1 * u2)^r \ (mod\ \text{n}) \\ &= \text{E(m1+m2 (mod n))} \end{aligned} \tag{18}$$

2.2.5 Additive Homomorphic Encryption (Paillier Cryptosystem):

In 1999, Paillier introduced another novel probabilistic encryption scheme based on composite residuosity problem [57]. Composite residuosity problem is very similar to quadratic and higher residuosity problems that are used in GM and Benaloh cryptosystems. It questions whether there exists an integer x such that $x^{\text{n}} \equiv$ a $(mod\ n^2)$ for a given integer a. Other authors have referred to it as the Decisional Composite Residuosity Assumption (DCRA) [52]. This enables Paillier cryptosystem to have numerous applications such as threshold schemes and e-voting systems.

KeyGen Algorithm: For large primes p and q such that gcd (pq, (p − 1) (q − 1)) = 1, compute n = pq and λ = lcm (p − 1, q − 1). Then, select a random integer g ∈ $\mathbb{Z}^*_{n^2}$ by checking whether gcd (L($g^{\lambda}\ mod\ n^2$),n)=1, where the function L is defined as L(u) = (u−1)/n for every u from the subgroup $\mathbb{Z}^*_{n^2}$ which is a multiplicative subgroup of integers modulo n^2 instead of n like in the Benaloh cryptosystem [57].

Pick two large primes p and q and let n=pq. Let λ denote the Carmichael function, that is $\lambda(n) = lcm\,(p-1, q-1)$. Pick random g$\in Z_{n^2}^*$ such that $L\left(g^{\lambda}\ mod\ n^2\right)\ is\ invertible\ modulo\ n$ ($where\ L(u) = \frac{u-1}{n}$). n and g are public; p and q (or λ) are private. For plaintext x and resulting ciphertext y, select a random r $\in Z_n^*$. Then,

$$e_k(\text{x, r}) = g^m.r^n\ mod\ n^2, \tag{19}$$

$$d_k(\text{y}) = \frac{L(c^{\lambda}\ mod\ n^2)}{L(g^{\lambda}\ mod\ n^2)}.\text{mod n} \tag{20}$$

Finally, the public key is (n, g) and the secret key is (p, q) pair. The Homomorphism: Suppose x1 and x2 are plaintexts. Then,

$$\begin{aligned} e_k(\text{x1,r1})\ e_k(\text{x2, r2}) &= g^{x1}.r1^n g^{x2}.r2^n\ mod\ n^2 \\ &= g^{x1+x2}.(r1r2)^n mod\ n^2 \\ &= e_k(\text{x1+x2, r1r2}) \end{aligned} \tag{21}$$

Note the following representations; c stands for the ciphertext, p and q are large prime numbers, KP is a public key, SK or λ is a private key, m is the plaintext message, L is the auxiliary function used in the decryption method to obtain m (plaintext message), E shows the encryption function, D is the Decryption function. Z_N denote the set of nonnegative integers less than n. Z_N^* denote the set of integers that are relatively prime to n. g is a random number where it has ordered multiple of n. n is the product of two large primes p and q. gcd is the greatest common divisor which is the same as lcm or lowest common multiple. **mod** denotes the modulo function which returns the remainder or signed

remainder of a division, after one number is divided by another (called the modulus of the operation).

To perform addition and multiplication on encrypted data stored in the cloud provider, the client must have two different key generators (one for RSA and one for Paillier) [51].

One limitation of the Paillier Encryption Scheme and other Partial Homomorphic Encryption (PHE) techniques is that you can only perform one computational operation, that is, either addition or multiplication and not both [34].

2.2.6 Fully Homomorphic Encryption Schemes:

In accordance with A. Acar et al, an encryption scheme is called Fully Homomorphic Encryption (FHE) scheme if it allows an unlimited number of evaluation operations on the encrypted data and resulting output is within the ciphertext space [57]. After almost 30 years from the introduction of privacy homomorphism concept, Gentry presented the first feasible proposal in his seminal PhD thesis to a long term open problem, which is obtaining an FHE scheme. Gentry's proposed scheme gives not only an FHE scheme, but also a general framework to obtain an FHE scheme. Hence, a lot of researchers have attempted to design a secure and practical FHE scheme after Gentry's work. Although Gentry's proposed ideal lattice-based FHE scheme is very promising, it also had a lot of bottlenecks such as its computational cost in terms of applicability in real life and some of its advanced mathematical concepts make it complex and hard to implement.

According to V. Biksham and D. Vasuma, another limitation is that FHE does not cater for multiple users [58]. The practical applications which involve the running of enormously large and complex algorithmic computations homomorphically have a

massive computational overhead, which makes the intermediate complex functional computations impractical. Therefore, many new schemes and optimization have followed his work in order to address aforementioned bottlenecks. The security of new approaches to obtain a new FHE scheme is mostly based on the hard problems on lattices.

In another paper, Y. Shi observes the following problems similarly. Firstly, FHE needs running an evaluation algorithm on the decryption function of a constructed bootstrappable homomorphic encryption scheme first [59]. However, this is computationally expensive. Secondly, it is less efficient than lattice-based schemes.

2.3 Spatial Search and Related Works

2.3.1 Knowledge Gap According to Available Literature:

According to G. M. Jacquez et al., access to spatially referenced health data continues to cause concerns among data guardians due to the consequences of privacy breaches or the careless release of sensitive information [60].

Census data has been a staple of spatial analysis in many countries, and continues to be seen as important [61]. The data is typically accurate and detailed with high spatial granularity, easily accessible and presented with high standards of documentation. Consumer data arising from the interaction between customers and service providers are becoming ubiquitous. These data are appealing for research because they are frequently collected and quickly released; they cover a wide variety of attitudes, lifestyles and behavioural characteristics; and they are often dynamically replenished and longitudinal. It is demonstrated that consumer data can make important contributions to understanding problems in Transport Geography and in solving applied problems ranging from

migration, infrastructure investment and retail service provision to commuting and individual mobility. However more effective exploitation of these data depends on the construction of bridges to allow greater freedom in the transfer of data from the commercial to the academic sector; it requires development of frameworks for privacy and ethics in the secondary use of personal data.

In order to realise the potential of consumer data, careful ethical controls and well-designed security protocols are required. New methods from data science may be valuable, but opportunities to restore and reinvigorate classical techniques in the light of new data should not be ignored [61].

A. Tondwalker and P. V. Jan identify the problem that is connected with data collection based on localization [62]. Battlefield surveillance or enemy tracking, rescue operations as well as monitoring of military facilities demand security and reliability of the location information. Similarly, spatial searching involves searching for data limited to a physical location or Geographical area. In their paper, they propose to deal with this problem of lack of security to sensor networks and reliability of the location information by using steganography.

A. AlDairi and L. Tawalbeh, in their paper, discuss that privacy is ensured by protecting privacy related issues such as: protecting identities that means protecting personnel and their confidential data, protecting people areas that indicates to protect each one's space and properties, protecting locations which means preventing spatial tracking, etc. [63].

In their paper, J. Ajayakumar and K. Ghazinour demonstrate the spatial privacy vulnerabilities created due to posting Location Based Services (LBS) check-ins in Twitter by using readily available open-source tools for mapping and reverse geo-coding [64]. Apart from demonstrating the vulnerabilities, they also analyze the privacy policies for Twitter and Swarm app (the LBS), and provide suggestions for improving spatial privacy based on policies documents. Finally, they provide suggestions to improve spatial privacy based on policies, and algorithmic techniques.

M. Kiedrowicz categorizes confidentiality as one of the security attributes that affects Spatial data [65]. He writes that spatial data acquired by remote sensing, photogrammetry, geodesy, cartography, services conducting observations and measurements, monitoring, inventory taking, statistical surveys, used for skillfully executed spatial analyses, allow to generate spatial information. This information applies to objects such as facts, events, objects, phenomena, processes and business activities. The characteristics (sets of features and properties) of these objects are stored in the databases concerned. In practical terms, the mentioned spatial database – geodatabase is a physical representation of real-world objects. It enables storage of spatial data (geometric, descriptive, raster, etc.) in Spatial Information Systems (SIP)) – in databases or repositories of the IT system. He concludes his article by presenting a description of the methods for ensuring the security of databases in SIP/Geographic Information Systems (GIS).

According to R. M. M. Pradeep and N. T. S. Wijesekera, spatial data represents information about the physical location, characteristics and shape of geometric objects [66]. These objects can be point features representing locations, line and polygon entities

representing countries, roads, lakes etc. Furthermore, the relationships between the geographic entities are usually stored as coordinates and topology. The Geographic Information System (GIS) is a computer system that can be used to capture, store, query, analyse and display spatial data. In a GIS, the associated spatial data are independent from the GIS software.

In their paper, R. M. M. Pradeep and N. T. S. Wijesekera highlight the fact that the security issue becomes one of the considerable challenges in spatial decision support systems, especially when the land ownership is involved in the associated changes. Therefore, any intentional or unintentional changes of a spatial data would affect the ownership and this could be advantageous or disadvantageous. Hence, the requirement was to develop a data security mechanism that enables the users to recognize the authenticity in GIS systems using desktop software [66].

In their paper, K. Chaturvedi et al focus on securing spatial data infrastructures for distributed smart city applications and services [67]. They describe Smart Cities as being complex distributed systems which may involve multiple stakeholders, applications, sensors, and Internet of Things (IoT) devices. In order to be able to link and use such heterogeneous data, spatial data infrastructures for Smart Cities can play an important role in establishing interoperability between systems and platforms. Based on the open and international standards of the Open Geospatial Consortium (OGC), the Smart District Data Infrastructure (SDDI) concept integrates different sensors, IoT devices, simulation tools and 3D city models within a common operational framework. However, such

distributed systems, if not secured, may cause a major threat by disclosing sensitive information to untrusted or unauthorized entities.

As the location-based applications are flourishing, R. Guo et al observe that privacy preserving is a critical issue in as far as spatial data is concerned [68]. To protect the confidentiality of the geographic information of individuals, the outsourced data stored at the cloud server should be preserved especially when they are queried. While the problem of secure range query on outsourced encrypted data has been extensively studied, the current schemes are far from the practice in terms of efficiency and scalability.

In their paper, R. Guo et al propose a novel geometrically searchable encryption scheme, MixGeo, by designing a flexible multi-level search structure for spatial data, which can efficiently retrieve points inside a specific geometric range while preserving private data and range queries from a cloud server. Instead of directly executing compute-then-compare operations over dataset, their scheme vastly weakens the connection between the search time and the size of retrieved points to achieve a fast retrieval, efficient updates and stable performance.

O. Kounadi and B. Resch write that Geoprivacy, location privacy, and spatial confidentiality are interrelated terms that explain privacy or confidentiality implications that are associated with “Sensitive” Spatial Data (SSD) [69]. SSD for data types that contain a location attribute which is connected to private or confidential information (e.g. location of an individual at a specific time, heart rate of an individual at a specific location and time, residencies of breast cancer patients). Thus, this kind of data can be

distinguished from other data because they can lead to a variety of disclosures (identity, inferential, or attribute disclosure).

The means to prevent such disclosures comprise protection methods and measures, and other privacy-preserving tasks during a research effort. However, literature review studies in spatial and a-spatial disciplines have shown that over the last 20 years several practitioners have not employed appropriate privacy-preserving measures when publishing their findings. These studies reveal the need to educate the practitioners and to offer them a complete privacy-preserving guidance for research efforts that use sensitive spatial data. In fact, while previous research has mainly focused on methods to preserve privacy and measures to examine information disclosure, limited efforts exist for practical privacy-preserving guidelines regarding the collection, storage, analysis, and dissemination of SSD. Thus in their paper, O. Kounadi and B. Resch present concrete geoprivacy guidelines that can be used for the Sensitive Spatial Data.

According to M. G. Fugini and G. C. Hadjichristofi, spatial data stored in distributed geographic repositories need to be provided according to security requirements [70]. This requires systems storing spatial data to not only prohibit different illegal attacks, but also to maintain information security when attacked and intruded and to provide data confidentiality based on the security requirements of each component when users access information. M. G. Fugini and G. C. Hadjichristofi ‘s paper provides principles of authorization design for spatial data based on RBAC (Role-Based Access Control) for users and security attributes defined for spatial objects. They propose to achieve confidentiality of spatial data repositories with respect to context analysis performed by

inspecting clusters of spatial data for a given purpose (e.g. economic analysis, environment analysis of a territory) while tuning to their security constraints. In order to let users access authorized spatial information, they introduce a selection methodology to filter the required context based on access permissions of users per role and per object [70].

According to D. Chen et al, Location Based Services (LBS) are gaining popularity on smart phones [71]. One fundamental LBS is range search, which returns all Point of Interests (POIs) within a user-specified range. However, people also leave their location privacy at risks when using LBS like range search. How a user can invoke such services without revealing his location is an interesting, yet challenging problem they target to solve. Most existing approaches blur a user's location into a cloaked region, so that LBS cannot figure out the exact location of the requesting user. However, this would make the returning results inaccurate, containing some out-of-range POIs. To this end, D. Chen et al propose a new method to provide location privacy for users of range search [71]. Their method leverages Homomorphic Encryption to let the user encrypt her location, and the LBS server can compute distances on ciphertext. In this way, the returning results by LBS are exactly the POIs within the specified range, while LBS learns nothing about user's real location.

A. Talha et al in their paper state that the increase of spatial data has led organizations to upload their data onto third-party service providers [72]. Cloud computing allows data owners to outsource their databases, eliminating the need for costly storage and computational resources. The main challenge is maintaining data confidentiality with

respect to untrusted parties as well as providing efficient and accurate query results to the authenticated users. A. Talha et al propose a dual transformation scheme on the spatial database to overcome this problem, while the service provider executes queries and returns results to the users. First, their approach utilizes the space-filling Hilbert curve to map each spatial point in the multidimensional space to a one-dimensional space. This space transformation method is easy to compute and preserves the spatial proximity. Next, the order-preserving encryption algorithm is applied to the clustered data. The user issues spatial range queries to the service provider on the encrypted Hilbert index and then uses a secret key to decrypt the query response returned. This allows data protection and reduces the query communication cost between the user and service provider.

While still dwelling on this point of confidentiality as a challenge, L. Kacha and A. Zitouni write that confidentiality refers to data protection from unauthorized access [73]. This challenge occurs when sensitive data is outsourced to the Cloud server. In a decentralized Computing context, the issues of confidentiality are much more important since the server hosting the data does not necessarily belong to the user. Confidentiality in Cloud systems is a major barrier to the user's adoption. Currently, Cloud offers are mainly public and therefore exposed to more attacks, compared to those hosted on private data centers.

In addition, B. R. Pushpa notes that confidentiality prevents unauthorized access of sensitive information [74]. It ensures that the necessary level of secrecy is enforced. Cryptography, which is a technique that transforms plain text into an unreadable format (encrypt) called ciphertext, then converting back the cipher text (decrypt) to plaintext, is

an example of attempts to ensure the confidentiality of data transferred from one computer to another. Cryptography provides a secure transmission protecting the sensitive data traversing across the shared medium.

P. Maniriho and T. Ahmad write that disguising the presence of communication has become a severe concern in this highly digitalized world due to the unauthorized data access and network policy violations that are emerging rapidly [75]. According to P. Maniriho and T. Ahmad, in security systems, information hiding is a broad discipline that encompasses a comprehensive range of several research areas.

M. Modak and R. Shaikh say that while data mining is an important and useful emerging trend, the possibility of it being distributed among various parties raises the issue of privacy [76]. According to X. Liao and C. Shu, nowadays, trust-management is a new security problem which cannot be solved by traditional techniques such as data backup, recovery backup, and firewall but by employing certain data hiding methods [77].

