A data corruption detection scheme based on ciphertexts in cloud environment

Sixu Guo[1*], Shen He[1*], Li Su[1], Xinyue Zhang[1], Huizheng Geng[1] and Yang Sun[1]
[1] China Mobile Communication Research Institute
Beijing, 100053, China
[e-mail: 792669696@qq.com, heshen@chinamobile.com]
[*]Corresponding author: Sixu Guo, Shen He

Received January 21, 2021; revised May 12, 2021; revised July 20, 2021; accepted August 22, 2021; published September 30, 2021

Abstract

With the advent of the data era, people pay much more attention to data corruption. Aiming at the problem that the majority of existing schemes do not support corruption detection of ciphertext data stored in cloud environment, this paper proposes a data corruption detection scheme based on ciphertexts in cloud environment (DCDC). The scheme is based on the anomaly detection method of Gaussian model. Combined with related statistics knowledge and cryptography knowledge, the encrypted detection index for data corruption and corruption detection threshold for each type of data are constructed in the scheme according to the data labels; moreover, the detection token for data corruption is generated for the data to be detected according to the data labels, and the corruption detection of ciphertext data in cloud storage is realized through corresponding tokens. Security analysis shows that the algorithms in the scheme are semantically secure. Efficiency analysis and simulation results reveal that the scheme shows low computational cost and good application prospect.

Keywords: Data corruption, Ciphertexts, Cloud environment, Cryptography, Corruption detection

This research was supported by the Ministry of Education China Mobile Research Fund under grant [MCM201805-2].

http://doi.org/10.3837/tiis.2021.09.016 **ISSN : 1976-7277**

1. Introduction

1.1 Background

In traditional way of data application, data are basically acquired from sources and then transmitted after being processed and analyzed through the platform, with a single propagation path and limited influence scope. With the development of various technologies such as big data and artificial intelligence (AI), data assets have been important assets for an enterprise and a person. Data play their roles through the data transfer among various system platforms. In addition, the scope of the data transmission is broadened, facing more security threats. Data corruption problems (such as data tampering and data loss) are likely to occur in each link of data transfer [1]. Owing to data present a wider influence scope and greater influence, in many cases, it will possibly be out of the control once data corruption appears, thus further causing unpredictable consequences. Therefore, data corruption problems are increasingly concerned.

Data corruption is a kind of damages to the integrity and authenticity of raw data caused by intentional or unintentional human operation, which distorts the true data [2,3]. Understanding on features of data corruption is considered as an important guarantee for solving the data corruption problems. Corrupted data are mainly characterized by numerous corrupted datum values, great confidentiality and strong dispersion effects of corruption. The data corruption is easily found sometimes even though it is highly confidential. Therefore, the most important feature of the data corruption problem lies in the presence of numerous corrupted datum values. Statistically, the corrupted datum values are generally outliers [4]. The outliers are probably found in the corrupted data. Therefore, how to accurately and efficiently detect outliers has been the key to solving data corruption problems.

1.2 Related works

In recent years, some progress has been made on research concerning the detection of corrupted data. A statistical model for describing the anomaly detection is proposed [5]; a method for online traffic anomaly detection based on software defined network (SDN) is put forward [6]; a sparse combination-based learning framework is proposed to detect abnormal behaviors [7]; a multi-scale non-parametric approach for abnormal behavior detection based on local spatio-temporal characteristics is put forward [8]; the application of the weighted conditional entropy in anomaly detection is explored [9]; an algorithm for crowd anomaly detection based on image textures represented by spatio-temporal information is proposed [10]; an anomaly detection algorithm based on chaotic radial basis function (RBF) neural network is put forward [11]; the anomaly events are detected by combining the Gaussian model with Markov chain [12]; A novel framework for anomaly detection in crowded scenes is presented[13]; [14] attempts to provide a comprehensive and structured overview of the existing research for the problem of detecting anomalies in discrete/symbolic sequences.

1.3 Problem statement

At present, much more users and enterprises choose to outsource their data storage and business computing to cloud servers to save expenses on data storage overhead and system maintenance. However, although the efficiency of cloud storage greatly improves, security accidents frequently occur due to the presence of the hidden danger pertaining to cloud service providers [15]. Therefore, the data encryption is first performed to ensure the confidentiality of data in cloud server. However, the encrypted data lose the features of raw data. Additionally, the encrypted data are also possibly corrupted due to human factors or statistical error, thus generating numerous outliers. Owing to encrypted data themselves are confidential and irregular, it is more difficult to detect the encrypted data if they are corrupted[16,17,18]. Nevertheless, existing technologies for data corruption detection are mostly used for plaintext data while not applicable to ciphertexts, which influences the practicability of technologies for data corruption detection in cloud environment[19,20,21]. Thus, the detection technology for ciphertext data corruption in cloud environment is increasingly concerned and demanded.

1.4 Our contribution

Aiming at the aforementioned problems, a data corruption detection scheme based on ciphertexts in cloud environment (DCDC) is proposed. The scheme can satisfy the users' demand for corruption detection of ciphertext data stored in cloud servers. The DCDC is established based on the anomaly detection method of Gaussian model [22,23,24,25]. Combined with statistics technologies (such as expected value, variance and F value) and cryptography means (such as RSA encryption), the encrypted detection index for data corruption and corruption detection token for each type of data are constructed according to the data labels [26,27]; on the premise of protecting the semantic security of the scheme and avoiding disclosure of data privacy due to the detection index and token, the data corruption detection of ciphertext data stored in cloud environment is efficiently realized, showing important theoretical value. By conducting the security analysis, the scheme is adaptively secure; through efficiency analysis, it is found that the scheme presents relatively ideal time complexity in three stages, index generation, token generation and corruption detection. The test analysis shows that the scheme can realize the corruption detection of ciphertext data within a short time, delivering a favorable application prospect.

2. Relevant knowledge

2.1 RSA encryption scheme

The RSA encryption algorithm, as a commonly used public-key encryption algorithm, presents its encryption process as follows:

1) Two large prime numbers p and q are stochastically selected, with $p \neq q$, and then $N = p \cdot q$ is calculated;

2) $r = \varphi(N) = (p-1)(q-1)$ is calculated, in which φ refers to the Euler function.

3) An integer e lower than r is selected, such that e and r are coprime. Afterwards, the value of d is calculated, which satisfies $ed \equiv 1 \bmod r$.

4) (N, e) and d separately represent the public key and private key.

The encryption and decryption processes are shown as $M^e \equiv C \bmod N$ and $C^d \equiv M \bmod N$, respectively.

