SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

Research on computer network data security storage technology in the era of big data

Liying Zhang[1], Xin Gu[1], Qiang Zhao[2]

1 Information Network Center,LuXun Academy of Fine Arts,Shenyang 110004,Liaoning, China

2 Infrastructure Office,ShengJing Hospital of China Medical University, Shenyang 110004,Liaoning, China

Abstract

In the burgeoning epoch of big data, the imperative for secure computer network data storage is confronted with formidable challenges, including the perils of data breaches and a paucity of robust security measures. An enhanced storage paradigm, predicated upon a refined Hash algorithm—termed H-AONT—is herein delineated. This methodology augments data storage security through the formulation of an apposite system model, the amalgamation of the merits inherent in conventional encryption algorithms, and the deployment of the H-AONT dual encryption algorithm in data processing. Empirical evidence substantiates that, vis-à-vis alternative approaches, the proposed method significantly bolsters data storage security, furnishes an elevated stratum of protection for computer network data repositories, ensures data storage reliability, and proffers a pertinent benchmark for the safeguarding of computer network data within the vast landscape of big data.

OPEN ACCESS

Published: 17/05/2024

Accepted: 06/05/2024

Submitted: 24/04/2024

DOI: 10.23967/j.rimni.2024.05.001

Keywords:
big data
cloud storage
secure data storage
H-AONT

1. Introduction

The swift advancement of information technology has ostensibly augmented the convenience of daily life. Concomitant with this advancement, however, is an escalated jeopardy of privacy data breaches. In the big data milieu, the voluminous repositories of data and information are besieged by grave threats; notably, the data storage process is beleaguered by security vulnerabilities, and the precision of data information screening is markedly deficient. Consequently, the exploration of computer network data security storage technology emerges as a pivotal endeavor for amplifying the caliber of data security storage. Traditional computer encryption, predominantly plaintext, is readily intercepted by malefactors, rendering the data storage security tenuous and the regional data transmission security feeble [1]. To ameliorate data storage security and guarantee storage accuracy, a novel schema predicated on attribute encryption has been posited. This schema encompasses a single server model outsourcing mode index calculation algorithm, thereby relegating the data encryption task to a cloud service provider and enabling the corroboration of encryption computation outcomes [2]. Furthermore, research has been conducted on a revocable and traceable KP-ABE scheme, predicated on cloud storage technology, which not only ensures attribute revocability but also facilitates the real-time traceability of user identities [3]. Additionally, a hierarchical remote data retrieval scheme has been scrutinized, which, to some extent, enhances the reliability of data storage. However, this scheme's authentication subject is predominantly user-centric, engendering substantial computational overhead [4]. To mitigate this overhead, studies have delegated data verification to a third party (TPA), albeit this incurs an elevated risk of data leakage due to the TPA's curiosity when users upload data [5]. To curtail the peril of data pilferage by the TPA, an identity-based public auditing scheme has been proposed, bolstering the trustworthiness of third-party services by increasing the computational overhead on the user's end [6]. While this scheme efficaciously diminishes the risk of third-party data theft, it does not alleviate the computational burden. Typically, users remunerate service fees to compel the third party to earnestly adhere to the service agreement, thereby diminishing the risk of data leakage. It is thus discerned that the third party's credibility is instrumental in determining the resilience of data information against supply chain vulnerabilities.

In the contemporary era of big data, this treatise advances a secure data storage protocol. Initially, it delineates the quandaries encountered in safeguarding computer network data, subsequently, it articulates the system and objective designs predicated on these quandaries. Thereafter, it advocates for a storage strategy that enhances the Hash algorithm (H-AONT), amalgamating it with conventional encryption techniques to elevate data security storage levels. Ultimately, the practicability of this computer data security storage blueprint is corroborated through the construction of an analogous system model, thereby furnishing a benchmark for the enhancement of data security storage within computer networks.