Data anonymization plays a major role in privacy preservation in non-interactive data sharing and releasing process. Data anonymization refers to hiding identity of sensitive data so that the privacy of an individual is effectively preserved even certain aggregate information can be still exposed to data users for diverse analysis and mining tasks [78].

2.3.2 Additional Information on Spatial Data:

According to P. B. Keenana and P. Jankowski, Spatial data are data connected to a location, a place on the earth. Spatial decision-making exploits the geographic relationships within this data to make decisions [79].

In their paper, M. Blistanova, P. Blistan and P. Lošonczi state that data processed in geographically oriented information systems further comprise a spatial component compared to conventional information systems [80]. The spatial component allows evaluating and presenting phenomena not only qualitatively and quantitatively, but in addition the spatial context.

According to X. Ma, spatial data are representations of facts that contain positional values, and geospatial data are spatial data that are about facts happening on the surface of the Earth [67]. Almost everything on the Earth has location properties, so geospatial data and spatial data are regarded as synonyms. Spatial data can be seen almost everywhere in the Big Data deluge, such as social media data stream, traffic control, environmental sensor monitoring, supply chain management, etc. Accordingly, there are various applications of spatial data in the actual world. For example, one may find a preferred restaurant based on the grading results on Twitter. A driver may adjust his route based on the real-time local traffic information. An engineer may identify the best locations for new buildings in an area with regular earthquakes. A forest manager may optimize timber production using data of soil and tree species distribution and considering a few constraints such as the requirement of biodiversity and market price.

In another research paper, P. McKeague et al show the potential value of spatial data in heritage which is currently not being realized [81]. Beyond the fundamental issue of ensuring the long-term preservation of digital data in general, there is a lack of recognition of the value and potential of spatial data held in reports and datasets, notwithstanding the

obvious benefit of standardizing and sharing spatial data for research and to inform environmental policies and activities that may impact the cultural heritage.

2.3.3 Summary of the Gaps in the Literature and how the Proposed Solution Addresses the Gaps.

Table 2.1 provides a summary of the gaps in the literature based on the knowledge gap and some of the solutions proposed by various authors in the articles presented.

Table 2.1: Summary of the gaps in the literature and how the gaps are addressed by the proposed solution

S/N	Gaps in the Literature	Solution
1	Privacy and security concerns over Big data available for aggregation [42]	Propose HE as a practical tool for secure processing of big data sets
2	Security concerns such as authentication, confidentiality and integrity in Electronic Voting Systems [43].	Propose HE to encrypt all the votes and perform the calculation of the votes without revealing any information about it to ensure the security of the votes and maintain the confidentiality.
3	Census data has been a staple of spatial analysis in many countries. No security in the transfer of data from the commercial to the academic sector [61].	Development of frameworks for privacy and ethics in the secondary use of personal data.
4	Spatial privacy vulnerabilities created due to posting Location Based Services (LBS) check-ins in Twitter by using readily available open-source tools for mapping and reverse geo-coding [64]	Provide anonymity to the person accessing Location Based Services and the providers of the services thereof.
5	Spatial data acquired by remote sensing, photogrammetry, geodesy, cartography, services conducting observations and measurements, monitoring, inventory taking, statistical surveys, used for skillfully executed spatial analyses lacks confidentiality [65].	Provide a description of the methods for ensuring the security of databases in SIP/Geographic Information Systems (GIS).

6	The security issue becomes one of the considerable challenges in spatial decision support systems [66]	Proposed to develop a data security mechanism that enables the users to recognize the authenticity in GIS systems using desktop software.
7	Smart Cities as being complex distributed systems which may involve multiple stakeholders, applications, sensors and Internet of Things (IoT) devices if not secured, may cause a major threat by disclosing sensitive information to untrusted or unauthorized entities [67].	Securing spatial data infrastructures for distributed smart city applications and services.
8	The problem of secure range query on outsourced encrypted data has been extensively studied, the current schemes are far from the practice in terms of efficiency and scalability [68].	Propose a novel geometrically searchable encryption scheme, MixGeo, by designing a flexible multi-level search structure for spatial data, which can efficiently retrieve points inside a specific geometric range while preserving private data and range queries from a cloud server.
9	Spatial data stored in distributed geographic repositories lack confidentiality when users access information [70].	Authors propose to achieve confidentiality of spatial data repositories with respect to context analysis performed by inspecting clusters of spatial data for a given purpose (e.g. economic analysis, environment analysis of a territory) while tuning to their security constraints and they introduce a selection methodology to filter the required context based on access permissions of users per role and per object.
10	People also leave their location privacy at risks when using Location Based Services (LBS) like range search [71]	Propose a method that uses Homomorphic Encryption to let the user encrypt her location, and the LBS server can compute distances on ciphertext.
11	The challenge of maintaining data confidentiality with respect to untrusted parties as well as providing efficient and accurate query results to the authenticated users when accessing spatial data [72].	An order-preserving encryption algorithm is applied to the clustered data which allows data protection and reduces the query communication cost between the user and service provider.

2.4 Ring Algorithms in Distributed Systems

B. K.Saraswat et al defines a Distributed System as a gathering of networked computers, which seems as a one large computer where there is no shared memory in the distributed system so they communicate and coordinate their actions with the help of message passing [82].

The Ring Election Algorithm is based on the ring topology with the processes ordered logically and each process knows its successor in a unidirectional way, either clockwise or anticlockwise [83].

According to M. Al-Refai et al, Distributed Systems consist of multiple independent computers that cooperate with each other to perform common tasks that are divided over it [84]. Tasks are distributed on multiple computers to increase the computational speed of problem solving. A group of processes that are communicating through various networks topologies in Distributed Systems requires one process to be a leader to coordinate and control their communications and actions. Any process in a purely distributed system has to communicate with all other processes to take a certain action. The basic idea for reducing the communication complexity is to choose one process from the current alive processes to be a centralized process, which in turn manages all processes communications over the system.

In the same paper by M. Al-Refai et al, Le Lann proposed an algorithm to elect a leader in a ring network, the algorithm assumes the processes are logically ordered and organized in a ring, where each process has a communication link to the next process in the ring in a unidirectional way. When any process detects that the leader is not functioning, it

initiates an election message that contains its ID and sends the message to the next process in the ring. Each process receives the message, puts its ID to the list in the message nominating itself a candidate to be elected as a leader. Finally, the message returns to the initiator process which started it; the initiator process selects the list member with highest ID as the new leader, and a new message known as leader message is circulated once again to announce the new leader and the members of the new ring [84].

S. Basu observes that as we consider distributed systems, some assumptions also need to be made about the communications network [85]. This is very important because nodes communicate only by exchanging messages between them. The following aspects about the reliability of the distributed communications network should be considered.

1. Messages are not lost or altered and are correctly delivered to their destination in a finite amount of time.
2. Messages reach their destination in a finite amount of time, but the time of arrival is not constant.
3. Nodes know the physical layout of all other nodes in the system and know the path to reach each other.

In his paper, S. Basu presents an algorithm for achieving mutual exclusion in Distributed Systems. The proposed algorithm is a betterment of the already existing Token Ring Algorithm, used to handle mutual exclusion in a Distributed System. His proposed algorithm does not allow the circulation of the token along the ring, when there is no need (i.e. when no process wants to enter in its critical section). Loss of a token in the ring can easily be detected, and regeneration of token can be done easily in this algorithm. The

process crash and recovery of crashed process can easily be managed using this algorithm and there is no chance of creation of duplicate tokens in the ring.

According to A. Dadlani et al, in Distributed Systems, nodes communicate with each other using shared memory or via message passing [86]. The key requirement for nodes to execute any distributed task effectively is coordination. In a pure distributed system, there exists no central controlling node that arbitrates decisions and thus, every node has to communicate with the rest of the nodes in the network to make an appropriate decision.

A. Dadlani et al, in Figure 2-6, highlights what happens when a node notices that the leader has crashed, it sends its ID number to its neighboring node in the ring. Thus, it is not necessary for all nodes to send their IDs into the ring. At this moment, the receiving node compares the received ID with its own and forwards whichever is the greatest. This comparison is done by all the nodes such that only the greatest ID remains in the ring. Finally, the greatest ID returns to the initial node. If the received ID equals that of the initial sender, it declares itself as the leader by sending a coordinate message into the ring. It can be observed that this method dramatically reduces the overhead involved in message passing. Thus, if many nodes notice the absence of the leader at the same time, only the message of the node with the greatest ID circulates in the ring thus, preventing smaller IDs from being sent. Note that in Figure 2-6, in part (a) Nodes 2 and 4 notice that the coordinator has crashed simultaneously. In part (b), they send their IDs into the ring. In part (c) and (d), the greatest ID always remains in the ring. In part (e), 5 is declared as the leader.

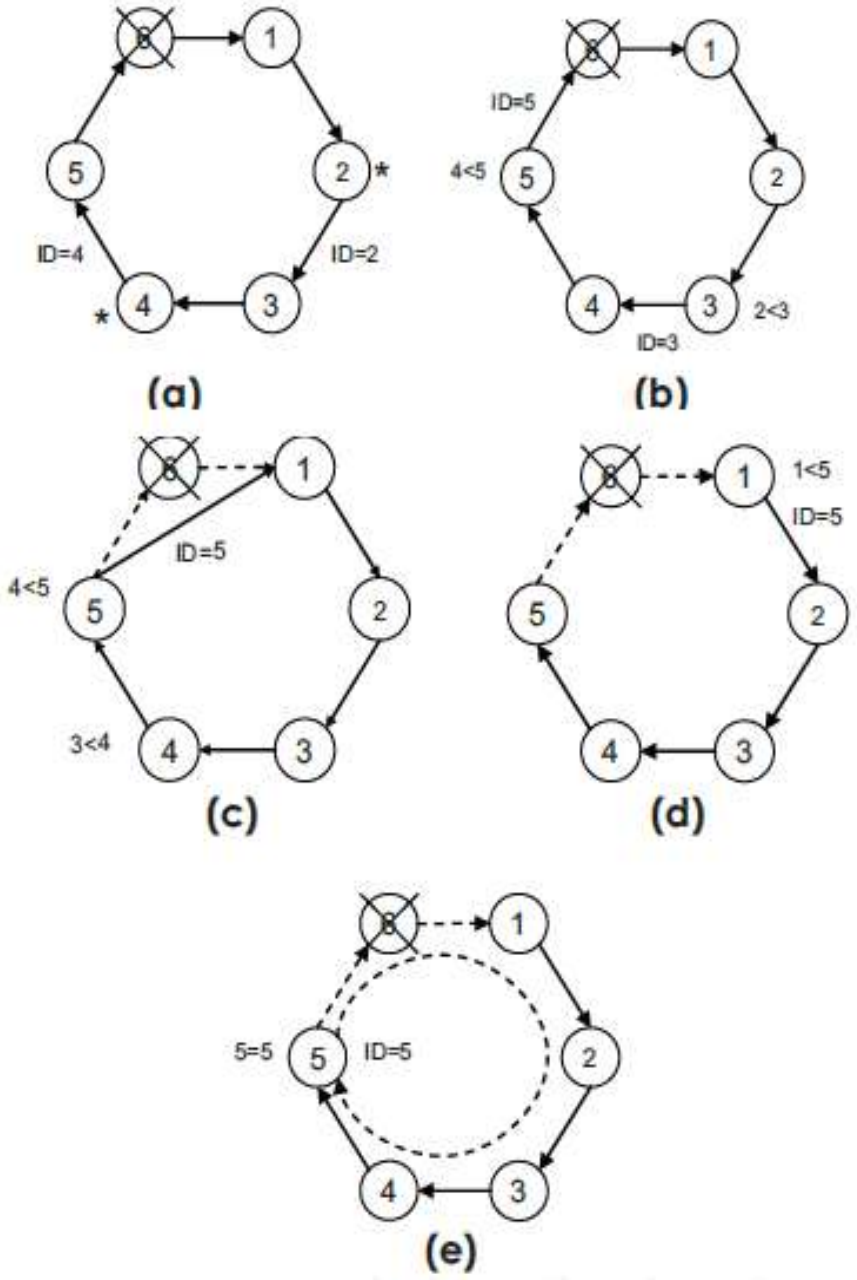

Figure 2-6: The Modified Ring Algorithm -'Adapted From [86]'.

S. Naseera notes in her paper that in Distributed Systems, nodes are connected at different geographical locations [87]. As a part of effective resource utilization, the data and

resources are shared among these nodes. In her paper she discusses the necessity of electing a leader to take care of this resource sharing process by eliminating conflicts among the nodes. As a result, she proposes a new distributed ring algorithm for coordinator election for a Distributed System. The algorithm is an extension of conventional ring algorithm. The performance of the algorithm is tested on a simulated distributed network and performance comparison is made between the conventional and the proposed algorithms. The proposed algorithm is fault tolerant and also exhibits better performance over the conventional ring algorithm. The proposed algorithm elects the new coordinator in a distributed manner and the results shows that the time taken by the proposed technique is less than the time taken by the original ring algorithm.

H. Shaheen writes that a Distributed System is a collection of independent computers, interconnected via a network, capable of collaborating on a task [88]. Among the characteristics of a Distributed System is one that has to do with it presenting a single system image. In Distributed Systems, the entire network will be viewed as a computer. The multiple systems connected to the network will appear as a single system to the user. Thus Distributed Systems hide the complexity of the underlying architecture to the user. This is true because a Distributed System deals with two aspects namely; the hardware and the software. For the hardware, the machines linked in a Distributed System are autonomous while for the software, a Distributed System gives an impression to the users that they are dealing with a single system. The second characteristic is that a Distributed System is easily expandable. This is so because the addition of new computers is hidden from the users as communication is hidden from the users. According to G. A. Qadir and S. R. M. Zeebaree, the Distributed System can easily connect more nodes to it [89]. Rather

than being restricted to only one, devices such as printers can be shared with many nodes. The third characteristic is that a Distributed System is continuously available. This means that a result of a failure in one component can be covered by the other components. M. Steen and A. S. Tanenbaum write that, in any case, an overlay network should in principle always be connected, meaning that between any two nodes there is always a communication path allowing those nodes to route messages from one to the other [90]. The fourth characteristic is a Distributed System is supported by the middleware. To assist the development of distributed applications, distributed systems are often organized to have a separate layer of software that is logically placed on top of the respective operating systems of the computers that are part of the system. This organization is shown in Figure 2-7, leading to what is known as middleware. According A. Z. Kintonova et al, Middleware is the intermediate software between the platform and the actual components of a distributed application [91]. Generally speaking, this level is not mandatory, but its presence is highly desirable. Its purpose is to conceal (disguise) the heterogeneity of platforms and to provide a convenient programming model.