The RSA encryption algorithm, belonging to a homomorphic encryption algorithm, shows the multiplicative homomorphism, which is specifically displayed as follows:
Combined with ciphertexts $Enc(m_1)$、$Enc(m_2)$, the ciphertext of $m_1 \cdot m_2$ can be attained by calculating $Enc(m_1 \cdot m_2) = Enc(m_1) \cdot Enc(m_2)$ on condition of not conducting the decryption.

2.2 Anomaly detection based on Gaussian model

Given a random variable X conforming to Gaussian distribution[28,29,30], that is, $X \sim N\left(\mu, \ \sigma^2\right)$, in which μ and σ^2 refer to the expected value of X and the variance, respectively, the probability density function of the variable is expressed as $P\left(x\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(x-\mu\right)}{2\sigma^2}\right)$. For a given training set $\left(x^{(1)},x^{(2)},x^{(3)},\ldots,x^{(m)}\right)$, that is, the sample data in each dimension belong to a n-dimensional vector, which conforms to Gaussian distribution, it is feasible to establish a model to estimate the probability density of samples according to the following method: $p\left(x\right)=p\left(x_1;\mu_1,\sigma_1{}^2\right)p\left(x_2;\mu_2,\sigma_2{}^2\right)\ldots p\left(x_n;\mu_n,\sigma_n{}^2\right)$.

The method for detecting whether a new datum $x' \sim N\left(\mu, \ \sigma^2\right)$ belongs to an outlier in $\left(x^{(1)},x^{(2)},x^{(3)},\ldots,x^{(m)}\right)$ is shown as follows:

1) As for expected values $\mu_1,\mu_2,\ldots,\mu_n$ and variances $\sigma_1{}^2,\sigma_2{}^2,\ldots,\sigma_n{}^2$ of data in each dimension, $\mu_j=\frac{1}{m}\sum_{i=1}^{m}x^{(i)}{}_j$ and $\sigma_j{}^2=\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}{}_j-\mu_j\right)^2$ are calculated, respectively;

2) $p\left(x'\right)=\prod_{j=1}^{n}p\left(x'_j;\mu_j,\sigma_j{}^2\right)$ is calculated;

3) A threshold ε is determined to compare $p\left(x'\right)$ with ε; in the case of $p\left(x'\right)<\varepsilon$, the datum is deemed as an outlier.

where, ε denotes an empirical value, which generally equals the datum that maximizes the evaluation index in the validation set.

2.3 Precision, recall rate and *F* value

There are two concepts: precision (*P*) and recall rate (*R*), in information retrieval theory[31]. Generally, precision refers to the proportion of accurate items (files and data) that are retrieved, which reflects the retrieval accuracy; the recall rate denotes the proportion of retrieved accurate items in all accurate items, which reflects the retrieval comprehensiveness, with the expressions as follows:

$$P=\frac{number\ of\ relationship\ instances\ of\ a\ class\ that\ are\ correctly\ classified}{total\ number\ of\ relationship\ instances\ determined\ to\ be\ of\ a\ certain\ class}$$
$$R=\frac{number\ of\ relationship\ instances\ of\ a\ class\ that\ are\ correctly\ classified}{total\ number\ of\ instances\ of\ a\ relationship\ in\ the\ test\ set} \tag{1}$$

During the retrieval, it is expected to attain both higher precision and recall rates of the retrieved results. As a matter of fact, the two parameters are contradictory in some circumstances. Thus, it is necessary to comprehensively consider them and the method for calculating *F* value is the commonest. *F* value (also called the comprehensive evaluation index) is used to comprehensively reflect the whole field, showing the expression as follows:

$$F=2\cdot\frac{P\cdot R}{P+R} \tag{2}$$

It can be seen that *F* value synthesizes the results of *P* and *R*. A high *F* value implies that the retrieval method is effective.

3. DCDC scheme

The model, formal definition and security definition of the DCDC scheme are described.

3.1 Scheme model

The DCDC scheme contains three entities: a data owner (DO), a data tester (DT) and a cloud server. The structure of the scheme model is shown in **Fig. 1**.

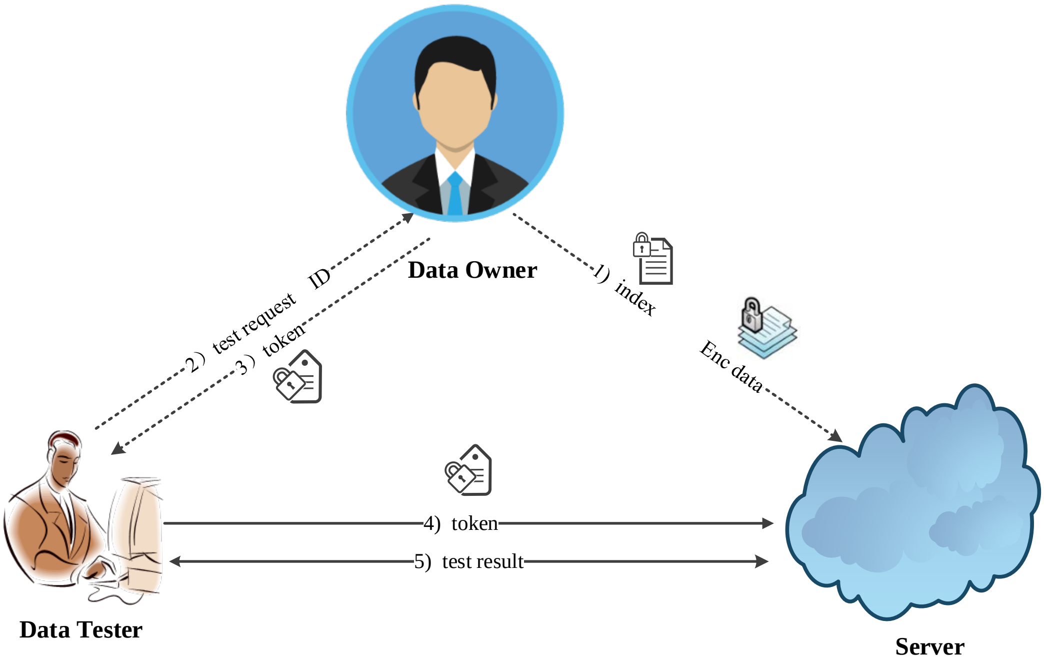

Fig. 1. DCDC scheme model

The local data are processed and uploaded to the cloud server first by the DO. In terms of the data processing, the data are classified and encrypted, their expected values and variances are calculated, and the detection indexes for data corruption are generated for each type of data; moreover, a series of thresholds for data corruption are generated based on the related statistical algorithms. Afterwards, the data corruption is detected by the DT according to data labels: the DT needs to submit the request for data detection and identity (ID) information to the DO at first; next, the DT submits the request for data detection to the cloud server by utilizing the detection tokens returned by the DO after successful certification. The related detection parameters are transmitted to the DT by the cloud server after receiving the detection tokens; then the DT decrypts the parameters and performs the corruption detection on the data to be detected to attain the detection results.

