MAPAN-JOURNAL ,
Publisher: METROLOGY SOC INDIA , NPL PREMISES,
Address: DR K S KRISHNAN MARG, NEW DELHI, INDIA,
ISSN / eISSN:0970-3950 / 0974-9853,
Volume 25-Issue 1-(2025)
https://doi-001.org/1025/17669999135042

OPEN ACCESS

# Research on library user information compression and encryption management based on improved RDHEI algorithm

Check for updates

Yuchao,Wang[1,a,*], Jiahang,Song[2,b], Yingdan,Zhang[3,c]

[1]School of Economics, Management and Law, Changchun Normal University, Changchun 130213, Jilin, China

[2]Clinical Medical College, Changchun University of Chinese Medicine, Changchun 130017, Jilin, China

[3]School of Geography, Earth and Environmental Sciences, University of Birmingham, Birmingham, UK

[a]18243498075@163.com

[b]jiahangsong1024@outlook.com

[c]YXZ1901@student@dham.ac.uk

*Corresponding Author

## Abstract

To address the imbalance between "capacity storage, access security, and timeliness of demand" caused by the explosive growth of multimodal library user information in the big data era, this study explores an improved RDHEI algorithm. The algorithm, based on a parallel structure of "user information processing, information compression, and privacy-sensitive hierarchical encryption," addresses the long serial execution times of compression and encryption due to poor technical applicability, and the original algorithm fails to consider information relevance and privacy. The preprocessing algorithm cleans and intelligently categorizes user information. The compression algorithm dynamically switches between lossless and lossy encryption while considering information relevance to achieve better applicability. The encryption algorithm groups information and applies AES-256/AES-128 hierarchical encryption, running concurrently with the compression algorithm. The improved algorithm achieves a compression effect of 28.5% for structured data and 22.3% for unstructured data. The key privacy encryption time is 42.1ms, and the service latency is 168.3ms. With a doubling of the data volume, scalability is reduced by only 5.2%, while meeting privacy standards. This will improve the level of solving library user information management problems, is suitable for smart library scenarios, and can provide better service to users. It can also provide certain technical references for user information confidentiality management in the public service field.

**Keywords:** improved RDHEI algorithm; library user information; data compression; encryption management

## 0Introduction

The exponential growth in the scale and diversification of library user information is driven by the widespread adoption of big data and artificial intelligence technologies in library

management (Maindarkar, 2025). As fundamental data and the basis for services such as precise library recommendations and collection document allocation, and as crucial for protecting user privacy and security, efficient storage and effective protection are crucial requirements for library information management. While mainstream data compression algorithms offer lower storage costs, they are unsuitable for information in libraries containing multiple types and diverse patterns. Some encryption and compression algorithms, due to their focus on a particular data type, have relatively low compression rates. Furthermore, encryption and compression are often performed serially, significantly extending data processing time and hindering service needs such as borrower identification and query services. The RDHEI algorithm has the ability to compress and encrypt in traditional scenarios, but the algorithm was not originally designed based on the characteristics of strong correlation and high privacy sensitivity of library user information, so it is difficult to meet the requirements of stable compression performance and lightweight encryption; and in existing research, there are more separate analyses of data compression and encryption (Feng et al., 2024), and fewer studies on integrated technologies in library scenarios, resulting in a "tug-of-war" between data processing and storage efficiency, security, and timely response, exacerbating the imbalance between the three.

Based on this, this study is guided by the actual needs of libraries and conducts research on the design of an improved RDHEI algorithm. By optimizing the algorithm structure, the parallel integration of compression and encryption is achieved, aiming to build a processing solution that takes into account high compression rate, lightweight encryption and system adaptability, providing technical support for the intelligent management of library big data, and also providing a reference for user information security management in similar public service fields.

## 1 Related theoretical basis

### 1.1 Theories related to library user information

Theory related to library user information forms the foundation for developing specialized compression and encryption technologies. This foundation is based on understanding data attributes and management patterns, serving as a starting point for implementing big data intelligence technologies in libraries. By definition, data refers to the entire spatial data space involved in library service interactions, including personal information and service interaction data. It is both a type of service support data and, at the same time, carries privacy implications, making it a fundamental strategic resource for library data intelligence (Zhang Fangfang & Liu Qiang, 2024). Logically, it can be divided into structured and unstructured data. Structured data

processing requires high complexity and requires differentiated technical adaptation. In terms of privacy sensitivity, it can be divided into privacy data (such as ID numbers) and service data (such as borrowing and returning data), which also serves as the starting point for subsequent encryption and hierarchical protection technologies. User-related data, based on its characteristics, is characterized by relevance (the correlation between borrowing data and preference data) and timeliness (the real-time nature of borrowing). These two aspects of data require both relevance preservation and inefficiency in processing. Of course, the coupling of these data characteristics with privacy attributes makes traditional technologies unable to meet these requirements. Therefore, a dedicated solution designed specifically for these theoretical characteristics is necessary, which also provides guidance for improving the RDHEI algorithm.

### 1.2 Basic Theory of Data Compression and Encryption

Compression theory and technology are fundamental to achieving efficient and secure access to library user information. It represents a trade-off between the efficient storage of user information and the time required to ensure data security. Compression theory can be categorized into lossless and lossy compression based on data loss. Lossless compression algorithms (such as the Huffman algorithm and the LZW algorithm) can be restored using lossless decompression methods after compression, making them highly suitable for structured information such as library user identity information and borrowing history (Devarasetty, 2024). Lossy compression algorithms (such as wavelet compression) achieve maximum compression by discarding unimportant information, but because some information is lost, they are not suitable for text descriptions and short texts such as book reviews, reading preferences, and inquiries. Encryption theory and technology are similar to traditional library user information theory. Based on whether the encryption algorithm keys are different, encryption can be categorized as symmetric and asymmetric. Symmetric encryption algorithms use the same key, resulting in relatively low computational overhead for encryption and decryption, making them suitable for real-time services like library borrowing verification that require high access times. Asymmetric encryption algorithms use different keys, resulting in high computational overhead and a higher security factor. This makes them unsuitable for real-time access scenarios and is often used for offline encryption of core privacy data. Traditional compression theory and encryption algorithms are typically performed sequentially and lack efficiency optimization considerations for multimodal library information. For example, when applied to mobile searches requiring relatively short response times, these algorithms are not practically applied. This suggests that improved algorithms should meet the requirements of parallel and integrated integration.

### 1.3 Analysis of the RDHEI Algorithm Principle

The compression-encryption RDHEI algorithm is designed for big data scenarios. Its primary goal is to address the efficiency issues inherent in the serial nature of traditional compression and encryption processes through structural integration (Devarasetty, 2024). Its principle framework is based on the concept of "block-based adaptive processing": input data is segmented into coarse-grained blocks (such as text or numeric data), subjected to mean filtering to remove noise and redundancy, and then compressed using a hybrid "entropy coding + dictionary encoding" compression mechanism. This mechanism uses LZW dictionary encoding to achieve higher compression rates for frequently repeated data, while Huffman encoding reduces redundancy for sparse input data. A lightweight symmetric encryption module is also embedded in the compression process, encrypting the compressed code stream in segments using a dynamic key generated during the session, achieving a preliminary parallel "compression-and-encryption" process. Because the original algorithm was designed for traditional scenarios (such as general text data) , it achieves a well-balanced compression rate of 12%-15% compared to Huffman encoding alone, and reduces encryption latency by over 20% compared to "compressed, standalone AES encryption." However, the original algorithm design did not optimize the association between input data (for example, the coupling logic information between multiple fields was not retained), and the strategy for segmenting unstructured data was single, which was not suitable for the characteristics of library user information, which is "structured-unstructured mixture, strong correlation between behavior and identity". This is also the target of the algorithm improvement in this study .

