International Journal of Cyber-Security and Digital Forensics (IJCSDF) 7(4): 463-479
The Society of Digital Information and Wireless Communications (SDIWC), 2018 ISSN: 2410-0439 (Online)

Privacy protection of Users' Data in Social Network Systems based on Homomorphic Cryptography Techniques

[1]Kosmas Kapis , [2]Maduhu Mshangi,
[1,2]*University of Dar es Salaam, Dar es Salaam, Tanzania*
[1]*E-mail: kkapis@gmail.com*
[2]*E-mail: mshangimaduhu@yahoo.com*

ABSTRACT

The use of social network systems (SNS) for interacting and sharing information across the globe has led to new vulnerabilities and cyber threats which results to privacy violation in SNS such as Facebook, Twitter, LinkedIn and the other likes. This paper presents an innovative approach for addressing privacy violation of users' data in SNS based on homomorphic cryptography techniques. It employed divide and conquers, use case analysis, controlled experiment, Big O notation and homomorphic cryptography techniques. The study proposes an algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques. The performance analysis of the proposed solution was performed using Big O notation and controlled experiment. It has a time complexity of $O(n^2)$. It has an execution time of shorter than 0.094 seconds for keys size of at most 1024 bit. Thus, the proposed solution can be used in the real-world environment for privacy protection of users' data in SNS. This research contributes security requirements for privacy protection of users' data in SNS using modern cryptography techniques. Additionally, it contributes empirical evidence for privacy protection of users' data in SNS using homomorphic cryptography techniques.

KEYWORDS

Homomorphic cryptography, homomorphic encryption, an algorithm for privacy protection, privacy protection of users' data, privacy in social network

1 INTRODUCTION

Privacy protection is concerned with the safeguarding of personally identifiable information (PII) or sensitive users' data from unauthorized access and disclosure or observation [1]–[3]. It provides the owner of the users' data with the ability to determine what data/information can be shared with the third party. The PII is concerned with information (or data) that can be used on its own or in conjunction with other information to identify, contact, or locate an individual in context [3], [4]. The increase of use of social network systems (SNS) for sharing information has raised security concern on how to ensure the privacy of data in SNS [5]–[7]. Privacy concerns in SNS are based on users' practices and cyber-attacks in SNS. The attacks in SNS include identity theft, data leaks, hacking, malicious codes such as viruses and Trojan [8].

Moreover, privacy preserving in SNS is becoming a challenge due to the practices of SNS providers of sharing of users' sensitive data to third parties [7]. SNS providers are creating open-holes by developing various tools (such as application program interface (API)) for sharing data with third parties [7], [9]. Likewise, SNS providers' leaks PII information by allowing third-party advertising and tracking companies [4] to associate the web surfing habits of users with a specific individual as it has been revealed in Cambridge analytica scandal [9]. The existing security and privacy controls are becoming a privacy nightmare. Many researchers have raised interest in conducting research to find out how to improve privacy in SNS; but the problem is long-standing yet [10]–[13]. Most of the early privacy-preserving techniques were based on normal encryption algorithmic techniques [14], access controls mechanisms and anonymity based techniques [15] for ensuring privacy in SNS.

The normal encryption techniques involve encrypting and decrypting sensitive data to perform various operations in SNS. This approach

leads to a loss of data privacy, and it poses a privacy concern for data processed, stored and transmitted in SNS [13], [14]. Now the trends in privacy preserving of data in a ubiquitous computing environment such as SNS has shifted to modern cryptography techniques [13], [16]. This involves the use of homomorphic cryptography (encryption) techniques [11], [17]. The homomorphic cryptography techniques allow manipulation and operations of encrypted users' data in SNS without a need of decrypting ciphertexts of users' data in advance. The study seeks to extend the application of homomorphic cryptography techniques to the privacy protection of users' data in SNS.

The paper addresses the problematic situation of privacy violation in SNS. It proposes an algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques. It is organized as follows. Section 1 presents the introduction. Section 2 presents the literature review in SNS. Section 3 presents the methodology employed in this study. Section 4 presents the analysis and design of the proposed solution. Section 5 presents the results and discussion of the proposed solution. Finally, section 6 presents the conclusion.

2 LITERATURE REVIEW

Literature view was reviewed in terms of homomorphic cryptography techniques and related work to the privacy protection of users' data in SNS.

2.1 Homomorphic Cryptography

Homomorphic cryptography is the modern cryptography technique which allows performing computation on encrypted data without its decryption in advance [16], [18]–[20]. The users of SNS operate on encrypted data by performing homomorphic operations without a need to decrypt ciphertexts in advance [17]. The homomorphic operations are based on additive and (or) multiplicative homomorphic property over ciphertext [14], [17]. The homomorphic cryptography includes homomorphic encryption schemes, homomorphic encryption primitives, homomorphic properties and homomorphic operations in SNS.

2.1.1 Homomorphic Encryption Schemes

Homomorphic encryption can be categorized mainly into three categories, namely partially homomorphic encryption, fully homomorphic encryption scheme and somewhat homomorphic encryption schemes (SHE).

(i) Partially Homomorphic Encryption Schemes

Partially homomorphic encryption (PHE) scheme is the one support either additional or multiplication homomorphic encryption, but not both at the same time [14], [21]. Some of the PHE schemes includes RSA invented in 1978 which is multiplicative homomorphic [22]; Goldwasser-Micali in 1982 [14] which is homomorphic over addition; El-Gamal in 1985 which is multiplicative homomorphic [14]; Paillier in 1996 which is additive homomorphic [21], [23]. PHE has been widely used in practical implementation to enhance privacy in areas such as cloud computing, big data computation in an untrusted environment, e-voting; as it requires less execution time, less memory and easy to implement compared to fully homomorphic encryption [14], [16].

(ii) Fully Homomorphic Encryption Schemes

Fully homomorphic encryption (FHE) is the one which allows both addition and multiplication operations at the same time over ciphertext [14], [21]. FHE has been debatable for longtime until [19] came out with first FHE based on ideal lattices [19], [24]; but it was very complex for practical implementation. Various researchers continued to improve Gentry FHE to get a FHE which can be practically implemented [16]. Another was FHE integer based scheme which was a simplification of FHE-Gentry by replacing ideal lattices with integer FHE based scheme [25]. Further researches have been going on to optimize the performance of the FHE [16], [24]. The further development in FHE schemes includes Gentry and Halevi Efficient FHE with learning with errors scheme (LWE) [26], Leveled-FHE scheme without Bootstrapping [27], The NTRU (N^{th} degree truncated polynomial ring) FHE based

scheme [16], [28], The Ring-LWE FHE based scheme [17], [29].

(iii) Somewhat Homomorphic Encryption
The shift has been to somewhat homomorphic encryption schemes (SHE) which in practice involve more homomorphic addition operations and few multiplication homomorphic operations [18]. The challenges of execution, memory consumption and noise in FHE have been addressed to practical implementable level through improvement and advancement in processors computing processing speed and further innovation of FHE and SHE schemes [16], [30].