The functions of various entities are described as follows:

1) As the owner of data to be detected, the DO is responsible for data preprocessing (including generating the encrypted data and the encrypted detection index for data corruption) and uploading the data to the cloud server; receiving the request for data detection and checking the ID information from the DT; generating the detection tokens and returning them back to the DT;

2) As the tester for data corruption, the DT needs to send the request for data detection and

the personal ID information, attain the authorization for data detection from the DO and receive the detection tokens; the DT sends the request for data detection to the cloud server with the aid of the detection tokens, receives the detection parameters and performs the data corruption detection.

3) The cloud server is a curious but honest entity and fails to attain the sensitive data, which is only in charge of the storage of encrypted data and detection index and assists the DT to detect the data corruption.

3.2 Symbol description and formal definition

Firstly, the symbols in DCDC scheme are described, and the definitions of each symbol are shown in **Table 1**.

Table 1. DCDC scheme symbol description

Symbol	definition
λ	security parameter required to generate key
MK	main key of DCDC
K_{DT}	key used by DT
D	raw datum
D'	encrypted datum
$\boldsymbol{I}$	corruption detection index
D_{T}	authenticated dictionary
$\boldsymbol{I}''$	encrypted detection index
$\boldsymbol{M}_{\mathrm{D}}$	dictionary matrix for corruption detection
C	corruption detection request by DO
ID	identity information of the DT
l_q	data label to be detected
T	detection token used by DT
RS	corruption detection result

Next, the algorithm in DCDC scheme is formally defined. The DCDC scheme comprises 5 algorithms and its formal definition is shown as follows: $\mathrm{SPDC}=\left(\mathrm{KeyGen},\mathrm{DataPre},\mathrm{IndexGen},\mathrm{TokenGen},\mathrm{Check}\right)$, with the specific descriptions as follows:

1) $\mathrm{MK},\mathrm{K}_{\mathrm{DT}} \leftarrow \mathrm{KeyGen}\left(\lambda\right)$, a key generation algorithm, is executed by the DO and DT. The security parameter λ is input and the main key MK and the key K_{DT} required by the DT are output, in which MK is stored in the whole life cycle of the scheme.

2) $\mathrm{D}',\boldsymbol{I},\mathrm{D}_\mathrm{T} \leftarrow \mathrm{DataPre}(\mathrm{D},\mathrm{MK})$, a data preprocessing algorithm, is executed by the DO. The raw datum D and the main key MK are input. Through calculation, the encrypted datum D', the corruption detection index $\boldsymbol{I}$ and the authenticated dictionary D_T are output.

3) $\boldsymbol{I}'',\boldsymbol{M}_\mathrm{D} \leftarrow \mathrm{IndexEnc}(\mathrm{MK},\boldsymbol{I})$, an algorithm for the encrypted index generation, is executed by the DO. The main key MK and the corruption detection index $\boldsymbol{I}$ are input and then the encrypted detection index $\boldsymbol{I}''$ and the dictionary matrix $\boldsymbol{M}_\mathrm{D}$ are output.

4) $T \leftarrow \mathrm{TokenGen}(C,\mathrm{ID},l_q)$, an algorithm for token generation, is executed by the DO. The request C for corruption detection, ID information of the DT and the data label l_q to be detected are input; afterwards, the detection token T is output.

5) $\mathrm{RS} \leftarrow \mathrm{Check}(T,\boldsymbol{I}'',\boldsymbol{M}_\mathrm{D})$, a corruption detection algorithm, which is executed by the server and the DT. The detection token T, the encrypted detection index $\boldsymbol{I}''$ and the dictionary matrix $\boldsymbol{M}_\mathrm{D}$ are input and then the detection result RS is output.

3.3 Security definition

In terms of the DCDC scheme, the DO is thought as a reliable entity and so does the DT after being certificated by the DO; it is also thought that the symmetric encryption algorithm used for the data is secure and therefore the data security itself is not discussed. Thus, the privacy protection required to be provided in the DCDC scheme involves two parts: detection index security and detection token security. The detection index security also involves two parts: the encrypted detection index $\boldsymbol{I}''$ security and the dictionary matrix $\boldsymbol{M}_\mathrm{D}$ security. That is, the related information of the index and data cannot be inferred by the server through the $\boldsymbol{I}''$ and $\boldsymbol{M}_\mathrm{D}$; moreover, the related information (including detection content, amount and whether the data to be detected have been detected or not) detected by the DT fails to be attained by the server based on the detection tokens. The security of the DCDC scheme is formally described through the security test.

Setup: The challenger C creates a label set **L** and selects some subsets from the corresponding data of **L** to make up a dataset D. At first, the KeyGen is operated to generate the key and then DataPre is executed according to D and L to generate the detection index $\boldsymbol{I}$; then, IndexEnc runs to generate the encrypted detection index $\boldsymbol{I}''$ and the dictionary matrix $\boldsymbol{M}_\mathrm{D}$; finally, C sends D, $\boldsymbol{I}''$ and $\boldsymbol{M}_\mathrm{D}$ to the adversary A

Check: The adversary A is allowed to apply for the detection token T from C. A can operate Check with T on $\boldsymbol{I}''$ and $\boldsymbol{M}_\mathrm{D}$ so as to attain the result RS.

Challenge: A selects two non-empty detection requests Y_1,Y_2 $(Y_1 \neq Y_2)$ and sends them to C. After receiving Y_1,Y_2, C chooses a parameter $\omega \xleftarrow{R} \{0,1\}$ to generate the detection token T_ω based on Y_ω and performs the data corruption detection on $\boldsymbol{I}''$ and $\boldsymbol{M}_\mathrm{D}$ with the aid of T_ω to obtain the result RS. The challenge for A is to judge the value of ω.

Response: The conjecture ω' of ω is output by A.

In the security test, the advantage of A for winning can be defined as $\mathrm{Adv}_\mathrm{A} = [\mathrm{pr}[\omega = \omega'] - 1/2]$. If the case that A wins the test at a non-negligible advantage does not exist, it can be thought that the scheme is semantically secure.