2. Challenges to the secure storage of computer network data in the context of the big data era

2.1 Security risks in data information

Amidst the swift proliferation of information technology, mobile terminals have become ubiquitous in daily life, engendering a dependency that, while facilitating routine activities,

Correspondence: Liying Zhang (zhangliying@lumei.edu.cn), Xin Gu (guxin@lumei.edu.cn), Qiang Zhao (zhaoq@sj-hospital.org).

concurrently amplifies the susceptibility to computer data breaches. The vast expanse of big data, with its diverse data types, furnishes malefactors with clandestine avenues for data exfiltration, imperiling the financial integrity of individuals and corporations alike. Despite the fact that information technology's widespread adoption has catalyzed corporate growth, it has concomitantly escalated the security risks associated with data storage. Instances abound where cybercriminals, leveraging malware, compromise corporate systems, adulterating the stored data and undermining its sanctity.

2.2 Insufficient precision in screening data information

In the big data epoch, the velocity of computer network data storage is accelerating, intensifying the volume of data and information stored and necessitating more stringent screening protocols. Traditional data screening methods, lacking in precision, fail to satisfy user requirements, complicating the accurate retrieval of information and leaving it prone to contamination by spurious data, which severely disrupts the maintenance of computer data by operational personnel.

2.3 Wide spread of spam data information

Furthermore, the big data era is characterized by an inundation of spam data, a byproduct of information technology's advancement that facilitates access to information but also intersperses copious amounts of unsolicited content, such as advertisements and emails, within legitimate data streams. This deluge of spam significantly hampers user data acquisition efforts. Although current computer technologies can filter out spam to some degree, eradication is elusive. Notably, as of 2019, spam constituted 57% of global email traffic, with significant contributions from various countries. These spam communications often harbor viruses, planted by attackers, which, upon interaction, compromise user data privacy and present a formidable threat to the security of data storage for both individuals and enterprises.

3. System modeling and design objectives

3.1 Modeling

In the present study, a network data security storage model is delineated, encompassing two principal facets: data security and data veracity. The model is underpinned by a tripartite framework comprising end-users, a third-party auditing agent (TAP), and a cloud service provider (CSP). Within this schema, end-users are tasked with the conversion and encryption of data prior to its transmission to the cloud repository. Concurrently, the TAP's role is pivotal in authenticating the keys and data block tags generated by users, thereby ascertaining the integrity and completeness of the stored data. The CSP, in turn, is instrumental in furnishing requisite storage solutions to the users [4].

3.2 Threat modeling

The role of the secure storage program designed in this research is to address the problem of usable information leakage from outsourced data. Assuming that TAP is serious about fulfilling public audit protocols, the threats modeled at this point specifically include the following five aspects:

(1) Intruder Attacks: Attackers obtain user cloud information through different channels;

(2) CSP attack: maintains the holding validation in spite of an attack on the server;

(3) Replacement attack. This form is mainly audited by the complete block of data to achieve the replacement of corrupted data blocks;

(4) Forgery attack. After data corruption, the attacker deceives the TPA by forging proof information;

(5) replay attack. the CS runs the previously verified information against any intact data block to prove that the corrupted or lost data block is held intact [5].

3.3 Design objectives

To ensure that data storage is secure and efficient and resistant to the threat models described above. The storage solution needs to be designed to meet the following objectives:

(1) Conversion encryption: when the user encrypts the data, the original data file needs to be converted using the H-AONT conversion mechanism, and the security encryption algorithm generates the corresponding ciphertext data;

(2) Lightweight architecture: After the ciphertext data is uploaded to the cloud, a trusted third party is authorized to perform data auditing, determine whether the data is complete or not, and generate corresponding data labels;

(3) Storage security: In the process of data storage, it must be ensured that TAP and the attacker can not be outsourced through the data block to obtain plaintext information;

(4) Low communication: only TPA and CSP communication is provided at the time of proof of possession.

4. Computer network data security storage program

The secure storage scheme proposed in this research consists of seven algorithms: *Lock*(·), *SliceGen*(·), *TagGen*(·), *ChallGen*(·), *Proof-Gen*(·), *Proof-Veri*(·), *Unlock*(·), which are mainly divided into the following stages.

4.1 File encryption

In the file encryption process, the algorithm is *Lock*(·), when the user encrypts the data through the scheme proposed in this paper, the pseudo-message data block is sent to the TAP. In this process, the asymmetric encryption algorithm then achieves the effective protection of outsourced data [6].

Assuming the original file is F, then the user can encrypt the original data file via *Lock(F)* $\rightarrow$ *UF*. The steps are as follows:

Step 1: Divide the original file to generate n files, i.e., $F = \{f_1, f_2, f_3, \cdots, f_n\}$.

