ขั้นตอนวิธีสำหรับการเข้ารหัสข้อมูลบนฐานข้อมูลแบบกระจาย

ธัญพร ศรีดอกไม้[1*] สมชาย ปราการเจริญ[2] และ นลินภัสร์ ปรวัฒน์ปรียกร[3]

บทคัดย่อ

งานวิจัยฉบับนี้ได้ทำการพัฒนาขั้นตอนวิธีสำหรับการเข้ารหัสข้อมูลบนฐานข้อมูลแบบกระจายเพื่อแก้ไขปัญหาการรั่วไหลของข้อมูลอันเกิดจาก DBA ทุจริตได้ โดยจะนำมาใช้เทคนิคการเข้ารหัสแบบ Homomorphic Polynomials Ring ร่วมกับSecret Sharing เพื่อแยกและกระจายชิ้นส่วนของข้อมูล ประโยชน์ที่จะได้รับจากงานวิจัยชิ้นนี้คือ การลดความเสี่ยงและความเสียหายที่อาจเกิดจากการขโมยและการใช้งานระบบในทางที่ผิด ทำให้ผู้ใช้มั่นใจในระบบรักษาความมั่นคงของข้อมูลมากขึ้นว่าสามารถปกป้องข้อมูลของตนจากภัยทั้งภายในและภายนอกองค์กร ส่งผลให้ทำงานได้อย่างมีประสิทธิภาพดีขึ้นกว่าเดิม

คำสำคัญ: การเข้ารหัส, การเข้ารหัสโฮโมมอร์ฟิค, การเข้ารหัสโฮโมมอร์ฟิคแบบแพริเออร์, การแบ่งปันความลับ, การแปลงค่า

[1] นักศึกหลักสูตรปรัชญาดุษฎีบัณฑิต ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยี พระจอมเกล้าพระนครเหนือ
[2] รองศาสตราจารย์ ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
[3] ผู้ช่วยศาสตราจารย์ ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
[*] ผู้นิพนธ์ประสานงาน โทร. 08-5244-3764 อีเมล: ladytanyaorm@gmail.com.

ACDD for Protecting Information leakage

Tanyaporn Sridokmai[1*] Somchai Prakancharoen[2] and Nalinpat Porrawatpreyakorn[3]

Abstract

The objective of research was to develop analgorithm for cryptic distribute databases (ACDD) for distributed database management of the encrypted data to solve the problem of information leakage caused by fraud DBAs. The used techniques were Paillier homomorphic Polynomials ring, Shamir's Secret Sharing scheme and transformation graph. The Advantages of this algorithm were that it can solve the problem of dishonest dealers and dishonest participants who try to deceive other participants data and enables perform calculations on encrypted data without decryption on which the calculation was carried out, with respect of the data confidentiality.

Keywords: Cryptosystem, Paillier encryption, Homomorphic encryption, Secret sharing, Transformations

[1] Doctoral Degree Student, Department of Information Technology, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok

[2] Associate Professor, Department of Computer science Faculty of applied science, King Mongkut's University of Technology North Bangkok

[3] Assistant Professor, Department of Information Technology, Faculty of Information Technology King Mongkut's University of Technology North Bangkok

* Corresponding Author Tel. 08-5244-3764 e-mail: ladytanyaorm@gmail.com.

1. Introduction

The research was arose from the problem that stored data in the databases were deliberately revealed by the client while data processing. This cause vulnerability as the DBA can in-depth access to the stored data. Also the whole data record was kept in only one database engine. Therefore, to solve this vulnerable threat, the stored data in the database should be encrypted. Moreover, the data should be divided into distinct secret parts then sent them to be stored in separate database engines rather than single database engine. So that when data is requested, each part is retrieved from different sites to be reconstructed to restore the original data. In addition, during the data processing period as the client updates data, the data will still be concealed. To solve this problem, Homomorphic Paillier Polynomial Ring method is used for Data Cryptosystem, the sharmir's secret sharing scheme method is used for separation of the data, and Transformation Polynomial Graph is used for Transformation of data in which this step was conducted before encrypting data under Administrator responsibility.

