Design and implementation of a cloud encryption transmission scheme supporting integrity verification

Zengyu Cai

School of Computer and Communication College,
Zhengzhou University of Light Industry,
Zhengzhou, 450002, China
Email: mailczy@163.com

Zuodong Wu

Software College,
Zhengzhou University of Light Industry,
Zhengzhou, 450002, China
Email: mailwzd@126.com

Jianwei Zhang*

Software College,
Zhengzhou University of Light Industry,
Zhengzhou, 450002, China
and
Henan Key Laboratory of Food Safety Data Intelligence,
Zhengzhou, 450002, China
Email: mailzjw@163.com
*Corresponding author

Wenqian Wang

School of Computer and Communication College,
Zhengzhou University of Light Industry,
Zhengzhou, 450002, China
Email: mailwwq@163.com

Abstract: In the cloud storage environment, the integrity of private data is one of the most concerned issues for users, which has become the focus of cloud storage research. For this kind of problem, the existing schemes usually sacrifice the communication efficiency of users for higher security, which often causes a lot of computing overhead. Therefore, the purpose of this paper is to achieve the coexistence of safety and efficiency, and adopts the ideas of Chinese commercial encryption algorithms SM2 and SM3, proposes a cloud encryption transmission scheme that supports integrity verification, and gives a security analysis under the assumption of discrete logarithm problem on elliptic curve and Diffie-Hellman problem. Finally, the actual test and comparative experiment results show that our scheme can realise the cloud data transmission encryption and cloud storage integrity verification functions at the same time without affecting the performance of the cloud server. Moreover, it can effectively resist all kinds of common attacks, reduce the storage and computing burden of cloud users, and has certain guiding significance for the research of user privacy protection in the cloud environment.

Keywords: cloud storage; Chinese commercial encryption algorithms; SM2; SM3; discrete logarithm; Diffie-Hellman; elliptic curve; cloud data transmission encryption; integrity verification; privacy protection.

Reference to this paper should be made as follows: Cai, Z., Wu, Z., Zhang, J. and Wang, W. (2021) 'Design and implementation of a cloud encryption transmission scheme supporting integrity verification', *Int. J. Embedded Systems*, Vol. 14, No. 3, pp.218–228.

Biographical notes: Zengyu Cai received his Master's in Computer Application Technology from the Northeast Normal University, Changchun, China, in 2006. He is currently an Associate Professor at the Zhengzhou University of Light Industry. His research interests include trusted computing, plan recognition and information security.

Zuodong Wu received his Bachelor's in Computer Science and Technology from the Henan Polytechnic University, Henan, China, in 2018. He is currently pursuing his Master's at the Zhengzhou University of Light Industry. His research interests include network security and cloud computing.

Jianwei Zhang received his PhD from the PLA Information Engineering University, Henan, China, in 2010. He is a Professor at the Zhengzhou University of Light Industry. His research interests include broadband information network and network security.

Wenqian Wang received her Bachelor's in Computer Science and Technology from the Henan University of Economics and Law, Henan, China, in 2018. She is currently pursuing her Master's at the Zhengzhou University of Light Industry. Her research interests include network security and artificial intelligence.

1 Introduction

With the continuous maturity of 5G technology and its in-depth application to the future market (Shahzadi et al., 2019), cloud computing services will grow rapidly in the next few years and are undergoing profound changes in the direction of intensification, scale and specialisation. According to the market share report of cloud computing issued by Gartner, an authoritative research institution in the USA, by 2020, the global cloud computing market has exceeded 260.2 billion dollars, a significant increase of 46.5% over last year (Mohammad et al., 2019). The application of cloud computing has penetrated traditional industries such as finance, industry, transportation, medical, and health care, providing enterprises and users with strong computing and storage capabilities (Azzedine and De Robson, 2018; Shynu and John, 2018).

However, cloud computing differs from traditional computing in that it mostly forces users to separate ownership and control of their private data (Liu et al., 2019; Mu et al., 2018). So most people have concerns about the security of private information stored in the public cloud, especially the integrity and privacy of cloud data. According to a survey by RedLock, 27% of users have encountered potential cloud storage security problems in the past year (Abubakr and Tian, 2019; Zhang and Mao, 2017), mainly due to the following aspects as shown in Figure 1:

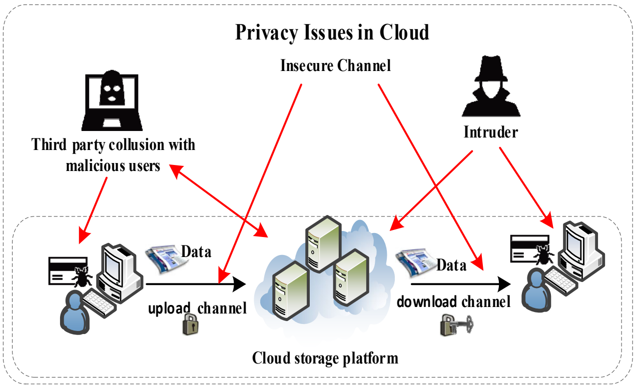
1 Sensitive data stored in the third party, lack of integrity verification and tamper-proof function, may be attacked by the server or untrusted third party to disclose the user's privacy.

2 When the private data is stored in the cloud, it is not encrypted, or the encryption technology is too simple, which makes the cipher-text easy to be cracked.

3 There is no access control mechanism in the cloud, and the attacker has unlimited access to resources.

It can be seen that cloud storage brings excellent flexibility and convenience to users, but without adequate security, its advantages and benefits will be meaningless. Therefore, many scholars begin to study the data security storage and transmission security in cloud computing (Xiao and Liu, 2019). Cui et al. (2018) proposed a cipher-text access control mechanism based on the CP-ABE algorithm. Under the premise that the service provider is not trusted, it can ensure the data security in the cloud storage system in the open environment, and reduce the complexity of permissions through attribute management. Jin et al. (2019) proposed a secure communication protocol in IoT convergence cloud environment. This protocol supports multi-user cipher-text computing and multi-user data sharing, supports fine-grained access control of cipher-text, and can resist collusion attacks. Moreover, under the random domain model of LWE, it can satisfy the IND-CPA security and is suitable for solving the privacy security problems in the cloud environment. Li and Pang (2016) proposed a cipher-text strategy based on CP-ABPRE that supports five features. In this project, the cloud proxy server can only use the heavy encryption key to encrypt the cipher-text specified by the user, which can resist the collusion attack between the user and the agent satisfying the massive encryption sharing strategy.