Theorem 1: If the homomorphic encryption algorithm used for the scheme is semantically secure, the DCDC scheme also shows the semantic security.

4. Detailed design of the scheme

The detailed design of the DCDC scheme is mainly introduced in this section. According to the formal definitions in the last section, the five algorithms involved in the DCDC scheme are separately elaborated.

4.1 Key generation algorithm

$\mathrm{MK},\mathrm{K}_{\mathrm{DT}} \leftarrow \mathrm{KeyGen}(\lambda)$ belongs to a probabilistic algorithm. The security parameter λ is input. The DO calculates the private key $sk = d$ and the public key $pk = (e, N)$ required by the RSA encryption algorithm at first; afterwards, the symmetric encryption keys K are generated to encrypt the data; finally, an invertible matrix $\boldsymbol{M}$, a pseudo-random function $f:\{0,1\}^{s} \to \{0,1\}^{n}$ and a hash function $H:\{0,1\}^{*} \to \mathrm{Z}_{N}^{*}$ are generated. $\boldsymbol{M}$, K, f, H and sk taken as the main keys MK are privately owned while the public key pk is published. Eventually, the DT generates the key $\mathrm{K}_{\mathrm{DT}} = (sk_{DT}, pk_{DT})$ required by RSA encryption to validate the signatures. The execution of the key generation algorithm is completed.

4.2 Data preprocessing algorithm

$\mathrm{D}',\boldsymbol{I},\mathrm{D}_{\mathrm{T}} \leftarrow \mathrm{DataPre}(\mathrm{D},\mathrm{MK})$ belongs to a probabilistic algorithm. The DO first encrypts the dataset $\mathrm{D} \in \mathrm{Z}_{N}^{*}$ by utilizing the symmetric keys K in MK. The commonly used symmetric encryption algorithm such as advanced encryption standard (AES) can be applied to attain the encrypted datum $\mathrm{D}' \in \mathrm{Z}_{N^2}^{*}$ and upload it to the server; subsequently, the DO classifies the dataset. It is supposed that the data are divided into ϑ classes, recording the data label as l_i. Aiming at each type of datasets $l_i = \left(x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(n)}\right)^{\mathrm{T}}$, that is, n-dimensional vector data, the data samples are processed (such as taking the logarithm) so that the samples all conform to Gaussian distribution, that is, $x^{(i)} \sim N\left(\mu, \sigma^2\right)$. The datasets are partitioned into the training set, cross-validation set and test set according to the following standards as the scientific way to divide data sets in machine learning (it can also be partitioned in other different ways):

$$\left[\begin{array}{l}\text{Training Set：60\% normal data}\\ \text{Cross-validation set：20\% normal data \& 50\% exception data}\\ \text{Test set：20\% normal data \& 50\% exception data}\end{array}\right]$$

It is supposed that there are m data samples in the training set. The expected values $\mu_1, \cdots, \mu_n$ and variances $\sigma_1^{\,2}, \cdots, \sigma_n^{\,2}$ of data in each dimension are calculated according to the following equations:

$$\begin{aligned} \mu_j &= \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \\ \sigma_j &= \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j^{(i)}\right)^2 \end{aligned} \tag{3}$$

After training all types of data, the corruption detection index $\boldsymbol{I}$ is constructed by using the data labels and the expected values and variances of data, showing the structure in **Fig. 2**.

$$\boldsymbol{I} = \begin{cases} l_1 \rightarrow (\mu_1,\sigma_1),(\mu_2,\sigma_2),\ldots,(\mu_{n_1},\sigma_{n_1}) \\ l_2 \rightarrow (\mu_1,\sigma_1),(\mu_2,\sigma_2),\ldots,(\mu_{n_2},\sigma_{n_2}) \\ \vdots \qquad\qquad\qquad \vdots \\ l_\vartheta \rightarrow (\mu_1,\sigma_1),(\mu_2,\sigma_2),\ldots,(\mu_{n_\vartheta},\sigma_{n_\vartheta}) \end{cases}$$

Fig. 2. Detection index architecture

Finally, the data corruption threshold ε_i is acquired through multiple attempts by calculating the F values of data samples in the cross-validation set, in the following ways:

1) First, an initial data corruption threshold ε_i' is determined by the data owner;

2) According to the threshold ε_i', the precision P and recall rate R of the samples are separately calculated according to (2);

3) Data owner calculate the *F* value of the samples according to (2);

4) According to *F* value definition in 2.3, data owner through multiple experiments, it is determined that $\varepsilon_i = \max(F)$.

The selected ε_i is tested based on the data in the test set. After all values of ε_i are determined, the authenticated dictionary $\mathrm{D_T} = [l_i : \varepsilon_i]$ is established by using the data label l_i and the corruption detection threshold ε_l. Above all, the execution of the data preprocessing algorithm finishes.

4.3 Encrypted index generation algorithm

$\boldsymbol{I}'',\boldsymbol{M}_{\mathrm{D}} \leftarrow \mathrm{IndexEnc}(\mathrm{MK},\boldsymbol{I})$ belongs to a probabilistic algorithm. The encrypted index generation algorithm aims to encrypt the corruption detection index $\boldsymbol{I}$ into a security index, and the algorithm is expressed as follows:

1) By applying the pseudo-random function f, each data label and the corresponding expected values and variances are encrypted to generate the encrypted label $f(l_i) = t_i$ and encrypted parameter $f\left(\mu_n^{(i)} \,\|\, \sigma_n^{(i)}\right) = \tau_i$. The sets of the encrypted labels are defined as Ω and the sets of the encrypted parameters are defined as Σ. The maximum length of the parameters in the set Ω is set as L to generate a series of random numbers $R_i = \{r_1,\ldots r_j\}$, with $r_j \notin \Omega$; afterwards, the R_i is filled into the parameter and then a polynomial $P_{\tau_i}(x) = \prod_{\tau_i \in \Sigma}(x-\tau_i)\prod_{r_j \in R_i}(x-r_j)$ is generated according to the parameter list after the data filling.

2) The vector $\boldsymbol{I}_{\tau_i} = \left(P_{\tau_1}, P_{\tau_2}, \ldots, P_{\tau_\vartheta}\right)^{\mathrm{T}}$ of the polynomial $P_{\tau_i}(x)$ is calculated and the coefficient $(\alpha_0,\alpha_1,\ldots,\alpha_L)$ in each polynomial P_{τ_i} is encrypted with the public key $pk = (e,n)$ in the RSA algorithm to attain $(Enc(\alpha_0), Enc(\alpha_1), \ldots, Enc(\alpha_L))$. After encrypting all polynomial parameters, the encrypted index $\boldsymbol{I}'' = Enc_{RSA}\left(\boldsymbol{I}_{\tau_i}\right)$ is acquired.