2.1.2 Homomorphic Encryption Primitives
Homomorphic encryption schemes are comprised mainly of three basic primitives' operations namely key generation, encryption, and decryption algorithms [14], [16], [21], [23]. The research study employed Paillier homomorphic encryption to illustrate the basic primitive operations involved in homomorphic encryption (PHE, FHE). Other homomorphic encryption schemes can be substituted to achieve the same results.

2.1.3 Homomorphic Properties
Homomorphic cryptosystem can be defined as follows. Let M denote the set of plaintext in plaintext space of message to be encrypted. Let C denote the set of the ciphertexts in ciphertext space in respect to plaintext space. Homomorphic cryptosystem (PHE, FHE) have homomorphic properties if it satisfies either or all of the following relations.

First, additively homomorphic relation

$$E(m_1 + m_2) = E(m_1) + \mathrm{E}(\mathrm{m}_2), \forall m_1, \ \mathrm{m}_2 \in M \qquad (1)$$

Second, multiplicatively homomorphic relation

$$E(m_1 * m_2) = E(m_1) * \mathrm{E}(\mathrm{m}_2), \ \forall m_1, \ \mathrm{m}_2 \in M \qquad (2)$$

Prior studies have revealed that 76% of users are not aware [31], [32] of the importance of privacy issues in SNS. Likewise, the existing privacy controls in SNS are inadequate to protect users' data [33]. Moreover, users of SNS cannot control what others may reveal about them [32], [33]. Moreover, SNS providers are sharing users' privacy data to third parties through API and web services without users consent [34]. The problem of privacy violation in SNS can be addressed using modern cryptography techniques [17], [35].

2.2 Related Works
Homomorphic cryptography is the modern cryptography technique which allows performing manipulation and operations on encrypted users' data in SNS without a need of decrypting users' data in advance [16], [18]–[20]. The users of SNS operate on encrypted data by performing homomorphic operations without a need to decrypt ciphertexts of users' data in advance [17]. The homomorphic operations are based on additive and (or) multiplicative homomorphic property over ciphertext [14], [17].

Homomorphic encryption despite its existence since 1978 as invented by Rivest, Shamir and Adleman [22], but it was not fully leverage due to its limitation of requiring large computing resources [14]. Currently, processing power has increased to outweigh the limitations [17], [30], [36], [37]. Many researchers are trying to address the problem of privacy violation in SNS, cloud computing, e-voting and other untrusted environment settings by using homomorphic encryption techniques [14], [29].

Homomorphic encryption allows performing operations on encrypted data without a need of decrypting first [16], [23]. Moreover, the generated results are encrypted ciphertext in which when decrypted the results is the same as if it was performed on plaintext [17], [31]. This research study discusses the related work to the research problem of privacy violation in SNS. A study by [10] proposed a secure privacy-preserving scheme for data sharing in online SNS with focus on revocation for deterring a contact's access right to the private data and privacy on searching using homomorphic encryption. It is limited to revocation for deterring a contact's access right to the private data and privacy on searching. Likewise, a research study by [38] proposed a privacy-preserving trusted feedback system for addressing feedback from friends based on the

question asked in SNS, using homomorphic encryption. It is limited to a scope of feedback for item recommendation in SNS. It lacks effective mechanisms and algorithm for addressing privacy violation in SNS.

A study by [11] proposed privacy chatting for preserving preserves based on homomorphic encryption technique for anonymous common interest verification in SNS. It is limited to improving the privacy of vehicle drivers while using wireless communication of Vehicular Ad Hoc Networks. Extension to SNS networks such as Facebook and Twitter is questionable. It lacks effective mechanisms and algorithm for addressing the privacy violation problem in SNS.

Studies by [39] and [31] proposed a privacy-preserving solution for user profile matching in SNS by preserving privacy on query used for searching and querying data results in SNS using homomorphic encryption techniques. Proposed solutions were limited to searching and protocols for profile matching. They lack effective mechanisms and algorithm for addressing the problematic situation of privacy violation in SNS. Moreover, a study by [12] proposed a solution for privacy-preserving for photo sharing in SNS using homomorphic encryption techniques. It is limited to privacy-preserving for photo-sharing service in SNS.

The existing privacy controls in SNS are inadequate for addressing privacy violation in SNS. Thus, this study proposes homomorphic cryptographic techniques for privacy protection of users' data in SNS. It uses research questions to address the research problem of privacy violation in SNS. The following research questions were used in the research problem.

i). What are the requirements for privacy protection of users' data in SNS?
ii). How to develop the algorithm for privacy protection of users' data in SNS?
iii). What is the performance of the proposed solution for the privacy protection of users' data in SNS?

3 METHODOLOGY

The study employed mixed research methods which include divide and conquers, use case analyses, controlled experiment and Big O notation. For analyzing the requirements for privacy protection of users' data in SNS, it employed use case analysis techniques. Moreover, for analysis of the performance of the proposed solution, it employed Big O notation and controlled experiment.

The divide and conquers was employed to address the problem of privacy violation. The research problem was split down into smaller sub-problems from the original problem [40], [41]. The smaller sub-problems were solved and the solutions to each sub-problem were combined to create a solution to the original problem. Use case analysis technique was employed to gather and analyze requirements [42], [43] for developing an algorithm for privacy protection of users' data in SNS. It was used to answer research question 2: "How to develop the algorithm for privacy protection of users' data in SNS?" It was used to design and understand the interactions of users in SNS by communicating the behaviour of users and specifying all external interactions in SNS.

The study employed Big O notation and controlled experiment to analyze the performance of the proposed algorithm for privacy protection of users data in SNS. Big O notation was used to analyze and determine the algorithm complexity [40] of the algorithm for privacy protection of users' data in SNS. Applying Big O notation to an algorithm for privacy protection of users' data in SNS); the algorithm performance can be deduced as follows. Let $f(n)$ and $g(n)$ be functions such that $f(n), g(n) \in \mathbf{Z}^{+}$. The Big O notation can be expressed as $f = O(g)$, if and only if $f(n) \leq c.g(n)$; where c is a constant, c>0. A controlled experiment was carried out in two phases to analyze the performance (algorithm execution time) of the proposed solution for the privacy protection of users' data in SNS.

4 ANALYSIS AND DESIGN OF THE PRIVACY PROTECTION OF USERS' DATA

The research study employed divide and conquers algorithm techniques to analyze the requirements for developing the algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques. It involves understanding the research problem: privacy violation in SNS and sub-dividing it into sub-problems. The research study applied a divide and conquers algorithm techniques compounded with use case analysis techniques.

Use case analysis technique was employed to gather and analyze requirements for developing an algorithm for privacy protection of users' data in SNS as presented in Figure 1. It was used to design and understand the interactions of users in SNS by communicating the behaviour of users and specifying all external interactions in SNS. Additionally, the requirements were clearly communicated based on how the algorithm for privacy protection of users' data in SNS should be developed. Moreover, it assisted in determining the roles the users play in the SNS and the response to stimulus users receives from the SNS.

Use case analysis technique was employed to understand the users' interactions [42], [43], as summarized in Figure 1. The users interact with SNS by requesting various services such as registering profiles, searching, chatting, sending messages, uploading and sharing multimedia content: photos, video, audio. Thus, causes privacy violation due to ineffective privacy controls in SNS such as Facebook, Twitter, and others. The problem of privacy violation can be sub-divided into sub-problems which include privacy violation due to users operations and usage of SNS; and privacy violation due to SNS provider. The sub-problem due to user operations can be further sub-divided into sub-problems which include privacy violation due to users' profile; privacy violation due to friending; privacy violation due to messing; privacy violation due to photo sharing; privacy violation due to searching and tagging.