Figure 1 Security threats of data sharing in mobile internet environment (see online version for colours)



China is also continually releasing its commercial cryptography, and the research and development of commercial cryptographic products has become an important topic for many scholars. Therefore, Chen et al. (2018) abandoned foreign cryptography technology and developed a cloud storage sharing encryption system based on SM9, SM3 and SM4 algorithms, which effectively improves the security of cloud files. Hu et al. (2018) proposed a solution to record and store data using blockchain and combined with the SM3 algorithm to improve the efficiency of executing shared ledger. Zhang and Peng (2019) proposed a blind signature method based on the SM9 algorithm, which guarantees that the information transmitted does not get tampered and realises the hidden protection of shared data.

Given the existing cloud storage security researches, there are two main problems: one is that the traditional cloud storage protection mechanism lacks the integrity verification technology for cloud data, and the third party may tamper with the sensitive data of the user to cause privacy disclosure. The other is that there is not any secure and efficient encryption algorithm with Chinese standards applied to cloud storage system. Traditional cloud data encryption algorithms are often faced with key management problems and complex requirements such as the inability to modify in parallel and fine-grained authorisation.

Therefore, based on the idea of SM2 signature algorithm, this paper proposes a cloud encryption transmission scheme that supports integrity verification and designs and implements it. The scheme can not only effectively reduce the fraud of cloud data in the process of sharing transmission and third-party storage, but also ensure the integrity, confidentiality, effectiveness, and non-repudiation of cloud sensitive data. The primary function innovation of this work is as follows:

1 It is the first time to adopt the joint encryption method of Chinese commercial encryption standard SM2 signature algorithm and SM3 hash algorithm to solve the problem that the cloud server is uploaded Trojan script due to vulnerability, thus losing the control right of the server and tampering with the cloud sensitive data in batches. In this paper, the distribution of the randomly generated key pairs on the elliptic curve does not conform to the principle of statistics, which makes it difficult for the attacker to reconstruct and predict the parameters and initial values determined by the prime number field and binary expansion field, thus greatly enhancing the confidentiality of sensitive data. More importantly, through joint encryption, the protection of hash value generated by SM3 is enhanced, which disrupts the cloud data nonlinearly, and the attacker cannot deduce the keyword information.

2 Combined with the advantages of fast encryption speed of SM3 hash algorithm, short key length, and simple key management of SM2, the cloud server needs to run the encryption program to solve the problem of slow efficiency of uploading and downloading files, batch downloading data jam, and crash. It solves the problems of slow file upload and download efficiency, data jam and crash caused by running encryption program on cloud server. Considering the SM2 signature algorithm for direct data signature, the transmission efficiency will inevitably be affected. In this paper, the SM3 algorithm is used to hash the cloud data first, and then make the system sign the hash value directly, which dramatically enhances the efficiency of the system in verifying the integrity, and forms a series of features with strong flexibility and high reliability.

3 While implementing the above scheme, effectively control the cost of resources, reduce the degree of network congestion, reduce delay, and adapt to the privacy protection mechanism of cloud data.

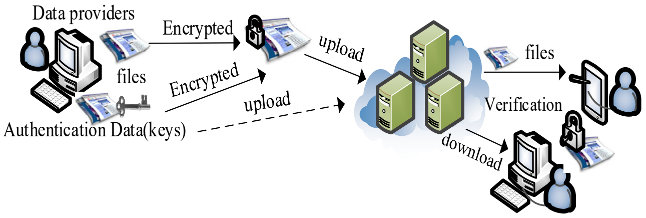
Organisation: the rest of the article is organised as follows. In Section 2, we provide some preliminary knowledge for readers to better understand the scheme. In Section 3, we introduce the design of the scheme in detail and analyse the security and efficiency of the scheme. In Section 4, we introduce the experimental process of the scheme in detail and prove the feasibility and advancement of the scheme according to the experimental results. Finally, Section 5 summarises this paper and elaborates on the future improvement direction.

2 Preliminaries

2.1 Chinese commercial cryptographic algorithms

In order to ensure the security of commercial cryptography, the Office of Security Commercial Code Administration (OSCCA) has formulated a series of cryptographic standards, including symmetric algorithm: SM1, SM4, SM7, ZUC, asymmetric algorithm: SM2, SM9, hash algorithm: SM3 (Cai, 2018). At present, these algorithms have been widely used in the internet of things (IoT), finance, and other related fields, completing functions such as identity authentication and data encryption and decryption.

Figure 2 Cloud data encryption transmission model (see online version for colours)



2.2 Cloud data encryption transmission

The cloud data encryption transmission mode is that the cloud service provider adopts the recognised encryption algorithm and authentication mechanism to ensure the uniqueness and integrity of the sensitive data of cloud users (Rajesh and Ashalatha, 2019; Xu et al., 2018). As shown in Figure 2, the primary function of the cloud data encryption transmission model is to protect the cloud data security sharing, and the cloud service provider and routing node are unable to obtain the decryption key at all. In this mode, the security of data storage, transmission, and sharing are improved, and the risk of data being monitored, tampered with, and leaked in network transmission are reduced (Su et al., 2017; Zhang and Mao, 2017).

2.3 SM3 hash algorithm

SM3 hash algorithm can compress messages of any length into fixed hash values (Wang and Yu, 2016). The algorithm uses different group operations, combined with the processing method of combining two words, to quickly spread and confuse the message in the local scope, and generate a hash value of 256-bit in the process of message pre-processing and compression to hash value, the algorithm is mainly composed of filling, iterative process, message expansion and compression function.

1 Data input: Accept data *w*, whose length is l bits and the maximum length is 264-bit.

2 Filling: First add bit '1' to the end of the message, add the *k*-bit data, the value is '0', where *k* is the minimum non-negative integer satisfying equation (1).

$$l + 1 + k \equiv 448 \bmod 512 \tag{1}$$

Finally, a 64-bit string is added to make the bit length of the filled message a multiple of 512.

