Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **6**, November **2018**

Secure Cloud Storage Model Based on TPKE and ECE

Hui Xia[a,1*] and Weiji Yang[b,1*]

[a]School of Software, Shenyang Normal University,
NO. 253 HuangHe North Street, Shenyang, China,

[b]School of Life Science, Zhejiang Chinese Medical University,
548 Binwen Road, HangZhou, China,

*freund_xia@126.com, *yangweiji@163.com

Received May, 2018; Revised July, 2018

Abstract. *The paper conduct the analysis and research aiming at the issue of data confidentiality and fault-tolerant in cloud storage environments. It is pointed out that the existing solutions can solve either of the confidentiality or fault-tolerance issues, but cannot consider both together.For this purpose, a secure cloud storage system with data confidentiality and fault-tolerant (SCSM-DCF) is proposedAwhich is based on threshold public key encryption scheme and erasure codes over exponents. The formal definition, the definition of security and the communication protocols between entities are given in the paper. Finally, the performance of the model is analyzed, and the result indicates that the model is not only correct and secure, but also has the higher efficiency.*
Keywords: Cloud storage; Threshold public key encryption; Erasure codes over exponent; Confidentiality; Fault-tolerant

1. **Introduction.** Cloud storage[1,2] is a combination of distributed storage technology and virtualization technology. Data confidentiality and fault tolerance are two important attributes and most concerned of data security in cloud storage area.Data confidentiality means that only the data owner and the authorized user can store data and access to data in plaintext, any other users or cloud storage service providers are unable to get the data in plaintext, theoretically end the possibility of all data leakage.Data fault tolerance[3,4] refers that to what extent the user's data can be available in the event of an accident (such as hard disk damage, IDC fire, network failure, etc.). At present, there are so many research about the confidentiality and fault tolerance of data, however, the two research programs which are considered comprehensively are relatively rare. So security cloud storage model based on Threshold Public Key Encryption(TPKE) and Erasure Codes over Exponent(ECE) is proposed.

1.1. **Motivation.** Cloud storage brings many benefits such as economies of scale and high availability, its core technical characteristics (virtualization, distributed, resource sharing, etc.) also determine its natural hidden dangers in terms of security. Therefore, the issue of data security in cloud storage has become one of the most important research topics in cloud security research. In particular, the problem of data confidentiality and fault tolerance in cloud storage has received more and more attention.

1.2. **Related work.** To ensure the confidentiality of data, the data should be stored in ciphertext in the cloud storage system. However, the encryption method brings the computational overhead. Therefore, reliable data confidentiality should be implemented with the lowest possible computational cost[5,6] Proxy re-encryption[7], broadcast encryption[8] and attribute-based encryption[9] are the main methods of data confidentiality protection in cloud storage.At the same time,considering the privacy of users, the server in the ciphertext of any operation can directly correspond to the corresponding plaintext operations by fully homomorphic encryption[10]. Encryption method to support search is also an important application of cloud storage applications. Recent literature[11] proposed a single-key to search public-key cryptosystem solution based on the bilinear pairing function. After that, many scholars proposed an encryption method to support search, including literature[12], which proposed a fuzzy search scheme for encrypted data. The literature[13] proposed an efficient search scheme for encrypted data that supports the ordering of returned results; the encrypted search scheme proposed in[14] supports multi-keyword search and is able to sort the returned results.

1.3. **Goal and organization.** Data fault tolerance is the key to the reliability of cloud storage systems. its mechanism includes two categories: one is a complete data backup mechanism, that is mirrired method, the other is based on erasure code method. Mirror method, also known as a complete copy of the method is to copy multiple copies of the stored data in order to achieve redundant backup. The mirror method is the easiest way to resist any server failure and maintain the flexibility of the system. However, the method is with very low storage efficiency and high cost. The erasure code technology is a kind of coding technology derived from channel transmission, which is introduced into the distributed storage area for its tolerance the loss of multiple data frames. In the distributed storage system, the erasure codes encode the data into blocks and check blocks, which are stored in different nodes, respectively. When the nodes in the system loses effectiveness or part of the data block is damaged, the storage system can still recover the original files according to the remaining data blocks, so that ensure the reliability of the data.