Step 2: After the division is completed, randomly select the file block and convert the secret key K_{tr}, t to indicate the replacement rule.

Step 3: With the help of H-AONT, $F = \{f_1, f_2, f_3, \cdots, f_n\}$ is converted and thus the strongly indivisible pseudo-message data block is obtained, i.e., $F' = \{f'_1, f_2, f_3, \cdots, f_{n'}\}$, where $n = n + 1$.

Step 4: In generating the pseudo-message data block, the private key K is randomly selected and encryption is realized by the symmetric encryption function $E(\cdot)$, i.e., $E_K(F) \rightarrow C$ and the ciphertext set $C = \{C_1, C_2, C_3, \cdots, C_{n'}\}$.

Step 5: After obtaining the set of ciphertexts, divide it into sets of

varying lengths, with the long set being α and the short set being β, then there is $C = \alpha + \beta$, where $|\alpha| \ll |\beta|$.

Step 6: Use β as an user file and upload it to a third party; α and K are saved on the user side.

4.2 Pre-processing

In the preprocessing process, the main algorithms used are *SliceGen*(·) and *TagGen*(·). After the user transmits the ciphertext to the TPA, the data is sliced and the corresponding labels are generated, and the random numbers and location information are added to them, which ensures the uniqueness of the data and avoids the attacks of the server. The specific process is:

4.2.1 The data is processed in slices, $SliceGen(UF, s, t) \rightarrow \{uf_{ij}\}$4.2.1 The data is processed in slices, S l i c e G e n (U F , s , t) → { u f i j } {\displaystyle SliceGen(UF,s,t)\to \{uf {ij}\}}

- The TPA divides UF into s data blocks, which gives: $UF = \{uf_i\}_{1 \le i \le s \le n'}$.
- Randomly select the t slices and construct the $s \times t$ data slice matrix $UF_{s \times t}$, then $uf_i = \{uf_{ij}\}_{1 \le i \le s, 1 \le j \le t}$

$$UF_{s \times t} = \begin{bmatrix} uf_{11} & uf_{12} & \cdots & uf_{1t} \\ \vdots & \vdots & \vdots & \vdots \\ uf_{s1} & uf_{s2} & \cdots & uf_{st} \end{bmatrix} \tag{1}$$

4.2.2 Generate labels, $TagGen(K_{ver}, \{uf_{ij}\}) \rightarrow \Phi$ 4.2.2 Generate labels, T a g G e n (K v e r , { u f i j }) → Φ {\displaystyle TagGen(K {\nu er},\{uf {ij}\})\to \Phi }

① The third party randomly selects the corresponding authentication key K_{ver}.

② Calculate the document identification $UF_{ID}: E = Hash(UFname \parallel s \parallel t)$.

③ Generate labels, then there:

$$\phi_i = g_{k_{ver}}^{\varepsilon + ri} \times h(uf_i) \bmod p = g_{k_{ver}}^{\varepsilon + ri} \times \prod_{i=1}^{s} g^{uf_i} \bmod p \tag{2}$$

In Eq. (2), $h(\cdot)$ denotes the homomorphic Hash function; ri denotes the random number; $w(\cdot)$ denotes the random function; $\{0,1\}^{|i|} \times \{0,1\}^{l} \rightarrow \{0,1\}^{l}$ generates, in $w(i) \rightarrow ri$, i denotes the position information of the data block in the matrix.

④ Finally, $(uf_i, \varepsilon, \phi_i)$ is uploaded to the GSP and the data blocks as well as labels are stored in the cloud server [7], respectively.

4.3 Proof of data holdings

The algorithms *ChallGen*(·), *Proof-Gen*(·) and *Proof-Veri*(·) are mainly used in this phase. After the third party sends a challenge to the GSP, the information returned in the GSP is verified for correctness.

Send a challenge to $ChallGen(\lambda \rightarrow)chall$ and a third party generates a challenge message:

① Randomly select the line parameter λ to challenge and obtain the index key $k_{ind} \leftarrow f(\lambda)$;

② The third party develops a challenge fast quantity, which is recorded as Z;

③ Generate challenge messages.