Privacy violation can result from SNS providers' failure to ensure security and privacy in SNS. Security and privacy controls based on access controls and legacy encryption by SNS providers do not ensure privacy in SNS. There is a need for establishing security and privacy controls which do not depend on the trust of SNS providers. The research study establishes privacy requirements for the development of an algorithm for privacy protection of users' data in SNS. These privacy controls based on homomorphic cryptography techniques would allow users in SNS such as Facebook, Twitter and the other likes to operate in untrusted environments without fears of violation of privacy due to SNS providers.

The SNS services as presented in Figure 1, involve the use of multimedia content for sharing information in SNS. A multimedia content involves the use of a combination of different content forms in SNS such as text, audio, images, animations, video and interactive content. The use of multimedia contents is becoming popular in SNS. Users of SNS can combine different types of media for sharing information such as text, audio, and video. Audio is compressed, stored and processed and transmitted in SNS. Audio files are encoded by sampling at 8 bit or 16-bit sample depth [44], [45].

The homomorphic encryption requirement is to encrypt them at 16-bit. The video is composed of a series of frames played over time. The video is compressed, stored, processed and transmitted as individual frames. For example, a frame with 720 x 576 pixels with a bit depth of 24 bits and a frame rate of 25 frames per second frames per second) [46]–[48]. Most digital video formats are decoded using a minimum of 8 bits depth sample, others use 16 bits, and 24 bits sample depth. The homomorphic encryption requirement for video is to encrypt individual frames of plaintext message using bit length of between 8 bits and 24 bits per sample [45], [47], [48].

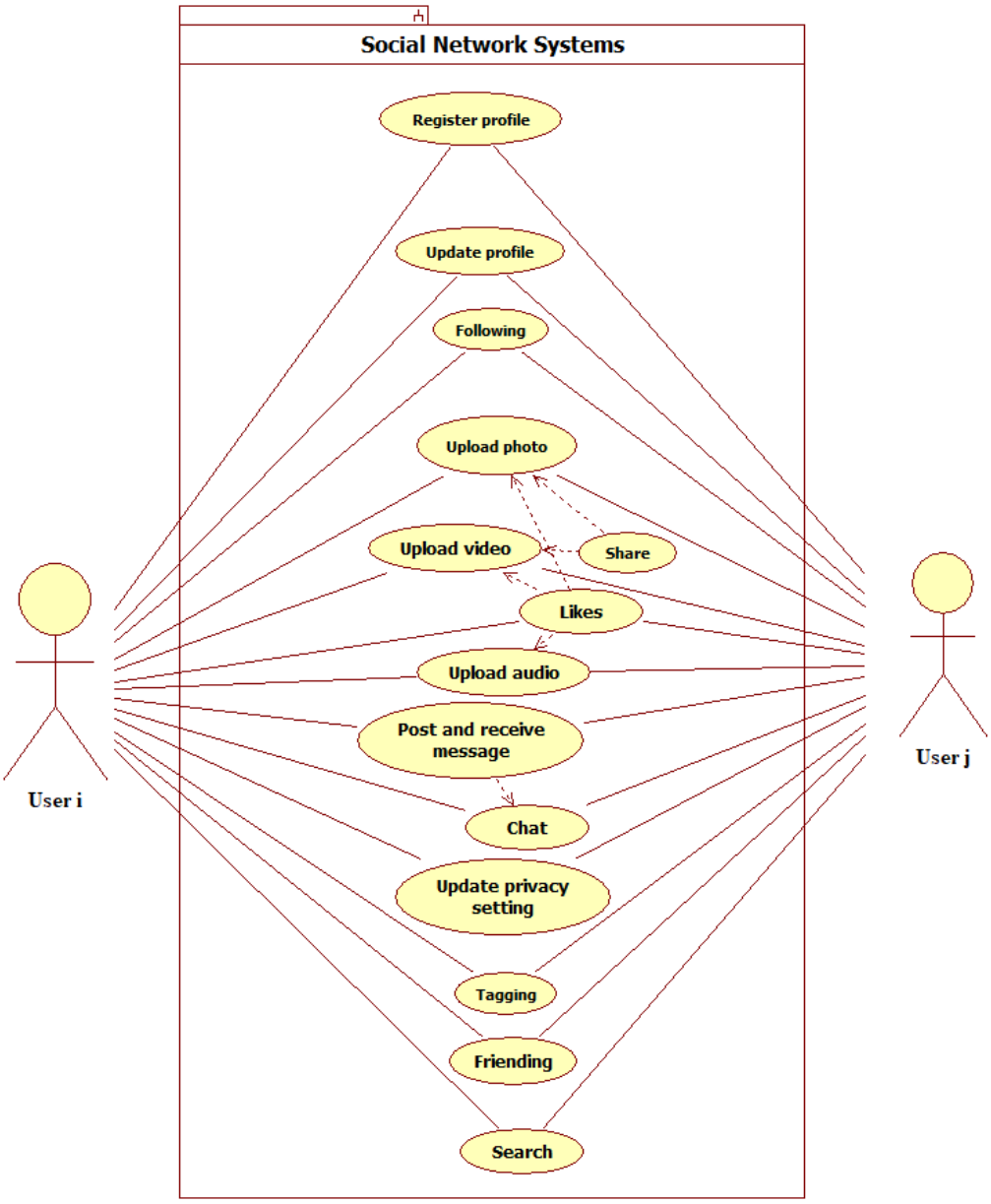

Figure 1. Users interactions in social network systems

The requirements for developing algorithm were gathered through literature review and divide and conquers algorithm techniques compounded with use case analysis. The requirements for the development of an algorithm for privacy protection of users' data in SNS are as follows.

i. Users of SNS performs operations on encrypted data using homomorphic cryptography operations.
ii. The users of SNS use homomorphic public key infrastructure by using the private key and public key
iii. Key generation: the $user_i$ of SNS generates the public key (PuK_i) and the private key ($\Pr K_i$)
iv. Encryption: The $user_i$ of SNS encrypts the data with the public key (Puk_i), and sends the encrypted data and Puk_i to the SNS provider server.
v. Storage: the encrypted data and Puk_i are stored in the SNS provider database.

vi. Request: the $user_i$ interacts with SNS interface to send a request to the SNS provider server to perform operations on encrypted data.

vii. Evaluation: SNS provider processes the request and performs the operations requested by the $user_i$

viii. Response: SNS provider returns to the $user_i$, processed encrypted result.

ix. Decryption: the $user_i$ decrypts the returned encrypted result using a private key, $\Pr k_i$.

4.1 The Proposed Solution for the Privacy Protection of Users' Data

The study employed divide and conquers to develop an algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques. It was developed based on requirements specifications. Figure 2 depicts algorithm architecture for privacy protection of users' data in SNS based on homomorphic cryptography techniques. It comprises of SNS services s_i with input mi requested by useri in SNS such as Facebook, Twitter, and other likes. The services in SNS can be defined as a set of services given by

S_x= {"friending", "news feed", "friends" ,"timeline" , "likes", "messages", "photo uploading, audio, video", "chatting", "friending", "news feed", "friends", "tagging", "tweet", "follow", search"}.

