Privacy Preserving Algorithm using Chao-Scattering of Partial Homomorphic Encryption

Saja J Mohammed 1* and Dujan B Taha 2

1 Department of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.
2 Department of Software, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.

Sj_alkado@uomosul.edu.iq, dr.dujantaha@gmail.com

Abstract. A big challenge in privacy-preserving is found when trying is to share data while protecting the information of personally identifiable, such as phone numbers, email addresses, and biometric. The same challenges have appeared when keeping sensitive data remotely in complete privacy, without recognizing them from any unauthorized persons. In such cases, Homomorphic encryption is the highlight and considered the most suitable solution. The goal of this paper is to keep the privacy of the data stored in a database system using a new Chao-modification to the partial homomorphic encryption algorithm. It uses the chaotic system to scatter the symbols of plain data before direct them to the encryption algorithm. This will be appending an additional level of security to partial homomorphic encryption algorithms that helps them to be more resisting attacking.

1 Introduction
For a long time ago, security became the most important characteristic that researchers try to supply while applying various methods and performing various techniques. Encryption has offered the most effective method to perform in order to ensure certain security, protection, and privacy to various present systems [1].

In the other side, databases are used in widely in our daily life, such as, Telephone directory, Dictionary, Library catalogue and so, where the data are arranged according to a classified order or alphabetically. A database can be defined as a regular set of related information. The organized information (or database) render as a base from which required information can be restored or decisions made by more recognizing or processing the data [2]. The database system is not anything greater than a record-keeping system which depending on the computer totally. In another word, it is a system whose total purpose is to document and maintain information/data [2].

Due to using databases in many domains to store information, it is necessary to ensure the security of data which stored in databases [3]. Data Privacy troubles are found wherever uniquely identifiable data bounded on a person or persons are gathered and stored, in digital form or anything else. The origin cause of privacy issues can be improper detection control of personal data [4].

2 Homomorphic Encryption
One of the most popular fields in cryptography now which used for keeping privacy is the homomorphic encryption. It has got major regard due to the uncountable features it provides.
Homomorphic encryption, applied to public clouds, presents the concept of a service provider who can carry out different operations on encrypted data without the need to decrypt it beforehand. Therefore, this encryption scheme is used in various systems, for various goals, it shows great fitness and adaptability while being applied in various ways [1]. The Homomorphic Encryption operations can be shown in the steps below [5]:

1. **Key generation**: The client generates the public key and the secret key (pk and sk respectively).
2. **Encryption**: The client encrypts the data with an encryption key then sends the encrypted data to database storage.
3. **Storage**: The encrypted data are stored in the server database.
4. **Request**: The client transmits a request to the server to carry out operations on encrypted data.
5. **Evaluation**: The processing server executes the request and carries out the operations requested by the client.
6. **Response**: The server returns to the client the processed result.
7. **Decryption**: The client decrypts the returned result using sk.

Figure 1 explains these steps.

![Figure 1. The Homomorphic Encryption Steps.](image)

### 2.1 Partial homomorphic encryption

A cryptosystem is called "partially homomorphic Encryption (PHE)" if it shows either multiplicative or additive homomorphism, but not both. Some of the most popular examples of partially homomorphic cryptosystems are: RSA and ElGamal (both are multiplicative homomorphism), and Paillier (additive homomorphism) we can simplify the homomorphic property as follows [6][7][8]:

- **Additive Homomorphic property**: HE is said to be additively if:

  \[
  \begin{align*}
  Enc(x \oplus y) &= Enc(x) \otimes Enc(y) & (1) \\
  Enc(\sum_{i=1}^{l} m_i) &= \prod_{i=1}^{l} Enc(m_i) & (2)
  \end{align*}
  \]