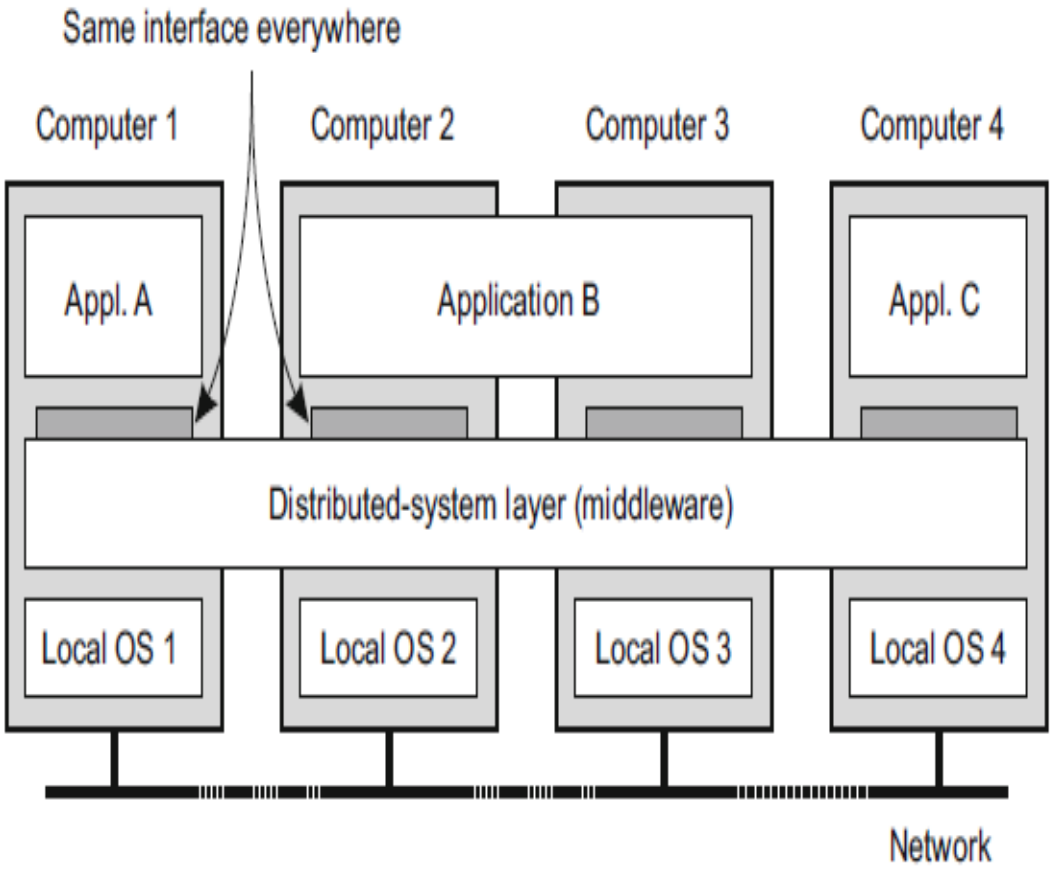

Figure 2-7: A Distributed System Organized as Middleware-'Adapted From [90]'

Figure 2-7 shows four networked computers and three applications, of which application B is distributed across computers 2 and 3. Each application is offered the same interface. Note that the middleware layer extends over multiple machines, and offers each application the same interface. The distributed system provides the means for components of a single distributed application to communicate with each other, but also to let different applications communicate. At the same time, it hides, as best and reasonable as possible, the differences in hardware and operating systems from each application. In a sense, middleware is the same to a distributed system as what an operating system is to a computer: a manager of resources offering its applications to efficiently share and deploy

those resources across a network. Next to resource management, it offers services that can also be found in most operating systems, including facilities for inter-application communication, security services, accounting services, masking of and recovery from failures, etc.

Distributed computing is computing performed in a Distributed System [88]. Distributed computing is widely used due to advancements in machines and faster and cheaper networks.

According to Figure 2-8 below, a Ring Algorithm provides a simplest way to arrange mutual exclusion between N processes without requiring an additional process to arrange them in a logical ring. Each process P_i in Figure 2-8 has a communication channel to the next process in the ring as follows: $P_{(i+1)}$/mod N. The unique token is in form of a message passed from process to process in a single direction clockwise. If a process does not require to enter the Critical Section (CS) when it receives the token, then it immediately forwards the token to its neighbor. A process requires the token, waits until it receives it, but retains it. To exit critical section, the process sends the token onto its neighbor.

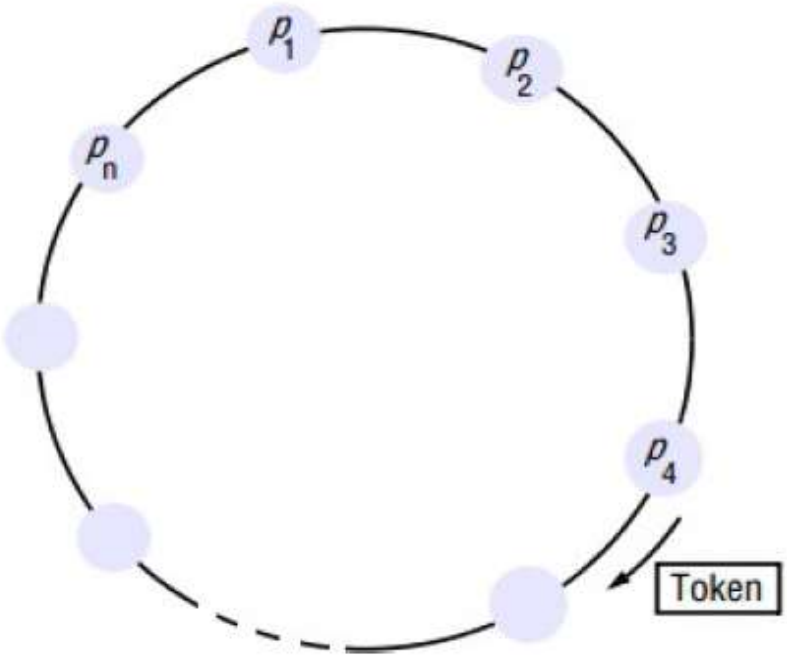

Figure 2-8: Ring Based Algorithm-'Adapted From [88]'.

Among the examples of a Distributed System application is a Web search. The task of a web search engine is to index the entire contents of the World Wide Web. Distributed search is a search engine model in which the tasks of Web crawling, indexing and query processing are distributed among multiple computers and networks. The search engines were supported by a single supercomputer. But in recent years, they have moved to a distributed model. Google search relies upon thousands of computers crawling the web from multiple locations all over the world [88].

Figure 2-9 shows that in Google's distributed search system, each computer is involved in indexing crawls and reviews a portion of the Web, taking a Universal Resource Locator (URL) and sends that information back to a centralized server in compressed format. The centralized server then coordinates that information in a database, along with information

from other computers involved in indexing. When a user types a query into the search field, Google's Domain Name Server (DNS) software relays the query to the most logical cluster of computers, based on factors such as its proximity to the user or how busy it is.

At the recipient cluster, according to Figure 2-9, the Web server software distributes the query to hundreds or thousands of computers to search simultaneously. Hundreds of computers scan the database index to find all relevant records. The index server compiles the results, the document server pulls together the titles and summaries and the page builder creates the search result pages.

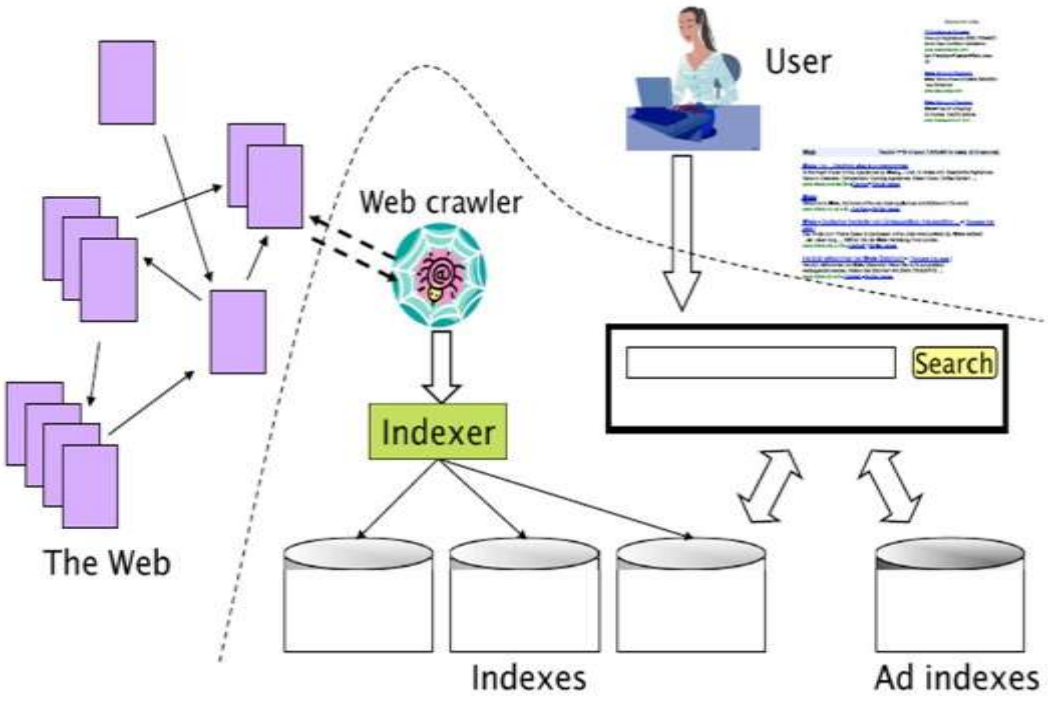

Figure 2-9: Google's Distributed Search System-Adapted From [92]

Summary

A literature review of Homomorphic Encryption and various Homomorphic Encryption Techniques provided insight into how HE works and its wide application in spatial data and helped us come up with an HE technique to make use of in identifying with which technique to employ in designing a protocol for a spatial search. We studied HE from the general knowledge available in the literature to sampling out a few techniques that seem to be readily available for use in the current age. After a general look at HE, we limited our focus on five types of HE from which we would eventually select what HE scheme to make use of in designing the proposed protocol. The five were Rivest, Shamir and Adleman (RSA), El Gamal cryptosystem, Goldwasser-Micali cryptosystem, Benaloh cryptosystem, Paillier cryptosystem and Fully Homomorphic Encryption (FHE). In our review of literature, we also considered Spatial Search and Related Works which reviewed the knowledge gap on which this study is based. The literature review shows that privacy breaches or lack of confidentiality is a common problem experienced by people when accessing or sharing spatial data online. In addition, we reviewed literature on Ring Algorithms in Distributed Systems. Our focus was on how we could incorporate the features/functionality of Ring Algorithms into our proposed protocol. One of the Ring Algorithm features attested to the fact that in distributed communications network, messages are not lost or altered and are correctly delivered to their destination in a finite amount of time. Secondly, messages reach their destination in a finite amount of time, but the time of arrival is variable. Thirdly, nodes know the physical layout of all other nodes in the system and they know the available path to reach each other. One of the key characteristics discussed was that a Distributed System presents a single system image.

This is true because a Distributed System deals with two aspects namely; the hardware and the software. For the hardware, the machines linked in a Distributed System are autonomous while for the software, a Distributed System gives an impression to the users that they are dealing with a single system. The critical point to note is that in Distributed Systems, nodes communicate with each other using shared memory or via message passing.

CHAPTER 3 METHODOLOGY

3.1 Objective 1: Literature Review Method to Identify Cryptosystem to Use

This was done to achieve objective 1 which has to do with identifying an HE technique that supports a spatial search.

In order to have a clear picture of an HE technique to use, a Systematic Literature Review (SLR) which presents an exhaustive summary of the literature was introduced. We followed the well-known guideline of Kitchenham [93], [94]. Figure 3-1 depicts the main phases that can be followed to perform a SLR. In the rest of this section, we explain each phase in detail.

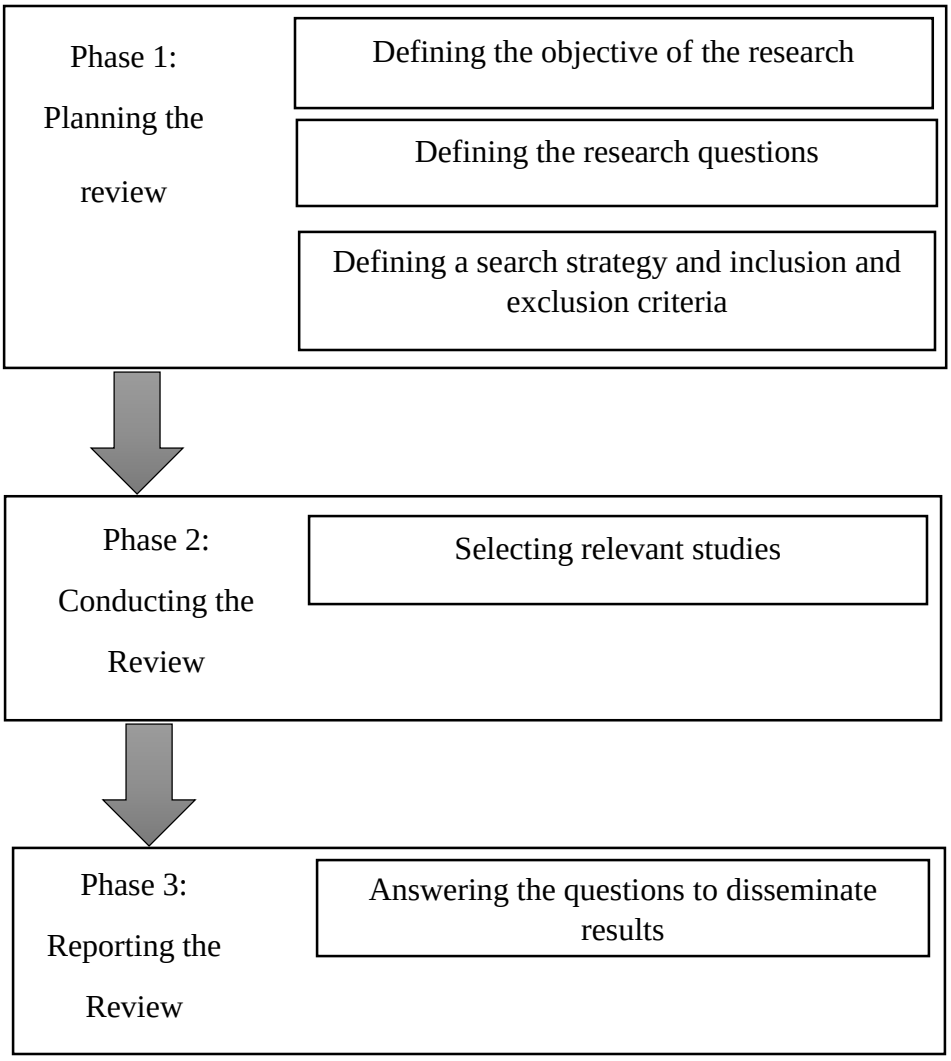

Figure 3-1: Systematic Literature Review Process-'Adapted From [95]'

3.1.1 Phase 1: Planning the Review

The goal of this phase was to investigate the studies that have focused on HE techniques so that the best technique is selected for the initial objective of finding the best technique to use which can support a spatial search among the various HE techniques studied. According to F. Fakhfakh et al, it required developing a review protocol in terms of the

review objective, research questions, search strategy and inclusion and exclusion criteria [95].

3.1.2 Protocol Review Questions

We examined the reviewed publications using the following Protocol Review Questions (PRQs):

PRQ1. Which and how many HE techniques are we going to base our study on?

PRQ2. What kind of properties do these HE technique possess?

PRQ3. Which among those properties can be analysed in comparison with the other HE techniques we have chosen to study?

PRQ4. What factors influence a spatial search?

PRQ5. Which technique among the techniques we have studied would be the best suited in conducting a spatial search to ensure anonymity of data and the data provider on either one side of the equation, that is, the client side or the server side or on both sides of the equation?

PRQ6. What are some of the advantages of this technique that stand out among the rest of the techniques?

3.1.3 Search Strategy

The improper identification of the reviewed articles might lead to inconsistent and inaccurate conclusions. So, recognizing the relevant terms plays a pivotal role in finding the best resources. We conducted an extensive search using the queries: “Homomorphic Encryption Techniques”, “Types of Homomorphic Encryption”, “Factors that influence a spatial search”, “Properties of Paillier Encryption Technique”, “Properties of Benaloh

Encryption", "Properties of RSA", "Advantages of FHE", "Homomorphic Encryption Algorithms", etc.

3.1.4 Inclusion and Exclusion Criteria

Our search was filtered to select only the papers that were written in the English language and published from 2013 to 2019. In addition, we had considered only papers published in journals, conferences and we had discarded other publications such as summaries and presented slides. We were focused on searching in relevant scientific sources (Science Direct, Research Gate, Springer, Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE)). We noticed a significant rise in the number of publications in the last five years on Homomorphic Encryption (HE). This emphasizes the importance of studying the different solutions that come along with various HE schemes to ensure security of data and its users by encryption.