4.3.1 Generation of supporting information, $Proof - Gen(\varepsilon, k_{ind}, Z) \rightarrow proof$4.3.1 Generation of supporting information, P r o o f − G e n (ε , k i n d , Z) → p r o o f {\displaystyle Proof-Gen(\varepsilon ,k {ind},Z)\to proof}

① After the GSP obtains the challenge information, calculate the Z set.

② Return proof information $proof = (\delta, \xi)$

$$\delta = \sum_{i=1}^{s} uf_i \bmod q$$
$$\xi = \prod_{i=1}^{s} \phi_i \bmod p \tag{3}$$

4.3.2 Integrity validation of challenge data blocks

Calculate the challenge index $[i]$ with the random number ri, ie:

$$i = \sigma_{k_{ind}}(z) \mid_{1 \le z \le Z}$$
$$ri = \omega(i) \tag{4}$$

The integrity verification equation can be expressed as:

$$\Phi_{z \mid (1 \le z \le Z)} = g_{k_{ver}}^{\sum_{i=1}^{s} (\varepsilon + ri)} \times h(\delta) \bmod p^* = \xi \tag{5}$$

If Eq. (5) holds, then it passes the verification, at which point the proof challenge block is correctly held; conversely, the challenge block is lost, or appears to have been tampered with [8].

4.4 Confidentiality of documents

This phase mainly uses the *Unlock*(·) algorithm, which combines and decrypts the corresponding outsourced data with the local data blocks after the user downloads it from the cloud server, thus obtaining the original file.

The user side performs the unlocking, i.e. $Unlock(\alpha, \beta, K) \rightarrow F$. The main steps are as follows:

Step 1: Download the complete outsourcing data from the cloud $UF = (\beta)$;

Step 2: Integrate the α and β ciphertexts to obtain the pseudo-message ciphertext data block;

Step 3: Decryption through the H-AONT mechanism.

4.5 Security analysis

In order to analyze the security of the above threat model, this paper evaluates the effectiveness of the model against intruder attacks in terms of strong indivisibility, holdout determinism of outsourced data [9].

4.5.1 Strong indivisibility analysis

If the H-AONT mechanism can realize resistance against intruder attacks, and the corresponding data information can be obtained through the outsourced information.

Suppose the stored pseudo-message data block is f_1 and the outsourced data block stolen by the attacker through different channels is $f_2, f_3, f_4, \cdots, f_n$, at this point the steps for the provider to restore the original data learning based on the outsourced data are:

Step 1: Sorting the acquired data according to the user's replacement rule t gives: $M = m_1 \| m_2 \| m_3 \| \cdots \| m_n$;

Step 2: Get the conversion key K_{tr};

Step 3: Based on the conversion key, compute the original data block.

An attacker can decrypt the original data only if all the above three conditions are met simultaneously.

The security analysis of outsourced data is specifically:

① Assuming that the replacement rule t is compromised and the amount of outsourced data stolen by the attacker is n, there are:

$$\begin{matrix} m_j = T_t(f_i), i, j = 2, 3, \cdots, n' \\ M' = m_2, m_3, \cdots m_n \end{matrix} \quad (6)$$

The lack of f_1 at the time of the attacker's substitution makes it difficult for $H(M') \neq H(M)$, at this point, to obtain the corresponding conversion key K_{tr}, making it difficult for the attacker to obtain the information [10].

② Suppose f_1 is leaked, and the acquired f_1 is $n+1$ one. At this point the attacker can complete the sequence reorganization by forging the alignment rules, i.e.:

$$\begin{matrix} m_j' = T_{k'}(f_i), i, j = 2, 3, \cdots, n' \\ M' = m_2', m_3', \cdots m_{n'} \end{matrix} \quad (7)$$

In Eq. (7), k denotes the alignment rule forged by the attacker. Due to $H(M') \neq H(M)$, the provider fails to steal the data message. From this, we know that it is difficult for the attacker to obtain $n+1 f_1$ to obtain the correct message sequence [11] without the user replacement rule t being compromised.

The comprehensive analysis shows that the completeness of network data storage improves with the increase in the number of converted data blocks after the corresponding conversion through the H-AONT mechanism.

4.5.2 correctness analysis

If the third party is serious about fulfilling the agreement with the service provider and the data block in the cloud is complete, at this point the certificate of possession can be used to determine whether the data block stored in the GSP is complete [12].