3) The dictionary matrix $\boldsymbol{M}'$ is constructed based on elements in the set Ω, which is shown as follows:

$$\boldsymbol{M}' = \begin{bmatrix} t^{\vartheta}{}_1 & t^{\vartheta}{}_2 & \cdots & t^{\vartheta}{}_{\vartheta} \\ t^{\vartheta-1}{}_1 & t^{\vartheta-1}{}_2 & \cdots & t^{\vartheta-1}{}_{\vartheta} \\ \vdots & \vdots & & \vdots \\ t^{1}{}_1 & t^{1}{}_2 & \cdots & t^{1}{}_{\vartheta} \end{bmatrix} \tag{4}$$

It can be obtained that $\boldsymbol{M}_{\mathrm{D}} = \boldsymbol{M} \cdot \boldsymbol{M}'$ by encrypting $\boldsymbol{M}'$.

The execution of the encrypted index generation algorithm is completed after uploading $\boldsymbol{I}''$ and $\boldsymbol{M}_{\mathrm{D}}$ to the server.

4.4 Token generation algorithm

$T \leftarrow \mathrm{TokenGen}\left(C, \mathrm{ID}, l_q\right)$ is a deterministic algorithm. The DT submits the request C for corruption detection to the DO. At first, the DT generates the signature $Sig = Enc_{sk_{DT}}\left(H\left(\mathrm{TS} \,||\, \mathrm{ID} \,||\, l_q\right)\right)$ of the data label l_q to be queried, personal ID information and timestamp TS with the aid of the private key sk_{DT} and the Hash function H and sends the $Sig \,||\, \mathrm{TS} \,||\, \mathrm{ID} \,||\, l_q$ to the DO. After receiving the information, the DO validates the effectiveness of ID and TS at first and then verifies the signature $ver\left(Sig, \left(TS \,||\, ID \,||\, l_q\right)\right) \Leftrightarrow H\left(TS \,||\, ID \,||\, l_q\right) = Dec_{pk_{DT}}\left(Sig\right).$ through the public key pk_{DT}. After being validated to be effective, the token T for data corruption detection is first generated by the DO according to l_q. The token generation algorithm is expressed as follows:

1) The set of the data labels to be queried is recorded as Q and the polynomial $p_Q\left(x\right) = \left(\prod_{l_i \in \Omega}\left(x - l_i\right) \Big/ \prod_{l_q \in \mathrm{Q}}\left(x - l_q\right)\right) \prod_{j=q+1}^{\vartheta}\left(x - r_j\right)$ is established. That is, the root of the polynomial corresponds to all data labels excluding those to be queried. In order to hide the length of the token, the length is fixed as ϑ and the random numbers are filled into the rest polynomial.

2) As for all coefficients $\left(\beta_0, \beta_1, \ldots, \beta_{\vartheta}\right)$ in $p_Q\left(x\right)$, $T\left[1\right] = \left(\beta_1, \ldots, \beta_{\vartheta}\right) \cdot \boldsymbol{M}^{-1}$ and $T\left[2\right] = \beta_0$ are calculated.

3) The DO acquires the corresponding corruption detection threshold ε_q in $\mathrm{D_T}$ according to the data label l_q to be detected, which is recorded as $T\left[3\right] = \varepsilon_q$. By assuming $T = \left(T\left[1\right], T\left[2\right], T\left[3\right]\right)$, a triple detection token T is generated.

The execution of token generation algorithm finishes after the DO sends T to the DT.

4.5 Corruption detection algorithm

$\mathrm{RS} \leftarrow \mathrm{Check}\left(T, \boldsymbol{I}'', \boldsymbol{M}_{\mathrm{D}}\right)$ belongs to a deterministic algorithm. The DT executes the data corruption detection on the encrypted detection index $\boldsymbol{I}''$ by utilizing the detection token T. The specific process of the algorithm is displayed as follows:

1) $T\left[1\right]$ is output from the server to first calculate $\mathbf{Y} = T\left[1\right] \cdot \boldsymbol{M}_{\mathrm{D}} = \left(\gamma_1, \gamma_2, \ldots \gamma_{\vartheta}\right)$;

2) $T\left[2\right]$ is output from the server. As for all parameters γ_i in $\mathbf{Y}$, $\gamma_i' = Enc_{RSA}\left(\gamma_i\right) \cdot Enc_{RSA}\left(T\left[2\right]\right)$ is calculated by the server through $pk = \left(e, n\right)$. A new vector

$\mathbf{Y}'=(\gamma_1',\gamma_2',\dots\gamma_g')$ is attained after completing the calculation.

3) $R(x)=\mathbf{Y}'\cdot\boldsymbol{I}''$ is calculated by the server and transmitted to the DT. The DT decrypts the $R(x)$ based on $sk=d$ under the help of the DO to attain the $R'(x)$. The root of $R'(x)$ is the detection index corresponding to l_q, that is, the DT acquires the set $\Delta=(\mu_1,\sigma_1),(\mu_2,\sigma_2),\dots,(\mu_{n_q},\sigma_{n_q})$ of the expected values and variances corresponding to the data to be detected.

4) $R_S(x)=\prod_{j=1}^{n_q}\frac{1}{\sqrt{2\pi\sigma_j}}\exp\left(-\frac{(x_j-\mu_j)}{2\sigma_j^2}\right)$ is calculated according to the parameters in the set Δ and the data $\mathrm{D}(l_q)=(x_1',x_2',\dots x_{n_q}')$ to be detected. $T[3]$ is output from the server and compared with $R_S(x)$. On condition of $R_S(x)<T[3]$, the data are corrupted, recorded as $\mathrm{RS{=}ture}$; otherwise, the data are qualified, recorded as $\mathrm{RS{=}false}$.

The execution of the corruption detection algorithm finishes.

5. Security proof and experimental test

The security proof is first conducted on the DCDC scheme according to the security definition in Section 3; afterwards, the efficiency analysis is performed on performances of various stages of the scheme; finally, the above stages are separately tested through simulation to verify the efficiency analysis.