Section 2 provides the formal definition of SCSM-DCF, security, and the inter-entity communication protocol. Section 3 analyzes the security of SCSM-DCF. SCSM-DCF performance is analyzed by calculation and storage cost in Section 4.Finally, Section 5 summarizes the paper.

2. SCSM-DCF design.

2.1. **Model description and formal definition.** SCSM-DCF is a secure cloud storage system with data confidentiality and fault-tolerant,Abbreviated as SCSM-DCF. SCSM-DCF includes the client and the server. The client is the cloud storage user, that is, the owner of the data. The server is divided into two kinds: storage server and key server.

SCSM-DCF mainly includes three processes: initialization, data storage and data acquisition.

1. Initialization

   Server A selects and calculates the common parameters. User A has its own storage space, public key PK_A and private key SK_A. User A will publish its own public key, and share its private key to key servers, the number of servers need to reach a minimum threshold t. Thus, each selected key server $KS_i(1 \leq i \leq m)$ has a key fragment $SK_{A,i}$ of the user private key SK_A.
2. Data storage

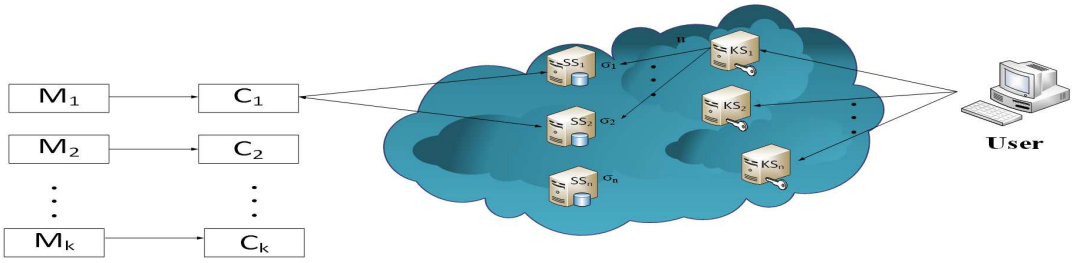

FIGURE 1. SCSM-DCF Architecture

User A divides file F into k blocks $(M_1, M_2, ..., M_K)(1 \leq i \leq k)$ and generates a session key K. The user encrypts the file block M_i using K to obtain the ciphertext $C_i = E(K, M_i)$. Then, the ciphertext is sent to n storage servers $SS_i(1 \leq i \leq n)$, and SS_i uses the EC-C to encode the ciphertext. For session key K, it is first divided into k blocks $(K_1, K_2, ..., K_K)$ and performs a threshold public key encryption to $K_i(1 \leq i \leq k)$ using the user's public key PK_A. The ciphertext of the session key is sent to the v storage server, and these storage servers is randomly selected and encoded using the EC-C to block the received ciphertext to form the final stored data σ_i.

3. Data collection

If you want to obtain k block data, user A first access to the file according to the encrypted content for the master key by the corresponding relationship between the file and the master key. Then the user send the instructions to the m key server. Each key server obtains the stored session key from the u storage server after receiving instructions, and then performs partial decryption on the acquired data, then the user A collects the partial decryption result from the key server, i.e., the decryption part. The result of these partial decryption is combined by the user A to restore the original session key K.

2.2. **Security definition.**

Definition 2.1. *If SCSM-DCF is secure, its key encryption scheme should be able to resist chosen plaintext attack(CPA for short).*

In the threat model of the SCSM-DCF, considering such an adversary A, he wants to hijack all the storage servers and less than t-1 key servers to destroy the confidentiality of a target user's key. The user will not tamper with the stored data,but try to infer the data content. This article uses the standard CPA to simulate this attack process, where the CPA is associated with the threshold public key encryption scheme.

2.3. **Key Technologies.** The following is the threshold public key encryption scheme implementation process.