- **Multiplicative Homomorphic property**: HE is said to be multiplicatively if:

  \[
  \begin{align*}
  Enc(x \otimes y) &= Enc(x) \otimes Enc(y) & (3) \\
  Enc(\prod_{i=1}^{l} m_i) &= \prod_{i=1}^{l} Enc(m_i) & (4)
  \end{align*}
  \]
2.1.1 RSA Algorithm. RSA is a public key cryptosystem. It classified as Multiplicative HE algorithm [7] [8]. In this paper two different types of RSA are used, deterministic RSA and probabilistic RSA. Probabilistic encryption indicates the use of randomness in encryption algorithms, so when encrypting the same plain message at different times, it will produce different cipher texts [9]. This type of encryption is resisting of many types of attacks [9], figure 2 display steps of RSA algorithm [7][9].

![RSA Algorithm](image1)

(a) Deterministic RSA  
(b) Probabilistic RSA

2.1.2 Paillier Algorithm. Paillier algorithm is also public key cryptosystem. It classified as additive HE algorithm [7][8]. Figure 3 display steps of Paillier algorithm [7].

![Paillier Algorithm](image2)
3 Chaotic System
A chaotic is a very known system for its sensitivity to initial parameters [10][11]. It can be defined as "the study of the unforeseen, of the nonlinear and the eccentric". It has high complexity and nonlinear behaviours and focuses on the behaviour of dynamic systems which are so influenced by initial values. This sensibility is commonly known as "the butterfly effect". Due to its above features, chaos was utilized in the encryption system very well. When initial values are set to a specific system, the future state of the system can be expected but it is impossible to forecast it for the long term [11].

In the last few years, chaos has been used to build-up hundreds of cryptographic primitives, the most popular usages are in hash functions, encryption algorithms, and watermarking[11]. The main advantage of using chaos is noted by a chaotic signal behaviour which looks like noise for unauthorized users, in addition to some interesting features, like mixing and sensitivity to initial conditions, which can ensure achieving Shannon's requirements of diffusion and confusion [12][13].

Many chaotic system are known, in this paper Lorenz system is selected to generate a pseudo-random chaotic number for confusion and diffusion procedure [14][15]. One of the first chaotic systems is the Lorenz system that discovered in 1963 by Edward N. Lorenz (1917-2008), the American meteorologist. The mathematical paradigm of this system consists of three-dimensional nonlinear ordinary differential equations and described by seven terms with two quadratic nonlinearity and described as the equation below (equation 5, 6, and 7 respectively) [16]:

\[
\begin{align*}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= r x - y - xz \\
\dot{z} &= xy - bz
\end{align*}
\]

In which \(x_i \in \mathbb{R}^3\) is the state variables, and \(a_i > 0, i=1,2,3\) are the positive parameters of the system. A system is sensitive to initial condition and has chaotic attractors with one positive Lyapunov exponents, \(LE_1=0.24784\) under the parameters \(a1=10, a3=8/3\) and \(a2=28\) [16] as depict in figure 4.

![Figure 4. The attractor of Lorenz system in: x1,x2,x3 space , x2,x3 plane.](image)

4 Related Works
In the past era, many papers concerned with information security scope were written, such as [8][11][12][17] and so. Many of them are interested in modifying the current algorithms to improve them and obtain better results in a certain direction, like time of encryption, algorithm complexity, or achieving higher secrecy. RSA and Paillier encryption algorithm got their share of these adjustments, Table 1 explain the found modifications to these two algorithms in the last years.
Table 1. Related work of modified RSA/Paillier papers.