3.1.5 Phase 2: Conducting the Review

A total of 200 papers were found in the first phase. These papers were stored in the INITIAL LIST. To select the relevant studies, we excluded the duplicate papers or the papers that were not clearly related to the protocol review questions. Then, the remaining papers were scored according to some quality criteria such as the quality of writing, the results clarity, the feasibility of the solution and the clearness of the research goal. Based on our exclusion criteria, 56 papers were stored in the FINAL LIST. Some of the findings, results and discussions presented in this dissertation/Thesis were extracted from the articles of the FINAL LIST.

3.1.6 Phase 3: Reporting the Review

In this phase, we used the five protocol review questions that we have already presented to assess each solution in 3.2.2. Based on the solution collected, the first objective of our research was achieved by us identifying the best HE technique that can support a spatial search. Some of the comparisons we came up with can be seen later on in Table 4.1 and Table 4.2 in Chapter 4 which follows this chapter.

3.2 Objective 2: Developing a protocol for distributed spatial searching using the Homomorphic Encryption technique identified in Objective 1.

To achieve objective 2 which was to do with developing a protocol for distributed spatial searching based on the identified Homomorphic Encryption technique, a protocol using the Paillier cryptosystem and distributed algorithm based on the Ring Algorithm (distributed ring algorithm principles) was designed. The protocol developed used the Paillier Encryption Algorithm which was imported as a free library online from libhcs, which is a C library implementing a number of partially Homormophic Encryption Schemes [96]. It must be noted clearly that, however, in this context, an encryption technique cannot be used to develop a protocol but a protocol can apply cryptographic primitives. A protocol describes how the algorithms should be used. The Paillier Encryption was used on both ends of the protocol, that is, on the client side and also on the server side. On the client end, the researcher ‘s attribute value which is a random value was encrypted before it was broadcast to the next machine which represents a data custodian. Encrypted also was the attribute value for the data custodian. Figure 3.1 shows that the one thing in common was that both ends were provided with the public key but

their only difference was that the server side where the data provider lies didn't have a private key while the client side where the researcher or searcher sends requests from had both the public and private keys. In Figure 3.1, R denotes the researcher or searcher on the client side while all the P_i s for i=1,2,3,4,……., n where n ∈ Z denote the data providers.

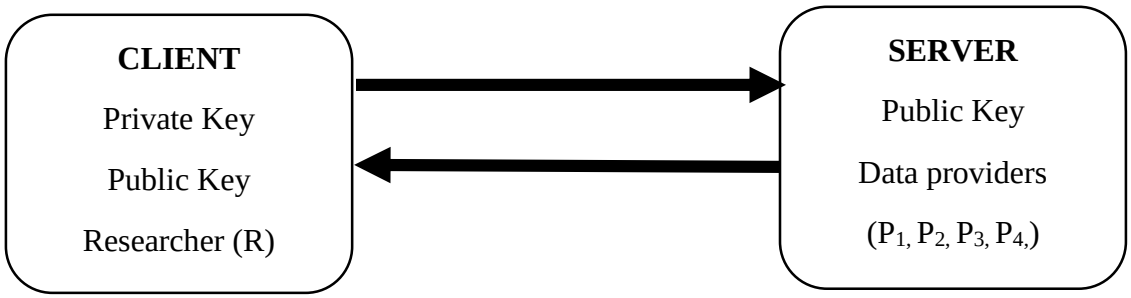

Figure 3.1: The Availability of Encryption/Decryption Keys in the Protocol

The protocol was designed on the basis of one of key properties of Paillier Homomorphic Encryption which states that the Paillier encryption functions are additively homomorphic. It leads to the following property:

The product of two ciphertexts gives the sum of their plaintexts on decryption [48].

$$D(E(m1, r1) \cdot E(m2,r2) \bmod \boldsymbol{n^2}) = (m1 + m2) \bmod n \quad (19)$$

Therefore, in the methodology, we put up steps that will help us to show that we can apply Paillier Encyrption on the client side and also on the server side and allow for computations to be done in the server without decrypting the data. This hypothesis would

then lead us to show whether this design would still be workable by us proving it using Paillier 's additive homomorphic property as stated above.

For this to be practical we had to demonstrate it by using a simple example of computing the mean. In our example, four data custodians namely P_1, P_2, $P_3$and P_4 were used. However, the approach applies in general when there are more than two P_i. Note also that the protocol is a set of steps in general that can be implemented in any programing languages that are available to be used by the Paillier Algorithm. Therefore, the following steps in the protocol were not limited to implementation in Java Programming alone but could be implemented in other programming language as well.

Here are some steps to show how the protocol is used;

Step 1: The researcher generates a set of public and private keys, and broadcasts its public key to all the data custodians, namely P_1, P_2, P_3, P_4, etc.

For key generation, the following was considered by the algorithm;

KeyGen Algorithm: For large primes p and q such that gcd (pq, (p − 1) (q − 1)) = 1, compute n = pq and λ = lcm (p − 1, q − 1). Then, select a random integer g ∈ $Z_{n^2}^*$ by checking whether gcd (L($g^{\lambda} \bmod n^2$),n)=1, where the function L is defined as L(u) = (u−1)/n for every u from the subgroup $Z_{n^2}^*$ which is a multiplicative subgroup of integers modulo n^2 instead of n.

Pick two large primes p and q and let n=pq. Let λ denote the Carmichael function, that is $\lambda(n) = lcm\,(p-1, q-1)$. Pick random g∈$Z_{n^2}^*$ such that

$L(g^{\lambda} \, mod \, n^2) \, is \, invertible \, modulo \, n$ ($where \, L(u) = \frac{u-1}{n}$). n and g are public; p and q (or λ) are private.

Step 2: The Researcher or searcher encrypts its value attribute R which becomes E(R). This encryption process is achieved by the following equation which referenced in the literature review already;

For plaintext x and resulting ciphertext y, select a random r $\in$ Z_n^*. Then,

e_k(x, r) = $g^m . r^n \, mod \, n^2$=y, where e_k is the encryption function or denoted as E in our general representation under step 2.

Step 3: The Researcher sends its encrypted value E(R) to P_1.

Step 4: P_1 encrypts its value using the public key received from R initially and multiplies it value with encrypted value received from R, that is, E(R) X E(P_1). Then it sends E(R) $\otimes$ E(P_1) to P_2. Note that the computations were to be done on encrypted data.

Step 5: P_2 encrypts its attribute value, applies computations on received encrypted value attribute and sends E(R) $\otimes$ E(P_1) $\otimes$ E(P_2) to P_3.

Step 6: P_3 encrypts its attribute value, applies computations on received encrypted value attribute and sends E(R) $\otimes$ E(P_1) $\otimes$ E(P_2) $\otimes$ E(P_3) to P_4.

Step 7: P_4 encrypts its attribute value, applies computations on received encrypted value attribute and sends E(R) $\otimes$ E(P_1) $\otimes$ E(P_2) $\otimes$ E(P_3) $\otimes$ E(P_4) to R.

Step 8: The researcher subtracts the random number R from the value obtained after decrypting the received encrypted value to retrieve the sum.

The decryption is done by using the following equation which was referenced in the literature review too.

$$d_k(\text{y}) = \frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)}.\text{mod n}$$

Finally, by dividing by n the researcher retrieves the mean value of the selected attribute.

The assumption we used in designing the protocol would be meaningful if we showed that the average obtained when doing computations on plaintext would be the same average obtained when doing computations on the encrypted text.

3.3 Objective 3: Developing a proof of concept prototype using the proposed protocol.

To achieve objective 3 which was to do with developing a proof of concept prototype using the proposed protocol, a prototype implemented as a distributed application was written in Java using the proposed protocol.

Note that in our discussion under literature review on Ring Algorithms in Distributed Systems, we established the following facts. Firstly, among the characteristics of a Distributed System is one that has to do with it presenting a single system image. In Distributed Systems, the entire network will be viewed as a computer. This is true because a Distributed System deals with two aspects namely; the hardware and the software. For the hardware, the machines linked in a Distributed System are autonomous while for the

software, a Distributed System gives an impression to the users that they are dealing with a single system. Therefore, we were able to work with two computers and yet established a number of participating nodes in the network. The other key principle in our discussion is that a computing element, which we will generally refer to as a node, can be either a hardware device or a software process [90]. A second element is that users (be they people or applications) believe they are dealing with a single system. So it was a number of nodes in terms of software processes that were generated by the use of two computers that formed a ring and not necessarily the two computers. This discussion helps to deal with the issue of distinguishing whether this was based on a peer to peer topology or a Distributed System. Yet in reality it was based on the later. While two computers cannot form a Ring in a peer to peer topology, two computers can form a Ring in a Distributed System based on the many number of software processes the two computers are able to generate which eventually appear as a single system.

Three experiments were conducted in Java using two computers that were acting as two ends, i.e. the client end and the server end. The first one experiment involved doing the spatial search without the proposed protocol. Key examples of the kind of spatial data the search is based on are data that can be aggregated from Secure Geocoding and also data that can be aggregated from cancer registries to calculate an average number of cases per geographic boundary. In short, it is statistical data. Therefore, in the experiments run in Java, we used integers that were converted to BigInteger values since the encryption algorithm uses BigInteger values in Java. The encrypted data was being transmitted as an object. In this one, the Java program was executed without having to send it to the next machine or computer.

The second one involved conducting the spatial search using the protocol with Paillier Encryption. In this one, using Java, a request was sent from one node to another as discussed in 3.3 where the random value for the client end is encrypted and the attribute value for server side is also encrypted so that the server side sends back an encrypted value to the client side to demonstrate what happens in a Ring algorithm. Recall that one of the key principles in our discussion is that a computing element, which we will generally refer to as a node, can be either a hardware device or a software process. As a result, it was a number of nodes in terms of software processes that were generated by the use of two computers that formed a ring and not necessarily the two computers. The application during the implementation was coded in such a way that it could generate an n number of nodes to participate in one Distributed System. Note that to discover the next node to send the encrypted value to, each node participating in the distributed computation checks its list of participating nodes to find the next active node.

The third option had to do with running it with a protocol without Paillier encryption at all. In this one we ensured that both ends didn't use the Paillier Encryption Algorithm at all. The researcher sent a value as a request that was not encrypted and vice versa the recipient could neither encrypt its value nor send back an encrypted sum. This was done to calculate the amount of resources in terms of processing time the protocol without encryption consumes.

3.4 Important Implementation Details of the Prototype

3.4.1 Step 1: Creation and Implementation of Java Application

The Java application was developed from the concept of Java Networking which involves connecting two or more computing devices together so that we can share resources. In our implementation, we used two computers. One acted as a client while the other was a server. In the server side, using principles of distributed systems, the programs was created in such a way that it generated n number of processes where n > 1. Note that in a distributed system a node can assume the role of a process in a single system image as we noted in the literature review under Distributed Systems. Therefore, it was the number of processes that formed a Ring Algorithm and not necessarily the two computers.

Since a client in socket programming must know the IP Address of Server, and Port number, we created a configuration file for each experiment that was to be done with experiment No.1 having three port numbers, experiment No. 2 with 4 port numbers, experiment number 3 with five port numbers and so and so forth. For example, for experiment no.1, we have the following;

IPAddress=127.0.0.1&PortNumber=4141

IPAddress=127.0.0.1&PortNumber=4142

IPAddress=127.0.0.1&PortNumber=4143

For experiment no.2, we have

IPAddress=127.0.0.1&PortNumber=4141

IPAddress=127.0.0.1&PortNumber=4142

IPAddress=127.0.0.1&PortNumber=4143

IPAddress=127.0.0.1&PortNumber=4144

For experiment No. 3, we have

IPAddress=127.0.0.1&PortNumber=4141

IPAddress=127.0.0.1&PortNumber=4142

IPAddress=127.0.0.1&PortNumber=4143

IPAddress=127.0.0.1&PortNumber=4144

IPAddress=127.0.0.1&PortNumber=4145

For experiment No. 1, to be executed, it meant that we had to consider the sender sitting on Port Number 4141 having to send requests to Port Number 4142 and Port Number 4143. The request was sent in encrypted form and as the data was sent from one port number to another and finally to the sender, it was still in its encrypted form. The port where the sender is located is where the decryption happens from. The program was designed in such a way as to calculate an average of encrypted values that were sent from the server after the final encrypted value was sent back to the sender (who represents the searcher). For, instance for experiment No.1 which was based on two nodes or processes, the average would be calculated based on two entries, that is, a and b where a, b$\in\mathbb{R}$ ($\mathbb{R}$ represents the set of all real numbers). For experiment No. 2, which was based on three nodes or processes, the average would be calculated based on three entries, that is, a, b and c. For experiment No.3, then there would four processes and four entries and so on

and so forth. This will be shown clearly in the next chapter under Table 4.1.1, Table 4.1.2, Table 4.1.3 and Table 4.1.4 respectively.

This process was repeated for more than ten experiments. It was these ten or more experiments that were used in the validation of results as discussed in step 2 below.

3.4.2 Step 2: How the Results Were Validated

Experiments were conducted to calculate an average based on the property of Paillier which states that the product of encrypted values gives us the sum of two plaintexts upon decryption. Our interest was to calculate not only the average value after decryption to prove that this property of Paillier was obeyed but we also calculated the average running time of the implementation of the whole protocol. This meant that for each experiment done, for example, experiment 1, we had to run it for ten times to obtain 10 different values of the run time in milliseconds from which sum an average was calculated. These details are shown in the next chapter under results in Table 4.1.1, Table 4.1.2, Table 4.1.3 and Table 4.1.4 respectively. From these results in the table, a graph was plotted. Figure 4-2 shows the graph that was obtained by using the tables of results as displayed in the next chapter. Using the graphical trend of results for ten experiments, it was easy for us to see that a similar trend would be obtained for 100, 1000, 10,000 experiments and so and so forth. The results of this graph helped us to show that the proposed protocol is scalable.

3.5 Ethical Consideration

The Paillier Encryption Algorithm (cryptosystem library) used during this study was obtained from an open source library which did not demand ethical clearance

before use [96]. Encryption Algorithms are either imported or bought at a price. When working with cryptosystem in any programming environment, we have to either buy a cryptosystem or use a free software. We didn't develop the Paillier Cryptosystem. We only used the existing one. The protocol designed used Paillier Cryptosystem and the technical aspect came when implementing the protocol using Java Programming. The use of the said software or a customized bundle of code/program did not infringe on any human participant because there was no human factor in the interpretation and use of the software. Nevertheless, the study was based on the Directorate of Research and Graduate Studies (DRGS) Ethical committee guidelines. Ethical clearance was obtained from the University of Zambia Postgraduate Studies Ethical committee and the reference number is **NASREC-2019 DEC-001**.

Summary

To guide the research, three objectives were set out as given in section 1.4. The research begun with a study of pre-existing HE techniques. Firstly, we identified a Homomorphic Encryption technique that can work best for a spatial search by reviewing literature on Homomorphic Encryption techniques. Among the Homomorphic Encryption techniques reviewed were Rivest, Shamir and Adleman (RSA), El Gamal cryptosystem, Goldwasser-Micali cryptosystem, Benaloh cryptosystem, Paillier cryptosystem and Fully Homomorphic Encryption (FHE).

Before we came to choosing the best technique to make use of in our research, we also looked at the factors that influence a spatial search using literature review. While taking note of these factors, we analysed the techniques according to their various properties to come up with the best technique to use in a spatial search.

Then, a protocol using distributed algorithm based on the Ring Algorithm coupled with the use of Paillier Homomorphic Encryption was designed.