For example: for data 011000010110001001100011, its length l is 24, is filled to get the bit string:

$$01100001\ 01100010\ 01100011\,1\overbrace{00\ldots00}^{423\,bit}\underbrace{\overbrace{00\ldots011000}^{64\,bit}}_{l}$$

3 Iteration and expansion: The filled data m' is grouped every 512-bit to get $m' = B^{(0)}\,B^{(1)}\ldots B^{(n-1)}$, where $n = (1 + K + 65) / 512$. Iterate according to the following equation (2), the purpose of iteration is to break up the data bits nonlinearly, so that each data bit can participate in diffusion and confusion quickly.

$$\begin{aligned}&\text{FOR } i = 0 \text{ TO } n-1\\ &V^{(i+1)} = CF\left(V^{(i)}, B^{(i)}\right)\end{aligned} \tag{2}$$

where *CF* is the compression function, $V^{(0)}$ is the initial value of 256-bit, $B^{(i)}$ is the message grouping after filling, and the result of iterative compression is $V^{(n)}$.

Each message group $B^{(i)}$ is expanded into 132 message words $W_0, W_1, \ldots, W_{67}, W_0', W_1', \ldots, W_{63}'$ for compressing function CF as follows: first, the group $B^{(i)}$ is divided into 16 words $W_0, W_1, \ldots, W_{15}$, where W_0 is the initial fixed value. Then the following equation (3) is extended, where P_1 is the permutation function:

$$\begin{aligned}&W_j \leftarrow P_1\left(W_{j-16} \oplus W_{j-9} \oplus \left(W_{j-3} \lll 15\right)\right.\\ &\left.\oplus\left(W_{j-13} \lll 7\right)\right) \oplus W_{j-6}\\ &W_i' = W_i \oplus W_{i+4}\end{aligned} \tag{3}$$

4 Compress: Let *A*/*B*/*C*/*D*/*E*/*G*/*H* be the word register, *SS*1/*SS*2/*TT*1/*TT*2 be the intermediate variable, calculated according to equation (4), the output is a 256-bit hash value *y* = *ABCDEFGH*.

$$V^{(i+1)} = CF\left(V^{(i)}, B^{(i)}\right), \quad 0 \le i \le n-1 \tag{4}$$

2.4 SM2 signature algorithm

SM2 signature algorithm is a public key cryptography algorithm designed by China (Liu et al., 2013). Compared with the RSA algorithm, which has similar functions to SM2, the SM2 algorithm is based on the discrete logarithm problem of point group on an elliptic curve, and the strength of encryption is higher than that of RSA. Our scheme is mainly implemented by the SM2 signature algorithm. The specific algorithm will be introduced in the third part in detail, which is mainly used to realise the anti-tampering and error detection functions (Zhang et al., 2017).

2.5 Hardness assumption

2.5.1 ECDLP assumption

Let *P* and *Q* be two points on the elliptic curve, both of which have an order of *n*, and *n* is a large prime, where *Q* = *k* × *P*, and $k \le n$. Given *P*, *Q* to get *k* is computationally infeasible.

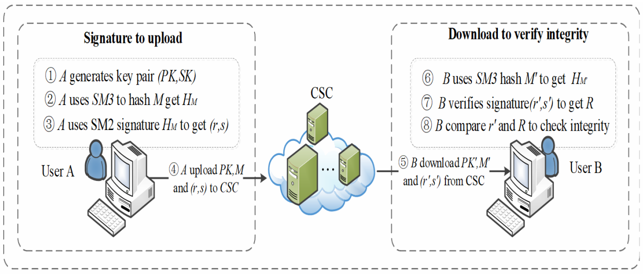
2.5.2 ECDHP assumption

G is the base point of order n on the elliptic curve, where *P* = *c* × *G*, and *Q* = *d* × *G*. Given points *P* and *Q*, it is computationally infeasible to get the point *k*.

3 The design of our scheme

The specific scheme of this paper is mainly operated in the process of data upload and download, which mainly consists of three parts: user *A* (upload and share data user), user *B* (download and consumer data user), and cloud storage centre (*CSC*). When user *A* wants to upload data *M*, the key derivation function of SM2 first generates a public-private key pair (*PK*, *SK*) for user *A* locally. User *A* uses the *SM*3 algorithm to generate a hash value H_M for the data *M*. Then, user *A* uses the received private key (*SK*) to sign the hash value H_M. The signature process uses the *SM*2 signature algorithm. Finally, user *A* uploads the data *M*, *PK*, and the signature result (*r*, *s*) to CSC, and the upload process is completed. When user *B* wants to download the shared data M' uploaded by user *A*, our scheme will uses SM3 and SM2 signature verification algorithms to jointly verify the integrity of data M', where the SM2 signature verification algorithm uses the public key (*PK*) and signed results (r', s'). If data M' has tampered, our scheme will actively remind user *B* that data tampered. User *B* can also remind user *A* to timely backup and update the data. The flow chart of cloud data encryption transmission scheme supporting integrity is shown in Figure 3.

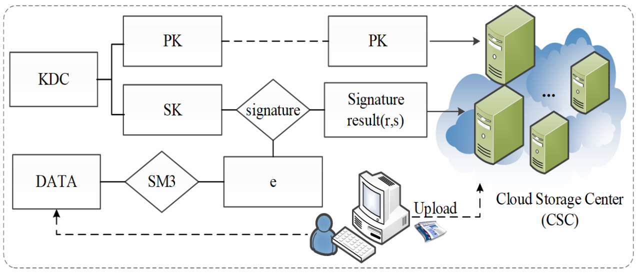
Figure 3 Cloud data encryption transmission scheme that supports integrity (see online version for colours)


Signature to upload
① A generates key pair (PK,SK)
② A uses SM3 to hash M get H_M
③ A uses SM2 signature H_M to get (r,s)
User A
④ A upload PK,M and (r,s) to CSC
CSC
Download to verify integrity
⑥ B uses SM3 hash M′ to get $H_{M'}$
⑦ B verifies signature(r′,s′) to get R
⑧ B compare r′ and R to check integrity
⑤ B download PK′,M′ and (r′,s′) from CSC
User B

3.1 Data upload process

In the process of data uploading, our scheme mainly consists of two parts, namely user *A* and *CSC*. The specific process is shown in Figure 4.