Where x=i or j, represent a given service requested by user i or j in SNS.

.

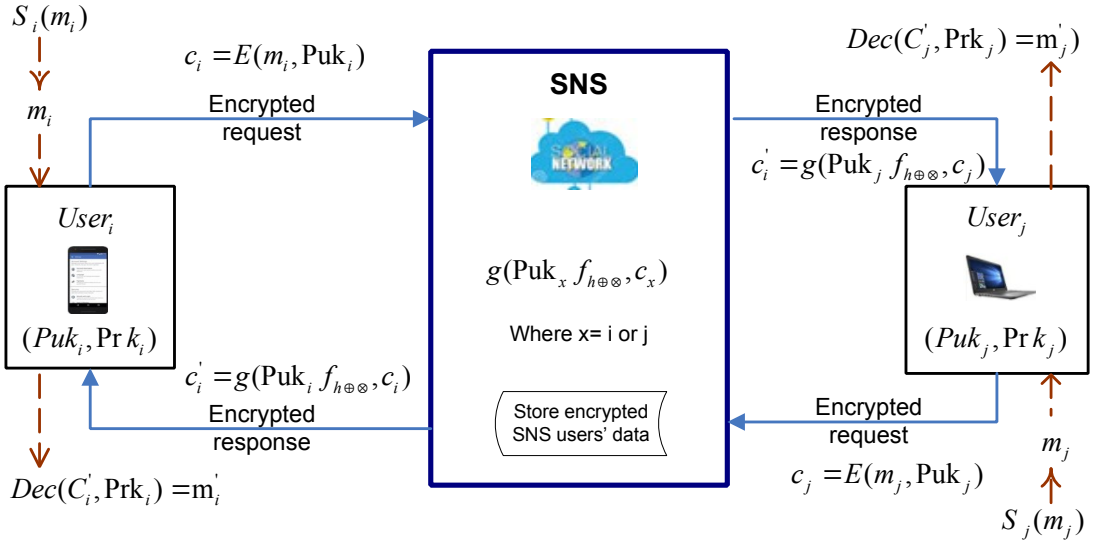

Figure 2. Algorithm architecture for privacy protection of users' data in social network systems based on homomorphic cryptographic techniques

Let M represents the plaintext space with, and let C represent the homomorphic ciphertext space with $c_x \in C$; $user_x$ represents users interacting in SNS ubiquitous computing environment; where x = i or j. From Figure 2, the following is deduced.

i. Key generation: the $user_x$ of SNS generates the public key (Puk_x) and the private key (Prk_x); where x= i or j.

ii. Input: the plain text m_x for the service is split into smaller n bit blocks of plaintext message $s_x(m_x) = m_1, m_2, ..., m_n$; where x= i or j; $m_x \in M$, M.

iii. Encryption: The $user_x$ of SNS encrypts the

plaintext m_x with the public key puk_x; and sends the encrypted data and his/her public key, Puk_x to the SNS provider server; where x=i or j.

iv. Request: the $user_x$ interacts with SNS interface to send a request to the SNS provider server to perform homomorphic operations on encrypted data.

v. Evaluation: SNS provider server processes the request from $user_x$ and performs the homomorphic operations $f_{h\oplus\otimes}$ on encrypted data c_x using the user public key Puk_x; the output is the encrypted result, $c_x^{'} = g(\mathrm{Puk}_x\, f_{h\oplus\otimes}, c_x)$, $c_x^{'} \in C$; where x=i or j.

vi. Storage: SNS provider stores the user's encrypted data $c_x^{'}$ and the user's public key Puk_x in the SNS provider database.

vii. Response: SNS provider returns to the $user_x$ processed encrypted result $c_x^{'}, c_x^{'} \in C$; where x= i or j.

viii. Decryption: the $user_x$ decrypts the returned encrypted result $c_x^{'}$ using his/her private key $\mathrm{Pr}k_x$; $Dec(C_x^{'}, \mathrm{Prk}_x) = \mathrm{m}_x^{'}, m_x^{'} \in M$; where x= i or j.

From Figure 2, using divide and conquers; the development of an algorithm for privacy protection of users' data in SNS was split into five blocks, namely algorithm block 1 for accepting input plaintext message in SNS, algorithm block 2 for generating public key and private key for users in SNS, algorithm block 3 for homomorphic encryption of message in SNS, algorithm block 4 for homomorphic evaluation of ciphertext in SNS, and algorithm block 5 for decryption of ciphertext results at user of SNS. The development of an algorithm for privacy protection of users' data in SNS was carried out for each algorithm block as summarized in Table 1.

Table 1. Algorithm for privacy protection of users' data in social network systems based on homomorphic cryptography techniques

Block 1: Accepting input plaintext message in SNS

1.Input: the plain text for the service $s_x(m_x)$

2.The message is split into smaller n bit blocks of plaintext $s_x(m_x) = m_1, m_2, ..., m_n$; where x= i or j; $m_x \in M$, M

3.For i=1 to n

{

4.Plaintext message

M= $s_x(m_x) = m_1, m_2, ..., m_n$

}

Block 2: Generating a public key and private key

5.Select the homomorphic encryption scheme, $HE = \min_{time,memory}\{HE_1, HE_2, ...HE_n\}$ with minimum complexity time and memory

6.Select key bit length to be used for generating public and private key pair

bit-length =bit-length value

7.Select certainty value for semantic security

certainty= certainty value

8.Call KeyGen (Prk, Puk) to generated private key and public key

{

9.The private key, Prk= HE.KeyGen(prkx, pukx)

10.The public key, Puk=HE. KeyGen(prkx, pukx)

}

Block 3:Homomorphic encryption of message in SNS

11.Pass the public key for the user of SNS generated by KeyGen (Prk, Puk)

12.Receive the plaintext message m_x for $user_x$

13.Encrypt message m_x for $user_x$ using public key Puk_x

14.For i=1 to n

{

15. $c_x = HE.E(\mathrm{Puk}_x, \mathrm{m}_x)$

}

16.Send the encrypted result c_x and the user public key Puk_x to SNS provider through the Internet

Block 4: Homomorphic evaluation in SNS

17.Receive ciphertext $C_x, C_x \in C$ together with the public key Puk_x from $user_x$; C=Ciphertext C_x

18.Perform required homomorphic operations on ciphertext

19.Homomorphic evaluation, $c_x^{'} = HE.eval(\mathrm{Puk}_x f_{h\otimes}, c_x)$

20.Perform homomorphic addition on ciphertext, $c_x^{'} = HE.eval(\mathrm{Puk}_x f_{h\oplus}, c_x)$

21.Perform homomorphic multiplication on ciphertext, $c_x^{'} = HE.eval(\mathrm{Puk}_x f_{h\otimes}, c_x)$

22. Give response to $user_x$ by sending processed ciphertext results, $c_x^{'} = HE.eval(\mathrm{Puk}_x, f_{h\oplus\otimes}, c_x)$
23. Store homomorphic evaluated privacy data for $user_x$ in SNS provider 's database