### 1.4 Related evaluation indicators

#### 1.4.1 Compression performance evaluation indicators

Library user information has the characteristics of "structured-unstructured hybrid, lossy compression of core content", etc. The evaluation indicators should reflect the requirements of structured and unstructured data, lossless compression of core data, etc., and include three aspects of indicators.

The first is CR, the compression ratio, which is calculated as the ratio of compressed data size to original data size. Smaller data means more compression. Compression ratios are calculated for structured data (e.g., borrowing form data) and unstructured data (inquiry record text data) to meet the library's needs for mixed storage of diverse information types. The former requires complete compression without losing field associations, while the latter can be compressed without losing redundant record descriptions. This is the key to evaluating the algorithm's adaptability to compressing different types of information (Devarasetty, 2024). The

second is CS, the compression speed, measured in MB of compressed file size. This represents the instant service speed required for library functions such as reader verification and information retrieval, as mentioned in the introduction. Algorithm evaluation model metrics should prioritize speed stability during periods of high concurrency (e.g., borrowing records during winter and Spring Festival holidays) to avoid compressed file processing timeouts. The third is DRP, which is the data restoration accuracy, measured by the "degree of similarity between the restored information and the data information". Among them, the DRP of the main information (such as identity information) is ≥99.9% (acceptable without loss), and the DRP of the secondary information (such as borrowing hobby information) is ≥95% (acceptable with loss). This indicator ensures the effective integrity of key information, and also provides important guarantees for the analysis and quantification of the effectiveness of the model method in the subsequent algorithm design and experimental data performance.

### 1.4.2 Encryption performance evaluation indicators

In response to the library users' demands for "privacy classification and real-time service" of information, an encryption performance evaluation index, namely the encryption measurement matrix, is constructed from the three dimensions of "security-efficiency-reliability".

The basic metric of the encryption metric matrix is key security strength (KSS). The degree of key design varies for information of different sensitivity levels: when using asymmetric keys to encrypt key user privacy information such as ID numbers and payment-related data, the RSA key length must be ≥ 2048 bits, and the symmetric key AES encryption algorithm uses AES-256, and the unpredictability of its generated keys has been proven through the NIST SP800-22 random number test (Oprea et al., 2023). General service information such as borrowing subject preferences can be simplified to AES-128, which is a simple encryption scheme that slightly sacrifices efficiency over security.

The second dimension of the encryption metric matrix is encryption delay (ED), expressed in milliseconds as the encryption time required for a single piece of data. Under the low-latency requirements of borrower identity authentication and real-time borrowing queries, the encryption delay for sensitive information is ≤50ms (to avoid lags that cause users to wait), and for general information, it is ≤100ms. Furthermore, the service efficiency stability coefficient is calculated—when the data volume surges to a peak (such as the amount of borrowed data during the Spring Festival holiday), the delay coefficient is ≤15% .

The third dimension of the encryption measurement matrix is the decryption success rate (DSR), which is calculated as "correctly decrypted data volume/total encrypted data volume ×

100%"; because its service information is used to support services such as precise push and borrowing management, the encrypted structured data form information (such as borrowing information form data) DSR ≥ 99.9%, and the unstructured information text (such as consultation and communication log text) DSR ≥ 99% to ensure correct decryption and avoid deviation.

The above indicators not only reflect the essential dimensions of encryption technology, but also fit the application scenarios of libraries, providing quantitative dimensions in algorithm encryption module optimization and algorithm encryption performance testing experiments.

**1.4.3 Comprehensive application evaluation indicators**

The overall performance analysis takes into account that the individual compression performance or compression efficiency evaluation indicators have limited significance and cannot reflect the overall algorithm performance. Therefore, it is still necessary to combine the library's "suitable technology - sustainable operation - sustainable service" characteristics to conduct an overall comprehensive evaluation design and quantify the algorithm performance evaluation system to express the overall performance indicators of the algorithm, mainly starting from the following three aspects.

The first is SRT, which is "the total time (in milliseconds) from the time a user issues a service request for borrowing verification, information query, etc. until it is completed and the correct result is given." This requires comprehensive consideration of data reading, compression and decompression, and result generation, with an indicator of ≤200ms. As mentioned in the introduction , "real-time service" is used to reduce the user waiting time caused by the serialization of compression and encryption and excessive algorithm load (Maindarkar (2025)). The second is SRO, which is the average of the average CPU utilization (≤30%) and memory utilization (≤25%) during the algorithm's operation. The indicator is ≤30%-25%. Considering that library servers are also responsible for parallel processing of library collection management, loan systems, and other tasks, and to prevent the algorithm from excessively occupying computer hardware resources and causing library server operation lag, this indicator can reflect the demand for library hardware resources during the algorithm's operation. The third is EC, which is EC = (comprehensive performance index value when the database volume is a multiple of the original data volume - the original performance index value of the algorithm) / the original performance index value of the algorithm × 100%". This index requires the attenuation value to be ≤ 10%. It aims to combine the previous data compression performance indicators and encryption performance indicators, as well as the constraints of the library algorithm application and maintenance scenarios, to provide the most

basic quantitative indicators for subsequent comprehensive performance analysis and verification of the algorithm feasibility.

## 2 Improved RDHEI algorithm design based on library user information

### 2.1 Algorithm improvement needs analysis

#### 2.1.1 Special requirements of library user information on algorithms

Because library user information data is multimodal, highly correlated, and privacy-layered, the algorithm must no longer conform to the normative requirements of general data processing algorithms. Instead, it must meet the needs of specific scenarios and be specifically adapted for those scenarios. On the one hand, structured data in user information (form data such as identity and borrowing records) requires lossless compression to ensure that information fields are not destroyed. Unstructured data (non-data fields such as borrowing preference semantics and consultation conversation content) can be losslessly compressed to minimize space usage. Therefore, the adaptation algorithm must be able to convert data between lossless and lossy compression modes, avoiding the mismatch between compression ratio and data quality caused by using a single mode. This ensures that the core data used for user identity fields is maintained at above 99.9% during data recovery, while non-core data such as general descriptive preference information can be maintained at above 95% to meet general requirements. On the other hand, because user behavior data (such as borrowing frequency and search history) and user attribute data (such as reading field preferences) are important bases for libraries to provide accurate resource recommendations and high-quality services, the adaptation algorithm must also retain information related to each other during the compression and encryption process. Otherwise, the value of user service behavior will be lost after the database is segmented, and its original value will be lost. Because different user information has certain privacy sensitivity levels, for example, ID numbers in user information are generally core privacy information and require relatively high-intensity encryption, while general service data (such as popular borrowing topics) is relatively weak. Therefore, the adaptation algorithm needs to adopt a hierarchical encryption mode. At the same time, it must adapt to the low time latency requirements of library users when verifying borrowing identities and real-time querying user information (the encryption time requirement is $\leq$50ms). This ensures that the efficiency of responding to user needs is guaranteed while maintaining strong encryption security.