5.1 Security proof

According to Theorem 1, it is supposed that the adversary A can win the security test mentioned above at a non-negligible advantage and a semantically secure algorithm B, which can break the encryption algorithm based on the random oracle model(ROM), is established according to A. B has access to the random oracle O_f and f can be a random algorithm or a multiplicative homomorphic algorithm. The encryption operation in the DCDC scheme is replaced with the algorithm B; afterwards, the semantic security of the DCDC scheme is validated through the security test given in the security definition in Section 3. The specific process of the test is described as follows:

```
$\mathrm{Game}_{A,\ B}(\lambda')$:
$\mathrm{D},\mathbf{L},\boldsymbol{I}\leftarrow B(\lambda')$
$\mathrm{MK},\mathrm{K}_{\mathrm{DT}}\leftarrow \mathrm{KeyGen}(\lambda')$
$\boldsymbol{I}'',\ \boldsymbol{M}_{\mathrm{D}}\leftarrow \mathrm{IndexEnc}(\mathrm{MK},\ \boldsymbol{I})$
$for\ 1\le i\le c, do:$
    $c_i\xleftarrow{\substack{one\ check\\ each\ time}}A(C_1,C_2,\dots,C_c)$
    $T_{C_i}\leftarrow \mathrm{TokenGen}(\mathrm{MK},\ C_i,\ \mathrm{ID})$
    $\mathrm{RS}\leftarrow \mathrm{Check}(T_{C_i},\ \boldsymbol{I}'',\ \boldsymbol{M}_{\mathrm{D}})$
    $\omega\leftarrow A(Y_0,Y_1\in\mathbf{L})\in\{0,1\}$
        $T_{C_\omega}\leftarrow \mathrm{TokenGen}(Y_\omega,\ \mathrm{MK},\ \mathrm{ID})$
        $\mathrm{RS}\leftarrow \mathrm{Check}(T_{C_\omega},\ \boldsymbol{I}'',\ \boldsymbol{M}_{\mathrm{D}})$
$output\ \omega'$
```

After ending the test, the parameter ω' is output as a conjecture by A. If the output result equals 0, it means that f in the random oracle O_f represents a random algorithm, recorded as $\Pr\left[P_f = 0\right]$; otherwise, f stands for a multiplicative homomorphic encryption algorithm, recorded as $P_f = 1$.

If f is regarded as a random algorithm, obviously $\Pr\left[P_f = 0\right] = 1/2$ exists; the adversary A shows the same probability of outputting 1 as B in the case that f is taken as a multiplicative homomorphic encryption algorithm. Therefore, B delivers the same advantage in distinguishing the semantically secure encryption algorithm in the random oracle model as A in winning the security test. However, the algorithm B does not exist according to the definition of the semantically secure encryption system. Furthermore, the conclusion is drawn that the adversary A who can win the security test at a non-negligible advantage does not exists. Thus, it is validated that the DCDC algorithm is semantically secure.

Next, the index security and token security of the algorithm are explored. When generating the detection index based on the DCDC scheme, the detection labels are encrypted by applying the pseudo-random function and random matrix and the detection parameters are encrypted according to the RSA encryption algorithm, which guarantees the confidentiality of the encrypted index $\boldsymbol{I}''$ and the matrix $\boldsymbol{M}_{\mathrm{D}}$; moreover, the encrypted index $\boldsymbol{I}''$ is filled with the random numbers to ensure the consistent lengths of all indexes. It guarantees that it fails to attain the effective information on the encrypted data according to the length of indexes. As for the detection token, $T[3]$ is not uploaded to the server but stored in the DT and the DT is regarded as a reliable entity after being certificated. Considering this, it is thought that $T[3]$ is secure in the algorithm. During the token generation, the coefficient of $T[1]$ is encrypted based on the random matrix while that of $T[2]$ is filled with random numbers and cannot be determined. Therefore, the value of the detection token varies for different detection behaviors. As a result, the adversary fails to deduce that the same label is used during multiple detections based on the detection operation, which guarantees the security of the detection process and mode of the DCDC scheme.

5.2 Efficiency analysis

In the test, the number of the datasets to be detected and the maximum dimension of the vector of data to be detected are separately set as m and n. The efficiency of the DCDC scheme is mainly analyzed from three aspects.

In the stage of the detection index generation, it is supposed that all indexes $\boldsymbol{I}$ have been generated and the calculation efficiency of the expected values and variances of data is not taken into account. The DO needs to encrypt the index $\boldsymbol{I}$, in which $\boldsymbol{I}$ contains the list of m data. Each datum corresponds to a $2n$-order polynomial. Owing to all coefficients of m polynomials are subjected to the RSA encryption, it is necessary to perform $2nm$ times of encryption operation. Subsequently, the matrix $\boldsymbol{M}'$ is generated and it is essential to conduct m^3 times of multiplication operation. Obviously, the encryption operation presents higher time complexity than multiplication operation. Thus, the computational complexity in the stage of detection index generation lies in $O(mn)$ times of encryption operation.

Subsequently, the detection token generation stage is analyzed. It is also supposed that the DT has been certificated by the DO, so its certification efficiency is not discussed. In the process of the detection token generation, it is necessary to conduct m^2 times of multiplication operation on $T[1]$ due to the involvement of the invertible matrix $\boldsymbol{M}^{-1}$ and $T[2]$ is subjected to one time of the multiplication operation; moreover, $T[3]$ shows the complexity of $O(1)$ owing to it is only retrieved in the dictionary D_{T}. Therefore, the computational complexity in the detection token generation stage is shown as $O(m^2)$ times of multiplication operation and $O(1)$ times of exponential operation.

During the corruption detection stage, as it is data labels that remain to be detected, the detection process is divided into two steps: in terms of the first step, the index content of the data to be detected is attained. For this purpose, it is necessary to conduct m^2 times of multiplication operation to obtain $\mathbf{Y}$ and m times of exponential operation to acquire $\mathbf{Y}'$. Moreover, it is essential to implement m times of multiplicative homomorphic operation. Therefore, the computational complexity of the detection process in the first step is shown as $O(m^2)$ times of multiplication operation and $O(m)$ times of exponential operation. In terms of the second step, the data are subjected to corruption detection. According to the detection equation, it is necessary to run n^2 times of multiplication operation and thus the computational complexity corresponds to $O(n^2)$ times of multiplication operation.

5.3 Experiment

Finally, the experimental analysis is conducted on the DCDC scheme. The experimental test is realized by using Java language under Win10 operating system, in which the samples of test data and the set of data labels are self-defined.

At first, the efficiency test is separately performed on three stages (detection index generation, detection token generation and data corruption detection, recorded as A, B and C, respectively) involved in the DCDC scheme according to the number m of the ciphertext data labels stored in the server. The test data are partitioned into five groups according to the number of data labels, in which one datum remains to be detected each time and the maximum dimension of the data vector equals 50. The test results are shown in **Fig. 3**.