Figure 4 Flow chart of data upload (see online version for colours)



Let the data to be signed be *M*. In order to obtain the digital signature (*r*, *s*) of the data *M*, key derivation function of SM2 generates a key pair (*PK*, *SK*) for user *A*, where the SK is an integer generated by a random number generator. The specific operation steps are as follows:

Data upload process: generate digital signature

Input: upload data *M*

Out: signature (*r*, *s*)

A1: Calculation $\bar{M} = Z_A \,\|\, M$, Z_A is the hash value of user A's identity;

A2: Calculation public key PK = *SKG*, *G* is the base point of the elliptic curve;

A3: Calculation, plain-text and identity are converted into 256-bit hash value;

A4: Generate random numbers $k \in [1, n - 1]$

A5: Calculate elliptic curve points: (x1, y1) = [k]G, G is the base point of the elliptic curve;

A6: Calculate $r = k(e + x_1) \bmod n$, if $r = 0$ or $r + k = n$, return A3;

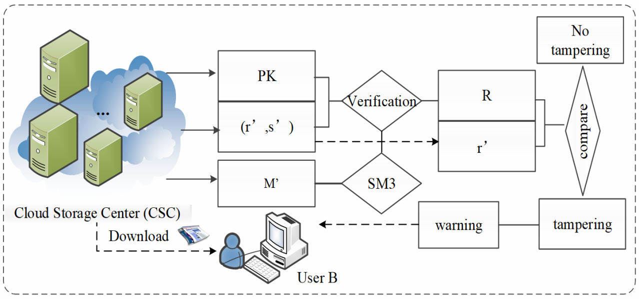
A7: Calculate $s = ((1 + SK)^{-1} \times (k - r \cdot SK)) \bmod n$, if s = 0, return A3;

A8: M, (r, s), PK upload and storage to cloud storage centre.

3.2 Data download process

In the process of data download, our scheme mainly consists of two parts, namely user *B* and *CSC*. The specific process is shown in Figure 5.

Figure 5 Flowchart of data download (see online version for colours)



In order to verify the integrity of the data *M′* to be downloaded, user *B* as the verifier first uses ZA to verify the identity of user *A* who has been negotiated, and then uses the signature result (*r′*, *s′*) and *PK* to verify whether the data has been tampered with. The specific operation steps are as follows:

Data download process: digital signature verification

Input: (*r′*, *s′*), PK, *M′*

Out: Integrity verification results

B1: Check if $r' \in [1, n - 1]$ is true, if it is not true, the verification fails;

B2: Check if $s' \in [1, n - 1]$ is true, if it is not true, the verification fails;

B3: Calculation $\bar{M}' = Z_A \,\|\, M'$, Z_A is the hash value of user A's identity;

B4: Generate random numbers $k \in [1, n - 1]$

B5: Calculation $e' = SM3(\bar{M}')$, *M′* and ID_A are converted into 256-bit hash value;

B6: Calculate $t = (r' + s') \bmod n$, if t = 0, the verification fails.

B7: Calculate elliptic curve points $(x_1', y_1') = [s']G + [t]PK$;

B8: Calculate $R = (e' + x_1') \bmod n$ and check whether $R = r'$ is true. If it is true, the verification is passed. Otherwise, the verification is not passed.

3.3 Scheme analysis

3.3.1 Security analysis

This paper mainly uses the assumption of discrete logarithm problem and Diffie-Hellman problem on elliptic curve (ECDLP, ECDHP) to give the security analysis of our scheme, thus proving that our scheme not only satisfies the confidentiality, authentication, unforgeability, integrity and non-repudiation, but also has forward security.

3.3.1.1 Confidentiality

In our scheme, the adversary must master k and SK to break the plain-text, but k and SK are random large prime numbers, which are difficult to speculate and obtain. Although attackers can get k and SK in many ways, these methods are based on solving the problems of ECDLP and ECDHP. Moreover, it can be seen from the preliminaries that these two problems are not feasible in calculation. This paper discusses the following situations:

- If the adversary wants to calculate the points (x_1, y_1) on the elliptic curve, he must get the secret random number k from equation (1) $(x_1, y_1) = k[G]$. However, if the adversary knows the PK and G, to get k is equivalent to solving the ECDLP problem.
- If the adversary can easily get r and PK. Suppose that the adversary wants to verify s according to equation (2) $s = ((1 + SK)^{-1} \cdot (k - r \cdot SK)) \bmod n$. If equation (3) $r = k\,(e + x_1) \bmod n$ and equation (4) $PA = SK \cdot G$ are known, it is equivalent to solving the ECDHP problem.
- If an adversary wants to get the private key SK from the equation (4), this is equivalent to solve the ECDLP problem. So it is tough for the adversary to forge s from equation (2), because he cannot get SK.
- If the adversary wants to verify x_1 according to equation (3), he must get the correct e. Since the SM3 hash algorithm in this paper is collision-free, it is impossible to get the correct x_1 without knowing the parameters and M.

3.3.1.2 Authentication

In our scheme, the user B can use the hash value of user A's identity to verify the validity of user A. When user B performs signature verification, it can verify the correctness of user A's information according to equation (5) $Z_A = SM3(ENTL_A \| ID_A)$. If equation (5) is true, user B can determine whether the identity has been tampered with in the transmission process, and the anti-repudiation is stronger.

3.3.1.3 Integrity

In our scheme, the user B can verify whether the data is the original data sent by the user A. In step 8 of the data upload process, the user A calculates the r and the CSC sends it to the user B. In equation (3), e is the output value of $SM3$ hash algorithm acting on M. If the adversary tampers the M stored in the cloud into M', then $e' = SM3(\overline{M}')$, due to the unidirectionality of the hash function, we can know that $R \neq r$, and the user B can directly verify that the data M has been modified during signature verification.

3.3.1.4 Unforgeability

In our scheme, the adversary cannot forge a valid signature without the user A's private key (Xu et al., 2018). If the adversary overhears that the signature (r, s) wants to forge the signature as (r', s'), then (r', s') must satisfy the equation $t = (r', s') \bmod n$. The adversary must calculate r' and s' by equations (3) and (2). However, neither the adversary nor the receiver can get the correct random prime number k and SK, so the corrects s' and r' cannot be generated. If the adversary or user B wants to get the SK and k from equations (3) and (4), he must solve the ECDLP problem, which is not feasible in calculation. Therefore, the scheme cannot be forged.