Step 1. Run SetUp(1^λ) algorithm, generate system parameters u, $u = (p, G_1, G_2, \widetilde{e}, g)$

Step 2. Run KeyGen(u) algorithm, generate public/private key pairs for users:

$$PK = g^x, SK = x, x \in_R Z_p \tag{1}$$

Step 3. Run ShareKeyGen(SK, t, m) algorithm. The key fragment is calculated by the polynomial $f(z)$:

$$f(z) = SK + a_1 z + a_2 z^2 + ...a_{t-1} z^{t-1} (\text{mod } p), a_1, a_2, ...a_{t-1} \in_R Z_P \tag{2}$$

Step 4. Run Enc(PK, M) algorithm. The message M$\in G_2$ is encrypted to obtain the ciphertext C, where C is calculated as follows:

$$C = (\alpha, \beta, \gamma) = (g^r, h, M\,\widetilde{e}(g^x, h^r)), r \in_R Z_p, h \in_R G_1 \tag{3}$$

Step 5. Run ShareDec(SK_i, C) algorithm, use the key fragmentation to perform partial decryption on a given key file and output decrypted slices. Let $C = (\alpha, \beta, \gamma)$, by using the key slice SK_i, the decryption slice of the ciphertext ς_i can be obtained as below:

$$\varsigma_i = (\alpha_i, \beta_i, \beta_i^{'}, \gamma_i) = (\alpha, \beta, \beta^{SK_i}, \gamma) \tag{4}$$

Step 6. Run Combine$(\varsigma_{i_1}, \varsigma_{i_2}, ..., \varsigma_{i_t})$, algorithm, by using the Lagrangian interpolation algorithm for the exponential position and the value of t decrypted slices $(\beta_{i_1}^{'}, \beta_{i_2}^{'}, ..., \beta_{i_t}^{'})$, we can get the equation $\beta^{SK} = \beta^{f_{(0)}}$, where

$$\beta^{SK} = \prod_{i \in S} \left((\beta_i^{'})^{\prod_{r \in S, r \neq i} \frac{-r}{i-r}} \right) S = \{i_1, i_2, ..., i_t\} \tag{5}$$

for any $1 \leq j \leq t$, there is $\varsigma_{i_j} = a_{i_j}, \beta, (\beta)_{i_j}^{'}, \gamma_{i_j}$.

The final output is $M = \gamma / \widetilde{e}(\alpha, \beta^{f(0)})$.

The same set of ciphertexts with the same h-value has the property of multiplicative homomorphism. That is, given the ciphertexts of M_1 and M_2, the ciphertext of $M_1 \times M_2$ can be calculated without knowing the private keys x, M_1 and M_2. Let $C_1 = Enc(PK, M_1)$, $C_2 = Enc(PK, M_2)$, where, $C_1 = (g^{r_1}, h, M_1\,\widetilde{e}(g^x, h^r))$. The ciphertext C of $M1 \times M2$ is calculated under the condition that the public key PK is known.

$$C = (g^{r_1} g^{r_2}, h, M_1\,\widetilde{e}(g^x, h^{r_1}) M_2\,\widetilde{e}(g^x, h^{r_2})) = (g^{r_1+r_2}, h, M_1 M_2\,\widetilde{e}(g^x, h^{r_1+r_2})) \tag{6}$$

2.4. **Inter - entity communication protocol.** SCSM-DCF inter-entity communication protocols include data storage protocols and data acquisition protocols.

2.4.1. *The data storage protocol.* In SCSM-DCF, the process of storing k messages is as follows:

Step 1. Data encryption. The user encrypts k messages with the same h_{ID} through the threshold public key encryption scheme TPKE, here, $h_{ID} = H(M_1 \parallel M_2 \parallel ...M_K)$ is the identifier of a set of messages $M_1, M_2, ..., M_k$, The ciphertext of message M_i is as below:

$$C_i = (\alpha_i, \beta_i, \gamma_i) = (g^{r_i}, h_{ID}, M_i\,\widetilde{e}(g^x, h_{ID}^r)), r_i \in_R Z_P, l \leq i \leq k \tag{7}$$

Step 2. Ciphertext distribution. For each C_i, the user randomly selects the v storage server and sends a copy of C_i to the selected server.