| Year | Author/s | Paper title | The suggested improvement |
|------|----------|-------------|--------------------------|
| 2013 | “Amare Anagaw Ayel and Dr. Vuda Sreenivasarao” | “A Modified RSA Encryption Technique Based on Multiple public keys”[18] | Used two public key pairs and used several mathematical logic instead of sending the value of public key directly. |
| 2013 | “Rahul Yogi and Mrs. Pushpa.G” | “An Enhanced Paillier’s Algorithm Using Homomorphic Encryption”[19] | Concatenating Paillier’s process the principle of the cryptographic Thresholding |
| 2014 | Jainul Abudin, etal | “Modified RSA Public Key Cryptosystem Using Two Key Pairs”[20] | Used two key pairs, one of them is a small size key pair for data encryption, and another is a large size key pair used to encrypt the small size key pair component |
| 2017 | “M. Thangavel, and P. Varalakshmi” | “Improved secure RSA cryptosystem for data confidentiality in cloud”[21] | Proposed an “improved secure RSA cryptosystem (ISRSAC)” which increase the complexity when trying to factorize the value of modulus ‘n’. |
| 2019 | Israa Al_Barazanchi, etal | “Modified RSA-based algorithm: a double secure approach”[22] | Added some complexity the key generation process by adding new arithmetic operation and appending a comparison process before using the RSA keys. |
| 2020 | Pallavi and Dr. Sandeep Joshi | “An Efficient Paillier Cryptographic Technique for Secure Data Storage on the Cloud”[23] | Used Paillier algorithm to develop an approach that can reduce the information measure over the file transmission and can create additional economical system. |

5 The Proposed Algorithm

In the proposed algorithm, a Chao-scattering approach is added to partial homomorphic encryption to ensure an additional level of security by scattering the ASCII code table before encrypting the plain message by the PHE chosen algorithms.

The first idea in this work is to use ASCII code value for each non-numeric character in the plain message instead of its alphabet order as it's known. The ASCII code value will be taken from a generated table, which will be Chao-scattering ASCII table. The step of the proposed algorithm can be explained in the following steps (It is worth noting that the decrypt operation is the inversion case of these steps):
Input: Chaotic initial parameter, plain message
Output: Encrypted message

Step 1. Input the initial parameters to the chaotic system (Lorenz chaotic system), turn the system, and wait for results. The results will be three floating one dimensional arrays (X, Y, and Z).

Step 2. Rearrange the three arrays (X, Y, and Z) then begin to get data from them in the zigzag format as figure 5 explained. As a result, a new single dimension array will be generated; this new one will help us to create a scattering index for the ASCII code table.

Step 3. Perform the pre-processing procedure on the new 1D array to get the final one (scatter array), the pre-processing procedure includes: fix array to get integer value only, multiply it by a constant value to reduce the duplicated value, remove duplicates, apply absolute operator, the result will be called scatter array.

Step 4. Sort scatter array to get Sort array, now from the difference between the indexes of scattering array and sort array we get the scattering indexes that used to scramble the table of code, the sort array will be considered as the original ASCII table index, where the scattered one will be the scattering index.

Step 5. Accept the plain message as character symbols.

Step 6. Convert plain message to a number according to the new index of the ASCII table.

Step 7. Use the Scattered index to scatter a numerical input also.

Step 8. Apply PHE algorithm on the scattering message

---

The proposed algorithm is applied to data stored in the MySQL server database system. The database contains information of an employee. Fields which have been taken into consideration are: id, employee name, and the salary. Now when trying to save encrypted data in the database the following steps will be executed:

1) Initialization step, it will be performed once at the beginning of work only, which includes:
   • Configure and create the database table (if not exist).
   • Choose one of the PHE algorithms (Deterministic /Probabilistic RSA, or Paillier) to encrypt with it.
   • Generate keys of encryption and decryption according to the chosen algorithm and the entire key length.
   • Turn on the chaotic system and get the scatter array.

2) Input data (id, employee name, salary).

3) Convert the data by the suggested Chao-Scatter algorithm that explained above.

4) Direct the result to the chosen PHE algorithm to encrypt with.

5) Save the encrypted data in the database table (each one to its specific fields) using the “INSERT INTO” query instruction.
6) When trying to fetch data from a database table, firstly execute the "SELECT – FROM" query then fetch the required data. The goal of this paper is to keep the privacy of the data stored in the database system. This will be gain when expecting that the encrypted data became in an unauthorized person. Therefore, any computation such as the addition or the multiplication operation on the numerical encrypted data can performed using homomorphic encryption successfully.