2. Objectives

2.1 To develop new model algorithms for cryptic distribute databases (ACDD) for distributed database management of the encrypted data

2.2 To test and compare the new model algorithm and the existing data encryption in performance and processing by using Cryptographic Algorithm Metrics

3. Research Methodology

Research method was shown in the Figure 1

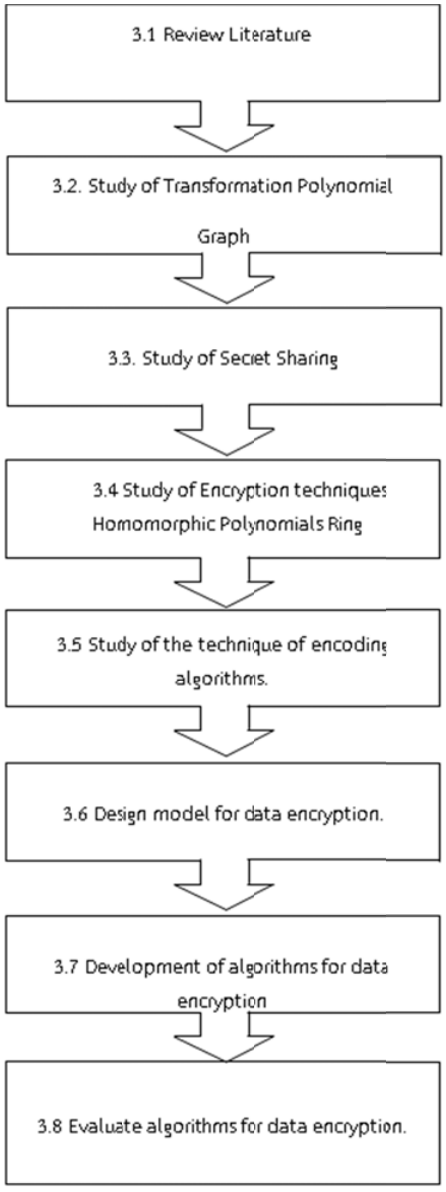

Figure 1 Research Methodology

3.1 Step I : Literature Review

In the past decade, with the advance of information technology, the need for data security increases steadily. Information leakage whether by accident or deliberation is a major problem. Therefore, to prevent the theft from DBA, data encryption is needed from plaintext to cipher text and further to coding and

decoding with keys [1], [2]. However, there are still some problems for the existing data encryption. Therefore, the research aimed to develop new model algorithms for cryptic distribute databases (ACDD) for distributed database management of the encrypted data.

3.2 Step II : Review Theory of Graph and Transformations

3.2.1 THE GRAPH OF ELEMENT FUNCTION

Element functions of graph are algebraic function: The algebraic functions are formed by applying algebraic operations to the identity function

$$y = f(x). \quad (1)$$

3.2.2 TYPES OF ALGEBRAIC FUNCTIONS

The general polynomial function as shown on equation (2)

$$f(x) = a_0 + a_1x + a_2x^2 + ... + a_{k-1}x^{k-1} \quad (2)$$

Linear Function: A linear function is a function defined by an equation of the form:

$$y = a_1x + a_0 \quad (3)$$

Cubic Function: A function defined by a polynomial of degree 3. The general form of cubic function is

$$y = a_3x^3 + a_2x^2 + a_1 + a_0 \quad (4)$$

3.2.3 THE TRANFORMATOPNS OF GRAPH ALGEBRAIC FUNCTION

In this section we will discuss how the graph of a function may be transformed either by shifting, stretching or compressing, or reflection. The form of the quadratic function in the equation is called vertex form, so named because the form easily reveals the vertex or "turning point" of the parabola [3], [4]. Each of the constants in the vertex that form of the quadratic function plays a role. As you will soon see, the constant a controls the scaling (stretching or compressing of the parabola), [5] the constant h controls a horizontal shift and placement of the axis of symmetry, and the constant k controls the vertical shift. [6]

3.3 Step III : Review Secret Sharing Based Threshold Scheme

In secret sharing [7],[8],[9] the essential thing is the number of participants required to reconstruct the secret. If the number of authorized participants k out of n , we call this scheme (k,n) -threshold access structure and defined as:

$$F = \left\{ A \in 2^p \mid |A| \geq K \right\}, 2 \leq K \leq n \quad (5)$$

Where F is a method which the dealer can use to distribute shares to each participant any access structure $F \subseteq 2^p$, p is the set of trustees and 2^p is the power set of p satisfies the following condition, A an arbitrary set of participants such that $|A| \geq K$.: if $A \in F$ and $A \subset A' \subset p$ *then* $A' \in F$, K is the least number of retrieved participants and n is the number of total participants in the system.