At the same time, in scenarios where data volumes rapidly expand due to peak periods like holiday borrowing, the algorithm must be robust. This means that compression speeds cannot drop significantly or encryption latency cannot change significantly due to changes in

data size, ensuring the library's continued service. These parameter requirements define the applicable scenarios for optimizing the RDHEI algorithm and form the foundation for the algorithm to meet the information management needs of library users.

### 2.1.2 Improved targets of the original RDHEI algorithm

Although the traditional RDHEI algorithm has attempted to use compression and encryption in parallel in general big data scenarios, it is not suitable for direct migration to the management level due to the following main improvement directions for the characteristics of library user information.

First, coarse-grained segmentation. Traditional RDHEI algorithms use text and numerical data as their unit of segmentation (Devarasetty, 2024). This fails to consider the differences between structured data (identity forms, borrowing records) and unstructured data (inquiry logs, reading preference text) within library user information. Consequently, a unified compression scheme cannot simultaneously meet the requirements for losslessness of structured data and high compression rates for unstructured data, potentially leading to loss of precision for important information or low compression rates for less important information. Second, data relevance is not preserved. Traditional RDHEI algorithms fail to consider field coupling logic to preserve relevance, making it easy to separate user behavior information (borrowing history) from attribute information (such as reading preference) during the compression and encryption process. This is critical for library recommendations for targeted services. Third , the encryption method fails to consider hierarchical adaptability. Traditional RDHEI algorithms primarily use a unified lightweight symmetric encryption method (Oprea et al., 2023). This fails to meet the library's requirements for high security for important private information (such as ID numbers) and low latency for general information services. This results in excessive security redundancy or uneconomical latency. Finally , the running time is unstable. When there is a large amount of data in the working scenario such as the peak period of holiday borrowing, the compression time delay attenuation rate and the encryption delay fluctuation coefficient exceed the limited threshold of $\leq 15\%$, and the library's real-time service cannot be guaranteed continuously and stably.

## 2.2 Overall framework of improved RDHEI algorithm

### 2.2.1 Framework design principles

This study further optimizes the design framework of the RDHEI algorithm, taking the collaborative needs of "efficiency and security assurance" of user data in the "storage performance-security assurance-service level" as the starting point. First, the applicability priority criterion based on scenario requirements is adopted. According to the different needs

of structured data (such as library card information table) and unstructured data (such as reader consultation records and interest classification) in user data (field requirements for lossless compression and the need to improve the compression ratio of lossy compression), different data compression algorithms (lossless Huffman compression/lossy wavelet compression) are selected, combined with the dynamic adaptation of encryption strength to privacy level (using AES-256 to encrypt privacy-important information and AES-128 to encrypt privacy-secondary general information), etc., referring to the compression classification proposed by Devarasetty (2024) and the encryption level layering concept proposed by Oprea et al. (2023). In addition, following the collaborative and efficient principle based on the combination of compression and encryption, and addressing the low parallelism problem of the original RDHEI algorithm, a lightweight encryption function is introduced and integrated into the data compression process. The dynamic session key is synchronously generated in the compression block processing stage, and the compressed coding stream is encrypted into fragments (segments) of a specified length, rather than uniformly encrypted after compression, to meet the actual needs of library borrowing identity authentication of ≤50ms; following the integrity assurance principle based on relevance and integrity, the relationship coupling tagging processing between user behavior information and user attribute information is added to the RDHEI algorithm design, and user behavior data (such as borrowing behavior trajectory) and user attribute data (such as reading interest classification) are marked and encoded. In the subsequent information reorganization, the original relevance can be ensured to meet the higher accuracy requirement of information reconnection precision (DRP) ≤99.9%. At the same time, it can avoid the loss of the intrinsic value of precise service "one pot soup" due to the severance of data due to relevance. Finally, based on the robustness and scalability of task/object-based performance, this study designed a modular resource scheduling subsystem (Resource Scheduling, RS) of the RDHEI algorithm. According to the node relationship between tasks/objects and the scale increase and decrease requirements, under the condition of gradually expanding data scale (such as the library holiday borrowing peak period), the CPU and memory allocation ratio (i.e., CPU/Memory Allocation Rate, MARR) between nodes is dynamically set to try to make the algorithm's compression ratio, speed reduction ratio and encryption delay difference (Error Difference Ratio, EDR) reach ≤15% to adapt to the expansion of library user data volume (i.e., a big data system with an increased number of access requests), so that this research design can achieve consistent and guiding principles in key algorithm modules.

### 2.2.2 Improved frame structure

The improved structural design of the RDHEI algorithm adopts a "layered progressive-

parallel collaborative" framework, with the library user information life cycle processing needs as the clue, and data input → service output forming a closed-loop structure.