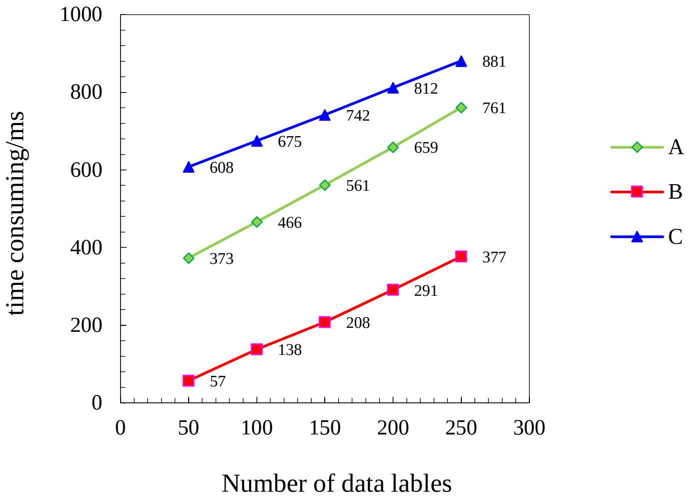

Fig. 3. The test results obtained according to the number of data labels

The following conclusion can be drawn based on the test results: the durations for the detection index generation, detection token generation and data corruption detection gradually increase with the growth of the number of data labels, basically showing a directly proportional relationship.

Subsequently, according to the maximum dimension n of the data vector stored in the server, the efficiency test is carried out on three stages (detection index generation, detection token generation and data corruption detection, recorded as A, B and C, respectively) involved in the DCDC scheme. The test data are divided into five groups based on the maximum dimension of the data vector, in which one datum remains to be detected each time and the number of data labels is always set as 100. The test results are displayed in **Fig. 4**.

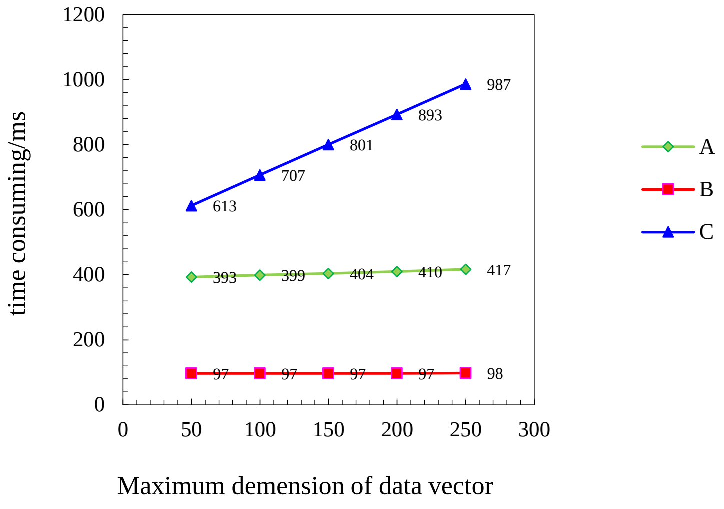

Fig. 4. The test results according to the maximum dimension of the data vector

According to the test results, it can be concluded that the durations for the detection index generation and data corruption detection gradually rise with the increasing maximum dimension of the data vector, basically delivering a directly proportional relationship; by contrast, the duration for the detection token generation does not show a close relationship with the increase of the maximum dimension of the data vector, and the detection token generation efficiency is basically a constant, this is also consistent with the result of token generation efficiency analysis in Section 5.2.

Since this scheme is mainly composed of the above three stages above (detection index generation, detection token generation and data corruption detection), it is obviously that the total time of DCDC scheme is the sum of A, B and C according to this work.

6. Conclusion

To solve the problem that data corruption cannot be detected on the ciphertext data stored in cloud environment in the majority of existing schemes, the DCDC is proposed. On the basis of the anomaly detection method based on the Gaussian model, a corruption threshold is separately calculated for each type of data in the scheme through various technologies such as precision, recall rate and F value. According to parameters (e.g. expected value and variance) of data and cryptography knowledge (such as RSA homomorphic encryption and pseudo-random function), the encrypted corruption detection indexes and tokens are constructed. It is possible to realize the corruption detection on the ciphertext data stored in cloud environment by utilizing the corruption detection tokens for each type of data. The random numbers are introduced in the processes of generating the detection indexes and tokens in the scheme, which protects the privacy of the indexes and tokens and guarantees the confidentiality of key information in the detection process. Through the security proof, it can be found that the algorithms in the scheme are semantically secure. Through efficiency analysis and experimental test, it is demonstrated that only multiplication and exponential operations are applied in the scheme during the token generation and corruption detection, showing low computational cost and promising application prospect.

References

[1] Daniel Barbará, Goel R, Jajodia S, "Using Checksums to Detect Data Corruption," *Lecture Notes in Computer Science*, vol. 1777, no. 1, pp. 136-149, Mar, 2000. Article (CrossRef Link)

[2] Chen C L P, Zhang C Y, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, no. 8, pp. 314-347, Aug, 2014. Article (CrossRef Link)

[3] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in telecommunication industry using data certainty," *Journal of Business Research*, vol. 94, pp. 290–301, Jun, 2019. Article (CrossRef Link)

[4] Jun Wu, Li Shi, Wen-Pin Lin, Sang-Bing Tsai, Yuanyuan Li, Liping Yang, Guangshu Xu, "An Empirical Study on Customer Segmentation by Purchase Behaviors Using a RFM Model and K-Means Algorithm," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8884227, p. 7, Nov, 2020. Article (CrossRef Link)

[5] D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: a case study of RFM model-based customer segmentation using data mining," *Journal of Database Marketing & Customer Strategy Management*, vol. 19, no. 3, pp. 197–208, 2012. Article (CrossRef Link)

[6] Y. Zhang, "An adaptive flow counting method for anomaly detection in SDN," in *Proc. of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 25-30, Dec, 2013. Article (CrossRef Link)

[7] C. Lu, J. Shi, J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proc. of the 2013 IEEE International Conference on Computer Vision*, pp. 2720-2727, Oct, 2013. Article (CrossRef Link)

[8] M. Bertini, A. Bimbo, L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Comput. Vis. Image Underst.*, vol. 116, no. 3, pp. 320-329, Oct, 2012. Article (CrossRef Link)

[9] L. Huang, Z. Zhao, K. Xing, "Web data extraction based on edit distance," *Journal of Computer Applications*, vol. 32, no. 6, pp. 1662 – 1665, 2012. Article (CrossRef Link)