Block 5: Decryption of evaluation results at users of SNS

24. Received the ciphertext processed results from SNS provider for service output ciphertext results, $\mathrm{c}_x^{'} = s_x(\mathrm{c}_x^{'})$
25. $user_x$ use his/her private key Prk_x to decrypt the ciphertext $c_x^{'}$ for homomorphic evaluated service $s_x(\mathrm{c}_x^{'}) = c_x^{'} = g(\mathrm{Puk}_x, f_{h\oplus\otimes}, c_x)$
26. HomoDecrypt (c, Prk)
{
27. $m_x^{'} = Dec(\mathrm{c}_x^{'}, \mathrm{Pr}\,k_x)$ where $m_x^{'}$ is the decrypted plaintext message result, $m_x^{'} \in M$
}

4.2 Illustration of the Proposed Solution for the Privacy Protection of Users Data

The developed algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques was translated into a computer program application using java application program. The study adopted the Paillier homomorphic encryption scheme to illustrate the developed algorithm; other homomorphic encryption schemes can be substituted to achieve the same objective of privacy protection of users' data in SNS. The developed algorithm was translated into a java program with four classes. The created classes are class for accepting plaintext message from a user of SNS (InputPlaintextMessage class); class for generating keys, encrypting and decrypting (HomoCrypto class); class for performing a homomorphic evaluation for ciphertexts (HomoEva class) and main class which call methods from the other three classes in the developed algorithm (HomoAlgorithmPrivacy class). Table 2 presents the summary of a description of each class in the developed algorithm for privacy protection of users' data in SNS.

Table 2. Classes in the algorithm for privacy protection of users' data

S/N	Class	Description
1	class InputPlaintextMessage	• Accepting plaintext messages from users of SNS. It has inputMessage() function for accepting an array of plaintext messages from users of SNS
2	class HomoCrypto	• This class performs homomorphic cryptography operations which include keys generation, encryption, and decryption. • It has KeyGeneration (int bitLengthVal, int certainty) function for generating public keys and private keys for homomorphic operations. It has a bit length parameter for determining the keys generated strength. • It has certainty parameter to ensure semantic security. • It has Encryption (BigInteger m) function for encrypting a plaintext message m from a user of SNS. • It has BigInteger Decryption (BigInteger c) function for decrypting processed ci-

S/N	Class	Description
		phertext results from SNS provider.
3	class HomoEval	• This class accepts ciphertext from users of SNS, performing a homomorphic evaluation on ciphertexts (addition & multiplication homomorphic operations) and output ciphertext responses to users of SNS. • It has ciphertext() function for accepting ciphertext requests from users of SNS • It has homomorphicAddition () function for performing homomorphic addition evaluation on ciphertext from the users of SNS at the provider of SNS. • It has homomorphicMultiplication () function for performing homomorphic multiplication evaluation on ciphertext from the users of SNS at the provider of SNS.
4	Class HomoAlgorithmPrivacy	• This is the main class & it calls methods from other classes to illustrate the developed algorithm for privacy protection of users' data in SNS. • It illustrates the input phase of plaintext message m from users of SNS, keys generation phase, encryption phase, homomorphic evaluation phase (homomorphic addition and multiplication of ciphertexts) and decryption phase of evaluated ciphertexts results.

5 RESULTS AND DISCUSSION

The performance of the developed algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques was analyzed using Big O notation and controlled experiment. It is presented as follows. The study employed Big O notation and controlled experiment to analyse the performance of the proposed solution for the privacy protection of users' data in SNS. The results and discussion are as follows.

5.1 Performance Analysis Using Big O Notation for the Developed Algorithm

The study employed Big O notation to analyze and determine the performance algorithm complexity for the developed algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques. Applying Big O notation to the developed algorithm. The performance of the developed algorithm for privacy protection of users' data in SNS can be deduced as follows. Let $f(n)$ and g(n) be functions such that $f(n), g(n) \in \mathbf{Z}^{+}$. The Big O notation can be expressed as $f = O(g)$, if and only if $f(n) \leq c.g(n)$; where c is a constant, c>0. From Table 1 the Big O notation for an algorithm for privacy protection of users' data in SNS was computed by applying divide and conquers to each algorithm block (block 1, 2,…5).

Algorithm block 1: accepting input plaintext message for users in SNS

$s_x(m_x) = m_1, m_2, ..., m_n$;

m<n; Big O =O(n).

Big O for algorithm block 1= O(n).

Algorithm block 2 for generating keys (public and private keys) for users in SNS

For keys generation, Big O = O(p*q) = O(1).

Big O for Algorithm block 2, Big O=O(1).

Algorithm block 3 for homomorphic encryption of message in SNS

Big O for algorithm block 3= $O(n^2)$

Algorithm block 4: homomorphic evaluation in SNS

Big O for homomorphic evaluation,

Big O $= O(m_1 + m_2) = O(n + n) = O(n)$ for addition;

Big O $= O(m_1 * m_2) = O(n * n) = O(n^2)$ for multiplication.

Big O for algorithm block 4= $O(n^2)$.

Algorithm block 5 for decryption of evaluation results at users of SNS

$m = L(c^{\lambda} (\mathrm{mod}\, n^2)).\mu (\mathrm{mod}\, n)$; applying the Chines remainder theorem [40],

Big O $= O(L(c^{\lambda} (\mathrm{mod}\, n^2)).\mu (\mathrm{mod}\, n)) = O(n^2)$,

Big O for algorithm block 5= $O(n^2)$

Thus, the Big O notation of algorithm for privacy protection of users' data in SNS

$$= \max\{O(\text{Algorithm } block1,..5)\}$$
$$= \max\{O(n), O(1), O(n^2), O(n^2), O(n^2)\} = O(n^2)$$

Big O for an algorithm for privacy protection of users' data in SNS is $O(n^2)$. Thus, the algorithm for privacy protection of users' data in SNS has algorithm complexity of $O(n^2)$, quadratic complexity.

5.2 Performance Analysis Using Experiment for the Developed Algorithm

The performance analysis of the developed algorithm for privacy protection of users' data in SNS was carried out using a controlled experiment. The experiment was performed in two phases. Phase I was carried out to examine the performance of the developed algorithm by varying input plaintext message m, and bit-length of keys was fixed. Phase II of the experiment was carried by varying bit-length of keys and, bit-length of plaintext message m was fixed. The following materials were prepared for conducting the experiment (phase I and II).

i. Developed algorithm for privacy protection of users' data in SNS with its description
ii. Laptop with Intel Core i5 processor, 2.4 GHz, 8 GB RAM, 64-bit processor, Windows 10 Pro.
iii. Bytecode and executable program of an algorithm for privacy protection of users' data in SNS
iv. Table template for recording results (Table 3 and Table 4).

5.2.1 Experiment Phase I

In a controlled experiment in phase I, keys (public and private keys) were generated at fixed bit-length such as 1024. Input plaintext message m_x was varied from bit-length of 1 to 30. The bytecode java program for the developed algorithm was executed by varying input plaintext message (m_1 and m_2) from bit-length of 1 to 30, while bit-length for keys was fixed (bit-length=1024). The execution time in milliseconds for each corresponding plaintext message m was recorded in table template as shown in Table 3**.** For a message with n bits; the size of plaintext message is 2^n.