Step 3. Code. The storage server SSj groups the received ciphertexts with the same h_{ID} into N_j. The storage server SSj randomly selects coefficients $g_{i,j}$ from Z_p for each ciphertext $C_i \in N_j$, for $C_i \notin N_j$, gte $g_{i,j} = 0$. Finally generation matrix $G = [g_{i,j}]_{1 \leq i \leq k, 1 \leq j \leq k}$ of erasure codes over exponent is obtained. Each storage server calculates (A_j, B_j) and stores data σ_j.

$$A_j = \prod_{C_i \in N_j} \alpha_i^{g_{i,j}}, B_j = \prod_{C_i \in N_j} r_i^{g_{i,j}}, \sigma_j = (A_j, h_{ID}, B_j, (g_{1,j}, g_{2,j}, \ldots g_{k,j})) \tag{8}$$

(A_j, h_{ID}, B_j) is the ciphertext of $\prod_{1\le i\le k} M_i^{g_{i,j}}$, because

$$\begin{aligned} A_j, h_{ID}, B_j &= \left(\prod_{C_i\in N_j} (g^{r_i})^{g_{i,j}}, h_{ID}, \prod_{C_i\in N_j} \left(M_i \widetilde{e}(g^x, h_{ID}^{r_i})\right)^{g_{i,j}} \right) \\ &= \left(g^{\prod_{C_i\in N_j} r_i g_{i,j}}, h_{ID}, \left(\prod_{C_i\in N_j} M_i^{g_{i,j}} \right) \left(\widetilde{e}\left(g^x, h_{ID}^{\prod_{C_i\in N_j} r_i g_{i,j}} \right) \right) \right) \\ &= \left(g^{\widetilde{r}}, h_{ID}, \left(\prod_{C_i\in N_j} M_i^{g_{i,j}} \right) \widetilde{e}\left(g^x, h_{ID}^{\widetilde{r}} \right) \right), \widetilde{r} = \prod_{C_i\in N_j} r_i g_{i,j}. \end{aligned} \tag{9}$$

After the information is encrypted, it is distributed to the storage server, and each storage server combines all the received ciphertexts, stores the final result and the selected coefficients.

2.4.2. *Data acquisition protocol.* Step 1. Get the instruction. The data owner issues a data acquisition instruction to the m key servers and sends the information identifier h_{ID} to the key server.

Step 2. Partially decrypted. Each key server SK_i randomly queries u storage servers for the data with the identifier h_{ID}, finally obtains up to u storage data σ_j from the storage server, then the key server SK_i uses its key fragment SK_i to execute the algorithm ShareDec on each received ciphertext to obtain decrypted fragments of the ciphertext. Assume that the key server SK_i receives the stored data σ_j. SK_i decrypts the ciphertext (A_i, h_{ID}, B_j) into decrypted fragments $(A_j, h_{ID}, h_{ID}^{SK_i}, B_j)$, then sends the data to the user (A_i, h_{ID}, B_j) is decrypted into decrypted slices, and the ciphertext $(A_j, h_{ID}, h_{ID}^{SK_i}, B_j)$ and then send the data to the user: $\widetilde{\varsigma}_{i,j} = (A_j, h_{ID}, h_{ID}^{SK_i}, \beta_j, (g_{1,j}, g_{2,j}, ..., g_{k,j}))$.

Step 3. Combine and decode. The user selects $\widetilde{\varsigma}_{i_1,j_1}, \widetilde{\varsigma}_{i_2,j_2}, ..., \widetilde{\varsigma}_{i_t,j_t}$ from all received data $\widetilde{\varsigma}_{i,j}$, using the Lagrangian interpolation for the exponential terms to calculate:

$$h_{ID}^{SK} = h_{ID}^{f(0)} = h_{ID}^x, i_1 \neq i_2 \neq ... \neq i_t, S = \{i_1, i_2, ..., i_t\}, h_{ID}^x = \prod_{i\in S} (h_{ID}^{SK_i})^{\prod_{r\in S, r\neq i} \frac{-i}{r-i}} \tag{10}$$

If the number of $\widetilde{\varsigma}_{i,j}$ received is greater than t, the user randomly selects the t part of them, if the number is less than t, the data acquisition fails. After h_{ID}^x is obtained, the user looks up all the data received and select $\widetilde{\varsigma}_{i_1,j_1}, \widetilde{\varsigma}_{i_2,j_2}, ..., \widetilde{\varsigma}_{i_t,j_t}, j_1 \neq j_2 \neq ...j_k$. By using h_{ID}^x, the user decrypts $\widetilde{\varsigma}_{i,j}$ as $w_j((i,j) \in \{(i_1,j_1),(i_2,j_2),...,(i_k,j_k)\})$, where $K = [g_{i,j}]_{1\le i\le k, j\in\{j_1,j_2,...j_k\}}$. If K is irreversible, the data acquisition process fails. Otherwise, the user succeeds in obtaining the following $M_i (1 \le i \le k)$:

$$w_j = \frac{B_j}{\widetilde{e}(A_j, h_{ID}^x)} = \prod_{c_i\in N_j} M_l^{g_{i,j}} \tag{11}$$