3.3.1.5 Forward security

Forward security means that if the user A's private key is disclosed, the signature information produced by it before this is safe and reliable (Michel et al., 2018). Even if the attacker has the private key, he cannot forge the signature. In this scenario, if the adversary wants to forge the signature successfully, he must construct the signature through the SK and k. However, without M, the adversary cannot find e through the equation $e = SM3(\overline{M})$. If the adversary wants to get the keys SK and k from PK and r, then he must solve the ECDLP problem. No one who obtains the user A's private key SK and the signature result (r, s) can use the equation (2) to verify s. In other words, the attacker cannot verify the integrity of the scheme. Therefore, in our scheme, even if the user A's private key is lost, the signature sent by the user A before is still safe and reliable, providing forward security.

3.3.2 Key size analysis

In the SM2 signature algorithm based on the elliptic curve cryptosystem, after determining the parameters of the elliptic curve system, the key pair generated by the base point cannot be predicted and can perform finite field operation, which can well hide the plain-text information (Ruma et al., 2017). Table 1 compares and analyses the key sizes of the RSA and ECC signature algorithms in detail with our scheme. It is easy to see that the key size of this paper has obvious advantages under the same level of confidentiality, and can effectively avoid exhaustive attacks. The attacker could not obtain any critical information from the signature results. In particular, the SM3 hash algorithm is used to pre-process the signature information, which further improves the encryption performance of the key. Any slight change of the signature information will spread to the full transmission process, resulting in the change of the whole hash value. Its good scalability can lead to the 'avalanche effect' (Cassio et al., 2017), thus enhancing the security of the scheme.

3.3.3 Calculate cost analysis

The calculation cost of our scheme is mainly reflected in the addition, multiplication, and inversion operations on the elliptic curve, which are respectively represented by T_{eadd}, T_{emul}, and T_{inv}. In the RSA algorithm, T_{add}, T_{mul}, T_{mod}, T_{exp} are used to represent the time required to perform an add, multiply, complementation, and power multiply operation, while T_{hash} represents the time required to perform hash operation in the scheme. *ES* represents the establishment of additional parameters, and *EX* represents the cost of calculating the key.

Table 1 Key size comparison of the same security level

Security level	*RSA*	*ECC*	*Ours*
10^4	512-bit	106-bit	98-bit
10^8	768-bit	132-bit	124-bit
10^{11}	1,024-bit	160-bit	147-bit
10^{78}	2,048-bit	210-bit	206-bit

As shown in Table 2, since the time required to run the exponential operation is about twice as long as the scalar multiplication operation on the elliptic curve (Yang et al., 2019; Zhang and Wang, 2018), and the addition operation on the elliptic curve is also relatively efficient (Wang and Zhang, 2016), this paper improves the efficiency compared with the RSA signature algorithm. Compared with the ECC signature algorithm, this paper adopts an SM3 pre-processing idea for signature message, which reduces the number of hash operations in integrity verification and improves the upload efficiency of big data.

Table 2 Calculation complexity analysis

Scheme	*RSA*	*ECC*	*Ours*
ES	5	7	8
EX	$3\ T_{mul} + T_{\mathrm{mod}}$	T_{mul}	T_{mul}
Signature	$3\ T_{mul} + T_{\mathrm{exp}} + 2\ T_{\mathrm{mod}}$	$3\ T_{emul} + T_{eadd} + T_{hash} + T_{\mathrm{mod}} + T_{inv}$	$T_{emul} + 2\ T_{eadd} + 2\ T_{hash} + 2\ T_{\mathrm{mod}} + T_{inv}$
Verification	$3\ T_{mul} + T_{\mathrm{exp}} + 2\ T_{\mathrm{mod}}$	$3\ T_{emul} + T_{eadd} + T_{hash} + T_{\mathrm{mod}} + T_{inv}$	$T_{emul} + 3\ T_{eadd} + T_{hash} + 2\ T_{\mathrm{mod}}$

3.3.4 Efficiency analysis

Our scheme mainly enjoys the advantages of the SM3 algorithm's fast encryption speed and SM2 signature algorithm's short key length and simple key management to improve the efficiency of verification integrity. Due to the fast generation of hash values by the SM3 algorithm, it avoids the less efficient implementation architecture and calculation methods that may be used in common solutions, and greatly improves the signature efficiency of the SM2 signature algorithm. More importantly, through joint encryption processing, it is found that the encrypted data generated by key derivation function is only related to the hash value length generated by hash algorithm, so when dealing with large data, the algorithm can directly encrypt hash without grouping, which improves the encryption rate of the scheme. At the same time, when encrypting the data with uncertain length, the scheme does not involve the operation of complement, which reduces the complexity. Table 3 compares the RSA and ECC signature algorithms with our scheme in detail. As shown above, our scheme has obvious advantages in signature efficiency.

Table 3 Efficiency comparison of RSA, ECC and our scheme

Scheme	*RSA*	*ECC*	*Ours*
Generate key	100 ms	48 ms	59 ms
Signature	896 ms	528 ms	600 ms
Upload	156 ms	78 ms	60 ms
Verification	766 ms	560 ms	400 ms
Download	1,456 ms	889 ms	946 ms

4 Scheme realisation and test result analysis

4.1 Scheme realisation

The scheme runs on a Windows 10 operating system configured with a 3.40 GHz Intel i7 processor and 8 GB of running memory. The system is implemented by Java programming in the Android Studio programming environment. Cloud storage platform is built with the cooperation of Hadoop and Alibaba cloud. The implementation of the system is: when user uploads data, first use the SM3 algorithm to hash the signature information to generate a hash value, and then use the private key generated by the SM2 signature algorithm to sign the hash value. When the user uploads the signature information to the Alibaba cloud, the corresponding public key and signature result will be placed in the Alibaba cloud background. Legitimate user downloads the signature information, and the background verifies the integrity of the data by comparing the hash value R obtained by the SM2 signature verification algorithm with the hash value r of the signature result. If the signature information has been tampered or the transmission is incomplete, the system will notify the user and recover the source data in time.

We select a set of data, such as "But a man is not made for defeat. A man can be destroyed but not defeated." Then take this as an example to explain the operation process of the system and demonstrate the feasibility of the system.

According to the system background parameters in Table 4, it can be seen that when the key derivation function of SM2 determines the parameters such as the elliptic curve and the base point, the generated key pair is random and irregular.