Table 3. Execution time of a developed algorithm for fixed keys size

bit length	1	2	3	4	5	6	7	8	9	10	11	12	..	30
m1	2	4	8	16	32	64	128	256	512	1,024	2,048	2^{12}	..	2^{30}
m2	2	4	8	16	32	64	128	256	512	1,024	2,048	2^{12}	..	2^{30}
time (ms)	100	100	100	99	94	97	101	103	96	97	97	101	..	102
$\log_{10} t$	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0		2.0

The study revealed that the logarithmic execution time ($\log_{10} t$) is a constant value for the developed algorithm for privacy protection of users' data in SNS at the fixed key size and variable bit-length of input plaintext message. The growth algorithm complexity is limited to a constant value as shown in Figure 3; when key size is fixed and the bit length of input text message is varied. This is consistent with a study by [49] which found that entropy of the graph limit for maximal entropy converges to maximizing graphon as limit approaches infinity.

In practice, the input plaintext message m is split into small n-bits blocks message size; $S(m_x) = m_1, m_2, ..m_n$. This shows that the algorithm for privacy protection of users' data in SNS based on homomorphic cryptography techniques can be used in a practical environment. Thus, the proposed solution is applicable to services which involve big plaintext message m, such as multimedia (sharing of photo, audio, and video) in SNS.

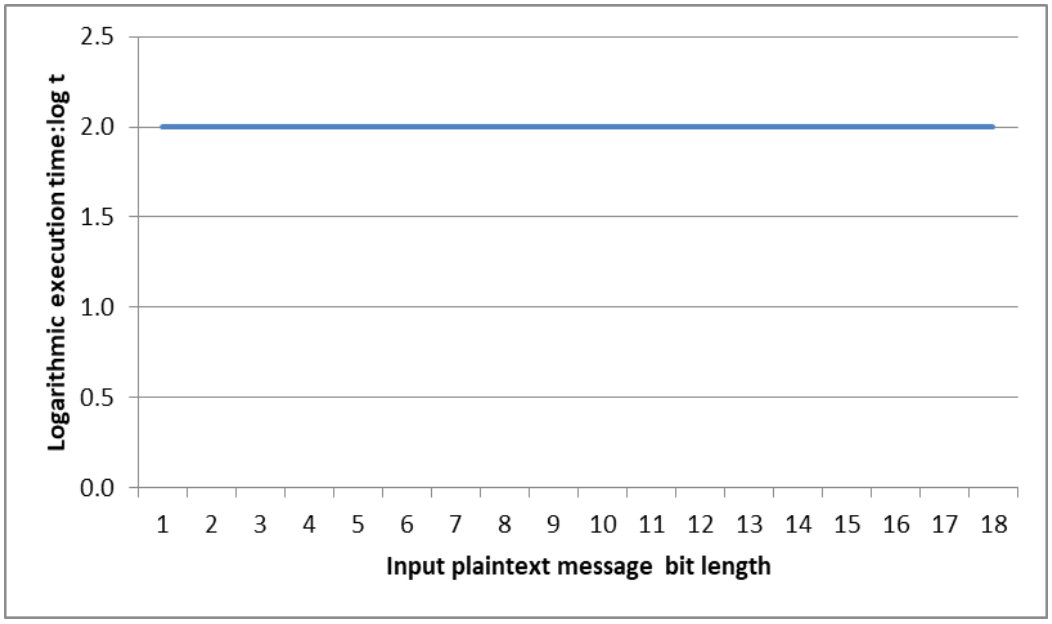

Figure 3. The execution time of the algorithm at variable input plaintext

5.2.2 Experiment phase II

The objective was to analyze the effect of varying keys size (bit-length) and execution time of the developed algorithm for privacy protection of users' data in SNS. In controlled experiment phase II, keys bit-length was varied. The bytecode java program for the developed algorithm was executed by varying keys' bit-length from 256 to 2048. The bit-length for input plaintext message (m1 and m2) was fixed as shown in Table 4. The execution time in milliseconds for each corresponding plaintext was recorded in table template as shown in Table 4.

Table 4. Variable keys size and execution time of the developed algorithm

Key size	**256**	**512**	**768**	**1,024**	**1,280**	**1,536**	**1,792**	**2,048**
time(t) (ms)	15	31	47	94	156	250	375	509
Input plaintext message, m1=33, m2=40								

The data were analyzed using MS Excel and visualized using a timeline graph as shown in Figure 4. This shows that the execution time of the algorithm for privacy protection of users' data in SNS increases with increase in bit-length of keys employed. Moreover, the results show that time execution for the developed algorithm increases in quadratic time approximately trends, y= 12.126x2 – 16.492x + 32. This result is similar to Big O performance analysis which found algorithm complexity of $O(n^2)$ for developed algorithm. The developed algorithm for privacy protection of users' data in SNS gives performance execution time of fewer than 0.05 seconds for keys size of at most 768 bit-length. Moreover, it gives execution time of fewer than 0.094 seconds for keys size of 1024 bit-length and has time execution of fewer than 0.6 seconds for keys size of 2048 bit-length.