$w_{j_1}^{d_{1,i}} w_{j_2}^{d_{2,i}} ... w_{j_k}^{d_{k,i}} = M_1^{\sum_{l=1}^k g_{1,jl}^{d_{l,i}}} M_2^{\sum_{l=1}^k g_{2,jl}^{d_{l,i}}} ... M_k^{\sum_{l=1}^k g_{k,jl}^{d_{l,i}}} = M_1^{T_1} M_2^{T_2} .. M_K^{T_K} = M_i$
where if $r = i$, $T_r = \sum_{l=1}^k g_{g_{r,jl}}^{d_{l,i}} = 1$; otherwise $T_r = 0$.

3. **Security analysis.** SCSM-DCF security depends on the key encryption scheme. The following theorem 1 proves that the proposed threshold public key encryption scheme TPKE is secure.

Theorem 3.1. *TPKE is based on the deterministic bilinear[15] Diffie-Hellman assumption in the standard model is chosen to be plaintext safe.*

Proof: This is proved by the anti-evidence method. If SCSM-DCF is secure, then its key encryption scheme should be able to resist the chosen plaintext attack(CPA for short). Assuming that the algorithm A can break TPKE with the advantage of 2ε and win the CPA security game, then the algorithm A' can be constructed to solve the deterministic bilinear[15] Diffie-Hellman problem with the advantage of ε.

1. Setup
   The input of algorithm A is (g, g_x, g^y, g^z, Q) and public parameters $(\widetilde{e}, G_1, G_2, p)$. Then A' will sent (u, PK, t, n) to A , where $u = (p, G_1, G_2, \widetilde{e}, g), PK = g^x$, t is the threshold, n is the number of private key key slices, and $SK = x$ is implied here.
2. Key share query
   Query $q_1, q_2, ..., q_{t-1}$ for t-1 key fragment, A' set $SK_{q1}, SK_{q2}, ..., SK_{q_{t-1}}$ as a random value and send to A. In general, assume that $q_1, q_2, ..., q_{t-1}$ are different from each other.
3. Challenge
   A gives two information M_0 and M_1. A' randomly select in $b \in \{0, 1\}$, calculate the encrypted $M_b : C = Enc(PK, M_b) = (g^y, g^z, M_b Q)$.
4. Output
   A' sends C to A and gets A's output b'. If b'= b, then A' guess $Q = Q_0 = \widetilde{e}(g, g)^{xyz}$ and output 0. If b'≠b, then A' guess $Q = Q_1 = \widetilde{e}(g, g)^r$, and output 1.

When $Q = Q_0 = \widetilde{e}(g, g)^{xyz}$, C is the cipher of M_b; therefore, A has the advantage of 2ε to win the game, then $\Pr[b' = b|Q = \widetilde{e}\,(g, g)^{xyz}] = 1/2 + 2\varepsilon$, For any value r, when $Q = Q_1 = \widetilde{e}\,(g, g)^r$, $(g^y, g^z, M_0 Q)$ and $(g^y, g^z, M_1 Q)$ is indistinguishable, because for any r, there is $M_0 \widetilde{e}\,(g, g)^r = M_1 \widetilde{e}\,(g, g)^{r'}$. So, there are $\Pr[b' = b|Q = \widetilde{e}\,(g, g)^r] = 1/2$.

A' has the advantage of being

$\Pr[A' \to 0|Q = Q_0] \Pr[Q = Q_0] + \Pr[A' \to 1|Q = Q_1] \Pr[Q = Q_1] - 1/2| = |(1/2 + 2\varepsilon) \times 1/2 + 1/2 \times 1/2 - 1/2| = \varepsilon$

Through the above proof, we can see that the proposed threshold public key encryption scheme is safe, so SCSM-DCF is also safe.

4. Cost and performance analysis.

4.1. Cost analysis.

Before performing the cost analysis, first give the relevant symbol description, see Table 1.