According to the background parameters in Table 5, it can be seen that the data transmission or storage process has been tampered with, such as M', the parameters R and r' of the signature verification will be completely different. Therefore, when downloading, the system only needs to compare the two parameters of R and r' to determine the integrity of the data M on the cloud storage. Moreover, we can see that because the data M has been hashed in advance, the attacker changes M slightly, and the two parameters will also have a huge difference.

Table 4 Experimental parameters generated by the system during upload

Name	*Parameters*
M	But a man is not made for defeat. A man can be destroyed but not defeated
e	BF88E785E883212690D0B7CADFA65EECD10413671E92B1E97AEA2C8DA79611BB
PK	(8DF211C1930BDF8844A65D891FA922,B7865BE94664D92E5ED70A855FE585D)
SK	66265DF192886518D95A6E32DE51169B830FB415AFE94C8E6F44EDA7EDD2CF98
r	B8FA332389474EF2A0BA19197F2693A9EF07F5E6641B12C3481C00555B87D51B
s	287C3500AB64F49CA02CFC71CCD7F5A9C38CC53B013A1BBE622BCE07732D9EAB
Z_A	F4A38489E32B45B6F871146FBFB7BC9A

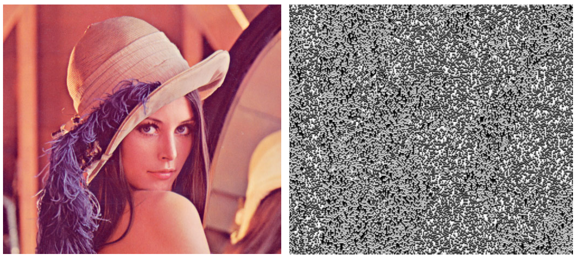
Table 5 Experimental parameters generated by the system during download

Name	*Parameters*
M′	0_But a man is not made for defeat. A man can be destroyed but not defeated.
e′	0384288B9E890161826385F572B8405B9621C74BED4D1C16B605F7C3D72378D9
r′	0C7314674460A5EEC88B8A413C08BA79C4AEDFD473C4EDF1F2FB3BE592F2E4E4
s′	4E32122384C5E2E72EFEFFBEBB408C09F0130AC5FA971483AC3B2DE411B5B356
R	C4C44C567182A01E8080885747D13131FAB14531F1E0621FDCCCA2CB50BA3E37

4.2 *Test result analysis*

In the security test, we perform correlation analysis and histogram analysis on the encrypted (signature) results to prove whether our scheme can resist common attacks, such as batch template attacks, features attacks, and statistical attacks. The standard test picture 'Lena' (picture size 463 kb) is used as the experimental reference document. In efficiency test, we selected data of different sizes as reference samples and selected RSA, ECC and Zhang signature algorithms for comparison with our scheme, reflecting the advantages and disadvantages of our scheme.

Figure 6 Image and signature results are visually compared, (a) the original image (b) signed (encrypted) image (see online version for colours)



(a) (b)

4.2.1 *Security test and analysis*

In this simulation experiment, we mainly use the picture 'Lena' as our analysis index to evaluate the privacy protection effect of our system. As shown in Figure 6, compared with the signature result, the image 'Lena' has completely lost the original characteristics of the data after a variety of operation theories on the quasi elliptic curve. The discrete data does not have any rules and does not have any practical application. This is due to the existence of a random number *k* in the SM2 signature algorithm. Even if the same plain-text is signed, it will cause different multiple point operations each time, and it can perfectly and effectively avoid batch template attacks.

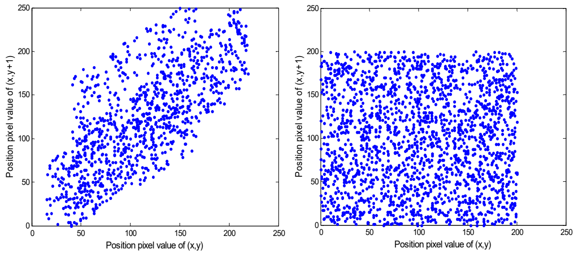
4.2.1.1 *Correlation analysis*

Image data generally has a very high data redundancy for all pixels, and the correlation between adjacent pixels is also very strong (Manjit and Vijay, 2018). Attackers can use it to analyse to construct attack information, which is called adjacent pixel correlation analysis (Haidar, 2019). In the test, 1,500 pairs of pixels were randomly selected from the plain-text image before the signature and the cipher-text image after the signature, and formula 5 was used to analyse the correlation between the signature information and the adjacent pixels of the signature image in the horizontal direction, as shown in the correlation Figure 7.

$$C = \frac{\left(N\sum_{j=1}^{N} x_j y_j - \sum_{j=1}^{N} x_j \sum_{j=1}^{N} y_j\right)}{\begin{pmatrix} N\sum_{j=1}^{N}\left(x_j\right)^2 \\ -\left(\sum_{j=1}^{N} x_j\right)^2 \end{pmatrix}\begin{pmatrix} N\sum_{j=1}^{N}\left(y_j\right)^2 \\ -\left(\sum_{j=1}^{N} y_j\right)^2 \end{pmatrix}} \tag{5}$$

x_j and y_j are two adjacent pixel points in the original image, and N is the total number of pixel points selected in the original image. It can be seen from Figure 7(a) that the original image is represented as a concentrated linear distribution, indicating that adjacent pixels are highly correlated. The encrypted image after signature is shown in Figure 7(b), and the signature algorithm covers all the features of the original image, showing excellent uniform distribution characteristics, which can resist the features attack.

Figure 7 Correlation analysis of image and signature image, (a) original image (b) signed (encrypted) image (see online version for colours)



(a) (b)

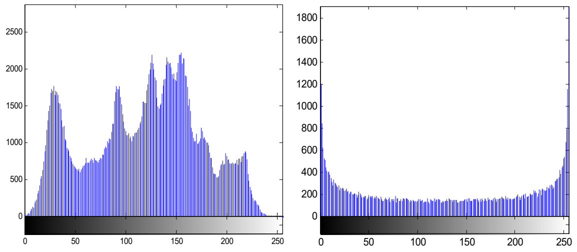
4.2.1.2 Histogram analysis

The ability of an image encryption algorithm to resist a uniform attack can be reflected by histogram (Zhao et al., 2017). The more advanced the encryption algorithm, the more uniform the histogram distribution of the encrypted image, and the more influential the ability to resist statistical attacks. Histogram variance is calculated as follows:

$$\text{var} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}\left(x_i - x_j\right)^2 \tag{6}$$

where x_i, x_j are the numbers of pixel values. Figure 8 shows the histogram of the image before and after the signature of our method. By comparing the histogram before and after signature, it is found that the histogram [Figure 8(a)] of the original image has more obvious statistical characteristics and is more sensitive to statistical attacks; while the histogram [Figure 8(b)] of the signed image tends to be more uniform and has stronger ability of defence statistical attacks.