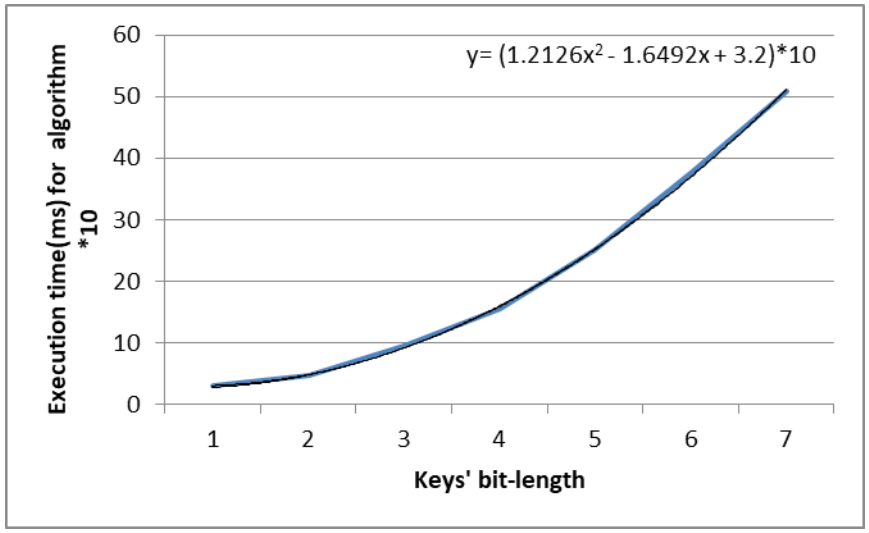

Figure 4. Keys size variation with execution time for the developed algorithm

6 CONCLUSION

The research study addressed the problematic situation of privacy violation in SNS using homomorphic cryptographic techniques. It proposes an algorithm for privacy protection of users' data in SNS based on homomorphic cryptographic techniques. The development of an algorithm for privacy protection of users' data in SNS was guided by divide and conquers techniques for design and development of the algorithm. Moreover, the study revealed that the developed algorithm has algorithm complexity of $O(n^2)$, quadratic complexity. It increases in quadratic time approximately trend. It gives performance execution time of shorter than 0.094 seconds for keys size of 1024 bit-length. In practice, the study recommends keys size of 1024 bit-length to be used for the proposed algorithm for privacy protection of users' data in SNS based on homomorphic cryptographic techniques. Thus, the proposed algorithm can be used in a real-world environment to protect the privacy of users' data in SNS. The study recommends the adoption of homomorphic cryptography techniques for services in SNS such as Facebook, Twitter, LinkedIn and the other likes. It allows manipulations and processing of ciphertexts users' data without decrypting in advance. Thus, manipulation, computation and evaluations can be carried out by untrusted third part in an untrusted environment such as SNS. The future research work recommendation is to extend the use of the proposed algorithm for privacy protection of users' data based on homomorphic cryptographic techniques to other sectors. This includes adopting e-voting through homomorphic cryptography techniques in Tanzania; adopting homomorphic cryptography techniques for privacy protection of users' data in National Internet Datacenter in Tanzania and other services accessible via untrusted environments.

REFERENCES

[1]. Conger, S., & Landry, B. J. L. (2008). The Intersection of Privacy and Security. Retrieved December 26, 2017, from https://www.researchgate.net/publication/253903294_The_Intersection_of_Privacy_and_Security

[2]. Mai, H. V. (2017). *Secure Privacy-preserving Computing Applications on Cloud Using Homomorphic Cryptography*. PhD. Thesis. The RMIT University. Retrieved from https://researchbank.rmit.edu.au/eserv/rmit:162139/Mai.pdf

[3]. Kapis, K. (2011). *Security and Privacy of Electronic Patient Records*. PhD Thesis. The Open University of Tanzania.

[4]. Venkatanathan, J., Kostakos, V., Karapanos, E., & Gonçalves, J. (2014). Online disclosure of personally identifiable information with strangers: Effects of public and private sharing. *Interacting with Computers*, *26*(6), 614–626. http://doi.org/10.1093/iwc/iwt058

[5]. Hartnett, H. A. (2016). The gift of privacy: How Edward Snowden changed the way I parent. Retrieved April 8, 2017, from http://www.salon.com/2016/07/24/the_gift_of_privacy_how_edward_snowden_changed_the_way_i_parent/

[6]. Ngambeket, G.-H. (2012). Social Networks and Privacy -Threats and Protection. *ISACA Journal*, *5*, 19–23. Retrieved from https://www.isaca.org/Journal/archives/2012/Volume-5/Documents/12v5-Social-Networks-and-Privacy.pdf

[7]. CBSNEWS. (2018). Mark Zuckerberg testimony: Facebook CEO open to regulation. Retrieved April 18, 2018, from https://www.cbsnews.com/live-news/watch-mark-zuckerberg-testimony-senate-judiciary-commerce-committee-facebook-data-breach-today-live/

[8]. Symantec. (2017). Internet Security Threat Report - ISTR. Retrieved January 20, 2018, from https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

[9]. C-SPAN. (2018). Facebook CEO Mark Zuckerberg Hearing on Data Privacy and Protection No Title. Retrieved April 18, 2018, from https://www.c-span.org/video/?c4722632/facebook-ceo-mark-zuckerberg-apologizes-data-privacy-failures

[10]. C-SPAN. (2018). Facebook CEO Mark Zuckerberg Hearing on Data Privacy and Protection No Title. Retrieved April 18, 2018, from https://www.c-span.org/video/?c4722632/facebook-ceo-mark-zuckerberg-apologizes-data-privacy-failures

[11]. Rabieh, K., Mahmoud, M., Siraj, A., & Misic, J. (2015). Efficient privacy-preserving chatting scheme with degree of interest verification for vehicular social networks. In *2015 IEEE Global Communications Conference, GLOBECOM 2015, 6-10 Dec. 2015, San Diego, CA, USA* (1–6). IEEE. http://doi.org/10.1109/GLOCOM.2014.7417514

[12]. Narendrareddy, A., & Gayathri, P. (2016). Efficient Mechanism Using Privacy Preserving Photo Sharing On SNSs. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, *23*(8), 4–7. Retrieved from http://www.ijetcse.com/wp-content/plugins/ijetcse/file/upload/docx/847Efficient-Mechanism-Using-Privacy-Preserving-Photo-Sharing-On-SNSs-pdf.pdf

[13]. Phong, L. T., Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2018). Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, *13*(5), 1333–1345. http://doi.org/10.1109/TIFS.2017.2787987

[14]. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2017). A Survey on Homomorphic Encryption Schemes: Theory and Implementation. Retrieved March 2, 2018, from http://arxiv.org/abs/1704.03578

[15]. Masoumzadeh, A. (2014). *Preserving Privacy in Social Networking Systems : Policy-Based Control And Anonymity*. Ph.D Thesis. The University of Pittsburgh School. Retrieved from http://d-scholarship.pitt.edu/22826/1/masoumzadeh_dissertation.pdf

[16]. Doröz, Y. (2017). *New Approaches for Efficient Fully Homomorphic Encryption*. PhD. Thesis. Worcester Polytechnic Institute. Retrieved from https://web.wpi.edu/Pubs/ETD/Available/etd-061417-201844/unrestricted/ydoroz.pdf

[17]. Feron, C. (2018). Fast Evaluation of Homomorphic Encryption Schemes based on Ring-LWE . In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 26-28 Feb. 2018, Paris, France* (1–5). IEEE. Retrieved from https://ieeexplore.ieee.org/document/8328693/

[18]. Belland, M., Xue, W., Kurdi, M., & Chu, W. (2017). Somewhat Homomorphic Encryption. Retrieved May 28, 2018, from https://courses.csail.mit.edu/6.857/2017/project/22.pdf

[19]. Gentry, C. (2009). *A Fully Homomorphic Encryption Scheme*. *Stanford University*. PhD Thesis. The Stanford University. Retrieved from https://crypto.stanford.edu/craig/craig-thesis.pdf

[20]. Paillier, P. (1999). Public Key Crypto Systems Based on Composite Degree Residuosity Classes. *Advances in Cryptography - {EUROCRYPT'99}*, *1592*, 223–238. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.4035&rep=rep1&type=pdf

[21]. Yi, X., Paulet, R., & Bertino, E. (2014). *Homomorphic Encryption and Applications*. Springer Cham Heidelberg, New York. http://doi.org/10.1007/978-3-319-12229-8

[22]. Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation*, *4*(11), 169–180. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.500.3989&rep=rep1&type=pdf

[23]. Harerimana, R., Tan, S. Y., & Yau, W. C. (2017). A Java implementation of paillier homomorphic encryption scheme. In *2017 5th International Conference on Information and Communication Technology (ICoIC7), 17-19 May 2017, Malacca City, Malaysia* (1–6). IEEE. http://doi.org/10.1109/ICoICT.2017.8074646

[24]. Gentry, C. (2009). Fully homomorphic encryption

using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09 , May 31 - June 02, 2009, Bethesda, MD, USA* (169–178). ACM New York, NY, USA. http://doi.org/10.1145/1536414.1536440