In traditional systems, when an employee want to execute a computation operation on his encrypted data he need to give the decryption key to server-side. The server side will decrypt the stored data, perform calculation then encrypt the result and store them finally again. With homomorphic encryption, the operation is done on the encrypted data without knowing the two cipher keys or tries to decrypt anything. This will be kept the privacy of the employee data stored in the database system at server side.

6  Practical Examples and results

Practically the project is performed on MySQL server using the three PHE algorithms and using the proposed algorithm. The examples below explain that in some details:

6.1  Create Chao-Scattering array
Using Lorenz equations (equ. 5, 6, and 7), determine the initial condition then turn the system on. We get 3 arrays some of their elements are:

X=[ 0. , 0.1 , 0.189 , 0.2708078, 0.34853155, 0.42475452,, 0.50167291, 0.58120506, 0.66507885, 0.7549022, 0.85222095,………..]
Y=[ 1., 0.99, 1.007078, 1.04804529, 1.11076127, 1.19393836,1.29699444, 1.41994293, 1.56331237, 1.72808975, 1.91568244, …………]
Z=[ 1.05,1.0219965, 0.99572985, 0.97107712, 0.94801668, 0.92660443, 0.90696319, 0.88928115, 0.87381681, 0.86090937, 0.85099431,…………………]

After pre-processing procedure and applying zigzag form on them the resulted array is:
New= [0, 22, 38, 4, 25, 36, 8, 32, 34, 1, 23, 37, 5, 27, 35, 9, 35, 33, 3, 24, 36,………..]

Finally we get two arrays the first for original ASCII table index and the second for scattered one.

Original array: [0, 22, 38, 4, 25, 36, 8, 32, 34, 1, 23, 37, 5, 27, 35, 9, 35, 33, 3, 24, 36,………..]
Scatter array: [0, 1, 3, 4, 5, 7, 8, 9, 11, 22, 23, 24, 25, 27, 29, 32, 33, 34, 35, …………]

6.2  PHE Encryption
Choose PHE algorithm and generate their keys, then direct the result to the chosen PHE algorithm to encrypt the data. The result will be store in MYSQL server using specific query language:

6.2.1 Deterministic RSA(DRSA). Key pairs (length=50 bit): ((809464169135917, 51446721154461339677232641569), (374241015738765855246164817493, 51446721154461339677232641569))
Id:1 Employee name: 'Momen Ali'
Salary : 450
The encrypted row by Chao-scattering DRSA:
Id=1 Employee name='22720036226935128327686808', Salary= 400509671486043299518135695928, 'DRSA'

6.2.2 Probabilistic RSA(PRSA). Key pairs (length=12 bit): ((3479, 8218283), (6984839, 8218283))
r=14
Id:1 Employee name: 'Ali'
Salary : 200
The encrypted data by Chao-scattering PRSA:
Id=1, Employee name='(3989672, 7419567)', Salary= (3989672, 7819764), 'PRSA'

6.2.3. Paillier. Key pairs (length=50 bit):((75994154880954348636433865911, 
207096409117282939672667301741868355695552712463389234795346),
(379972257440476298998501584756, 437596639608369959495274381397))

Id:1
Employee name: 'Momen Ali'
Salary: 550

The encrypted row by Chao-scattering Paillier:
Id=1, Employee name='46154883130403692452710703239871227954639745314544859255941',
salary= 4191625356788365051043841977935428179211242960618783706457
According to equ.(1)
Enc(550 ⊕ 5) = Enc(550) ⊗ Enc(5)
i.e.
Dec(Enc(550 ⊕ 5)) = Enc(550) ⊗ Enc(5)
555                     =         555