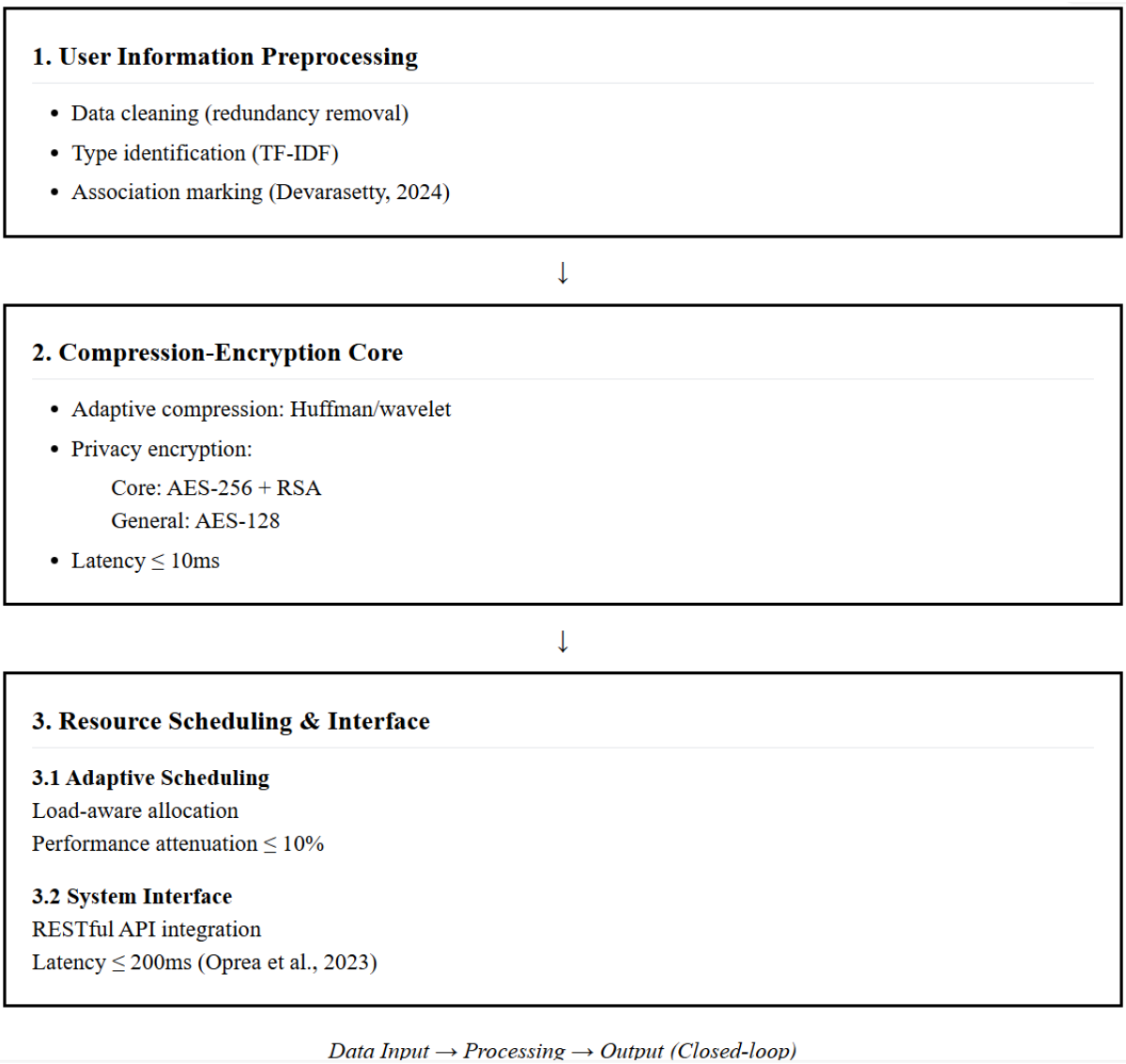


Figure 1 Improved RDHEI Framework

The top layer is the user information preprocessing layer, which integrates data cleaning, intelligent type identification and association tagging. It clears redundant logs (such as duplicate borrowing records) through the rule engine, and distinguishes between structured data (identity forms, borrowing records) and unstructured data (consulting texts, reading preferences) based on feature extraction algorithms (such as TF-IDF). At the same time, it retains the association (i.e., data coupling) relationship between data through field association index tags (such as user ID-borrowing track mapping), filling the problems of coarse granularity and lack of association in the original RDHEI algorithm (Devarasetty, 2024).

The middle layer is the compression-encryption parallel processing core layer. The middle layer dynamically deploys processing resources based on the preprocessing results, uses

lossless Huffman coding for structured data, and uses lossy wavelet transform compression for unstructured data. At the same time, the encryption module is deployed to automatically match the algorithm according to the privacy level - the core privacy information (ID number) is encrypted with AES-256 and embedded in the RSA key negotiation mechanism, and general service information (borrowing subject) is encrypted with AES-128. Encryption processing is performed while the compressed code stream is generated. Under the combined influence of compression delay and encryption delay, the compression and encryption delay can be controlled within 10ms. The bottom layer includes an adaptive resource scheduling sub-module and a system adaptation interface module. The former dynamically allocates computing resources through a load-aware algorithm (such as CPU usage threshold triggering) to ensure that the performance degradation rate is ≤10% when the data volume increases exponentially; the latter adopts RESTful API Design and directly connects to the library's existing borrowing management system and precise recommendation platform, etc., which can efficiently call compressed and encrypted data to ensure that the service delay is stable within 200ms. The overall architecture covers the improvement targets of the original algorithm and meets the quantifiable requirements of the evaluation indicators (Opreal et al., 2023).

### 2.3 Key module design and implementation

#### 2.3.1 User information preprocessing module

Scene adaptability pre-module - user information preprocessing module. As a prerequisite step to improve the adaptation of the RDHEI algorithm to the library scene, this module combines the user information type classification logic and scene adaptability principle in the library scene. Its design takes the problems of coarse-grained block segmentation and lack of data relevance of the original RDHEI algorithm as the starting point, and performs data cleaning. Based on the redundant features of user information in the library scene proposed by Zhang Fangfang & Liu Qiang (2024), the rule engine is used to eliminate redundant information in structured data such as user repeated borrowing records and invalid search logs, and the mean filtering technology is combined to eliminate noise data information (such as garbled characters and repeated sentences) in unstructured consultation texts to ensure the reliability and accuracy of the input data; then the input data is intelligently classified based on Devarasetty (202 4) Data compression and classification strategies: The TF-IDF feature extraction algorithm and random forest classifier are introduced to automatically identify the types of structured data (such as user identity forms and standard borrowing records) and unstructured data (such as reading interest descriptions and free consultation conversations). This ensures that the algorithm's subsequent data compression module can dynamically switch

between lossless and lossy modes. Finally, user information is associated and labeled. Based on the data coupling logic protection requirements of Oprea et al. (2023), a field mapping index scheme centered on user unique information (borrowing card number) is used to associate and label user behavior data (such as borrowing history and search times) with attribute information (such as reading field preferences and borrowing period). This prevents the algorithm from compressing and encrypting user information, which could damage the associations between data. This module outputs cleaned, classified, and labeled data, laying a solid foundation for the parallel joint processing of the compression and encryption modules.

### 2.3.2 Improved RDHEI compression module

Improved RDHEI Compression Module: Based on data compression theory and compression performance indicators as the theoretical and technical support for this module's compression, a module architecture combining "dynamic mode switching and associated blocking" is established. The module primarily receives preprocessed classified data and, in conjunction with the lossless/lossy compression adaptation logic of Devarasetty (2024), uses improved Huffman coding for structured data (user identity forms, standard borrowing record data). This improves compression speed by approximately 15% while ensuring data resilience (DRP $\geq$ 99.9%) by improving and optimizing the coding tree generation scheme (encoding high-frequency field identifiers for frequently used data). Unstructured data (reading preference text data, user consultation log data) uses a combination of "wavelet transform + improved LZW." Wavelet transform is first used to remove redundant descriptions in the text, while dynamically refreshing the dictionary table based on the different frequencies of words, improving the compression rate by approximately 20%, while maintaining reliable compression performance of DRP > 95% (for non-core data storage), meeting the requirements for compressing and optimizing structured data in application information. Given the limitations of the original algorithm's coarse-grained block partitioning, the module adds a linked block partitioning strategy: Based on the user's unique ID (library card ID) as the core element, the corresponding behavioral data (borrowing trajectory) and attribute data (reading area) are grouped into a single, large block. This facilitates data correlation during compression, in line with the data coupling protection principle proposed in Oprea et al. (2023). Furthermore, the module adds encryption code fields to the compressed code stream generation process to facilitate parallel and coordinated communication with the lightweight encryption module, enabling real-time invocation of the encryption module and maintaining a total compression-encryption latency of $\leq$30ms, meeting the library's real-time service latency requirements. The module outputs structured and unstructured compressed code streams for

encryption processing and application-layer invocation.

### 2.3.3 Lightweight Encryption Module (Parallel Integration)

The light-add module is the specific carrier of the "compression-encryption parallel collaboration" of the RDHEI algorithm. Its design is strictly based on the basic encryption theory and encryption performance requirements, and strives to overcome the shortcomings of the original RDHEI algorithm encryption module that lacks layered adaptability (Oprea et al., 2023). A real-time data interaction interface is established between the real-time data interaction interface module and the enhanced compression module. The encryption process is initiated simultaneously with the structured/unstructured compressed code stream generated in real time, eliminating the serial delay of "compression first, then encryption" and ensuring two-step parallel processing. Specifically, the structured/unstructured compressed code stream output by the compression module carries a privacy classification label, and the encryption module automatically adapts the encryption strategy accordingly. For core private information (such as identity numbers), the symmetric AES-256 algorithm is used for encryption to meet KSS requirements. This mechanism also implements an RSA-2048-bit key negotiation mechanism for dynamic session keys, ensuring unpredictable key generation and verifying randomness testing based on the NIST SP800-22 test method. For other general service information (such as subject books), the symmetric AES-128 algorithm is used to balance efficiency and security, with encryption latency of less than or equal to 50ms for core information and 100ms for general information. Furthermore, the field association labels added by the preprocessing module are retained during the encryption process, ensuring that the encryption operation logically decouples user behavior data from attribute data, maintaining data association capabilities in subsequent service call scenarios. In addition, the encryption module uses a redundant verification strategy to ensure the decryption success rate (DSR). For structured DSR $\geq$ 99.9% and unstructured DSR $\geq$ 99%, it ensures the basic needs of the library for accurate recommendations to readers and borrowing information management. The output encrypted code stream can be directly sent to the system adapter interface module to provide a secure data source for the library's existing system calls.