Assuming that a block of data in the cloud is corrupted or lost, at this point the GSP can perform a proof of holding on the basis of the data held by the TPA, i.e:

$$\begin{aligned} \Phi &= g_{k_{ver}}^{\sum_{i=1}^{s}(\varepsilon+ri)} \times h(\delta) \bmod p \\ &= g_{k_{ver}}^{\sum_{i=1}^{s}(\varepsilon+ri)} \times h(\sum_{i=1}^{s} uf_i) \bmod p \\ &= g_{k_{ver}}^{\sum_{i=1}^{s}(\varepsilon+ri)} \times \prod_{i=1}^{s} g^{uf_i} \bmod p \\ &= \prod_{i=1}^{s} (g_{k_{ver}}^{(\varepsilon+ri)} \times g^{uf_i}) \bmod p \\ &= \zeta \end{aligned} \quad (8)$$

In the case of packet loss, it is difficult for GSPs to pass third-party hold validation [13].

Assuming that the data storage scheme proposed in this research is resistant to GSP attacks, in the case of uf_i loss, the GSP is attacked through the complete uf_t as well as ø as a way to pass the third-party's holdability verification, at which point it returns that the evidence returned by the GSP contains:

$$\begin{aligned} \delta &= (uf_t + \sum_{i=1, t\neq i}^{s} uf_i) \bmod p \\ \zeta &= (\phi_t + \sum_{i=1, t\neq i}^{s} \phi_i) \bmod p \end{aligned} \quad (9)$$

This shows that the scheme proposed in this paper enables GSP attack resistance.

It is assumed that the data storage scheme proposed in this research is resistant to the GSP forgery attack, in the case of uf_i loss GSP through the third party deceived by the forgery of uf_t, based on the Hash function, only when $uf_i = uf_t$ can be obtained from $h(uf_i) = h(uf_t)$, and then through the proof of holdability [14,15].

Assuming that the data storage scheme proposed in this research is resistant to GSP multiple attacks, and that the GSP, in order not to disclose the loss of data blocks to the third party, will operate on the evidence messages that pass integrity, the data blocks that are not lost, and the corresponding labels, and return the evidence to the third party, there is:

$$\begin{matrix} \delta' = (uf_t + \delta_1) \bmod p \\ \zeta = (\phi_t \times \zeta_1) \bmod p \end{matrix} \quad (10)$$

At this time $g_{k_{ver}}^{\sum_{i=1}^{s}(\varepsilon+ri)} \times g^{\delta'} \bmod p \neq \xi'$. As a result, it is known that the data location is then added to the data label, making it difficult for the GSP to access the data, which in turn effectively improves the security of data storage.

5. System test

5.1 Test environment

In this paper, we further verify the performance of the proposed scheme in this paper by building the corresponding system model. The built system model mainly consists of a computer with IntelCorei7 CPU, 16GBRAM, and an AliCloud server [16].

5.2 Test indicators

System storage overhead, communication overhead, computation overhead, are selected as test metrics.

5.3 Analysis of results

(1) Comparative analysis of communication overhead

In the communication overhead comparison process, challenge-answer is used as the benchmark. For the scheme in this paper, its main common index key, the number of challenge blocks, etc. constitute the corresponding index set, and the data block own set and label set returned by the GSP are used as evidence, and the communication load is relatively low.

(2) Comparative analysis of storage overhead

In the scheme of this paper, after the user transmits the ciphertext to the cloud server, the locally reserved copy can be deleted, and only the private key and the short ciphertext data block collection are retained. Wherein, the size of the private key is 128bit; the size of the short ciphertext data block collection α can be calculated by the following formula, i.e.:

$$|\alpha| = a \times qbit \tag{11}$$

In Eq. (11), α and q represent the number and length of ciphertext data blocks, respectively. In practice, the size of q is set to 128bit. In order to ensure the security of data storage, the user usually only retains 1 block of ciphertext data, and the size of α is 128bit. It is thus known that the storage overhead of the proposed scheme in this paper is relatively low [17].