Finally, a proof of concept prototype using the proposed approach was implemented by conducting some experiments using Java. The first experiment involved doing the spatial search without the proposed protocol, the second one being doing the spatial search using the protocol with Paillier Encryption while the third option involved running it with a protocol without Paillier encryption at all.

Table 3.1 Summarises the methodology in relation to the objectives and research questions.

Table 3.1: Summary Table of Objectives, Research Questions and Methodology

Item No.	Objective	Research Question	Methodology
1	To identify a Homomorphic Encryption technique that can support a spatial search.	Which one among the preexisting Homomorphic Encryption techniques can support a spatial search?	Existing literature on Homomorphic Encryption was reviewed. Analysed encryption techniques. Then the best approach for use in a secure distributed spatial search was picked.
2	To develop a protocol for distributed spatial searching based on the identified Homomorphic Encryption technique in "1.".	How can a protocol be designed that uses the cryptographic primitives of the identified Homomorphic Encryption technique in "1."?	A protocol was designed that applies the cryptographic primitives of the Paillier cryptosystem and Distributed Ring Algorithm principles.
3	To develop a proof of concept prototype using the proposed protocol in "2."	What proof of concept prototype can be developed using the proposed protocol in "2."?	A prototype implemented as a distributed application was written in Java using the proposed protocol. Some experiments were done, the first one being doing the spatial search without the proposed protocol, the second one being doing the spatial search using the protocol with Paillier Encryption and the third option of running it with a protocol without Paillier encryption at all.

CHAPTER 4 RESULTS

4.1 Objective 1: Selection of the cryptosystem to use

The selection of the cryptosystem to use was based on performance analysis selected based on functional requirements and non-functional requirements. A functional requirement describes a functional behavior that a system or system component should be able to perform. In other words, functional requirements specify what the system should do; an activity that the system must do to provide its users with the required functionality [97]. Another definition for the Functional Requirement (FR) is a requirement that specifies an action that a system must be able to perform, without considering physical constraints; a requirement that specifies input/output behavior of a system [98]. The FR describes the user functionalities of the future system, and general conditions specify restrictions or design decisions to be observed in the development. A function is nothing but inputs to the software system, its behavior and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behaviour of the system.

On the other hand, non-functional requirements, express how good a software system must work. It has been widely acknowledged that a quality attribute such as reliability, security, performance, or usability is a non-functional requirement of a software system. That is why non-functional requirements are sometimes called quality attributes or quality

requirements. Non-functional requirements relate to properties or qualities that the software system must have when performing one or some functions. A non-functional requirement defines the performance attribute of a software system. It judges the software system based on responsiveness, usability, security, portability and other non-functional standards that are critical to the success of the software system. How fast a website is able to load is an example of nonfunctional requirement. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. H. Kaur and A. Sharma writes that a non-functional requirement (NFR) is defined as a software requirement that describes not what the software will do, but how the software will do it, for example, software performance requirements, software external interface requirements, design constraints, and software quality attributes [99]. Non-functional requirements are difficult to test; therefore, they are usually evaluated subjectively. It has been also acknowledged that the achievement of non-functional requirements along with functional requirements is critical to the success of a software system. Furthermore, to achieve a high quality software system, both FR and NFR should be addressed. Therefore, in our study both types of requirements were taken into confederation.

After going through the literature on various HE techniques, an analysis on the factors that influence a spatial search was done through literature review. Here are some of the factors that were identified.

The most popular class of such services is k-nearest neighbor (kNN) queries where users search for geographical points of interests (e.g., restaurants, hospitals) and the corresponding directions and travel-times to these locations. Accordingly, numerous

algorithms have been developed to efficiently compute the distance and route between objects in large road networks [100].

In the query processing of spatial-keyword search, indexing techniques for both text and geographic data are used.

In a GeoSN, a variety of spatial objects (e.g. restaurants, hotels, businesses) are marked on the map and annotated with user generated tags. GeoSN users can search for interesting spatial objects, and share information about their location and activities [101]. More importantly, users with similar interests can plan for social activities collaboratively, such as going somewhere for dining and shopping, or taking a cycling tour together. To make such plans, it is essential to identify a group of spatial objects, such as restaurants, shops and parks, which can maximally satisfy the users' needs.

Another factor that influences a spatial search is scalability in terms of dataset size. In order to simulate the real geo-social networking in which the number of objects and tags continuously increase, J. Zhong, X. Meng, X. Zhou and D. Liu [101] conducted a set of experiments to evaluate the scalability of three algorithms by varying the number of objects.

A summarization of this discussion is presented in Table 4.1.

Table 4.1: Factors That Influence a Spatial Search

S/N	FACTORS THAT INFLUENCE A SPATIAL SEARCH
1.	Point Data (Longitude/Latitude)
2.	Geocoding-When there are no coordinates, the search can still be done by ggmap package
3.	Search Algorithms -such as the kNN (k-nearest neighbor)
4.	Time Dependent-how fast and how slow after running query
5.	Query Length-Handling queries of various length
6.	Scalability.

The Paillier cryptosystem was selected for distributed spatial search for the following reasons:

a) It has a smaller expansion rate and lower cost of encryption and decryption than FHE and better security than RSA, El Gamal, Goldwasser-Micali and Benaloh. To ensure that the system remains protected and secure, the lower bound of this expansion rate should be four. Improved schemes have developed with the expansion factor being lowered to increase efficiency. For instance, Paillier cryptosystem allowed efficient decryption by enabling encryption of many bits during a single calculation with a better expansion rate of two [4].

b) It is computationally cheaper and can be used in practice.

c) Paillier's scheme is the most efficient among currently known additively homomorphic schemes, i.e. it requires simple operations in the encryption, decryption and addition procedures and so achieves high performance.

d) Calculations necessary to reconstruct the original message can be performed on the encrypted data even without having access to the private key.

e) Analyses can be conducted on encrypted data with potentially little if any risk of revealing confidential information [60].

f) It has a self-blinding property which property allows mapping a plaintext into possibly many different ciphertexts and the same plaintexts cannot be recognized from their ciphertexts [102].

g) Since Paillier encryption is probabilistic, the encrypted files on the different peers are not linkable to each other for anyone not knowing the private decryption key [103].

h) Paillier Encryption is probabilistic in that its encryption algorithm uses pseudorandom number generators. Hence, encrypting the same message several times will produce different ciphertexts, making it very cumbersome for even an informed adversary to compare encrypted messages in order to ascertain the original value that was encrypted [60].

The discussion presented is high level. Details can be found in the literature cited. A summarization of this discussion is presented in Table 4.2.

Here is a brief account on the meaning of the subheading, Security Assumption,

Computational Cost, Probabilistic Nature, Expansion Factor, Encryption Cost and Decryption Cost as identified in Table 4.2 below. In Table 4.2, security assumption is a phrase used to describe the security of the most practical homomorphic encryption schemes. The security of these schemes is based on the Ring-Learning With Errors (RLWE) problem, which is a hard mathematical problem related to high-dimensional lattices. Namely, the security assumption of these encryption schemes states that if the scheme can be broken efficiently, then the RLWE problem can be solved efficiently. For example, Paillier Encryption is secure under the assumption of the Decisional Composite Residuosity Assumption (DCRA) [4]. DRCA states that for given integers N ∈ Z and x ∈ $Z_{n^2}^*$, it is "hard" to decide whether there exists y ∈ $Z_{n^2}^*$ such that x ≡ y^n mod N. Here, the notation Z_n^* denotes the set of integers modulo N for all N ∈ N [24]. A long line of peer-reviewed research confirming the hardness of the RLWE problem gives us confidence that these schemes are indeed at least as secure as any standardized encryption scheme.

Computation cost is the total time taken for the algorithm to run which can be measured in milliseconds, seconds, minutes, or even hours. X. Wang, T. Luo and J. Li write that computational cost is the amount of time an algorithm takes to complete all additions and multiplications [104]. Computational cost associated with the features can be measured in computing power, memory occupation and Central Processing Unit (CPU) time. It is common that a more computationally expensive feature usually provides more discrimination power in classifying ambiguous cases than a cheap feature.

Encryption procedure is a set of steps of converting a message from readable form into unreadable form. In other words, it is a process used to convert a plaintext into a ciphertext

taking into account that without a secret key, no unauthorized users can access the original message [105]. Therefore, encryption cost is the time taken for the algorithm to complete the encryption procedure. Similarly, decryption procedure is the process of taking encrypted or encoded text then, converting it back into text that you can read and understand. Basically, decryption is the inverse process of encryption. Therefore, decryption cost is the time taken for the HE algorithms in our context to complete the decryption procedure. However, it must be noted that the difference between computational cost and encryption/decryption cost is that for computational cost it involves the whole length of time the algorithm runs from step A to step Z while for encryption/decryption involves the total time the algorithm takes to do finish the encryption or decryption process alone respectively, either of which may be subset of the whole computational process (computational cost).

The probabilistic encryption nature is a property of encryption which for every encryption key, a given plaintext will be encrypted into a different ciphertext because the algorithm uses pseudorandom number generators, for example, Paillier Encryption is probabilistic, whereas almost all the well-known cryptosystems are deterministic. Deterministic simply means that for a fixed encryption key, a given plaintext will always be encrypted into the same ciphertext under these systems. However, this may lead to some security problems. RSA scheme is a good example for explaining this point.

According to J. Sen, expansion factor is defined as the ratio of the length of the ciphertext and the corresponding plaintext (in bits) [54]. The value of this parameter is of paramount importance in determining security and efficiency tradeoff of a probabilistic encryption

scheme. In Paillier's scheme, an efficient probabilistic encryption mechanism has been proposed with the value of expansion less than 2.

In addition to this, Table 4.2 shows that the security standard of Paillier is relatively higher than that of RSA, El Gamal, Goldwasser-Micali and Benaloh because it has a higher encryption and decryption costs than these encryption algorithms [7]. However, its security standard is relatively lower than FHE since its encryption and decryption costs are lower than FHE. The higher the encryption and decryption costs the better the security since it is hard and takes more time to break the algorithm. Therefore, computational costs, encryption and decryption costs are thus key to the selection of a good algorithm to use. FHE may have a better encryption and decryption costs in terms of security strength but it has a higher computational costs due to complex mathematical computations created by its ability to perform both multiplications and additions. Hence, Paillier Encryption was preferred in the choice of the best algorithm to use in the designing of our protocol for a spatial search for it has a higher security standard than RSA, El Gamal, Goldwasser-Micali and Benaloh and also a better computational cost than FHE.

Table 4.2: Security Assumption of the HE Schemes and Comparisons on Properties.

HE SCHEME	**HOMOMORPHIC NATURE**	**SECURITY ASSUMPTION**	**PROPERTIES OF THE SCHEMES**				
			COMPUTATIONAL COST	***PROBABILISTIC NATURE***	***EXPANSION FACTOR***	***ENCRYPTION COST***	***DECRYTION COST***
RSA	Multiplicative	Integer factorization problem [105].	High [55]	Deterministic	3	Low	Low
El Gamal	Multiplicative	Diffi-Hellman problem [52]	Low	√	3	Low [7]	Low [7]
Goldwasser-Micali	Additive	Quadratic residuosity problem [56]	Low	√	3	Low	High (heavier) [54]
Benaloh	Additive	Higher residuosity problem [57]	Low	√	3	Low	Complex
FHE	Additive & Multiplicative	Sparse Subset Sum (SSSP) assumption [54]	Higher [58]	√	Complex or >3	Complex/Higher	More complex (Higher)
Paillier	Additive	Decisional Composite Residuosity Assumption (DCRA) [52]	Low	√	2 [54]	High [7]	High

4.2 Objective 2: Developing a protocol for distributed spatial searching using Paillier HE which was identified in Objective 1-How the protocol works

The result for Objective 2 were based on the designing of the protocol and showing how the protocol works.

To illustrate how our platform works, we demonstrated by using a simple example of computing the mean. The example used four data custodians namely P_1, P_2, $P_3$and P_4, but the approach applies in general when there are more than two P_i.

1. The client end had a searcher of data or a researcher.
2. The researcher or person doing the spatial search issued an online request for particular data by sending an encrypted random number which can be symbolic for a particular search request. He also sent a public key to all the locations identified, i.e. P_1, P_2, P_3 and P_4 when we consider the four distributed points as per the four locations. Hence the value of n = 4.
3. The request was then sent as an encrypted random value to the first point P_1.
4. P_1 received the encrypted value of the request, i.e. E(R).
5. P_1 encrypted its attribute value and sent $E(R) \otimes E(P_1)$ to P_2.
6. P_2 encrypted its attribute value and sent $E(R) \otimes E(P_1) \otimes E(P_2)$ to P_3.
7. P_3 encrypted its attribute value and sent $E(R) \otimes E(P_1) \otimes E(P_2) \otimes E(P_3)$ to P_4.
8. P_4 encrypted its attribute value and sent $E(R) \otimes E(P_1) \otimes E(P_2) \otimes E(P_3) \otimes E(P_4)$ to R.

9. The researcher subtracted the random number R from the value obtained after decrypting the received encrypted value to retrieve the sum.

Finally, by dividing by n the researcher retrieved the mean value of the selected attribute. This whole process is illustrated in Figure 4-1.

To discover the next machine to send the encrypted value to, each machine participating in the distributed computation checked its list of participating machines to find the next active machine. Extensive detail about this mechanism can be found in literature on Ring Algorithms in Distributed Systems under section 2.5.

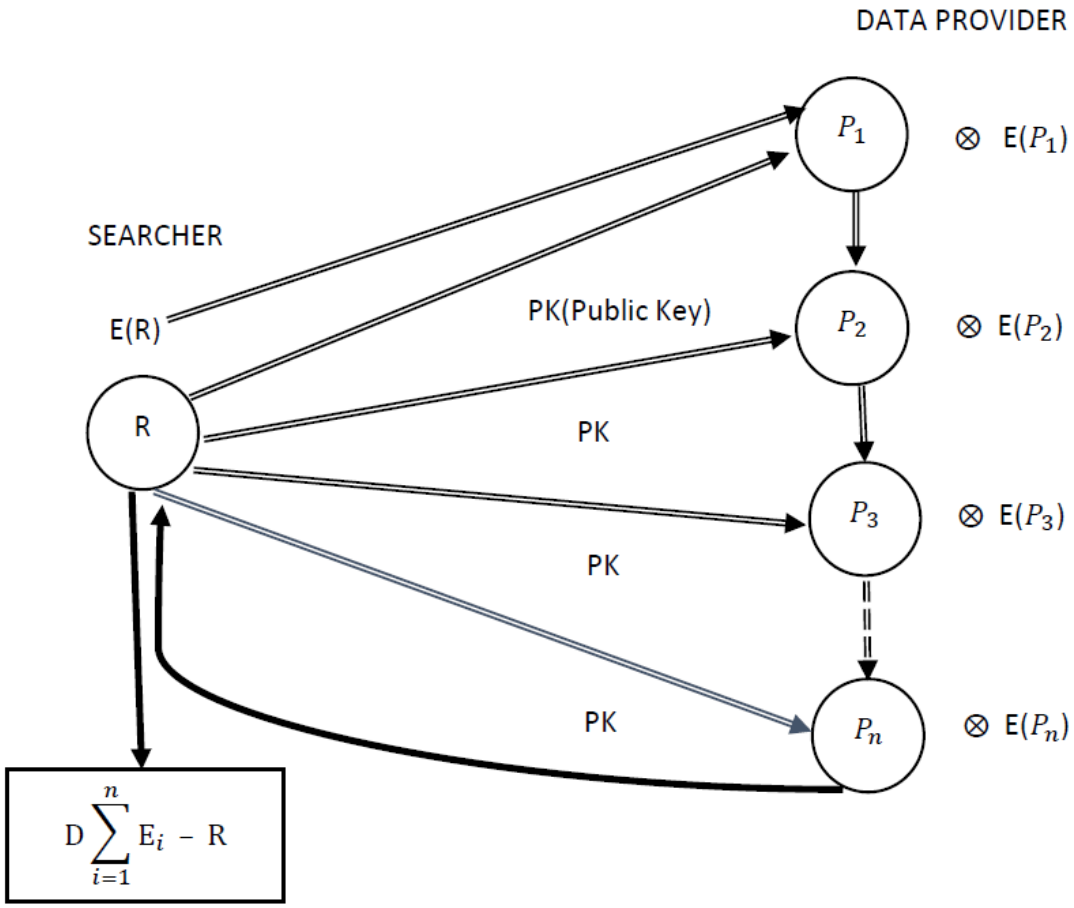

Figure 4-1: How the proposed protocol works that uses a distributed ring algorithm

In the demonstration shown in Figure 4-1, Paillier Encryption scheme is applied on both the searcher end and data provider end to ensure anonymity on both sides. The only key which is broadcast to everyone (data providers) by the researcher (client) is the Public Key.