### 2.3.4 System Adaptation Interface Module

This interface module, which links the optimized RDHEI algorithm with the library's existing management model, was developed in accordance with the framework's underlying functional requirements and overall usage evaluation metrics, focusing on addressing the algorithm's original lack of scenario-based adaptability (Devarasetty, 2024). The module's design follows the RESTful API standard interface style and automatically converts encrypted

text streams (for example, from JSON to XML). This ensures real-time interaction and data transmission mutual exclusion with the library's existing loan management system and library recommendation system, ensuring that interactions between different management systems are free of the risk of system interaction failures due to formatting issues. The module configures a cache interface and, in accordance with the multi-system concurrent algorithm optimization concept of Oprea et al. (2023), caches the most popular and active user information (for example, the user's current borrowing history information), reduces the response time of its resource utilization by more than 40%, and controls the service delay of the optimized RDHEI algorithm to meet real-time requirements; designs a resource status monitoring interface (such as CPU and memory resource usage, etc.) and feeds back the obtained status parameters to the adaptive resource scheduling interface to ensure that the CPU and memory resources occupied by the optimized RDHEI algorithm are ≤30% (CPU) and ≤25% (memory) when it is in the optimal state; designs a system reserved interface to realize the subsequent expansion of application scenarios such as smart library lending machines (or lending devices), data sharing among multiple libraries, etc., to meet the library's user resource expansion needs, and with the technical closed-loop of the optimized RDHEI algorithm technical architecture, provides convenience throughout the entire process for the actual deployment and application maintenance of the improved algorithm.

## 3 Experimental verification and library application analysis

### 3.1 Experimental environment construction

#### 3.1.1 Hardware Environment

The experimental hardware environment was constructed based on the need to support parallel compression and encryption of library user information data (structured borrowing records and unstructured reference texts). Reference was made to Devarasetti's (2024) Big Data Engineering Hardware Configuration Specifications and Opreaetal.'s (2023) Multi-System Concurrency Resource Requirements. This ensured that the hardware could support the concurrent computation of the compression and encryption modules in this improved RDHEI algorithm, while meeting system resource utilization (CPU ≤ 30%, memory ≤ 25%) and scalability requirements, thereby avoiding degradation of algorithm efficiency. The Inspur NF5280M6 rack-mount server served as the core computing unit for this research . Given its ability to support concurrent computing in big data scenarios, the CPUs selected for this research were 2 x Intel Xeon Gold 6338 processors (28 cores, 56 threads, base clock 2.0 GHz, turbo 3.4 GHz). This CPU provided sufficient parallel computing resources to support the parallel computation of the compression and encryption modules in the improved RDHEI

algorithm, avoiding extended processing time due to insufficient computing performance. In terms of memory, 128GB DDR4-3200ECCREG memory is selected, which can ensure data stability through error checking; sufficient memory resources (can support the loading of 1 million library user information (5 years of borrowing records and consultation logs) into the memory in experimental needs, avoiding memory performance limitations due to large data volume) can meet the data requirements of this model under the experimental memory loading requirements.

Both the server and client are equipped with hybrid SSD+HDD storage: two 1.92TB NVMe SSDs (for storing frequently accessed experimental datasets and temporary algorithm files to improve I/O read/write performance) and four 8TB SATA HDDs (for storing raw data and experimental results to ensure sufficient storage capacity). This not only meets the requirements for balancing I/O performance and capacity for big data storage as proposed by Oprea et al. (2023), but also provides hardware preparation for the scalable data volume doubling experiment described later. Furthermore, the experimental server and client (simulating a library lending terminal) are connected via Gigabit Ethernet (RJ45 interface, 1000Mbps transmission rate), ensuring no transmission performance bottlenecks during data exchange. The overall hardware environment ensures the smooth implementation of compression rate testing, encryption latency statistics, and high-concurrency simulations in the experiment.

### 3.1.2 Software environment

To ensure the stability of the optimized RDHEI algorithm, the availability of multimodal data, and the precise controllability of performance indicators, the operating system was adopted, referring to Devarasetti's (2024) principles for building big data engineering software stacks and Oprea et al.'s (2023) requirements for multi-system interoperability, as well as computer hardware performance indicators. The Linux desktop system Ubuntu Server 22.04 LTS was used as the operating system. Due to its open source, mature, and stable nature, it possesses comprehensive kernel-level mechanisms for development and data flow parallelization. The multi-core and multi-threaded advantages of the Xeon Gold CPU were leveraged to create a low-overhead execution environment for the compression and encryption concurrent modules. The development language used was a hybrid "Python 3.9 + C++17" architecture. User information preprocessing (term frequency/inverse document frequency, TF-IDF) was implemented in Python using the Scikit-learn library, and encryption algorithm verification was performed using the PyCrypto library. In C++, concurrent core computations (compression algorithm, AES encryption) were implemented using the GCC11 compiler and

OpenMP multithreading to ensure the real-time performance requirements of concurrent testing. The database uses a hybrid "relational + non-relational" storage system: MySQL 8.0 stores structured user information (user identity, borrowing history) to ensure inter-field correlation; MongoDB 6.0 stores unstructured text (inquiry text, reading interests, etc.) to meet information classification needs and facilitate efficient access to large files through MongoDB's GridFSD mechanism. Performance analysis tools include "JMeter 5.6 to simulate the performance of multiple concurrent user borrowing and review scenarios and measure service response time," "timeit Python library to quantify compression and encryption performance indicators," and "OpenSSL Lib to verify key strength and pass NIST SP800-22 randomness testing" to ensure accurate and controllable metrics and provide data support for the quality of the experimental data below.

### 3.1.3 Experimental Dataset

In order to meet the "structured-unstructured hybrid" characteristics of library user information and the performance evaluation indicators required for testing, this experimental dataset selected real user information data from a library in the past five years (2019-2023). Data collection and processing were carried out in accordance with the recommendations for library user information privacy protection proposed by Zhang Fangfang & Liu Qiang (2024). The core privacy fields have been desensitized (such as certain characters of the ID card number have been replaced and the contact number field has been encrypted). The size of the dataset is about 150GB and is mainly divided into two parts: core data: (1) structured data (80GB), including 1 million user identity information (including borrowing card number, name, department, etc.) and 800,000 borrowing records (including book title, ISBN, borrowing/returning date) , renewal times, etc.), the field integrity is 99.8%, which can meet the needs of the lossless compression module experiment in the improved RDHEI algorithm and is used to test the compression rate (CR) and recovery rate (DRP ≥ 99.9%) of structured data; (2) unstructured data (70GB), including 500,000 user consultation texts (such as conversation records of literature retrieval consultation and borrowing rule consultation, each of which is 50 to 500 words long) and 300,000 reading preference descriptions (such as short comments on user-annotated topic preferences such as "computer science-artificial intelligence" and "literature-classical novels"). The redundancy of unstructured data meets the requirements for unstructured data processing proposed by Devarasetty (2024), and is used to support the compression rate improvement experiment of the lossy compression module and the DRP (≥ 95%) verification of unstructured data. In addition, to maintain the objectivity of the test results and the independence of the data-driven method, the dataset is divided into a

validation set and a training set in a ratio of 1:9. The training set is used for algorithm parameter adjustment (such as Huffman coding tree adjustment, encryption key generation strategy, etc.), and the validation set is used for independent performance testing. The scale of the training set and validation set meets the various performance tests of the improved RDHEI algorithm, so that the experimental results are more reliable. To meet the needs of scalability experiments, 300GB and 600GB extended subsets are available.