6.3 Apply Homomorphic Property:
6.3.1. Additive. Try to add the salary encrypted by Paillier algorithm, this will be equivalent to:
550+5 = 555
550 = 4191625356788365051043841977935428179211242960618783706457
5 = 55436312473673458931828316685435761340032502458468668072797
According to equ.(1)
Enc(550 ⊗ 5) = Enc(550) ⊗ Enc(5)
i.e.
Dec(Enc(550 ⊗ 5)) = Enc(550) ⊗ Enc(5)
555                     =         555

6.3.2. Multiplicative. When trying to multiply the salary encrypted by DRSA/PRSA algorithm, this will be equivalent to:
450*2 = 900
450 = 400509671486043299518135695928
2 = 137844079460427484125811007398
According to equ.(3)
Enc(450 ⊗ 2) = Enc(450) ⊗ Enc(2)
i.e.
Dec(Enc(450 ⊗ 2)) = Enc(450) ⊗ Enc(2)
900                     =         900

The proposed Algorithm runs at various parameters (key size, plain-data size, plain-data type) and the time of the execution is calculated. Note that the plain-size can increase as the key size increase (Directly Relative Dependencies). Figure 6. explains the time of key generation for the three algorithms. Where figure 7 explains the time of encryption and decryption in DRSA and Paillier algorithms. In this figure PRSA is neglected because it takes time more than the other two algorithms. The time of execution in PRSA became very long when the key size is exceeds 15 bit. Figure 8 explains time consumed by PRSA algorithm in small key size (5, 8, 10, 12, and 15) where values of ‘r’ are: 678, 4905, 91078, 3678856, and 323468291 respectively. Time in the three algorithms is depending on the random generated key and its length. Therefore the time may be differed in each run.

![Figure 6. The time of key generation](image)
Conclusion

The proposed algorithm append new level of secrecy when add chao-scattering of message code by using the proposed chao-scattering ASCII table. Using chao-scatter is preferred to normal random function due its chaotic behaviour and the provided flexibility to produce many random sequence from single system. Also The three proposed algorithms shows stability in execution time when generate their keys, time is increased when key size is increased as explained in figure 6. When calculate the time of encryption and decryption process , the proposed DRSA and Paillier algorithms shows stability too as explained in figure 7. As well as, they achieve the PHE properties as well to keep the privacy of database system in MYSQL server, and therefore they achieve the required privacy preserving mechanism to keep the data secure in remote data base system.

The proposed PRSA algorithm, in the other hand, is a very suitable and resists algorithm against chosen-plaintext attack. But practically PRSA takes a long time to encrypt and especially for decrypt when compared with the proposed DRSA and Paillier algorithm. This belongs to the multi exponential operation, of ‘r’ in encryption process and ‘d’ in decryption operation. As declared above ‘r’ and ‘d’ is increased when the key size is increased, this will has an effect to encryption and decryption time.

As the explained example(in section 6) , if PRSA has a 12-bit key it takes: 0.0009989738464355469 seconds for key generating, 0.006995677947998047 seconds for encrypting when ’r’= 14, 394.99323177337646 seconds for decrypting operation.

In addition to that, when trying to save the PRSA encrypted data in the remote database system, we must append other fields to the database table since the cipher text contain two cipher texts (c1 and c2), in order to retrieve and decrypt the data correctly from the database. Therefore, in this application, we don't advise using the PRSA algorithm with the proposed modification to encrypt data using long key size, if the DRSA is working well.

Acknowledgements
The authors are very grateful to the University of Mosul/ College of Computer Science and Mathematics for their provided facilities, which helped to improve the quality of this work.