The equation that was used to send requests to the data providers by the searcher is derived from Paillier's properties which states that the Paillier encryption functions are additively homomorphic. It leads to the following property which referenced as equation (19) in the literature review: The product of two ciphertexts gives the sum of their plaintexts on decryption and is as shown below;

D(E(m1, r1) . E(m2,r2) mod $\boldsymbol{n^2}$) = (m1 + m2)mod n

4.3 Objective 3: Developing a proof of concept using the proposed protocol.

Furthermore, a proof of concept prototype was implemented using the java programming language based on the proposed protocol. We demonstrated by using the BigInteger values in Java how large mathematical computations can be carried out to transfer encrypted values, through a distributed system, that are sent as requests without decrypting them since we ensured that both ends didn't have the secret key. The fact that the Paillier Cryptosystem uses modulo arithmetic (mathematical calculations involving functions that return the remainder) and BigInteger values in Java makes it complicated and impossible for any third party or intruder to decrypt and reach the mean value of the selected attribute. The BigInteger class refers to a Java class used for mathematical operation which involves very big integer calculations that are outside the limit of all available

primitive data types [106]. For instance, factorial of 100 contains 158 digits in it which can't be stored in any available primitive data type in Java. Note also that BigInteger is one word and should not be separated as in the context of the English Language or grammar.

Another important emphasis to make is that Homomorphic Encryption, particularly Paillier in this case, seeks to aid in this encryption process by allowing specific types of computations to be carried out on ciphertext which produces an encrypted result which is also in ciphertext [23]. Its outcome is the result of operations performed on the plaintext. Case in point, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers.

Table 4.3 shows the analysis of performance when we consider three scenarios as we did the research, i.e. the first one being doing the spatial search without the proposed protocol, the second one being doing the spatial search using the protocol with Paillier Encryption and the third option of running it with a protocol without Paillier encryption at all. Table 4.3 shows the average run time in milliseconds calculated after executing each experiment for ten (10) different runs. The byte size of the data for Table 4.3 is 16 bytes. From the results, it can be concluded that the encryption and protocol result in an increased running time. It can, however, be argued that they collectively introduce an overhead of 39.7% in processing time. The bigger expense comes from the network communication overhead. The overhead is calculated by the following formula;

$$\text{OVERHEAD \%} = \frac{B-A}{B} \text{ X } 100\%, \quad (20)$$

$$= \frac{981-702}{702} \text{ X } 100\% = 39.7\%.$$

where A=Average run time for spatial search without protocol without Paillier Encryption and

B = Average run time for spatial search with protocol with Paillier Encryption as shown in Table 4.3. Note that the third experiment which was to do with conducting a spatial search with protocol without Paillier Encryption was done with a view to calculating the running time for the protocol without the encryption algorithm applied on it.

Table 4.3: Analysis of Performance During a Spatial Search.

DESCRIPTION	**AVERAGE RUN (EXECUTION TIME)-JAVA PROGRAM**	**ANONYMITY/SECURITY**
i. Spatial search without protocol without Paillier Encryption	702 milliseconds	NIL
ii. Spatial search with protocol with Paillier Encryption	981 milliseconds	Guaranteed
iii. Spatial search with protocol without Paillier Encryption	767 milliseconds	NIL

4.3.1 Validity Test

In addition to these results in general, a validity test was done by doing experiments to calculate an average using anonymous data from a varying number of participating nodes. Experiments carried out involved running the proposed protocol in Java using n number of processes where n > 1. Data passed between nodes was encrypted in nature. It was only the result which was decrypted. More than 10 experiments were done. Experiment No. 1 had 2 participating nodes, experiment no. 2 had 3 participating nodes and experiment No. 3 had 4 nodes, experiment No. 4 had 5 nodes, etc. As explained in the methodology, it was a number of nodes in terms of software processes that were generated by the use of two computers that formed a ring and not necessarily the two computers. For each experiment, it was run for ten times to obtain the run time in milliseconds for the time taken to run the Java application as shown in Table 4.1.1, Table 4.1.2, Table 4.1.3, Table 4.1.4, etc. below. Here are the tables for Experiment No1., Experiment No.2, Experiment No. 3, Experiment No 4., respectively;

Table 4.1.1: Results for Experiment No.1

S/N	EXPT 1: INPUT VALUES	RESULTS FOR 10 RUNS (RUN-TIME IN MILLISECONDS)	NUMBER OF PARTICIPATING NODES
1	10	681	2
2	20	679	
3		674	
4		776	
5		663	
6		740	
7		717	
8		696	
9		697	
10		700	
AVERAGE	**15**	**702**	

Table 4.1.2: Results for Experiment No.2

S/N	EXPT 2: INPUT VALUES	RESULTS FOR 10 RUNS (RUN-TIME IN MILLISECONDS)	NUMBER OF PARTICIPATING NODES
1	10	794	3
2	20	802	
3	30	799	
4		879	
5		820	
6		819	
7		805	
8		791	
9		810	
10		804	
AVERAGE	**20**	**812**	

Table 4.1.3: Results for Experiment No.3

S/N	EXPT 3: INPUT VALUES	RESULTS FOR 10 RUNS (RUN-TIME IN MILLISECONDS)	NUMBER OF PARTICIPATING NODES
1	10	906	4
2	20	826	
3	30	888	
4	40	863	
5		978	
6		962	
7		876	
8		886	
9		911	
10		980	
AVERAGE	**25**	**908**	

Table 4.1.4: Results for Experiment No.4

S/N	EXPT 4: INPUT VALUES	RESULTS FOR 10 RUNS (RUN-TIME IN MILLISECONDS)	NUMBER OF PARTICIPATING NODES
1	10	964	5
2	20	991	
3	30	1,078	
4	40	1,038	
5	50	994	
6		1,000	
7		1,038	
8		924	
9		1,004	
10		934	
AVERAGE	**30**	**997**	

The results of the above tables and many more tables were grouped together to come with the results to prove the validity of the proposed protocol. These results were then plotted in a graph as shown in Figure 4-2 below.

The validity test results proved that the proposed protocol is scalable, i.e. it can be used, for example, for 10, 100, 1,000, 10,000, 100,000 nodes, etc. The results of the validity test are displayed graphically in Figure 4-2 below. Figure 4-2 below shows results for ten different experiments that were conducted for n nodes where $2 \leq n \leq 11$.

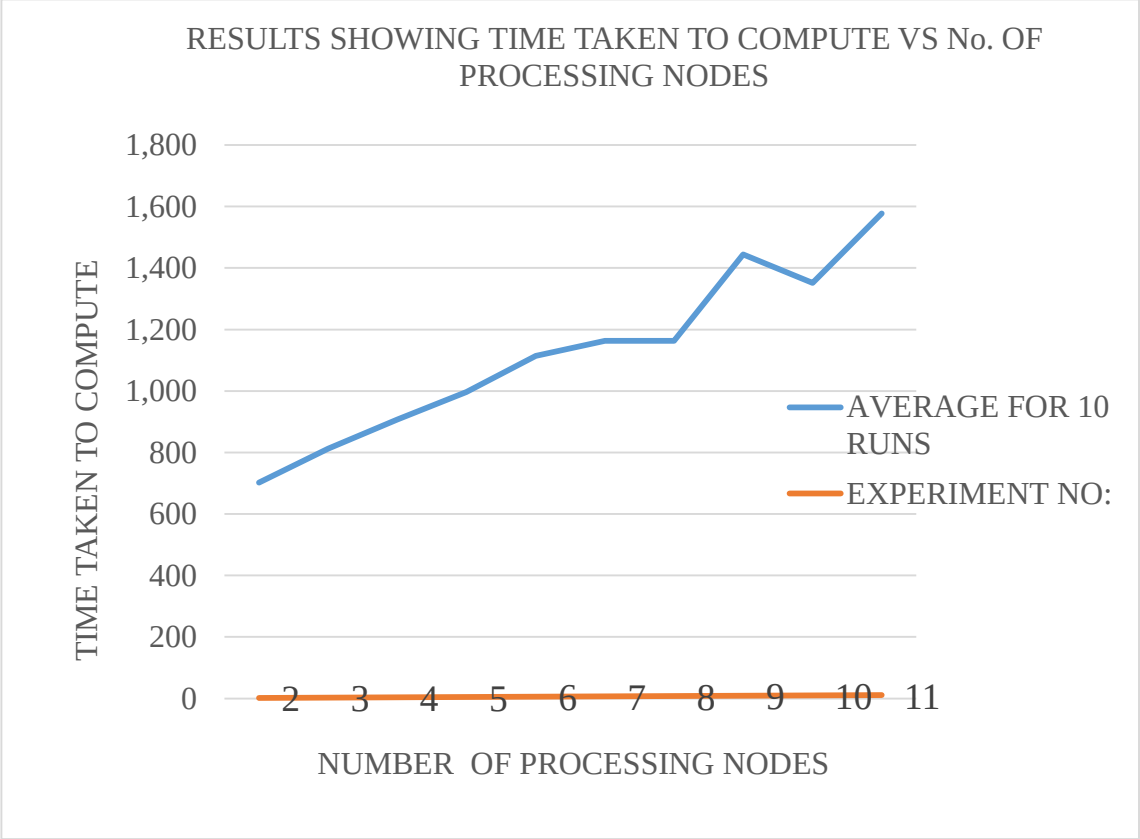

Figure 4-2: Showing results compiled when experiments are conducted on n number of processing nodes where n > 1

Summary

The first objective, which was to do with selecting the type of Homomorphic Encryption to employ in our efforts to develop a protocol for spatial searching, was achieved by reviewing literature systematically on HE and HE schemes. Paillier Homomorphic

Encryption was selected as the best technique to employ after a thorough comprehensive review and comparative analysis based on the literature selected. The second objective, which was to do with developing a protocol based on the selected HE scheme which later emerged to be Paillier Homomorphic Encryption, was achieved by our designing of a protocol using a distributed algorithm based on the Ring Algorithm (distributed ring algorithm principles) and making use of the Paillier cryptosystem. The third objective, which was to do with developing a proof of concept prototype using the proposed protocol, was attained by conducting some experiments in Java. Here, the experiments were done by using a prototype implemented as a distributed application which was written in java using the proposed protocol. The first experiment was to do with conducting a spatial search without using the proposed protocol. The second one involved doing the spatial using the protocol with Paillier Encryption. The third option involved conducting a spatial with a protocol without Paillier encryption at all. The purpose of the third option was to calculate the running time for the protocol without the encryption algorithm applied on it. A validity test was also done by doing experiments to calculate an average using anonymous data from a varying number of participating nodes. This test proved that the proposed protocol is scalable, i.e. it can be used for example for a 1,000, 10,000, 100,000 nodes, etc.

The other important point to note is that even though the proposed solution introduces a 39.7% overhead, this is outweighed by the benefits of the proposed approach.

CHAPTER 5 DISCUSSION AND CONCLUSION

5.1 Discussion

The research looked at the development of a protocol for secure distributed spatial searching using homomorphic encryption. The study was addressing the issue of privacy and security. Specifically, the study was looking at the lack of anonymity of data and confidentiality to the one who is providing the data and also whether the one who is providing the data is anonymous to those who are doing the searching. We were able to develop a protocol to ensure anonymity of the person conducting a spatial search online and data exchange with the service providers. The protocol was developed based on Homomorphic Encryption and Distributed Ring Algorithm Principles using Paillier Homomorphic Encryption technique. Using the developed protocol, a prototype was implemented using Java language. The results show that the developed prototype proved useful for collecting statistical data with guaranteed confidentiality. It also proved that, by putting into application Homomorphic Encryption, the person who was doing the search became anonymous to the providers of the data and the data providers became anonymous to the person who was doing the searching. The study was important and the problem being addressed was dealt with.

One of key areas where the results of the study can be applied is in collecting statistical data. Statistical information such as election results, number of HIV AIDS patients in certain geographical areas, data for use in cancer clustering and surveillance, etc. can be accessed by researchers online. The reason for accessing it are numerous including sharing

and combining information, broadcasting it, identifying patterns for statistical analysis, logistical support and planning, market surveys and business initiatives, further scientific research to mention but a few. Approaches like that proposed by [107], [108] have made access to spatial searches a highly feasible approach. However, this access can be problematic to both a data provider and a searcher as a result of privacy breaches and sometimes careless release of sensitive information to the public. Research in Mathematical Cryptography [109] has opened many doors for researchers to study Homomorphic Encryption techniques to identify possible solutions so that anonymity is fully guaranteed on both ends, that is, anonymity of the person doing the searching and also the data provider.

After analyzing the encryption schemes using existing literature on Homomorphic Encryption, Paillier Encryption Scheme became the focus for the three experiments. In the first experiment, the spatial search was conducted without using the protocol and encryption. The average execution time after running the program for ten times is shown in Table 4.3. In the second experiment, the spatial search was conducted by using the protocol with Paillier Encryption Sheme. After running the program for ten times, the average execution time is shown in Table 4.3. In the third experiment, the spatial search was done using the protocol but without Paillier Encryption Scheme. The average result after running the program for ten times is displayed in Table 4.3. A validity test was also done by doing experiments to calculate an average using anonymous data from a varying number of participating nodes. During the validity test, more than ten experiments were conducted. Each experiment done had $n > 1$ number of participating nodes where nodes in

our reference to Distributed Systems was applied in the context of number of processes which present one system image. In addition, each experiment was run for ten times so that we could calculate the average run time in milliseconds as displayed in Table 4.1.1, Table, $.1.2, Table 4.1.3, etc. It was these processes that formed a ring. The validity test proved that the proposed protocol is scalable, i.e. it can be used for example for a 10, 100, 1,000, 10,000, 100,000 nodes, etc.

Note that experiments **were not** conducted with other schemes here because the focus was on using Paillier Encryption Scheme after selecting Paillier when the other encryption schemes were analysed during the literature review of Homomorphic Encryption putting into consideration desirable qualities of a spatial search. It was during the literature review that the comparisons in Table 4.2 were made and Paillier was chosen as the best scheme to be used in our research.

The key principle in Homomorphic Encryption is that we can achieve addition and subtraction of the data using Additive Homomorphism and we can also get multiplication and division of the data using Multiplicative Homomorphism [110].