Table 1. Symbols

Symbolic	Symbolic meaning	Symbolic	Symbolic meaning
l_1	The length of the elements in group G1	l_2	The length of the elements ingroup G2
Exp_1	Modulus Operators in Group	Exp_2	Modulus Operators in Group
$Mult_1$	Modular multiplication in group G1	$Mult_2$	Modular multiplication in group G2
F_p	*Arithmetic operations on GF (p) domain*	*Pairing*	*Arithmetic operations on e domain*

4.1.1. *Calculate the cost.* The calculation cost of the SCSM-DCF is analyzed by the logarithmic operations,modular exponentiation operations and modular multiplication operations in G_1 and G_2, as well as arithmetic operations on the GF(p) domain,Pairing, Exp_1 Exp_2, $Mult_1$, $Mult_2$ and Fp are represented these operations respectively. This is because the data storage and retrieval process is for a group of data containing k messages, so the computational cost is studied in k units of information. In fact, F_p cost is much lower than that of $Mult_1$ and $Mult_2$. On average (By using the fastest squares and multiplication algorithms), Exp_1 is approximately equal to 1.5 $(log_2p)Mult_1$. Similarly, Exp_2 is approximately equal to 1.5 $(log_2p)Mult_2$.

Since the coefficients can be selected from a smaller set during actual operation, the calculation cost of Exp_1 and Exp_2 is higher than that of the actual situation. The computational cost of Pairing is much higher than that of Exp. However, in order to improve the speed of logarithmic operations, scholars have proposed improved algorithms. The calculation cost of SCSM-DCF is shown in Table 2.

Table 2. Computation cost of SCSM-DCF

operating	Calculate the cost
k data encryption	$kPairing + 2kExp_1 + kMult_2$
coding	$kExp_1 + kExp_2?k1)Mult_1?(k1)Mult_2$
Partially decrypted	$tExp_1$
Data combination	$kPairing + kMult_2 + O()t^2F_p$
decoding	$kExp_2^2?(k1)kMult_2 + O(k)^3F_p$

4.1.2. *Storage cost.* The storage cost of a particular user at the key server is log_2p because the key server only needs to store the key fragments of the private key. The primary storage overhead is the storage server.

Use the bit value as a unit to measure the average storage cost of the storage server. If storing k pieces of information, each storage server SS_i needs to store $A_j, h_{ID}, B_j(A_j, h_{ID} \in G_1; B_j \in G_2)$ and coefficient vector $(g_{1,j}, g_{2,j}, ..., g_{k,j})$. The total cost of the storage server is: $2l_1 = l_2 = k(log_2p)$

Thus, the average storage cost for each message is $(2l_1 + l_2 + k(log_2p))/kl_2$ bits. For a large enough k, this value depends on log_2p/l_2.

4.2. **Performance analysis.** Through the experiment, the performance of this model is analyzed, and the experimental environment is introduced before the experiment is carried out. The client and the server in the model run on two hosts with the same configuration, and the host is loaded with Windows 7 Ultimate 64-bit OS,Pentium E5800 CPU, and 4G memory. Each set of data for the test is the record of the result of the program running in a separate thread, and each data is the average of the 10 running results, excluding the obvious error data.

Experiment 1. Test the file size on the system performance. The experimental data was divided into 4 groups, the first group of 1KB $\sim$ 10KB size of file, recorded as a class A file, the second group of 100KB $\sim$ 200KB size of file, recorded as class B file, the third group of 300KB $\sim$ 400KB size of file, recorded as class C file, and the fourth group of 1000KB 2000KB size of file, recorded as class D file. The number of files of each class is 10. Using the above file for storage testing, the time overhead for encryption (AES encryption) and encoding operations was obtained. The results are shown in Figure 2.

Experiment 2. The effect of the number of test files on system performance. The experimental data are divided into 5 groups. The first group has 10 files, the second