Figure 8 Shows the histogram of the image before and after the signature of our scheme, (a) original image histogram (b) signed image histogram (see online version for colours)



(a) (b)

4.2.2 Efficiency test and analysis

In the efficiency test, first, select signature data of different sizes for integrity verification and compare and analyse with Zhang, RSA, and ECC signature algorithms. The results verify that our method has efficiency advantages. Second, in the case where the signature data is 256 kb, test the implementation efficiency of the above ways, and analyse the usability of our scheme. The specific experimental results are shown in Figures 9 and 10.

Figure 9 Data integrity verification efficiency test of different sizes (see online version for colours)

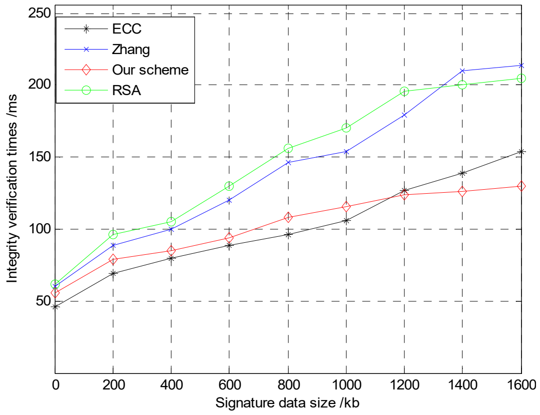
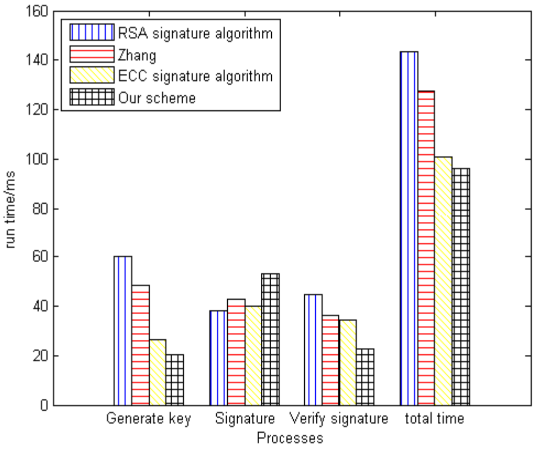


Figure 10 Compare the implementation efficiency of the scheme (see online version for colours)



As can be seen from Figure 9, this system is faster than Zhang and RSA signature algorithms in the integrity verification process, but slower than the ECC signature algorithm in the integrity verification of small data volume. This is because in order to enhance the transmission efficiency of signature data and reduce the problem of batch download data jam and crash, the SM3 signature algorithm is adopted to hash the signature data, which increases the time complexity of our scheme. However, for big data messages, this speeds up the signing and verification process. Although Zhang also applies the Chinese commercial signature algorithm, it is mainly an identity-based signature algorithm, which increases the time cost of authentication. Compared with other schemes, the RSA signature algorithm has a considerable key length and low efficiency. Therefore, this paper sacrifices time complexity for higher system stability.

Besides, in terms of test execution efficiency, the implementation efficiency of our scheme is equivalent to that of the ECC, but the inversion operation $(1 + SK)^{-1}$ mod n in our scheme can be pre-computed. Still, the inversion operation of ECC cannot, so the efficiency of signature algorithm in this scheme is higher than that of the ECC. In the process of comparing with the RSA efficiency, we find that the efficiency of our scheme is better than the length of the key, saving the transmission bandwidth, the length of the public key and the length of the signature are better than the RSA, so although the RSA verification signature speed is similar to our scheme, the total time is more than our scheme. Because Zhang dynamically signs based on the identities of the communicating parties, he sacrifices time complexity for higher encryption security, resulting in low efficiency. The specific experimental results are shown in Figure 10.

5 Conclusions

The security of cloud storage affects the development of cloud computing applications. A reasonable and effective privacy protection method can not only improve the trust of cloud storage service to service users, but also consider the performance cost of cloud storage systems. Therefore, this paper introduces the idea of a commercial encryption algorithm of China. Under the condition of not affecting the performance of the cloud server, we use the joint form of SM3 and SM2 signature algorithms to conduct data integrity verification, and also to realise the secure sharing of cloud storage data. In order to demonstrate the advanced nature of our scheme, we not only verify the confidentiality, authentication, integrity, and unforgeability of our scheme, but also provide forward security. Moreover, it has sufficient advantages in computing cost and communication cost. Finally, we simulated the project on Alibaba cloud and proved that the project can resist various attacks and efficient execution capabilities, which is very suitable for the privacy protection of cloud storage data. The next research focus is to apply the scheme to different cloud types (public cloud, private cloud, hybrid cloud) and different scenarios (such as: internet of vehicles, intelligent medical care), and optimise and improve the scheme according to the actual experimental results. Furthermore, we should optimise the verification efficiency, save the cost, expand the local storage resources, and design a more efficient real-time dynamic integrity verification scheme to enhance the user's experience.

Acknowledgements

This research was financially support by the following fund: The National Natural Science Foundation of China (Grant No. 61672471), Henan Province University Science and Technology Innovation Team (Grant No. 18IRTSTHN012), Scientific and Technological Projects of Henan Province (Grant No. 202102210176), LiaoNing Provincial Department of Education Project (Grant No. LQN201912).

References

Abubakr, O.A. and Tian, L. (2019) 'TTLoC: taming tail latency for erasure-coded cloud storage systems', *IEEE Transactions on Network and Service Management*, Vol. 16, No. 4, p.1.

Azzedine, B. and De Robson, E.G. (2018) 'Vehicular cloud computing: architectures, applications and mobility', *Computer Networks*, Vol. 1, No. 4, pp.171–189.