From the comparative analysis that was done in the literature review, it was identified that computational costs, encryption and decryption costs are key to the selection of a good algorithm to use. The higher the encryption and decryption costs the higher the security since it is hard and takes more time to break the algorithm. The less the computational cost the better the performance of a given algorithm. It was noted that the security standard of Paillier is relatively higher than that of RSA, El Gamal, Goldwasser-Micali and Benaloh because it has a higher encryption and decryption costs than these encryption

algorithms [7]. However, its security standard is relatively lower than FHE since its encryption and decryption costs are lower than FHE. FHE has a better encryption and decryption costs in terms of security strength but it has a higher computational cost due to complex mathematical computations created by its ability to perform both multiplications and additions. In terms of the computational cost, Paillier has a higher computational cost than RSA, El Gamal, Goldwasser-Micali and Benaloh. However, it has a lower computational cost than FHE. In striking a balance, Paillier was selected on the basis of security strength and also computational cost. Hence, Paillier Encryption was preferred in the choice of the best algorithm to use in the designing of our protocol for a spatial search for it has a higher security standard than RSA, El Gamal, Goldwasser-Micali and Benaloh and also a better computational cost than FHE.

Paillier Cryptosystem was identified as the best for supporting a spatial search because it is computationally cheaper to be used in practice and is relatively highly secure. Paillier's scheme is the most efficient among currently known additively homomorphic schemes because it requires simple operations in the encryption, decryption and addition procedures and hence achieves high performance. Another observation is that calculations on the encrypted data can be performed without necessarily reconstructing the original message and without having access to the private key [110].

5.2 Conclusion

The risk of lack of anonymity and confidentiality is what a client or a data provider may experience. A key limitation is that both the user and the data provider focus on either getting information or providing data without being careful about their anonymity

respectively. Therefore, protecting both the searcher and the data provider side is of greater importance for spatial searches.

Homomorphic Encryption, particularly the Paillier Homomorphic Encryption, supports a spatial search by providing anonymity of data of both parties involved in a spatial search while allowing analyses to be conducted on encrypted data in the encrypted space. Statistical data such as data to be aggregated in secure geocoding, data aggregation in cancer registries, election results were key examples in our research where analysis of data can be used to find total sums, averages, differences and so on and so forth. In secure geocoding, data aggregation such as calculation of county-specific incidence rates and analysis can be done based on the physical addresses provided. In cancer registries, researchers may aggregate all cases of a specific cancer across geographic boundaries to collect data from a sufficient number of cases for statistical analyses, for example, to find an average number of cases per geographic boundary. Analyses of large data sets such as election results can be achieved while ensuring anonymity of the one casting a vote and protecting the numbers of the votes cast. This can be a reality when such a scheme as a Paillier cryptosystem is used to encrypt both running ends of the searcher and the data provider. The Paillier cryptosystem is cheap and computationally capable and viable compared to other cryptosystems. Even though the proposed solution introduces a 39.7% overhead this is outweighed by the benefits of the proposed approach.

5.3 Recommendation

A recommendation to the relevant stakeholders including the Government Republic of Zambia (GRZ), medical institutions like Ministry of Health (MoH), Electoral

Commission of Zambia (ECZ) and various universities like University of Zambia Copperbelt University, Mulungushi University and many other institutions to adopt this protocol to use it in implementing applications that can be used for data aggregation and analysis. Such applications would ensure anonymity of the person conducting a spatial search as well as the providers of data since the protocol comes with it an additional layer of security based on the use of Paillier Homomorphic Encryption. In addition, the proposed protocol is scalable due to the use of Distributed System Principles based on the Ring Algorithm.

This is very useful for collecting statistical data with guaranteed confidentiality

5.4 Future Work

A proof of concept prototype was developed that can be implemented in most of the programming languages that can make use of Paillier Homomorphic Encryption. There is need to complete the work and build a full-fledged system that can be used by data providers and online researchers as they conduct spatial searches, search for or exchange data online.

There is also need to find an alternative for the operation limitation of the Paillier cryptosystem. Paillier only allows for one computational operation i.e. either addition or multiplication and not both addition and multiplication. It lacks the ability to perform both [34].

References

[1] S. Chauhan and N. K. Panda, "Hacking Web Intelligence, Online Anonymity", pp. 147-168, 2015. [Online]. Available. Feb 15, 2021. https://doi.org/10.1016/B978-0-12-801867-5.00008-2.

[2] K. Baxter, C. Courage and K. Caine, "Ethical and Legal Considerations, Anonymity vs. Confidentiality," pp. 64-78, 2015. [Online]. Available. Feb 15, 2021. https://doi.org/10.1016/B978-0-12-800232-2.00003-1.

[3] K. Balasubramanian and M. Jayanthi, "A Homomorphic Crypto System for Electronic Election Schemes," Circuits and Systems, Vol.7 No.10, 2016, pp. 1-11. [Online]. Available. Feb 15, 2021. DOI: 10.4236/cs.2016.710272

[4] M. Alkharji, H. Liu, M. A. Hammoshi, "A Comprehensive Study of Fully Homomorphic Encryption Schemes," International Journal of Advancements in Computing Technology (IJACT) Volume10, Number1, pp.1-24, Mar. 2018.

[5] A.A. Izang, A.O. Adebayo, O.J. Okoro and O.O. Taiwo, "Security and Ethical Issues to Cloud Database," The Journal of Computer Science and its Applications Vol. 24, No. 2, pp. 1-12, December, 2017.

[6] N. Jain, S. K. Pal and D. K. Upadhyayi, "Implementation and Analysis of Homomorphic Encryption Schemes," International Journal on Cryptography and Information Security(IJCIS), Vol.2, No.2, pp. 1-18, June 2012.

[7] M. Zhao and Y. Geng, “Homomorphic Encryption Technology for Cloud Computing,” 8th International Congress of Information and Communication Technology (ICICT-2019)- Procedia Computer Science 154 (2019) 73–83. [Online]. Available. Accessed on: Aug 16, 2019.

[8] R. ALmarwani, N. Zhang and J. Garside, “An effective, secure and efficient tagging method for integrity protection of outsourced data in a public cloud storage,” pp. 1-47, Nov 2020. [Online]. Available. Accessed on: Aug 16, 2019.
https://doi.org/10.1371/journal.pone.0241236

[9] R. Suwandi, S. M. Nasution and F. Azmi, “Secure E-voting System by Utilizing Homomorphic Properties of the Encryption Algorithm,” TELKOMNIKA, Vol.16, No.2, pp. 862-867, April 2018. [Online]. Available. Feb 16, 2021.
DOI: 10.12928/TELKOMNIKA.v16i2.8420

[10] A.M. Vengadapurvaja, G. Nisha, R. Aarthy and N. Sasikaladevi, “An Efficient Homomorphic Medical Image Encryption Algorithm for Cloud Storage Security,” 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22- 24 August 2017, Cochin, India. [Online]. Available. Accessed on: Aug 16, 2019.

[11] I. Ahmad, H. Bakht and U. Mohan, “Cloud Computing – A Comprehensive Definition,” Journal of Computing and Management Studies Volume 1. Issue 1. pp. 1-9, Jan 2017. [Online]. Available. Feb 17, 2021.
https://www.researchgate.net/publication/314072571

[12] I. Jabbar and S. Najim, "Using Fully Homomorphic Encryption to Secure Cloud Computing. Internet of Things and Cloud Computing," Vol. 4, No. 2, 2016, pp. 13-18, May 2016. [Online]. Available. Accessed on: Dec 16, 2018. doi: 10.11648/j.iotcc.20160402.12

[13] O. Waart and J. Thijssen, "Traditional Cryptography," pp. 1-11, Feb 2015. [Online]. Available. Feb 17, 2021. https://www.researchgate.net/publication/314072571

[14] S. M. P. C. Souza and R. S. Puttini, "Client-side encryption for privacy-sensitive applications on the cloud," Procedia Computer Science 97 (2016) 126 – 130. [Online]. Available. Accessed on: Aug 19, 2019.

[15] F. Kerschbaum. Client-controlled cloud encryption. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security; CCS '14. New York, NY, USA: ACM. ISBN 978-1-4503-2957-6; 2014, p. 1542–1543

[16] K. E. Makkaouia, A. Ezzatia, A. Beni-Hssaneb and S. Ouhmada, "A swift Cloud-Paillier scheme to protect sensitive data confidentiality in cloud computing," The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018). Procedia Computer Science 134 (2018) 83–90. [Online]. Available. Accessed on: Aug 19, 2019.

[17] S. S. Mathew and C. A. Hafsath, "Aiding Effective Encrypted Document Manipulation Incorporated with Document Categorization Technique in Cloud,"

International Conference on Information and Communication Technologies (ICICT 2014). Procedia Computer Science 46 (2015) 668 – 675. [Online]. Available. Accessed on: Aug 19, 2019.

[18] M. M. Poteya, C. A. Dhoteb and D. H. Sharmac, "Homomorphic Encryption for Security of Cloud Data," 7th International Conference on Communication, Computing and Virtualization 2016. Procedia Computer Science 79 (2016) 175 – 181. [Online]. Available. Accessed on: Aug 19, 2019.

[19] S. Shah, Y. Shah and J. Kotak, "Somewhat Homomorphic Encryption Technique with its Key Management Protocol", Dec 14 2014,s Volume 2 Issue 12, International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 4180 – 4183.

[20] A.M. Vengadapurvaja, G. Nisha, R. Aarthy and N. Sasikaladevi, "An Efficient Homomorphic Medical Image Encryption Algorithm for Cloud Storage Security," 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22- 24 August 2017, Cochin, India, Procedia Computer Science 115 (2017) 643–650.

[21] S. Dasgupta, "Design of a polynomial ring based symmetric homomorphic encryption scheme," Perspectives in Science (2016) 8, 692—695, pp. 1-4, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.pisc.2016.06.061

[22] F. Farokhi, I. Shames and N. Batterham, “Secure and Private Cloud-Based Control Using Semi-Homomorphic Encryption,” IFAC-PapersOnLine 49-22 (2016) 163–168, pp. 1-6, 2016.

[23] M. Ogburn, C. Turner and P. Dahal, “Homomorphic Encryption,” Procedia Computer Science 20 (2013) 502 – 509, 2013. [Online]. Available: doi: 10.1016/j.procs.2013.09.310

[24] F. Farokhi, I. Shames and K. H. Johansson, “Private and Secure Coordination of Match-Making for Heavy-Duty Vehicle Platooning,” IFAC PapersOnLine 50-1 (2017) 7345–7350.

[25] S. Ramezanian, T. Meskanen, M. Naderpour, V. Junnila and V. Niemi, “Private membership test protocol with low communication complexity,” Digital Communications and Networks, pp. 1-12, May. 2019. [Online]. Available: https://doi.org/10.1016/j.dcan.2019.05.002

[26] J. W. Bos, K. Lauter and M. Naehrig, “Private predictive analysis on encrypted medical data,” Journal of Biomedical Informatics 50 (2014) 234–243, Apr. 2014.

[27] R. Hayward and C. C. Chiang, “Parallelizing fully homomorphic encryption for a cloud environment,” Journal of Applied Research and Technology 13 (2015) 245-252, Aug. 2014.

[28] M. S. Darup, A. Redder and D. E. Quevedo, “Encrypted cloud-based MPC for linear systems with input constraints,” IFAC PapersOnLine 51-20 (2018) 535–542, 2018.

[29] K. Xu, W. Zhang and Z. Yana, “A privacy-preserving mobile application recommender system based on trust evaluation,” Journal of Computational Science 26 (2018) 87–107, Apr. 2019.

[30] C. Ma and C. W. Chen, “Nearby Friend Discovery with Geo-Indistinguishability to Stalkers,” The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014), Procedia Computer Science 34 (2014) 352 – 359.

[31] M. Z. Hasan, M. S. R. Mahdi, M.N. Sadat and N. Mohammed, “Secure count query on encrypted genomic data,” Journal of Biomedical Informatics 81 (2018) 41–52, Mar. 2018.

[32] K. E. Makkaoui, A. Ezzati, A. Beni-Hssane and S. Ouhmad, “Fast Cloud–Paillier homomorphic schemes for protecting confidentiality of sensitive data in cloud computing,” Journal of Ambient Intelligence and Humanized Computing, May. 2019. [Online]. Available: https://doi.org/10.1007/s12652-019-01366-3

[33] K. Muhammad, K. A. Sugeng and H. Murfi, “Machine Learning with Partially Homomorphic Encrypted Data,” Journal of Physics: Conf. Series 1108 (2018) 012112 doi :10.1088/1742-6596/1108/1/012112

[34] T. Oladunni and S. Sharma, “Homomorphic Encryption and Data Security in the Cloud,” Proceedings of 28th International Conference on Software Engineering and Data Engineering EPiC Series in Computing Volume 64, 2019, pp. 129–138, Oct. 2019.

[35] A. Malika, H. Wang, T. Chenc, T. Yang, A. N. Khand, H. Wue, Y. Chena and Y. Huf, "Reversible data hiding in homomorphically encrypted image using interpolation technique," Journal of Information Security and Applications 48 (2019) 102374, pp. 1-8. Oct. 2019.

[36] D. Archer, L. Chen, J. H. Cheon, R. Gilad-Bachrach, R. A. Hallman, Z. Huang, X. Jiang, R. Kumaresan, B. A. Malin, H. Sofia, Y. Song and S. Wang, "Applications of Homomorphic Encryption," pp. 1-15, Jul. 2017. [Online]. Available:

https://www.researchgate.net/publication/320976976

[37] W. T. Al-Sit, H. Al-Zoubi and Q. Al-Jubouri, "Cloud Security based on the Homomorphic Encryption," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 8, 2019, pp. 1-9, Jan. 2019.

[38] Z. Cao and L. Liu, "On the Weakness of Fully Homomorphic Encryption," pp. 1-11, Nov. 2015. [Online]. Available: https://www.researchgate.net/publication/284219690

[39] Z. Min, G. Yang and J. Shi, "A privacy-preserving parallel and homomorphic encryption scheme," Open Phys. 2017; 15:135–142, Dec. 2016. DOI 10.1515/phys-2017-0014

[40] P. V. Parmar, S. B. Padhar, S. N. Patel, N. I. Bhatt and R. H. Jhaveri, "Survey of Various Homomorphic Encryption algorithms and Schemes," International Journal of Computer Applications (0975 – 8887) Volume 91 – No.8, April 2014, pp. 1-8.

[41] S. Leela and P.Nithyanandam, “Secured Change Detection of Satellite Images Using Homomorphic Encryption,” Global Journal Of Engineering Science and Researches, pp. 1-9, Aug. 2016. DOI-10.5281/zenodo.59610

[42] R. A. Hallman, M. H. Diallo, M. A. August and C. T. Graves, “Homomorphic Encryption for Secure Computation on Big Data,” In Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018), pp. 340-347, 2018.

[43] M. K. Ibrahem, “Robust Electronic Voting System using Homomorphic Encryption Protocol and Zero-Knowledge Proof,” International Journal of Enhanced Research in Science, Technology & Engineering ISSN: 2319-7463, Vol. 5 Issue 1, January-2016, pp. 1-11.

[44] M. Ibtihal, E. O. Driss and N. Hassan, “Homomorphic encryption as a service for outsourced images in mobile cloud computing environment,” pp. 1-9, Apr. 2017. DOI: 10.4018/IJCAC.2017040103

[45] G. M. Penn, G. Potzelsberger, M. Rohde and A. Uhl, “Customisation of Paillier Homomorphic Encryption for Efficient Binary Biometric Feature Vector Matching,” pp. 1-12, 2014.

[46] T. Zhao, Q. Ran, L. Yuan, Y. Chi and J. Ma, "Key Distribution and Changing Key Cryptosystem Based on Phase Retrieval Algorithm and RSA Public-Key Algorithm," Mathematical Problems in Engineering, pp.1-13, Jun 2015. [Online]. Available. Accessed on: Oct 8, 2019.

http://dx.doi.org/10.1155/2015/732609

[47] N. F. H. Al Saffar, “Steganography Algorithm Based RSA Cryptosystem,” Journal of Engineering and Applied Sciences, pp. 1-5, April 2019. [Online]. Available. Accessed on: Jan 20, 2021.
DOI: 10.36478/jeasci.2019.2240.2243

[48] K. Mallaiah and S. Ramachandram, “Applicability of Homomorphic Encryption and CryptDB in Social and Business Applications: Securing Data Stored on the Third Party Servers while Processing through Applications,” International Journal of Computer Applications (0975 – 8887) Volume 100– No.1, p1-15, August 2014.