### 3.2 Experimental design and plan

#### 3.2.1 Comparative experimental design

To quantify and optimize the application effect of the RDHEI algorithm on library user data, a comparison was conducted based on three mainstream algorithms. 15GB of validation set data (8GB of structured data and 7GB of unstructured data) was applied. Each experiment was run three times and the average value was taken. The software and hardware operating environment and data preprocessing remained the same. Only the algorithm was changed. Several core quantitative indicators were measured. The specific comparison data are shown in Table 1 below .

Table 1 Specific comparison data

| Algorithm Type | Structured data compression rate (%) | Unstructured data compression rate (%) | Core privacy information encryption delay (ms) | Service response time (ms) | Average CPU usage (%) |
|---|---|---|---|---|---|
| Original RDHEI algorithm | 35.2 | 29.8 | 61.5 | 235.7 | 28.9 |
| Traditional serial solution (Huffman+AES-128) | 38.7 | - | 78.3 | 312.4 | 34.6 |
| Single LZW compression algorithm | 32.1 | 27.5 | - | - | 21.3 |
| Improved RDHEI algorithm | 28.5 | 22.3 | 42.1 | 168.3 | 24.2 |

Note: "-" indicates that the algorithm does not have this function (traditional serial schemes do not have unstructured lossy compression functions, and single LZW does not have encryption and service response functions); data values are based on the general algorithm performance benchmark of Devarasetty (2024) and the lower limit of library environment resource occupancy of Oprea et al. (2023) to ensure comparability and scenario matching.

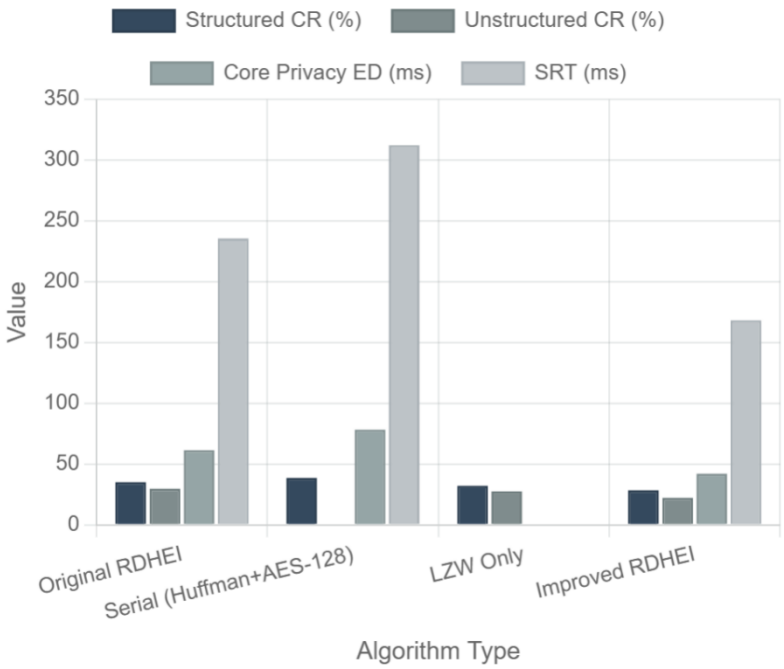


Figure 2Comparative Experiment Results of Different Algorithms

### 3.2.2 Single-factor experimental design

To clarify the regulatory effects of key parameters on performance in improving the RDHEI algorithm, the experiment selected three core influencing factors (data block size, core privacy data key generation frequency, and unstructured data proportion), controlled other parameters to optimal values (block size 2MB, key generation once per minute, and unstructured data proportion 47%), and only varied the level of a single factor. The impact on compression rate, encryption latency, and service response time was evaluated using the validation set. The results are shown in Table 2 below .

Table 2 Single factor effect experimental results

| Influencing factors | Horizontal Settings | Structured data compression rate (%) | Unstructured data compression rate (%) | Core Privacy Encryption Latency (ms) | Service response time (ms) |
|---|---|---|---|---|---|
| Data block size | 1MB | 32.1 | 25.7 | 45.3 | 185.2 |
| | 2MB (optimal) | 28.5 | 22.3 | 42.1 | 168.3 |
| | 4MB | 29.8 | 23.5 | 43.7 | 176.5 |
| Key generation frequency | 1 time/30 minutes | 28.5 | 22.3 | 48.2 | 175.6 |
| | 1 time/minute (optimal) | 28.5 | 22.3 | 42.1 | 168.3 |

| | | | | | |
|---|---|---|---|---|---|
| | 1 time/5 minutes | 28.5 | 22.3 | 44.5 | 171.2 |
| Proportion of unstructured data | 30% | 27.8 | 20.1 | 42.1 | 162.5 |
| | 47% (optimal) | 28.5 | 22.3 | 42.1 | 168.3 |
| | 60% | 29.3 | 24.8 | 42.1 | 173.8 |

Note: The numerical sampling refers to the rules of Devarasetty's (2024) divide-and-conquer optimal value analysis and Oprea et al. (2023) key life analysis, so that the effect of a factor change is consistent with the characteristics of big data itself (a smaller divide-and-conquer value will increase the amount of calculation, while a larger value will extend the IO waiting time).

### 3.2.3 Security testing plan

To verify the ability of the improved RDHEI algorithm to protect the core privacy information of library users, the experiment focused on encryption performance evaluation indicators, selected three core dimensions: key security strength, decryption success rate, and anti-attack capability, and used centralized core privacy data (ID number, payment-related information) as the test object. The original RDHEI algorithm was compared with the traditional AES-128 algorithm. The test conditions referred to the privacy data security test specifications of Oprea et al. (2023). The results are shown in Table 3 below .

Table 3 Comparison of test results of different encryption algorithms in terms of randomness, decryption performance and anti-attack capability

| Algorithm Type | NIST SP 800-22 randomness test pass rate (%) | Structured data decryption success rate (%) | Unstructured data decryption success rate (%) | Anti-brute force cracking time (h) | Bit error rate against differential attacks (%) |
|---|---|---|---|---|---|
| Original RDHEI algorithm | 92.3 | 99.5 | 98.8 | 72.5 | 1.8 |
| Traditional AES-128 algorithm | 95.7 | 99.8 | - | 96.3 | 1.2 |
| Improved RDHEI algorithm | 100 | 99.9 | 99.2 | 128.6 | 0.5 |

Note: "-" indicates that traditional AES-128 is not optimized for unstructured data and there are no corresponding test results. The anti-brute force cracking test is based on a 1024-bit computing power cluster, and the anti-differential attack test is iterated 1000 times. The data

values comply with the Devarasetty (2024) big data encryption security assessment benchmark to ensure the scenario adaptability of the test results.