Cai, C.H. (2018) 'The design, implementation and application of OpenSSL with Chinese cryptographic algorithms', *Journal of Information Security Research*, Vol. 6, No. 3, pp.21–27.

Cassio, M.O., Fernando, P.M. and Roney, L.T. (2017) 'The 'avalanche effect' of an elasto-viscoplastic thixotropic material on an inclined plane', *Journal of Non-Newtonian Fluid Mechanics*, Vol. 247, No. 7, pp.165–177.

Chen, Z., Qi, F. and Ye, C.Y. (2018) 'Research on cloud data encryption scheme based on Chinese cryptographic algorithms', *Journal of Information Security Research*, Vol. 4, No. 7, pp.646–650.

Cui, H., Deng, R.H., Lai, J., Yi, X. and Nepal, S. (2018) 'An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited', *Computer Networks*, Vol. 133, No. 14, pp.157–165.

Haidar, R.S. (2019) 'An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling', *Multimedia Tools and Applications*, Vol. 78, No. 18, pp.26073–26087.

Hu, W., Wu, Q.H. and Ye, C.M. (2018) 'Design of mobile security eID and authentication protocol based on Chinese encryption algorithm and blockchain', *Information Network Security*, Vol. 7, No. 211, pp.13–21.

Jin, B.W., Park, J.O. and Mun, H.J. (2019) 'A design of secure communication protocol using RLWE-based homomorphic encryption in IoT convergence cloud environment', *Wireless Personal Communications*, Vol. 105, No. 2, pp.599–618.

Li, H.X. and Pang, L.J. (2016) 'Efficient and adaptively secure attribute-based proxy reencryption scheme', *International Journal of Distributed Sensor Networks*, Vol. 12, No. 5, pp.1–12.

Liu, D.B., Peng, X. and Li, Y.J. (2019) 'Saving energy consumption for mixed workloads in cloud platforms', *International Journal of Computational Science and Engineering*, Vol. 20, No. 3, pp.376–386.

Liu, M.J., Chen, J.Z. and Li, H.X. (2013) 'Partially known nonces and fault injection attacks on SM2 signature algorithm', *International Conference on Information Security and Cryptology*, pp.343–358.

Manjit, K. and Vijay, K. (2018) 'An efficient image encryption method based on improved Lorenz chaotic system', *Electronics Letters*, Vol. 54, No. 9, pp.562–564.

Michel, A., Fabrice B. and David, P. (2018) 'On the tightness of forward-secure signature reductions', *Journal of Cryptology*, Vol. 8, No. 32, pp.1–67.

Mohammad, U.B., Shallal, Q. and Tamandani, Y.K. (2019) 'Cloud computing service models: a comparative study', *International Conference on Computing for Sustainable Global Development (INDIACom)*, pp.890–895.

Mu, Y., Hu, Y.P. and Zhang, L.Y. (2018) 'Fully secure hierarchical inner product encryption for privacy preserving keyword searching in cloud', *International Journal of High Performance Computing & Networking*, Vol. 11, No. 1, pp.45–55.

Rajesh, K.R. and Ashalatha, N. (2019) 'Data residency as a service: a secure mechanism for storing data in the cloud', *International Journal of Embedded Systems*, Vol. 11, No. 4, pp.397–418.

Ruma, K., Ajeena, K. and Sarah, J.Y. (2017) 'The integer sub-decomposition method to improve the elliptic elgamal digital signature algorithm', *International Conference on Current Research in Computer Science and Information Technology*, pp.14–20.

Shahzadi, R., Niaz, A. and Ali, M. (2019) 'Three tier fog networks: enabling IoT/5G for latency sensitive applications', *China Communications*, Vol. 16, No. 3, pp.1–11.

Shynu, P.G. and John, S.K. (2018) 'Privacy preserving secret key extraction protocol for multi-authority attribute-based encryption techniques in cloud computing', *International Journal of Embedded Systems*, Vol. 10, No. 4, pp.287–300.

Su, C.C., Wang, Y.Z. and Shen, Y.L. (2017) 'Improving database storage usability with the cloud-based architecture', *IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp.494–499.

Wang, X.Y. and Yu, H.B. (2016) 'SM3 cryptographic hash algorithm', *Journal of Information Security Research*, Vol. 78, No. 6, pp.18–30.

Wang, Z.H. and Zhang Z.F. (2016) 'Overview on public key cryptographic algorithm SM2 based on elliptic curves', *Journal of Information Security Research*, Vol. 11, No. 2, pp.972–982.

Xiao, P. and Liu, C.S. (2019) 'A novel virtual disk bandwidth allocation framework for data-intensive applications in cloud environments', *International Journal of Computational Science and Engineering*, Vol. 20, No. 1, pp.1156–1162.

Xu, G.P., Mao, Q.F., Li, H. and Li, S.L. (2018) 'Towards optimisation of replicated erasure codes for efficient cooperative repair in cloud storage systems', *International Journal of Computational Science and Engineering*, Vol. 16, No. 2, pp.108–116.

Yang, H.Y., Ning, Y.G. and Wang, Y. (2019) 'Research on RSA and hill hybrid encryption algorithm', *International Journal of Computational Science and Engineering*, Vol. 20, No. 1, pp.976–982.

Zhang, J.H. and Mao, J. (2017a) 'Anonymous multi-receiver broadcast encryption scheme with strong security', *International Journal of Embedded Systems*, Vol. 9, No. 2, pp.177–187.

Zhang, K.Y., Xu, S. and Gu, D.W. (2017b) 'Practical partial-nonce-exposure attack on ECC algorithm', *13th International Conference on Computational Intelligence and Security (CIS)*, IEEE, pp.248–252.

Zhang, X. and Wang, X. (2018) 'Digital image encryption algorithm based on elliptic curve public cryptosystem', *IEEE Access*, Vol. 6, No. 9, pp.70025–70034.

Zhang, X.F. and Peng, H. (2019) 'A blind signature scheme based on SM9 algorithm', *Information Network Security*, Vol. 8, No. 4, pp.61–67.

Zhao, Y.L., Yuan, Q.D. and Meng, X.P. (2017) 'Double-image encryption algorithm based on discrete fractionalrandom transform and weighted histogram cross permutation', *Journal of Applied Optics*, Vol. 38, No. 6, pp.937–946.