(3) Comparative analysis of computational overhead

In the scheme proposed in this paper, the plaintext is converted and encrypted by H-AONT algorithm after dividing it accordingly. Among them, the size of each data block is 128bit, compared with the traditional AONT algorithm, H-AONT replaces the 2s a 1 time different-or operation by the Hash value. Since the Hash function has strong practicality, the calculation process is more convenient. To ensure the accuracy of the test, this paper uses the same structure of pseudo-random function AONT, ASE, H-AONT + ASE instead of the encryption algorithm to analyze the computational overhead, the results are shown in Figure 1.

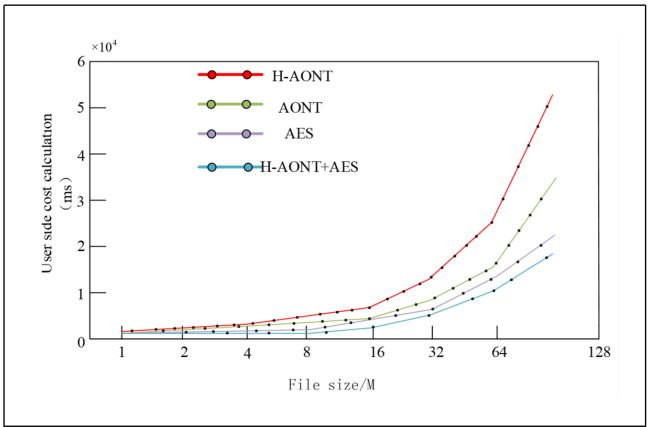

Figure 1. Comparison of computational overhead between this paper's algorithm and traditional encryption algorithm

Upon scrutiny of Figure 1, it becomes evident that the algorithm delineated herein incurs a computational overhead that is comparatively diminutive for an equivalent file magnitude. Postulation of a file dimensionality at 20M permits the inference that subsequent to the partial data transmission by the user to the cloud server, the TPA engenders the pertinent data labels. Thereafter, the data's veracity undergoes validation. Concomitantly, Figure 2 elucidates the computational overhead juxtaposition at the user's terminus vis-à-vis the TPA's locus.

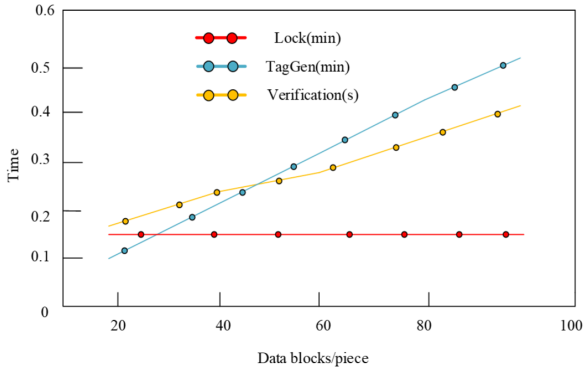

Figure 2. Computational overhead of user side and TPA in data preprocessing stage

Figure 2 elucidates that the encryption's epicenter resides with the user, while label generation for data is executed by the TAP. It is observed that when encrypted data spans 20M, the user's computational burden remains invariant despite an escalation in the count of authenticated data blocks. This phase witnesses a progressive augmentation in the third party's computational load. Notably, for fewer than 30 files of 20M size, the user's overhead is pronounced, attributable to the third party's predominant role in data block-based label generation, where fewer labels equate to diminished computational demands on the TPA.

The analysis substantiates that the algorithm enhances data storage security post H-AONT key transformation without imposing significant computational overhead. Moreover, post-encryption and cloud upload by the user, the TPA's verification of cloud data holdings mitigates the user's computational load, thereby bolstering data storage security.

5.4 Security Comparison

Literature 1 and literature 2 schemes are used, respectively, to compare with the scheme designed in this paper, and then analyze the safety of the scheme designed in this paper, and the comparison results are detailed in Table 1.