3.3.1 SHAMIR'S THRESHOLD SECRET SHARING SCHEME

Shamir's secret sharing scheme is a threshold scheme based on polynomial interpolation. It allows a dealer ***D*** to distribute a secret value *s* to *n* users, such that at least $k < n$ users share are required to reconstruct the secret. The algorithm is information theoreticallysecure, i.e., any fewer than ***k*** users cannot gain any information about the secret by themselves. To share the secret ***s*** among users $P_1, P_2, ..., P_n$ such that ***k*** users are required to reconstruct the secret. Dealer *D* creates a random polynomial $f(x)$ of degree ***k-1*** and constant term ***s***. [8]

$$f(x) = a_0 + a_1x + a_2x^2 + ... + a_{k-1}x^{k-1} \quad (6)$$

This polynomial is constructed over a finite field, such that the coefficient a_0 is the secret *s* and all other coefficients are random elements in the field; the field is known to all authorized participants.

Dealer D publicly chooses n random distinct evaluation points: $X_j \neq 0$, and secretly distributes to each user P_j the share

$share_j(s) = (X_j, f(X_j)), j = 1 \ldots n.$ (Remark: The evaluation point X_j could be any publicly known value, therefore for our convenience, we assume $X_j = j$, hence the shares are denoted as) [4]

$$f(1), ..., f(j), ..., f(n) \quad (7)$$

To reconstruct the secret from each ***k*** shares out of n shares, without loss of original plaintext, we have to retrieve at least ***k*** shares: $f(1), f(2) ..., f(k)$.

By Lagrange interpolation [10], the ***k*** participants function value (Range) can be used to recomputed the coefficients a_0 of polynomial equation then we can recall the original plaintext S. The correctness and security conditions are also maintained.

Limitations of the scheme are computationally hard and become impractical with large number of shares K , stack overflow despite Shamir's shares are only pairs (x, y), their sizes increased with the power K .

3.4 Step IV : Review The Paillier Cryptosystem

The Paillier Cryptosystem is a modular, public key encryption scheme, created by Pascal Paillier [11]-[14]

3.4.1 Properties of scheme

Paillier scheme is the most known, and maybe the most efficient partially homomorphic encryption scheme. Although Paillier interestingly comes back to a standard RSA modulus $N = pq$ it is the direct continuation of the work of Okamoto and Uchiyama. In the following description, $\lambda(N)$ denotes the Carmichael function (in our case, $\lambda(N) = lcm(p-1, q-1)$), and $L_N(\bullet)$ denotes the function defined over the set

$$SN = \{x < N^2 \mid x = 1 \bmod N\} \text{ as } LN(x) = (x-1)/N.$$

3.4.2 Threshold Paillier Cryptosystem

Leave one line space of 10 pts and then give the abstract. Before A recent popular public key cryptosystem is a system proposed by Paillier [3] in 1999. The Paillier cryptosystem is based on properties of Carmichael function in Z_{N^2}. Security of the cryptosystem is based on the intractability of computing discrete logarithms in Z_{N^2} without the Carmichael number $\lambda(N)$.

Below are the steps for the key generation, encryption and decryption used.

3.4.3 USING THE PAILLIER CRYPTOSYSTEM

The following is the description of the Paillier steps that have been produced from [15]:

Steps for Key Generation

1). Choose two large prime numbers p and q randomly and independently of each other.

2). Compute the modulus n ; the product of two primes $n = p.q$ and (λ is Carmichael's function)

3). Select random integer g where $g \in \mathbf{Z}^*_{n^2}$ and g's order is a non-zero multiple of n (since $g = (1+n)$ works and is easily calculated - this is the best choice).

4). Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:

$u = L(g^{\lambda} \bmod n^2)^{1} \bmod n$, where function L is defined as (Lagrange function)

$L(u) = u - 1 \backslash n$ for $u = 1 \bmod n$

- The public (encryption) key is (n, g).
- The private (decryption) key is (λ).

Steps for Encryption

1). Plaintext is m where $m < n$.
2). Find a random r
3). Let cipher text $c = g^{m} . r^{n} \bmod n^2$.

Steps for Decryption

1). The cipher text $c < n^2$.
2). Retrieve plaintext

$$m = L\left(c^{\lambda} \bmod n^2\right) / L(g^{\lambda} \bmod n^2) \bmod n$$

Or in same expression

$$m = L\left(c^{\lambda} \bmod n^2\right) . u \bmod n$$

3.5 Step V : Review Homomorphisms Encryption

The concept of privacy homomorphism was first introduced by Rivestet. al. [15], that defined privacy homomorphisms as homomorphisms functions which permit encrypted data to be operated on without preliminary decryption of the operands. According to the correspondence between the operation in the cipher text domain and the operation in the plaintext domain, a cryptosystem can be additively homomorphic or multiplicatively homomorphic: in this paper we are interested in the former. Additively homomorphic cryptosystems allow, in fact, to perform additions, subtractions and multiplications with a known (non-encrypted) factor in the encrypted domain. More extensive processing would be allowed by the availability of an algebraically homomorphic encryption scheme, that is a scheme that is additive and multiplicative homomorphic.

The most common definition is the following. Let M (resp., C) denote the set of the plaintexts (resp., cipher texts).

An encryption scheme is [14] said to be homomorphicif for any given encryption key k *the* encryption function E satisfied

$$\forall m_1, m_2 \in M, E(m_1 \, e_M \, m_2) \leftarrow E(m_1) \, e_C \, E(m_2)$$

for some operators e_M in M and e_C in C, where means "can be directly computed from," that is, without any intermediate decryption. If (M, e_M) and (C, e_C) are groups, we have a *group homomorphism*. We say a scheme is *additively homomorphic*if we consider addition operators, and *multiplicatively homomorphicif we* consider multiplication operators.

A lot of such homomorphic schemes have been published that have been widely used in many applications. Note that in some contexts it may be of great interest to have this property not only for one operator but for two at the same time.

Hence, we are also interested in the design of ring/algebraic homomorphisms. Such schemes would satisfy a relation of the form

$$\forall m_1, m_2 \in M, E(m_1 +_M m_2) \leftarrow E(m_1) +_C E(m_2),$$
$$E(m_1 \times_M m_2) \leftarrow E(m_1) \times_C E(m_2) \quad (8)$$

As it will be further discussed, no convincing algebraic homomorphic encryption scheme has been found yet, and their design remains an open problem. Less formally, these definitions mean that, for a field key k, *it is* equivalent to perform operations on the plaintexts before encryption or on the corresponding ciphertexts after encryption. So, we require a kind of commutatively between encryption and some

data processing operations. Of course, the schemes we will consider in the following have to be probabilistic ciphers, and we may consider *E* to behave in a probabilistic way in the above definitions

3.5.1 The homomorphic property of the Paillier Cryptosystem

The given ciphertexts are valid encryptions of plaintexts m_i, $= Enc(m_i) = g^{m_i} r^{n_i} \bmod n^2$. The following properties hold

$$c_1 = g^{m_1} x_1^n \bmod n^2$$
$$c_2 = g^{m_2} x_2^n \bmod n^2$$
$$c_1 \times c_2 = g^{m_1} x_1^n \times g^{m_2} x_2^n \bmod n^2$$
$$= g^{m_1+m_2} (x_1 x_2)^n \bmod n^2 \quad (9)$$

So, Paillier cryptosystem public key cryptosystem is homomorphic cryptosystem. We can also get:

$$D\left[E_{x_1}(m_1)E_{x_2}(m_2)\right] \bmod n^2 = (m_1 + m_2) \bmod n$$
$$D\left[E_{x_1}(m_1)^k \bmod n^2\right] = (km_1) \bmod n$$
$$D\left[E_{x_1}(m_1) g^{m_2} \bmod n^2\right] = (m_1 + m_2) \bmod n$$
$$D\left[E_{x_1}(m_1)^{m_2} \bmod n^2\right] = (m_1 m_2) \bmod n \quad (10)$$

3.6 Design of Algorithm for Cryptic Distribute Databases

The Construction of Secret Part.