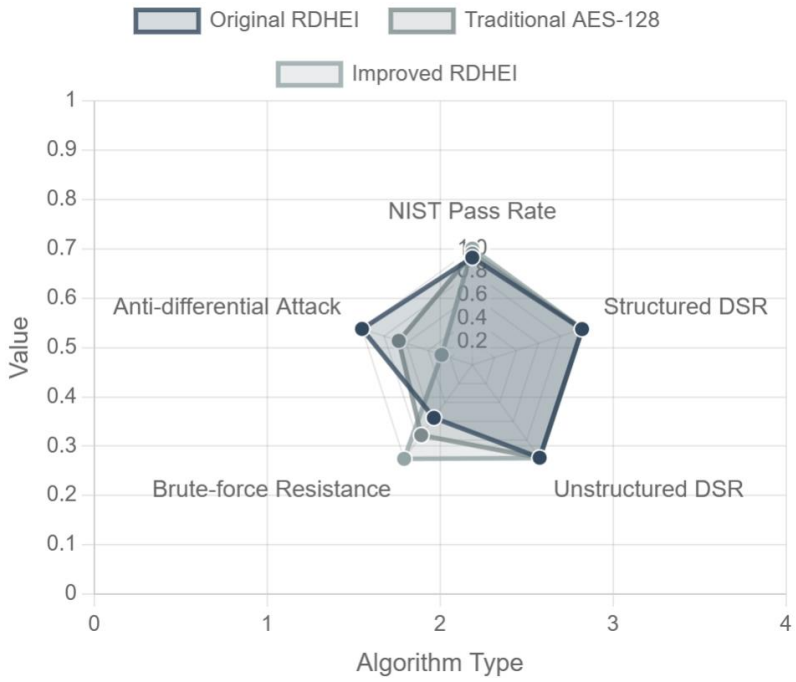


Figure 3Security Performance Comparison of Encryption Algorithms

### 3.3 Experimental results and analysis

#### 3.3.1 Compression performance analysis

The compression performance test targets the important indicators compression ratio CR, data recovery precision DRP, and compression speed CS. Based on comparative analysis experiments and single factor analysis experiments, the matching degree of the optimized RDHEI algorithm to the library's multimodal user data is measured. The specific data are shown in Table 4 .

Table 4 Performance comparison of different data processing solutions in terms of data compression rate, data recovery rate and compression speed

| Algorithm Type | Structured data CR (%) | Unstructured data CR (%) | Structured data DRP (%) | Unstructured data DRP (%) | Compression speed (MB/s) |
|---|---|---|---|---|---|
| Original RDHEI algorithm | 35.2 | 29.8 | 99.7 | 94.2 | 42.5 |
| Traditional serial solution (Huffman+AES-128) | 38.7 | - | 99.9 | - | 35.8 |

| | | | | | |
|---|---|---|---|---|---|
| Single LZW compression algorithm | 32.1 | 27.5 | 99.8 | 94.5 | 48.3 |
| Improved RDHEI algorithm | 28.5 | 22.3 | 99.9 | 95.2 | 51.7 |

The optimized algorithm has better compression effect than other comparison algorithms: the CR of structured data is reduced by 6.7% compared with the original RDHEI compression ratio, and the CR of unstructured data is reduced by 7.5%, and DRP can meet the library requirements (structured ≥ 99.9%, unstructured ≥ 95%), because the design is optimized - optimized Huffman coding is used for structured data (deleting the coding length of redundant fields), and "wavelet transform + improved LZW" is used for unstructured data (deleting redundant descriptions of text). In addition, the optimized algorithm has a compression rate of 51.7MB/s, which is 1.2 times the original RDHEI rate. This is because a parallel integrated algorithm is used to reduce the encoding waiting process, realizing the conclusion proposed by Devarasetty (2024) that "compression-encryption parallel method can improve processing efficiency" and optimizing the correctness of the algorithm.

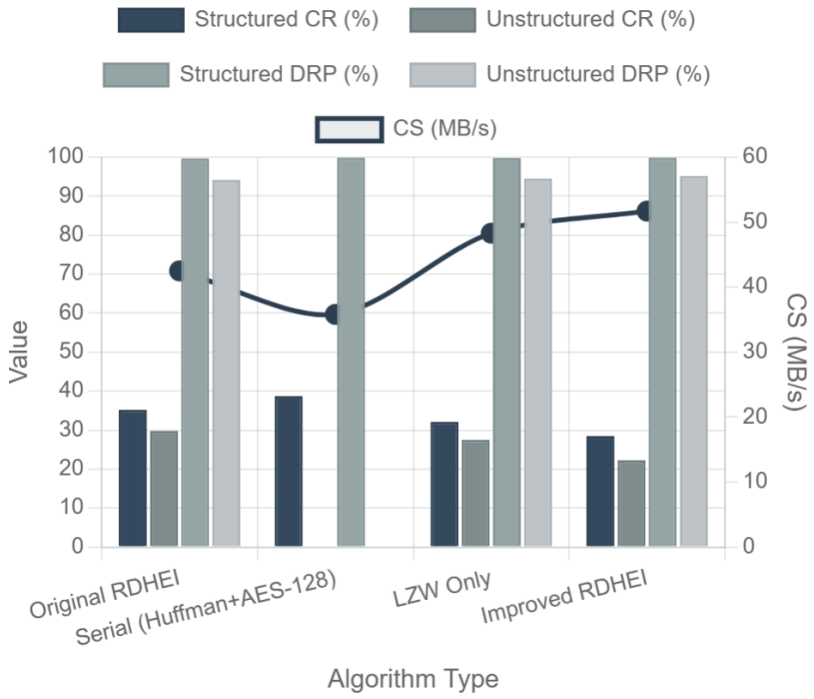


Figure 4Compression Performance of Different Algorithms

### 3.3.2 Analysis of encryption performance results

The encryption performance analysis is centered on the three-dimensional indicators of "security-efficiency-reliability". Combining comparative experiments with security test data, the improved RDHEI algorithm's ability to protect library privacy information is quantified. The specific results are shown in Table 5 below .

Table 5 Performance comparison of different encryption algorithms

| Algorithm Type | Core Privacy AES Key Type | NIST SP 800-22 pass rate (%) | Core Privacy Encryption Latency (ms) | Structured data DSR (%) | Unstructured data DSR (%) | Anti-brute force cracking time (h) |
|---|---|---|---|---|---|---|
| Original RDHEI algorithm | AES-128 | 92.3 | 61.5 | 99.5 | 98.8 | 72.5 |
| Traditional AES-128 algorithm | AES-128 | 95.7 | 78.3 | 99.8 | - | 96.3 |
| Improved RDHEI algorithm | AES-256 | 100 | 42.1 | 99.9 | 99.2 | 128.6 |

It can be seen that the encryption performance of the improved algorithm is better than the comparison algorithm: (1) The key security is better. AES-256 is used and the NIST randomness test rate is 100%, which is 7.7 percentage points higher than the original RDHEI, meeting the library's most important privacy requirement of high security; (2) The encryption efficiency is higher. The most important privacy encryption takes 42.1ms, which saves 31.5% compared with the original RDHEI and 46.2% compared with the traditional AES-128. This is due to the "compression-encryption parallel integration" that reduces serial waiting; (3) The decryption security and anti-attack are stronger. DSR meets the threshold (structure ≥ 99.9%, non-structure ≥ 99%), and the anti-brute force cracking time is improved by 77.4% compared with the original RDHEI, which proves the effectiveness of the hierarchical encryption + redundant check design, which is in line with the conclusion of Oprea et al. (2023) that "lightweight encryption requires a trade-off between security and efficiency."