Table 1. Comparison of security data storage security of different schemes

Safety	Program of this paper	Literature [3] program	Literature [4] program
Resistant to Replacement Attacks	be	be	be
Resistant to forgery attacks	be	be	be
Resistant to replay attacks	be	be	be
Resistant to key attacks	be	clogged	clogged
Availability of enhanced credibility of external audit services	be	clogged	be

Analysis of Table 1 reveals that the triad of schemes proffer robust security capabilities, effectively thwarting substitution, forgery, and replay assaults. However, the methodologies delineated in Fu et al. [3] and Xue et al. [4] exhibit a pronounced reliance on third-party audit services. The encryption algorithms employed are of a conventional ilk, inheriting intrinsic

limitations that predispose data to potential compromise upon key exposure. Concurrently, while the strategy in Xue et al. [4] marginally bolsters third-party audit service trustworthiness, it fails to impose requisite constraints on the auditor, thereby attenuating its reliability. In contrast, the approach advocated in this study fortifies data storage security, adeptly circumventing key leakage risks and augmenting audit service viability, thereby significantly contributing to the secure storage of computer network data.

6. Suggestions for the development of computerized data storage in the era of big data

6.1 Sound computer data information security prevention system

In the era of burgeoning big data, the escalation of computer network data necessitates stringent security measures. To foster the salubrious evolution of data storage technologies, an enhancement of the computer security apparatus is imperative to safeguard information integrity.

Firstly, the refinement of the information security management system is paramount. This involves a methodical standardization of the technical protocols associated with secure data storage and a bolstered scrutiny of potential security hazards. Implementing preventative measures facilitates the prompt identification of vulnerabilities within the data storage continuum.

Secondly, the professional acumen of IT personnel is crucial. Given their pivotal role in data storage operations, it is essential to augment their proficiency in information security and elevate their training in professional competencies. This will enhance their cognizance of security protocols, thereby advancing the state of computer data storage technology.

Lastly, the persistent advancement and application of computer information security technologies are vital. In an epoch characterized by rapid technological progression, attention to the development of data confidentiality, obfuscation, and authentication technologies is critical. Confidentiality technologies aim to shield user privacy effectively, while obfuscation techniques, such as covert communication and content verification, enhance the stability of data security storage. Authentication technologies serve as bulwarks against cyber threats, mitigating the risks of system infiltration and data manipulation.

6.2 Strengthening computerized data storage and screening efforts

In the era of big data, the intricate interconnectivity of networked data has exacerbated the challenges associated with data storage and filtration. The proliferation of superfluous data—manifested as spam and junk advertisements—compounds these difficulties. Such data, upon integration into computer systems, opens avenues for malicious entities to launch cyber-attacks or disseminate viruses, thereby imperiling data integrity and precipitating potential system collapse. To circumvent these threats, it is imperative to implement precise data curation and storage strategies. Leveraging the multifaceted nature of big data and aligning with user requisites, it is essential to amalgamate data, employing sophisticated, intelligent methodologies to excise non-essential data, thereby economizing storage capacity while fortifying computer system security. This approach not only ensures a seamless network milieu but also augments the efficacy of data conveyance and conservation. Furthermore, the deployment of data mining and cloud computing technologies is crucial for the meticulous excavation and sifting of voluminous datasets. Given the comprehensive and bespoke attributes of data mining technology, its application in data processing—tailored to data idiosyncrasies and mining extant correlations—can be seamlessly integrated with computer storage technology. This integration is pivotal in enhancing the efficiency and security of data storage, concurrently satisfying the concrete data storage needs of users.

6.3 Emphasize the construction of computer hardware performance and quality

Amidst the swift advancement of information technology, the robustness and quality of computer hardware are recognized as pivotal determinants in the evolution of data storage technologies. Failures in computer hardware are typically categorized into three distinct phases: pre-failure, failure, and post-failure, with causative factors attributed to external, internal, and human elements. These factors collectively pose a formidable risk to the secure storage of networked computer data. Consequently, to substantially elevate the security level of data storage within computer networks, it is incumbent upon IT professionals to incessantly fortify the architecture of computer hardware. This necessitates a two-pronged approach:

Firstly, the optimization of the computer network is essential. In comparison to international benchmarks, China's development in computer hardware performance is perceived as lagging, thereby impeding the nation's technological progression. Addressing this, it is imperative to harness cutting-edge technologies, such as cloud computing, to effectuate a comprehensive enhancement of the computer network. This strategy promises not only to bolster hardware performance but also to refine the overall quality of hardware development.

Secondly, the design and development of computer hardware must be integrally aligned with the evolving landscape of the computer industry. This entails a collaborative, multi-faceted design paradigm that leverages collective strengths and mitigates weaknesses. Such an approach, deeply embedded in the research and development of computer technology, aims to amplify hardware performance. Concurrently, it seeks to augment the construction quality of the hardware, thereby fostering an environment conducive to the secure storage of computer network data.