1) Input- secrete key number of participant

2) Select random values $a_1, a_2 \ldots, a_n$

3) Generate polynomial string to share the secrete into parts.

Transformations of Polynomial Function

4) Transformations of polynomial graph function

5) Sent Transformation Polynomial to Database

The Reconstruction

6) From Database Server to Secret sharing Data

7) Use by Lagrange Polynomials Return to Secret Sharing

The Construction Homomorphic Property of the Paillier Cryptosystem

8) Steps for Key Generation

8.1) Choose two large prime numbers p and q randomly and independently of each other.

8.2) Compute the modulus n ; the product of two primes $n = p.q$ and (λ is Carmichael's function)

8.3) Select random integer g where $g \in \mathbf{Z}^*_{n^2}$ and g's order is a non-zero multiple of n (since $g = (1+n)$ works and is easily calculated - this is the best choice).

8.4) Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:

$u = L(g\lambda \bmod n^2)^1 \bmod n$,where function *L* is defined as (Lagrange function) $L(u) = u - 1 \backslash n$ for $u = 1 \bmod n$

- Thepublic (encryption) key is (n, g).
- The private (decryption) key is (λ).

9) Steps for Encryption

9.1) Plaintext is m where $m < n$.

9.2) Find a random r

9.3) Let cipher text $c = g^m . r^n \bmod n^2$.

10) Steps for Decryption

10.1) The cipher text $c < n^2$.

10.2) Retrieve plaintext

$$m = L(c\lambda \bmod n^2) / L(g\lambda \bmod n^2) \bmod n$$

Or in same expression

$$m = L(c\lambda \bmod n^2) . u \bmod n$$

STEP of Algorithm for cryptic distribute databases

The Construction of Secret Part (Step1-3)

First, the Data owner *(D)* has to define the original plaintext (*S*) which would be assigned as

a_0 of the arbitrary polynomial equation. Second, this polynomial function $(f(x))$ will be calculated of $f(x)$ on y_i far each value of x_i .these share S_i ,on $(x_i, f(x_i))$,were then represent as secret share part of original secret S.

1: D defines $a_0 = s$

Step2: D select *t-1* random, independent coefficients

$a_1, ..., a_{t-1}, 0 \le a_j \le p-1$ Defining the random polynomial Z_p ,

$$f(x) = \sum_{j=0}^{t-1} a_j \times x^j. \quad (11)$$

Step3: D publicly chooses n random distinct evaluation points: $X_j \neq 0$, and secretly distributes to each users P_j the share

$$share_j(s) = \left(X_j, f\left(X_j\right)\right), j = 1...n. \quad (12)$$

Example 1 Polynomial Secret Sharing Let S = 10000, n = 6, a_1 =16, a_2 = 9, $k = 3$

Secret share points

$$f(x) = 10000 + 16x + 9x^2 \quad (13)$$

(1,1494), (2,1942), (3,2598), (4,3402), (5,4414), (6,5614)

For each participant a different single point of share $j(S); j = 1,6$ (both x and $f(x)$).

Let us consider (2,1924),(4,3402),(5,4414) $(x_0, y_0) = (2,1924), (x_1, y_1) = (4,3402), (x_2, y_2)$ = (5,4414)

Transformations of data using polynomial graph function (Step4)

Step4: Transformations of data using polynomial graph function

1. Bring the original data are not encrypted. M

2. To convert data into forms $f(m')$.

This is in the process transformation polynomial Function

$$M' = f(x - k) + h \quad (14)$$

Example 2 Let $m_1 = 9900, m_2 = 90$ =2, and $f(m') = (m_i - 1) + 2$

INPUT: $m_1 = 9900, m_2 = 90$

Process:

1. $f(m_1') = (m_1 - 100) + 200$
2. $f(m_2') = (m_2 - 10) + 20$

OUTPUT: $m_1' = 10000, m_2' = 100$

Step5: Sent Transformation data to database Administrator saves m_1', m_2' to Local Data Base and Log File and sent return to Users

The Reconstruction

Step6: From Database Server to Secret sharing Data The Secret sharing 3 points after Return to Polynomial Secret Sharing

(2,1924),(4,3402),(5,4414)

$(x_0, y_0) = (2,1924), (x_1, y_1) = (4,3402),$

$(x_2, y_2) = (5,4414)$

Step7: Use by Lagrange Polynomials Return to Secret Sharing

We will compute Lagrange basis polynomials:

$$l_0 = \frac{x-x1}{x_0-x_1} * \frac{x-x2}{x_0-x_2} = \frac{x-4}{2-4} * \frac{x-5}{2-5} = \frac{1}{6x^2} - \frac{11}{2x} + \frac{31}{3}$$

$$l_1 = \frac{x-x0}{x_1-x_0} * \frac{x-x2}{x_1-x_2} = \frac{x-2}{4-2} * \frac{x-5}{4-5} = \frac{-1}{2x^2} - \frac{31}{2x-5}$$

$$l_2 = \frac{x-x0}{x_2-x_0} * \frac{x-x2}{x_2-x_1} = \frac{x-2}{5-2} * \frac{x-4}{5-4} = \frac{1}{3x^2} - 2x + \frac{22}{3}$$

$$f(x) = \sum_{j=0}^{2} y_j l_j(x)$$

$$= 1942\left(\frac{1}{6x^2} - \frac{11}{2x} + \frac{31}{3}\right) + 3402\left(\frac{-1}{2x^2} - \frac{31}{2x} - 5\right)$$

$$+4414\left(\frac{1}{3x^2} - 2x + \frac{22}{3}\right)$$

We could recover original polynomial equation with '15' as an original

$$f(x) = 10000 + 16x + 9x^2 \quad (15)$$

The Construction Homomorphic Property of The Paillier Cryptosystem

Step8: Steps for Key Generation

Assume $p=137, q=139, n=pq=19043\ 5$, $q=7, n^2=362635849, \varphi(n)=18768, \lambda(n)=9384, g=(n+1)=19044, \mu=6163$,

$m_1'=10000, r_1=488, m_2'=100, r_2=657$

The product of these cipher texts, c_1', c_2' are 221470276 and 159093273

Example 3

INPUT: $c_1'=221470276, c_2'=159093273$

Process:

$c_1 \times c_2 =$ (221470276 * 159093273) mod362635849 = 193070066

$\equiv (221470276 * 159093273) \bmod 362635849$

$\equiv 193070066$

$$2. \equiv \left(\frac{L\left(193070066^{9384} \bmod 362635849\right)-1}{19043} \times \mu \right) \bmod 19043$$

OUTPUT:=10100=10000+100

And can be return Transformation Data back to the original data m_i

Example 4

INPUT: $m_1'=10000, m_2'=100$

Process:

1. $m_1'=10000$

   $f(m_1)=\left(m_1'+100\right)-200$

2. $m_2'=100$

3. $f(m_2)=\left(m_2'-10\right)+20$

OUTPUT: $m_1'=9900, m_2'=90$

STEP OF FLOW ALGORITHM FOR CRYPTIC DISTRIBUTE DATABASES WORK

step of flow algorithm for cryptic distribute databases work as shown in Figure 2 were as follows'

1) The administrator performs Transformed Data as Users request.

2) An administrator record data to the local database and log file.

3) The administrator sends data through the Transformed to Server.

4) Users begin their duties and made a separate data by using Secret Sharing.

5) The data is divided into a series by the Message ID in each batch of confidential information, and all information will be distributed to each of the DBA.

6) When the DBA to get secret data, comparing to verify the accuracy of the data that was not changed during transmission with Compare.

7) DBA checks the accuracy of the information and the record into the database.

8) Users want information on Update They will make a request to all DBAs will send Message ID. Go to Search

9) Users perform data encryption, Ciphertext, and Update with paillier homomorphic encryption process.