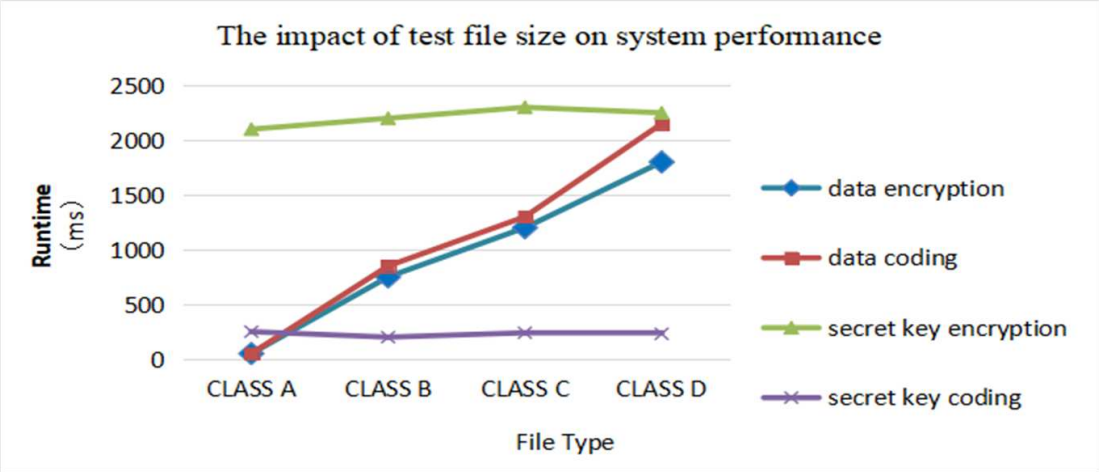

FIGURE 2. Effects of file size on system performance

group has 20 files, and the fifth group has 50 files in turn, the size of each group of files are between 100KB 200KB, the experimental results are shown in Figure 3.

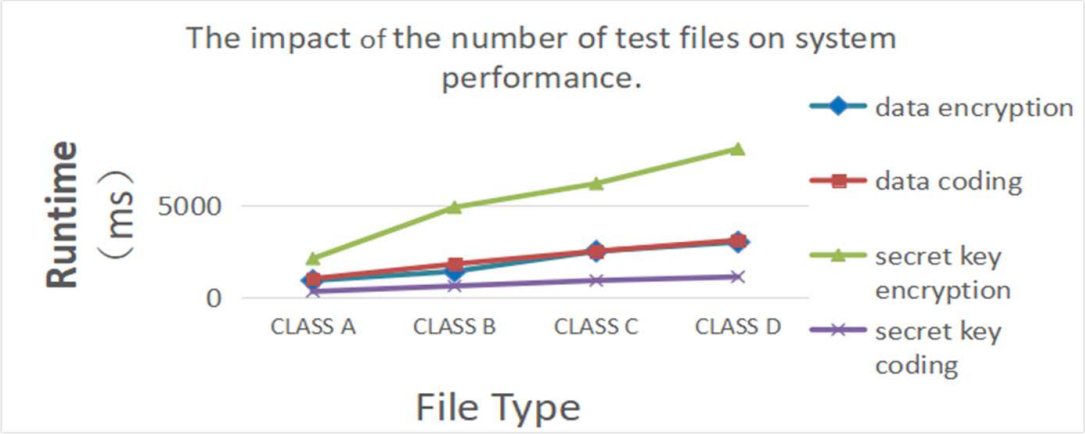

FIGURE 3. Effects of the number of files on system performance

As can be seen from Figure 2, with the gradual increase in file size, data encryption and coding operation of the time-consuming show an increasing trend. This is because the AES encryption belongs to packet encryption.As the volume of the file increases, the number of groups increases and processing time will increase accordingly. EC-E encoding is also read by the packet, when the file size exceeds the buffer settings, the encoding time increases with the file volume growth.Key encryption and coding operation show horizontal trends with the increase of file size.The size of the key is fixed, so the number of keys is related to the number of files. Under the condition that the number of files is the same, the volume of the file has no effect on the related operation of the key.

As can be seen from Figure 3, as the collection of files continues to increase, data encryption and coding operations are increasing time-consuming. Key encryption and coding is of the same time.

5. **Concluding remarks.** Aiming at the problem of data security in existing cloud storage[16-18], a secure storage model that meets the requirements of both confidentiality

and fault tolerance is proposed. It not only solves the problem of data confidentiality in cloud storage systems, but also can resist problems such as server failures. In the model, data is stored in encrypted and encoded form. The storage server cannot have the decoding key and its access rights management is completely controlled by the user. Even if all servers are controlled by the adversary at the same time, the confidentiality of the data can be fully guaranteed. Due to erasure codes over exponents technology, the system is fault-tolerant at the same time. Even if the system data is destroyed, it can still be effectively restored within the allowable range. Therefore, this model proposes data for the current cloud storage. Security issues have a certain theoretical significance and practical application value.