[25]. Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In *Proceeding Eurocrypt'10 Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques, May 30 - June 03, 2010 , French Riviera, France* (24–43). Springer-Verlag Berlin, Heidelberg. http://doi.org/10.1007/978-3-642-38348-9_20

[26]. Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient fully homomorphic encryption from (standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, 22-25 Oct. 2011, Palm Springs, CA, USA* (97–106). IEEE. http://doi.org/https://doi.org/10.1109/FOCS.2011.12

[27]. Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory (TOCT) - Special Issue on Innovations in Theoretical Computer Science 2012 - Part II*, *6*(3), 309–325. http://doi.org/10.1145/2633600

[28]. Hoffstein, J., Lieman, D., Pipher, J., & Silverman, J. H. (1998). NTRU: A ring-based public key cryptosystem. Retrieved March 20, 2018, from https://pdfs.semanticscholar.org/252a/18dcc149ed0fd2627936446da05fc836a197.pdf

[29]. Gentry, C., Sahai, A., & Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8042 LNCS*(PART 1), 75–92. http://doi.org/10.1007/978-3-642-40041-4_5

[30]. Makkaoui, K. E. L., & Ezzati, A. (2015). Challenges of Using Homomorphic Encryption to Secure Cloud Computing. In *2015 International Conference on Cloud Technologies and Applications (CloudTech), 2-4 June 2015, Marrakech, Morocco* (1–7). IEEE. http://doi.org/10.1109/CloudTech.2015.7337011

[31]. Yi, X., Bertino, E., Rao, F. Y., & Bouguettaya, A. (2016). Practical privacy-preserving user profile matching in social networks. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE), 16-20 May 2016, Helsinki, Finland* (373–384). IEEE. http://doi.org/10.1109/ICDE.2016.7498255

[32]. Ho, A., Maiga, A., & Aïmeur, E. (2009). Privacy protection issues in social networking sites. In *2009 IEEE/ACS International Conference on Computer Systems and Applications,10-13 May 2009, Rabat, Morocco* (271–278). IEEE. http://doi.org/10.1109/AICCSA.2009.5069336

[33]. Bodriagov, O. (2015). *Social Networks and Privacy*. PhD. Thesis. The KTH Royal Institute of Technology. http://doi.org/10.1007/978-1-4939-1740-2_2

[34]. CNN. (2018). Why meeting with Zuckerberg won't fix the Facebook problem. Retrieved April 18, 2018, from https://edition.cnn.com/2018/04/10/opinions/zuckerberg-congress-facebook-opinion-james-ball-intl/index.html

[35]. Mshangi, M., Kapis, K., & Cosmas, J. (2018). Enhancing Privacy in Social Networks Systems using Homomorphic Cryptographic Techniques in Ubiquitous Computing Environment. In *2018 IST-Africa Week Conference (IST-Africa), 9-11 May 2018, Gaborone, Botswana* (1–9). IEEE. Retrieved from https://ieeexplore.ieee.org/document/6809839/

[36]. Moore, G. E. (1970). Moore's Law. Retrieved April 22, 2018, from http://www.mooreslaw.org/

[37]. Debenedictis, E. P., Badaroglu, M., Chen, A., Conte, T. M., & Gargini, P. (2017). Sustaining Moore's Law with 3D Chips. *Computer*, *50*(8), 69–73. http://doi.org/10.1109/MC.2017.3001236

[38]. Basu, A., Corena, J. C., Kiyomoto, S., Marsh, S., Vaidya, J., Guo, G., … Miyake, Y. (2014). Privacy preserving trusted social feedback. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14, Gyeongju, Republic of Korea, March 24 - 28, 2014* (1706–1711). ACM New York, NY, USA. http://doi.org/10.1145/2554850.2554860

[39]. Zhu, H., Du, S., Li, M., & Gao, Z. (2013). Fairness-aware and privacy-preserving friend matching protocol in mobile social networks. *IEEE Transactions on Emerging Topics in Computing*, *1*(1), 192–200. http://doi.org/10.1109/TETC.2013.2279541

[40]. Cormen, T. H., Leiserson, C. E., & Rivest, R. L. C. S. (2002). *Introduction to Algorithms* (Second Edi). Cambridge, Massachusetts London, England McGraw-Hill: The Massachusetts Institute of Technology.

[41]. Hou, D., & Zhang, W. (2017). Multi - Warehouse Location of Logistics Based on Dijkstra and Divide-and- Conquer Algorithm. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID), 9-10 Dec. 2017, Hangzhou, China* (442–447). http://doi.org/10.1109/ISCID.2017.204

[42]. Faitelson, D., & Tyszberowicz, S. (2017). UML Diagram Refinement (Focusing on Class-And Use Case Diagrams). In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 20-28 May 2017, Buenos Aires, Argentina* (735–745). IEEE. http://doi.org/10.1109/ICSE.2017.73

[43]. Haoues, M., Sellami, A., & Ben-Abdallah, H. (2017). Predicting the functional change status in UML activity diagram from the use case diagram. In *2016 IEEE/ACS 13th International Conference of Computer Systems*

and Applications (AICCSA),29 Nov.-2 Dec. 2016, Agadir, Morocco. IEEE. http://doi.org/10.1109/AICCSA.2016.7945783

[44]. Mirghani, M., & Madane, H. O. (2017). Evaluation of the quality of encoded Quran digital audio recording. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), 16-18 Jan. 2017, Khartoum, Sudan* (1–4). IEEE. http://doi.org/10.1109/ICCCCEE.2017.7867657

[45]. Zheng, P., & Huang, J. (2013). An Efficient Image Homomorphic Encryption Scheme with Area Chair : Rita Cucchiara. In *Proceeding MM '13 Proceedings of the 21st ACM international conference on Multimedia, October 21 - 25, 2013, Barcelona, Spain* (803–812). ACM New York, NY, USA. http://doi.org/10.1145/2502081.2502105

[46]. Westwate, R., & Furht, B. (1997). Real-Time Video Compression: Techniques and Algorithms. Retrieved May 20, 2018, from https://www.researchgate.net/publication/247345079_Real-time_video_compression

[47]. Xu, K., Huang, B., Liu, X., Tu, X., Wu, Z., Yan, Z., … Li, Y. (2018). A Low-power Pyramid Motion Estimation Engine for 4K @ 30fps Realtime HEVC Video Encoding. In *2018 Systems of Signals Generating and Processing in the Field of on Board Communications,14-15 March 2018, Moscow, Russia* (2–5). IEEE. http://doi.org/10.1109/SOSG.2018.8350571

[48]. Ziad, M. T. I., Alanwar, A., Alzantot, M., & Srivastava, M. (2017). CryptoImg: Privacy preserving processing over encrypted images. In *2016 IEEE Conference on Communications and Network Security (CNS), 17-19 Oct. 2016, Philadelphia, PA, USA* (570–575). IEEE. http://doi.org/10.1109/CNS.2016.7860550

[49]. Hatami, H., Janson, S., & Szegedy, B. (2017). Graph properties, graph limits, and entropy. *Journal of Graph Theory*, (February), 1–22. http://doi.org/10.1002/jgt.22152