[49] X. Ye, C. Liu and D. Ga, “Weakness of RSA Cryptosystem Characteristic,” AIP Conference Proceedings 2040, 130005 (2018), pp. 1-9, Dec 2018. [Online]. Available. Accessed on: Jan 20, 2021.
doi: 10.1063/1.5079187.

[50] G.A.V.R. Rao, P.V. Lakshmi and N. Ravi Shankar, “RSA Public Key Cryptosystem using Modular Multiplication,” International Journal of Computer Applications (0975 – 8887) Volume 80 – No5, p1-5, October 2013.

[51] M. Tebaa and S. EL Hajii, “Secure Cloud Computing through Homomorphic Encryption,” International Journal of Advancements in Computing Technology(IJACT) Volume5, Number16, December 2013.

[52] M. Alkharji and H. Liu, “Homomorphic Encryption Algorithms and Schemes for Secure Computations in the Cloud,” ICSCT 2016 - International Conference on Secure Computation and Technology, Virginia International University, Fairfax, VA, 2016. Available. Accessed on: Oct 11, 2019.

[53] R. Shruthi, P. Sumana and A. K. Koundinya, “Performance Analysis of Goldwasser-Micali Cryptosystem,” International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7, July 2013.

[54] J. Sen, “Theory and Practice of Cryptography and Network Security Protocols and Technologies,” Homomorphic Encryption-Theory and Application, Chapter1, pp. 2-34, July 2013. [Online]. Available. Feb 23, 2021.
DOI: 10.5772/56687

[55] H. Yu and Y. Kim, “New RSA Encryption Mechanism Using One-Time Encryption Keys and Unpredictable Bio-Signal for Wireless Communication Devices,” Electronics 2020, pp. 1-10, Feb 2020. [Online]. Available. Feb 23, 2021.
doi:10.3390/electronics9020246

[56] J. Bringer, H. Chabanne, M. Izabache’ne and D. Pointcheval, “An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication,” Jun 2014. [Online]. Available. Accessed on: Oct 8, 2019.

[57] A. Acar, H. Aksu, A. S. Uluagac and M. Conti, “A Survey on Homomorphic Encryption Schemes: Theory and Implementation,” pp.1-35, Oct. 2017.

[58] V. Biksham and D. Vasuma, “Homomorphic Encryption Techniques for securing Data in Cloud Computing: A Survey,” International Journal of Computer Applications (0975 - 8887) Volume 160 - No.6, February 2017.

[59] Y. Shi, “Data Security and Privacy Protection in Public Cloud,” pp. 1-9, Dec. 2018.

[60] G. M. Jacquez et al., “Geospatial Cryptography: Enabling researches to access private, spatially referenced human subjects’ data for cancer control and prevention, pp. 1-26, Jul. 2017. [Online]. Available. Accessed on: Oct 8, 2019.

[61] M. Birkin, “Spatial data analytics of mobility with consumer data,” Journal of Transport Geography 76 (2019) 245–253, pp. 1-9, Apr. 2019.

[62] A. Tondwalker and P. V. Jan, “Secure localisation of wireless devices with application to sensor networks using steganography,” International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA, pp. 1-7.

[63] A. AlDairi and L. Tawalbeh, “Cyber Security Attacks on Smart Cities and Associated Mobile Technologies,” The International Workshop on Smart Cities Systems Engineering (SCE 2017), Procedia Computer Science 109C (2017) 1086–109.

[64] J. Ajayakumar, K. Ghazinour, “I am at home: Spatial Privacy Concerns with Social Media Check-ins,” The 4th International Symposium on Emerging Information, Communication and Networks (EICN-2017), Procedia Computer Science 113 (2017) 551–558.

[65] M. Kiedrowicz, “Methodology of Ensuring the Security of GIS Spatial Data,” 26th Geographic Information Systems Conference and Exhibition “GIS ODYSSEY 2019” Conference proceedings, pp. 1-13, Nov. 2019.

[66] R. M. M. Pradeep and N. T. S. Wijesekera, “Development of Security Stamp for Desktop Spatial Data Modification in Unrestricted Access Platform,” pp. 1-8, Sept. 2015.

[67] X. Ma, “Spatial Data,” pp. 1-7, Jan. 2017. [Online]. Available. Accessed on: Aug 13, 2019.
DOI: 10.1007/978-3-319-32001-4_192-1

[68] R. Guo, B. Qin, Y. Wu, R. Liu, H. Chen and C. Li, “MixGeo: Efficient Secure Range Queries on Encrypted Dense Spatial Data in the Cloud,” pp. 1-11, Jun. 2019.

[69] O. Kounadi and B. Resch, “Towards geoprivacy guidelines for spatial data,” pp. 1-8, Jan. 2018. [Online]. Available: DOI: 10.3929/ethz-b-000225618

[70] M. G. Fugini and G. C. Hadjichristofi, “Principles of Authorization Design for Spatial Data,” pp. 1-16, 2013.

[71] D. Chen, P. Zhang, C. Hu, H. Wang, S. Wu and N. Xing, “PAPERS: Private and Precise Range Search for Location Based Services,” IEEE ICC 2015 - Communication and Information Systems Security Symposium, pp. 1-6, 2015.

[72] A. Talha, I. Kamel and Z. A. Aghbari, “Enhancing Confidentiality and Privacy of Outsourced Spatial Data,” pp. 1-8, Nov. 2015.

[73] L. Kacha and A. Zitouni, “An Overview on Data Security in Cloud Computing,” Advances in Intelligent Systems and Computing, pp. 1- 13, Sep. 2018. [Online]. Available: DOI: 10.1007/978-3-319-67618-0_23

[74] B. R. Pushpa, “Enhancing Data Security by Adapting Network Security and Cryptographic Paradigms,” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 1319-1321, pp. 1-4, Aug. 2019.

[75] P. Maniriho AND T. Ahmad, “Information hiding scheme for digital images using difference expansion and modulus function,” Journal of King Saud University – Computer and Information Sciences 31 (2019) 335–347. [Online]. Available. Accessed on: Aug 13, 2019.

[76] M. Modak and R. Shaikh, “Privacy Preserving Distributed Association Rule Hiding Using Concept Hierarchy,” 7th International Conference on Communication, Computing and Virtualization 2016- Procedia Computer Science 79 (2016) 993 – 1000, Mar 2016. [Online]. Available. Accessed on: Aug 13, 2019.

[77] X. Liao and C. Shu, “Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels,” J. Vis. Commun. Image R. 28 (2015) 21–27. [Online]. Available. Accessed on: Aug 13, 2019.

[78] S. Vennila and Priyadarshini. J, “Scalable Privacy Preservation in Big Data a Survey,” Procedia Computer Science 50 (2015) 369 – 373, Apr 2015. [Online]. Available. Accessed on: Aug 16, 2019.

[79] P. B. Keenana and P. Jankowski, “Spatial Decision Support Systems: Three decades on,” Decision Support Systems 116 (2019) 64–76, pp. 1-13, Oct. 2018.

[80] M. Blistanova, P. Blistan, P. Lošonczi, “Possibilities of Application of Geographic Information Systems to Security Education,” 5th World Conference on Learning, Teaching and Educational Leadership, WCLTA 2014, Procedia - Social and Behavioral Sciences 186 (2015) 744 – 748.

[81] P. McKeague, R. V. Veer, I. Huvila, A. Moreau, P. Verhagen, L. Bernard, A. Cooper, C. Green and N. V. Manen, “Mapping Our Heritage: Towards a Sustainable Future for Digital Spatial Information and Technologies in European Archaeological Heritage Management,” Journal of Computer Applications in Archaeology, 2(1), pp. 89–104, 2019. DOI: https://doi.org/10.5334/jcaa.23

[82] B. K.Saraswat, R. Suryavanshi, and D.S.Yadav, “A Comparative Study of Checkpointing Algorithms For Distributed Systems,” International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 1595-1603, pp. 1-11, Sep. 2019.

[83] P. B. Soundarabai, J. Thriveni, K. R. Venugopal and L. M. Patnaik, “An Improved Leader Election Algorithm for Distributed Systems,” International Journal of Next-Generation Networks (IJNGN) Vol.5, No.1, March 2013, pp. 1-9.

[84] M. Al-Refai, Y. Alraba'nah, M. Alauthman, A. Almomani, M. Al-Kasassbeh and M. Alweshah, “A Novel Leader Election Algorithm for Honeycomb Mesh Networks,”

Journal of Theoretical and Applied Information Technology 31st July 2019. Vol.97. No 14, pp. 1-14.

[85] S. Basu, "Token Ring Algorithm to Achieve Mutual Exclusion in Distributed System – A Centralized Approach," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011 ISSN (Online): 1694-0814, pp. 1-6.

[86] A. Dadlani, A. Khonsari, M. Effatparvar, N. Yazdani and M. Effatparvar, "Improved Algorithms for Leader Election in Distributed Systems," pp. 1-6, 2010. DOI: 10.1109/ICCET.2010.5485357 · Source: IEEE Xplore.

[87] S. Naseera, "A Distributed Ring Algorithm for Coordinator Election in Distributed Systems," ICTACT Journal On Communication Technology, September 2016, Volume: 07, Issue: 03, ISSN: 2229-6948(ONLINE) DOI: 10.21917/ijct.2016.0197

[88] H. Shaheen., Distributed Systems. Hyderabad: BONFRING Intellectual Integrity, 2019.

[89] G. A. Qadir and S. R. M. Zeebaree, "Evaluation of QoS in Distributed Systems: A Review," International Journal of Science and Business Volume 5. Issue 2. pp. 89-101, Jan 2021. [Online]. Available. Feb 18, 2021.
DOI: 10.5281/zenodo.4481463

[90] M. Steen and A. S. Tanenbaum, "A Brief Introduction to Distributed Systems," Computing 98. pp. 1-43, June 2016. [Online]. Available. Feb 18, 2021.
DOI 10.1007/s00607-016-0508-7

[91] A. Z. Kintonova, B. Z. Andassova, M. A. Ermaganbetova and E. K. Maikibaeva, "Development of Distributed System for Electronic Business Based on Java-Technologies," International Journal of Environmental & Science Education 2016, Vol. 11, No. 10, 3861-3883. pp. 1-23, June 2016. [Online]. Available. Feb 18, 2021. https://files.eric.ed.gov/fulltext/EJ1114675.pdf

[92] Anatomy of a Search Engine, "The Various Components of a Web Search Engine," [Online]. Available. Feb 18, 2021. https://www.google.com/url?sa=i&url=https%3A%2F%2Fslideplayer.com%2Fslide%2F7550257%2F&psig=AOvVaw39nMhjAlkT1Um1z1mx00yD&ust=1613754337954000&source=images&cd=vfe&ved=2ahUKEwjW3-jB9fPuAhWTQkEAHVq6CYYQjhx6BAgAEBI

[93] M. J. Escalona, F. J. Domínguez-Mayo, J. A. García-García, N. Sánchez and J. Ponce, "Evaluating Enterprise Content Management Tools in a Real Context," Journal of Software Engineering and Applications, 2015, 8, 431-453, pp. 1-23, Aug 2015. [Online]. Available. Feb 22, 2021. http://dx.doi.org/10.4236/jsea.2015.88042

[94] K. Abawi, "Systematic Review-From Research to Practice: Training in Sexual and Reproductive Health Research 2015," pp. 1-12. [Online]. Available: https://www.gfmer.ch/SRH-Course-2015/research-methodology/pdf/Systematic-review-Abawi-2015.pdf

[95] F. Fakhfakh, M. Tounsi, M. Mosbah and A. H. Kacem, “Formal Verification Approaches for Distributed Algorithms: A Systematic Literature Review,” International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018, 3-5 September 2018, Belgrade, Serbia. Procedia Computer Science 126 (2018) 1551–1560.

[96] A Java Implementation of Paillier cryptosystem, “Paillier Cryptosystem,” [Online]. Available. Feb 19, 2021.
https://github.com/kunerd/jpaillier.git

[97] M. Dabbagh, S. P. Lee and R. M. Parizi, “Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches,” Soft Computing, pp. 1-25, July 2015. [Online]. Available. Feb 22, 2021.
DOI 10.1007/s00500-015-1760-z

[98] S. Alsaleh and H. Haron, “The Most Important Functional and Non-Functional Requirements of Knowledge Sharing System at Public Academic Institutions: A Case Study,” Lecture Notes on Software Engineering, pp. 1-6, May 2016. [Online]. Available. Feb 23, 2021.
DOI: 10.7763/LNSE.2016.V4.242

[99] H. Kaur and A. Sharma, “Non-Functional Requirements Research: Survey,” International Journal of Science and Engineering Applications Volume 3 Issue 6, 2014, ISSN-2319-7560, pp. 1-12, Sept 2015. [Online]. Available. Feb 22, 2021.
DOI: 10.7753/IJSEA0306.1003

[100] U. Demiryurek, F. Banaei-Kashani and C. Shahabi (2010), "Efficient K-Nearest Neighbor Search in Time-Dependent Spatial Networks," International Conference on Database and Expert Systems Applications DEXA 2010: Database and Expert Systems Applications, pp 432-449. https://doi.org/10.1007/978-3-642-15364-8_36

[101] J. Zhong, X. Meng, X. Zhou and D. Liu, "Co-spatial Searcher: Efficient Tag-Based Collaborative Spatial Search on Geo-social Network," DASFAA 2012, Part I, LNCS 7238, pp. 560–575, 2012.

[102] I. San, N. At, I. Yakut, H. Polat, "Efficient Paillier Cryptoprocessor for privacy-preserving data mining," Feb. 2016.

[103] M. Nassar, A. Erradi, Q. M. Malluhi, "Paillier's Encryption: Implementation and Cloud Applications," pp. 1-6, Oct 2015. [Online]. Available. Accessed on: Feb 9, 2021. DOI: 10.1109/ARCSE.2015.7338149

[104] X. Wang, T. Luo and J. Li, "A More Efficient Fully Homomorphic Encryption Scheme Based on GSW and DM Schemes," Security and Communication Networks, pp. 1-15, Dec 2018. [Online]. Available. Feb 23, 2021. https://doi.org/10.1155/2018/8706940

[105] G. A. AL-Rummana, G. N. Shende, "Homomorphic Encryption for Big Data Security: A Survey," International Journal of Computer Sciences and Engineering, pp. 1-10, Oct 2018. [Online]. Available. Feb 23, 2021. DOI: 10.26438/ijcse/v6i10.503511

[106] Geeksforgeeks.org. BigInteger Class in Java. [Online]. Available. Nov 12, 2019. https://www.geeksforgeeks.org/biginteger-class-in-java/

[107] M. Nyirenda, H. Arimura and K. Ito, "Relaxing the data access bottleneck of geographic big-data analytics applications using distributed quad trees," 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, 2016, pp. 189-195.

[108] M. Nyirenda and D. Zulu, "Speeding up construction of distributed quadtrees for big-data analytics applications using dilated integers and hashmaps," 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, 2017, pp. 1-6.

[109] N. Hamlin, "(2017) Number in Mathematical Cryptography," Open Journal of Discrete Mathematics, 7, 13-31, pp. 1-19, Jan. 2017. [Online]. Available. Accessed on: Nov 12, 2019. http://dx.doi.org/10.4236/ojdm.2017.71003

[110] J. Katambo, M. Nyirenda and D. Zulu, "A Protocol for Secure Distributed Spatial Searching Using Homomorphic Encryption," Proceedings of The International Conference in ICT (ICICT2019) - Lusaka, Zambia (20th - 21st November 2019), pp. 178-183, Dec. 2019.