10) SplitedCiphertext of process paillier homomorphic encryption process. Using by Secret Sharing method.

11) When the DBA to get secret data, comparing to verify the accuracy of the data that was not changed during transmission with Compare.

12) DBA checks the accuracy of the information and the record into the database.

13) Users want information on Update They will make a request to all DBAs will send Message ID. Go to Search

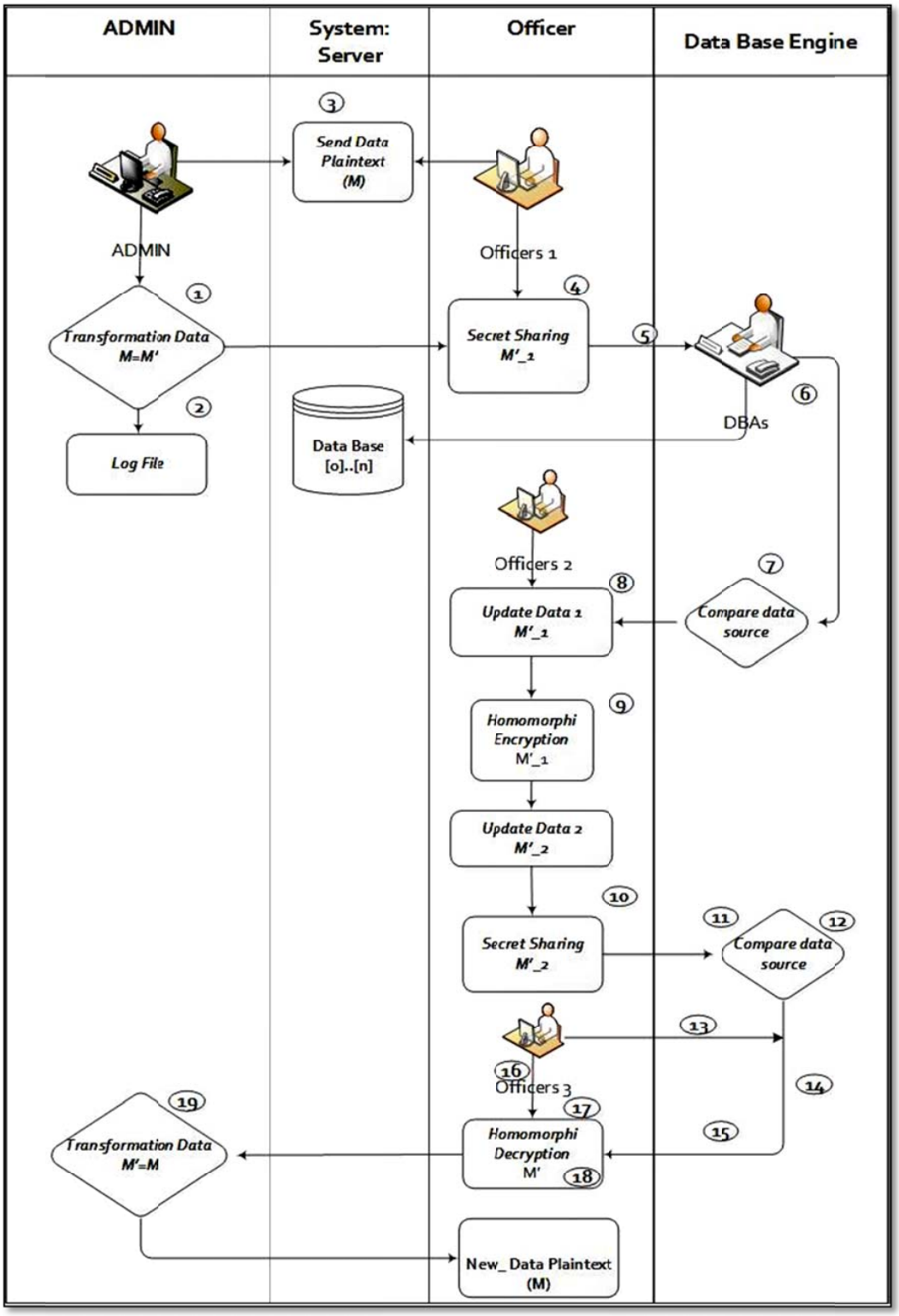

Figure 2 Flow of Algorithm for Cryptic Distribute Databases work

14) DBAs everyone brings MessageID to data search.