References:
[1] Alaya, B., Laouamer, L. and Msilini, N. 2020 Homomorphic encryption systems statement: Trends and challenges Comp. Sci. Rev. 36 100235
[2] Avi S Korth H and Sutarashan S 2005 Database System Concepts Fifth Edition McGraw-Hill, ISBN 0-07-295886-3.
[3] Deore, P and Chaudhari T 2017 Hybrid Encryption for Database security Inter. Res. J. Eng. Technol. 4
[4] Bertino E Byun J and Li N 2005 Privacy-preserving database systems Found. Sec. Ana. Des. III Springer, Berlin, Heidelberg 178
[5] El Makkauki K Ezzati A and Hssane A 2015 Challenges of using homomorphic encryption to secure cloud computing Inte. Con. Cloud Technol. App.(CloudTech) IEEE
[6] Morris L 2013 Analysis of partially and fully homomorphic encryption Roch. Inst. Technol
[7] Patel N Oza P and Agrawal S 2019 Homomorphic cryptography and its applications in various domains Inter. Conf. Innovat. Comp. Comm. Springer, Singapore 269-78
[8] Mohammed S. and Taha D. From Cloud Computing Security towards Homomorphic Encryption: A Comprehensive Review TELKOMNIKA Telecomm. Comp. Elec. Cont. on press
[9] Geng Y Li S and Zhou S 2019 Probabilistic RSA with Homomorphism and Its Applications JJ Net. Sec. 21 1042-1053
[10] Sun Sh 2018 A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scattering and bit-level scattering IEEE Photonics J. 10 1-14
[11] Al-Khateeb Z and Jader M 2020 Encryption and hiding text using DNA coding and hyperchaotic system Indonesian J. Elect.1 Eng. Comp. Sci. 19 766-74
[12] Mouafak A Jusem S and Jader M 2013 Apply new algorithm for chaotic Encryption using CBC&CFB Australian J. Basic App. Sci. 7 221-28
[13] Elashry I El-Shafai W Hasan E El-Rabaie S Abbas A Abd El-Samie F El-sayed H and Faragallah O 2020 Efficient chaotic-based image cyrptosystem with different modes of operation Multimedia Tools Appl.79 20665-87
[14] Khan F Ahmed J Khan J Ahmad J and Khan M 2017 A novel image encryption based on Lorenz equation, Gingerbreadman chaotic map and S 8 permutation J. Intell. Fuzzy Sys. 33 3753-65
[15] Sharma N Saini I Yadav A and Singh P 2017 Phase-image encryption based on 3D-Lorenz chaotic system and double random phase encoding 3D Res. 8 1-17
[16] Al-Azzawi S 2012 Stability and bifurcation of pan chaotic system by using Routh–Hurwitz and Gardan methods App. Math. Sci. 219 1144-52
[17] Mohammed S. 2017 Using biometric watermarking for video file protection based on chaotic principle Inter. J. Comp. Sci. Info. Sec. 15 201-6
[18] Ayele A and Sreenivasarao V 2013 A Modified RSA Encryption Technique Based on Multiple public keys Inter. J. Innovat. Res. Comp.Eng. 1 859-64
[19] Yogi R and Pushpa G 2013 An Enhanced Paillier’s Algorithm Using Homomorphic Encryption Thresholding Inter. J. Eng. Res. Comp. Technol. 2 2447-54
[20] Abudin J Keot S Malakar G Borah N Rahman M 2014 Modified RSA PublicKey Cryptosystem Using Two Key Pairs Inter. J. Comp. Sci. Info. Technol. 5 3548-50.
[21] Thangavel M and Varalakshmi P 2017 Improved secure RSA cryptosystem for data confidentiality in cloud Inter. J. Info. Sys. Change Manag. 9 261-77
[22] Al-Barazanchi I Shawkat Sh Hameed M Al-badri Kh 2019 Modified RSA-based algorithm: A double secure approach TELKOMNIKA Telecomm. Comp. Elec. Cont. 17 2818-25
[23] Joshi S 2020 An Efficient Paillier Cryptographic Technique for Secure Data Storage on the Cloud Inter. Conf. Intell. Comp. Cont. Sys. IEEE