Acknowledgment. This work is partially supported by Scientific Study Project for Institutes of Higher Learning, Ministry of Education, Liaoning Province(LQN201720), and Natural Science Foundation of LaioNing Province, China(20170540819). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] H. Li, WH Sun, F. H. Li, BY Wang, Secure and privacy-preserving data storage service in public cloud, *Journal of Computer Research and Development*, vol. 51, no. 7, pp. 1397-1409, 2014.

[2] Y. X. Fu, S. M. Luo, J. W. Shu, Survey of secure cloud storage system and key technologies *Journal of Computer Research and Development*, vol. 50, no. 1, pp. 136-145, 2013.

[3] N. Cao, C. Wang, M. Li, K Ren, W. J. Lou, Privacy-Preserving multi-keyword ranked search over encrypted cloud data, *In: Proc. of the INFOCOM. Shanghai: IEEE Computer Society*, pp. 829-837, 2011.

[4] K. Julisch, M. Hall, Security and control in the cloud, *Information Security Journal: A Global Perspective*, vol. 19, no. 6, pp. 299-309, 2010.

[5] M. Irfan, M. Usman, Z. Yan, et al, A critical review of security threats in cloud computing, *International Symposium on Computational and Business Intelligence. IEEE*, pp. 105-111, 2016.

[6] R. Padilha, F. Pedone, Confidentiality in the cloud, *IEEE Security & Privacy*, vol. 13, no. 1, pp. 57-60, 2015.

[7] C. Wang, J. Fang, Y. Li, An improved cloud-based revocable identity-based proxy re-encryption scheme[J], 2015.

[8] W. T. Zhu, Towards secure and communication-efficient broadcast encryption systems, *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 178-186, 2013.

[9] H. J. Wei, W. F. Liu, X. X. Hu, Forward-Secure ciphertext-policy attribute-based encryption scheme, *Journal on Communications*, vol. 35, no. 7, pp, 38-45, 2014.

[10] C. Wang, Y. Li, J. Fang, et al. Cloud?aided scalable revocable identity?based encryption scheme with ciphertext update[J], *Concurrency & Computation Practice & Experience*, 29, 2017.

[11] YM Tseng, TT Tsai, SS Huang, et al, Identity-based encryption with cloud revocation authority and its applications[J], *IEEE Transactions on Cloud Computing*, pp. 99, 2017.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. J. Lou, Fuzzy key word search over encrypted data in cloud computing, *In: Proc. of the INFOCOM 2010. San Diego: IEEE Press*, pp. 441-445, 2010. [doi: 10.1109/INFCOM.2010.5462196]

[13] C. Wang, N. Cao, J. Li, K. Ren, W. J. Lou, Secure ranked keyword search over encrypted cloud data, *In: Proc. of the ICDCS 2010.Genova: IEEE Computer Society*, pp. 253-262, 2010.

[14] N. Cao N, C. Wang, M. Li, K. Ren, W. J. Lou, Privacy-Preserving multi-keyword ranked search over encrypted cloud data, *In: Proc. of the INFOCOM. Shanghai: IEEE Computer Society*, pp. 829-837, 2011.

[15] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, *In: Proc. of the EUROCRYPT 2004. Interlaken: Springer-Verlag*, pp. 506-522, 2004. [doi: 10.1007/978-3-540-24676-3_30]

[16] Y. R. Sun and W. M. Zheng, An identity-based ring signcryption scheme in ideal lattice, *Journal of Network Intelligence*, vol. 3, no. 3, pp. 152-161, August 2018.

[17] T. Y. Wu, CM Chen, K. H. Wang, J. S. Pan, W. M. Zheng, S. C. Chu, and J.F. Roddick, Security analysis of rhee et al.'s public encryption with keyword search schemes: a review, *Journal of Network Intelligence*, vol. 3, no. 1, pp. 16-25, Feb 2018.

[18] C. M. Chen, Y. Y. Huang, E. K. Wang, and TY Wu, Improvement of a mutual authentication protocol with anonymity for roaming service in wireless communicatins, *Data Science and Pattern Recognition*, vol. 2, no. 1, pp. 15-24, 2018.