15) DBAs have to send Secret data from database to Users.

16) Users get the data will be decrypted with process of paillier homomorphic decrypted to verify the accuracy of data that is not changed during transmission Compare.

17) Combine the secret data to Ciphertext.

18) Then take that decrypting the ciphertext to come to take the same step before encryption in the first

19) Administrator transformed data return to before encrypted.

3.8 Evaluate algorithms for data encryption

The results of efficient evaluation of algorithms for cryptic distribute databases (ACDD) for distributed database management of the encrypted data showed that it was 1.27 times more complex than the compared method. This led to its performance time was 1.35 times higher. In addition in the algorithm Paillier Homomorphic Encryption, the new developing algorithm was 1.3 times more efficient than the compared method in which the performance time increased 1.54 times.

4. CONCLUSION

The presented method, this is a part of the whole model Algorithm security data. ACDD algorithm can solve the problem of dishonest dealers and dishonest participants who try to deceive other participants and security enhancement by enables providing results of calculations on encrypted data without knowing the raw data on which the calculation was carried out. Homomorphic encryption is secure cryptosystem. In this scheme, server can perform arithmetic addition, subtraction, multiplication and division on Homomorphic cipher data. Disadvantage in this ACDD is that ACDD is slightly calculation time more than ordinary method In the future, ACDD should be modified to be effective in providing more security and rapidly mathematical calculation.

9. References

[1] Akinwande, M. (2009). “Advances in Homomorphic Cryptosystems.” J. UCS. Vol.15 No.3 : 506–522.

[2] Sridokmai, T., Prakancharoen, S. (2015). “The homomorphic other property of Paillier cryptosystem.” Science and Technology (TICST). 2015 International Conference. IEEE, 356–359

[3] Borba, M. C., et al (1996). “A student’s construction of transformations of functions in a multiple representational environment.” Educational Studies in Mathematics. Vol. 31 No. 3 : 319–337.

[4] Löwe, M. (1993). “Algebraic approach to single-pushout graph transformation.” Theoretical Computer Science. Vol. 109 No. 1 : 181–224.

[5] Leinhardt, G., et al (1990). “Functions, graphs, and graphing: Tasks, learning, and teaching”. Review of Educational Research. Vol. 60 No. 1 : 1–64.

[6] Godsil, C., et al (2013). Algebraic graph theory . New York : Springer Science & Business Media.

[7] Blakley, G., et al (1979). “Safeguarding cryptographic keys.” In Proceedings of the national computer conference. (313–317).

[8] Shamir, A. (1979). “How to share a secret.” Communications of the ACM. Vol.22 No.11 : 612–613.

[9] Al-Ssulami, A. M. (2013). "A novel threshold secret sharing scheme using FFT algorithm." International Journal of Information Security Science. Vol.2 No.1 : 19–27.

[10] Kovács, A. (2011). "The Lagrange Interpolation Formula for Analyzing Fluid Movement in Network Profiles." Annals of the Faculty of Engineering Hunedoara. Vol.9 No.4 : 53.

[11] Paillier, P. (1999). "Public-key cryptosystems based on composite degree residuosity classes". In International Conference on the Theory and Applications of Cryptographic Techniques. Springer. (223–238).

[12] Paillier, P. (2011). "Paillier encryption and signature schemes." In Encyclopedia of Cryptography and Security. Springer. (902–903)

[13] Parmar, P. V., et al (2014). "Survey of various homomorphic encryption algorithms and schemes." International Journal of Computer Applications. Vol. 91 No. 8 : 26-32.

[14] Fontaine, C. and Galand, F. (2007). "A Survey of Homomorphic Encryption for Nonspecialists." EURASIP J. Inf. Secur. Vol. 2007 : 1–15.

[15] Rivest, R. L., et al (1978). "On data banks and privacy homomorphisms". Foundations of Secure Computation. Vol. 4 No. 11 : 169-180.