7. Conclusion

In an epoch characterized by accelerated technological progression, the safeguarding of data within computer networks has emerged as both a novel developmental frontier and a domain susceptible to data breaches. This manuscript delineates an enhanced secure storage protocol predicated on an augmented All-Or-Nothing Transform (AONT), utilizing the Hybrid-AONT (H-AONT) dual encryption algorithm. This schema not only fortifies the security of network data repositories but also, to a measurable degree, bolsters the trustworthiness of third-party audit services. Concurrently, it facilitates the transference of computational burdens from the user to the third party, thereby alleviating the data processing load on the user end. Moreover, the third party is equipped to generate requisite duplicate files tailored to user specifications, which not only solidifies system dependability but also incrementally fortifies the system's resilience against risks. To ensure the unimpeded advancement of secure data storage technologies, it is incumbent to perpetually refine the technical acumen of personnel tasked with data operation and maintenance management. This encompasses continual enhancements in computer hardware infrastructure, rigorous data vetting, and

the reinforcement of computer network data security storage levels, all while amplifying data storage precision. Given the current climate of burgeoning information technology and the escalation of malevolent cyber activities, future research must integrate user-centric data storage requisites with ongoing refinements in network data storage methodologies to effectively mitigate data leakage risks.

References

[1] Lei L., Wang Y., Meng F. Computer data security storage technology and application. Network Security Technology and Application, 4:33-36, 2012.

[2] Chanhyuk L., Jisoo K., Heedong K., et al. Addressing Io Tstorage constraints. A hybrid architecture for decentralized data storage and centralized management. Internet of Things, 25:101014-101020, 2024.

[3] Fu A.-m., Li Y.-h., Yu Y., et al. DoR: an IDA-based dynamic proving rebability scheme for cloud storage systems. Journal of Network and Computer Applications, 104:97-106, 2018.

[4] Xue J.-t., Xu C.-x., Zhao J.-n., et al. Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. Science China Information Sciences, 62(3):45-60, 2019.

[5] Dastgeer G., Nisar S., Rasheed A., et al. Atomically engineered, high-speed non-volatile flash memory device exhibiting multi bit data storage operations. Nano Energy, 119, 109106, 2024.

[6] Cervantée E.K.W., Ngauru T.R., Ken T., et al. In safe hands: child health data storage, linkage and consent for use. Health Promotion International, 38(6):1-10, 2023.

[7] Verdesoto I., Navajas F.Á., Roca B.J.P., et al. Preventive conservation of a short theatre skit (Valencian "Sainete") with cloud data storage and Internet of Things. Sensors, 23(24):134-141, 2023.

[8] Famutimi R.F., Oyelami M.O., et al. An empirical comparison of the performances of single structure columnar in-memory and disk-resident data storage techniques using healthcare big data. Journal of Big Data, 10(1):1123-1129, 2023.

[10] Sivakumaran S. Easing the data sharing burden: perspectives and principles to successfully leverage AD data repositories, infrastructures and functionality. Alzheimer's & Dementia, 19(S21), 2023.

[11] Wen X., Liu Q. The use of artificial intelligence technology in computer network data security storage. Digital Communication World, 11:148-150, 2023.

[12] Liu F., Wang B., Jiang F., et al. A secure storage system for hydropower station data based on cloud computing. Automation Technology and Application, 42(3):97-100, 2023.

[13] Chou T.B. (2023). Research on data security storage and permission verification scheme based on blockchain technology. Software, 44(7):86-88, 2023.

[14] Han S.-P. Data security storage method of electronic labor contract based on blockchain technology. Information Technology and Informatization, 5:169-172, 2023.

[15] Jinshan. Design and experimental analysis of network data security storage retrieval system for medical big data. Science and Technology Innovation, 8:96-99, 2023.

[16] Wang L., Wang Z., Wang L. Exploration of data security storage strategy based on cloud computing. Network Security Technology and Application, 6:68-70, 2021.

[17] Liu Y., Wang H., Zhang M., et al. Collaborative model for secure data storage in cloud computing environment. Computer Application Research, 35(10):3091-3095, 2018.