[10] J. Wang, Z. Xu, "Spatio-temporal texture modeling for real-time crowd anomaly detection," *Comput.Vis. Image Underst.*, vol. 144, pp. 177-187, Mar, 2009. Article (CrossRef Link)

[11] Moscoso M, Novikov A, Papanicolaou G et al., "Imaging with highly incomplete and corrupted data," *Inverse Problems*, vol. 36, no. 3, pp. 035010, Feb, 2020. Article (CrossRef Link)

[12] R. Leyva, V. Sanchez, C. T. Li, "Video anomaly detection with compact feature sets for online performance," *IEEETrans. Image Process*, vol. 26, no. 7, pp. 3463-3478, Jul, 2017. Article (CrossRef Link)

[13] Mahadevan V, Li W X, Bhalodia V et al., "Anomaly detection in crowded scenes," *Computer Vision & Pattern Recognition*, IEEE, pp. 13-18, Jun, 2010. Article (CrossRef Link)

[14] Chandola V, Banerjee A, Kumar V, "Anomaly Detection for Discrete Sequences: A Survey," *IEEE Transactions on Knowledge & Data Engineering*, vol. 24, no.5, pp. 823-839, 2012.

Article (CrossRef Link)

[15] Tahir S, Rajarajan M, "Privacy-Preserving Searchable Encryption Framework for Permissioned Blockchain Networks," in *Proc. of 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1628-1633, Jun, 2019. Article (CrossRef Link)

[16] Park E, Jeong J, Han W S et al., "Estimation of Groundwater Level Based on the Robust Training of Recurrent Neural Networks Using Corrupted Data," *Journal of Hydrology*, vol. 582, no. 1, pp. 124512, Mar, 2020. Article (CrossRef Link)

[17] Biswas S, Babu RV, "Real time anomaly detection in H. 264 compressed videos," in *Proc. of the fourth National Conference on Computer Vision, Pattern Recognition., Image Processing and Graphics*, pp. 1-4, Dec, 2013. Article (CrossRef Link)

[18] J. Liu, Y. Feng, W. Liu, D. Orlando and H. Li, "Training Data Assisted Anomaly Detection of Multi-Pixel Targets In Hyperspectral Imagery," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3022-3032, Apr, 2020. Article (CrossRef Link)

[19] Shehab D, Ammar H, "Statistical detection of a panic behavior in crowded scenes," *Machine Vision and Applications*, vol. 30, no. 5, pp. 919-931, Jul, 2019. Article (CrossRef Link)

[20] Yue G, Jun ping D, Meiyu, "Abnormal event detection in tourism video based on salient spatio-temporal features and sparse combination learning," *World Wide Web*, vol. 22, no. 2, pp. 689-715, Mar, 2019. Article (CrossRef Link)

[21] Pan, X., S. Tang, and Z. Zhu, "Privacy-Preserving Multilayer In-Band Network Telemetry and Data Analytics," in *Proc. of IEEE/CIC International Conference on Communications in China (ICCC) 2020 IEEE*, Jul, 2020. Article (CrossRef Link)

[22] A Ouafa Amira, "Weighted-capsule routing via a fuzzy gaussian model – ScienceDirect," *Pattern Recognition Letters*, vol. 138, no. 1, pp. 424-430, Aug, 2020. Article (CrossRef Link)

[23] Nick Iliev, Alberto Gianelli, Amit Ranjan Trivedi, "Low Power Speaker Identification by Integrated Clustering and Gaussian Mixture Model Scoring," *Embedded Systems Letters IEEE.*, vol. 12, no. 1, pp. 9-12, May, 2020. Article (CrossRef Link)

[24] Qu J, Du Q, Li Y et al., "Anomaly Detection in Hyperspectral Imagery Based on Gaussian Mixture Model," *IEEE Transactions on Geoscience and Remote Sensing*, pp.1-14, Dec, 2020. Article (CrossRef Link)

[25] Liang J M, Shen S Q, Li M et al., "Quantum Anomaly Detection with Density Estimation and Multivariate Gaussian Distribution," *Physical Review A*, vol. 99, no. 5, pp. 52310-52310, 2019. Article (CrossRef Link)

[26] Cao N, Wang C, Li M et al., "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222-233, Jan, 2014. Article (CrossRef Link)

[27] Zarezadeh M, Mala H, Ashouri-Talouki M, "Multi-keyword ranked searchable encryption scheme with access control for cloud storage," *Peer-to-Peer Networking and Applications*, vol. 13, no. 1, pp. 207-218, Jun, 2020. Article (CrossRef Link)

[28] Lu Q, Yao X, "Clustering and learning Gaussian distribution for continuous optimization," *IEEE Transactions on Systems Man & Cybernetics Part C Applications & Reviews*, vol. 35, no. 2, pp. 195-204, May, 2000. Article (CrossRef Link)

[29] Birra T, Li T, Daniel B et al., "Detection of Isocitrate Dehydrogenase Mutated Glioblastomas Through Anomaly Detection Analytics: A Pilot Study," *Neurosurgery*, vol. 89, no, 2, pp, 323-328, Apr, 2021. Article (CrossRef Link)

[30] Meng L, Zhang J, "Process Design of Laser Powder Bed Fusion of Stainless Steel Using a Gaussian Process-Based Machine Learning Model," *JOM*, vol. 72, no. 1, pp. 420-428, Mar, 2020. Article (CrossRef Link)

[31] Velupillai S, Dalianis H, Hassel M et al., "Developing a standard for de-identifying electronic patient records written in Swedish: Precision, recall and F-measure in a manual and computerized annotation trial," *International Journal of Medical Informatics*, vol. 78, no. 12, pp. e19-e26, Dec, 2009. Article (CrossRef Link)

Guo Sixu, born in 1996. Master degree. His main research interests include cryptography and information security.

Email: 792669696@qq.com, guosixu@chinamobie.com

He Shen, born in 1980, PhD, senior engineer. His main research interests include network security, internet of things security, mobile internet security, trusted computing and other aspects of research.

Email:heshen@chinamobile.com

Suli, born in 1981, PhD, professor level senior engineer. His main research interests include the research of mobile communication network security and data security.

Email: suli@chinamobile.com

Zhang Xinyue, born in 1994, Master degree. Her main research interest is searchable encryption.

Email: zhangxinyue@chinamobile.com

Geng Huizheng, born in 1988, Master degree. His main research interests include the research of mobile communication network security and data security.

Email: genghuizheng@chinamobile.com

Sun Yang, born in 1983, Master degree. Her main research interests include NLP, AI and data security.

Email: sunyangyjy@chinamobile.com