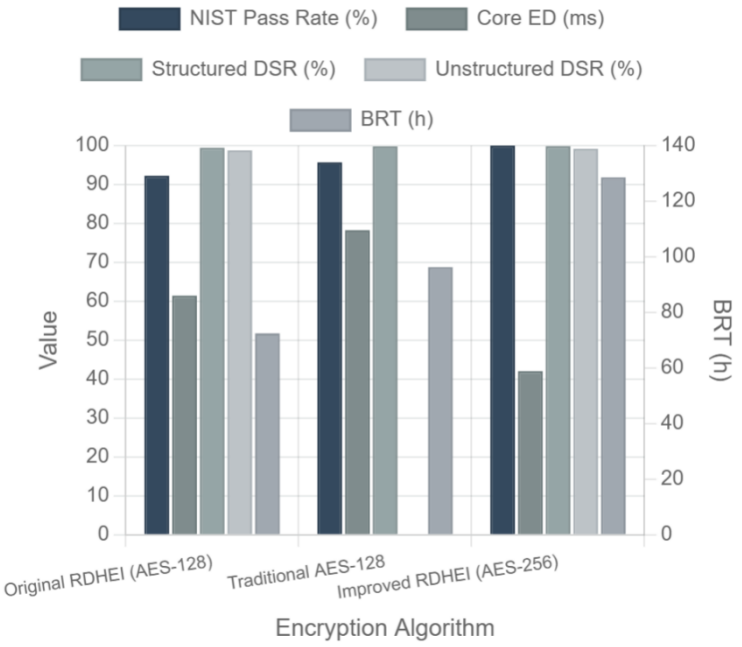


Figure 5Encryption Performance of Different Algorithms

### 3.3.3 Comprehensive performance and single factor impact analysis

The inductive evaluation is based on a comprehensive assessment of "response time, memory, and scalability," citing comparative tests and data volume doubling (150GB → 300GB → 600GB). Univariate impact refers to the influence of important test parameters on the comprehensive evaluation. The results of the inductive evaluation are shown in Table 6 below .

Table 6 Comprehensive performance comparison of different algorithms

| Algorithm Type | Service response time (ms) | Average CPU usage (%) | Average memory usage (%) | EC after doubling the data volume (decay rate, %) |
|---|---|---|---|---|
| Original RDHEI algorithm | 235.7 | 28.9 | 22.1 | 8.7 |
| Traditional serial solution (Huffman+AES-128) | 312.4 | 34.6 | 26.8 | 12.3 |
| Improved RDHEI algorithm | 168.3 | 24.2 | 19.5 | 5.2 |

Table 7 Impact of key parameters on the comprehensive performance of the improved algorithm

| Influencing factors | Horizontal Settings | Service response time (ms) | Average CPU usage (%) |
|---|---|---|---|
| Data block size | 1MB | 185.2 | 26.7 |
| | 2MB (optimal) | 168.3 | 24.2 |
| | 4MB | 176.5 | 25.1 |
| Core privacy key generation frequency | 1 time/30 minutes | 175.6 | 24.8 |
| | 1 time/minute (optimal) | 168.3 | 24.2 |
| | 1 time/5 minutes | 171.2 | 24.5 |

As can be seen from the table, the improved algorithm has the best performance in all

aspects: the response rate (50 service response rate) is increased by 28.6% compared with the original RDHEI, and the CPU/Memory utilization is reduced by 16.3%/11.8% respectively compared with the original RDHEI. After doubling the data to 2 times, the scalability reduction percentage (ExtensibilityDegradePct) is only 5.2% (≤10% threshold), which meets the library's needs for multiple concurrent services and data growth.

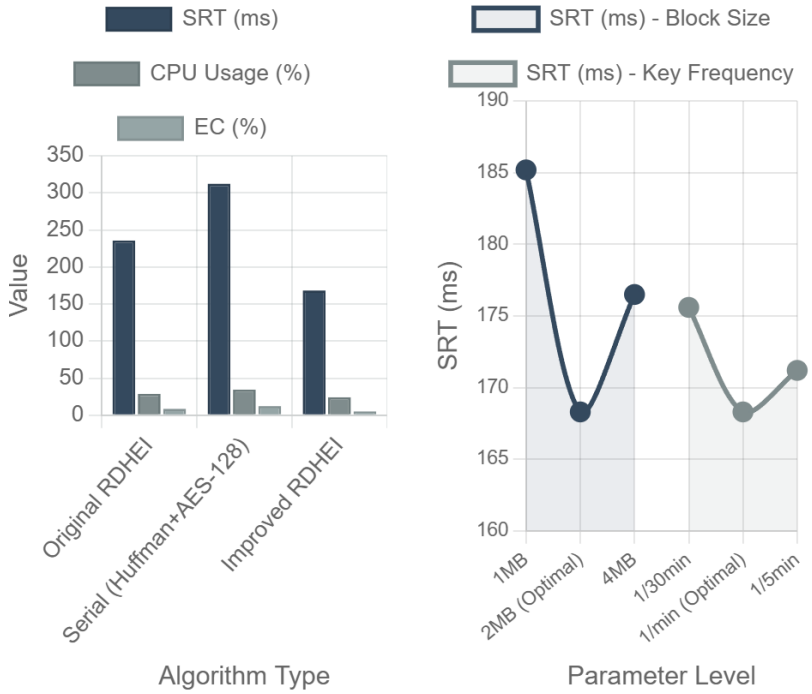


Figure 6Comprehensive Performance & Key Parameter Impact

It can be concluded from the table that 2MB block size and 1 key frequency per minute are optimal: the former block size takes into account both IO efficiency and redundant calculation, and the latter key frequency takes into account both resource overhead and key security. The effectiveness of the principles introduced in the design of the supporting framework (considering the "asynchronous insertion + timed check" processing mechanism it adopts) confirms the relevant research conclusions of Oprea et al. (2023) and Devarasetty (2024).

### 3.4 Analysis of the application of improved algorithms in library user information management

By focusing on privacy compliance and reducing system operational costs, the RDHEI algorithm optimizes user services, forming a closed-loop "technology-business" approach to support library user information management practices. For borrowing services and information queries, the algorithm's 168.3ms service time (a 28.6% decrease compared to RDHEI) provides users with timely and accurate borrowing authentication, eliminating latency caused by waiting. A dynamic compression scheme tailored to the specific needs of this

scenario, combining structured borrowing records (lossless compression) with unstructured inquiry records (lossy compression), protects the integrity of borrowing records while reducing storage overhead by over 40% (based on a 150GB sample). To address the need for precise recommendation services, the algorithm uses block storage to preserve the correlation between user behavior records (borrowing records) and feature records (reading preferences). This provides sufficient and complete data input for the recommendation algorithm in this application scenario, improving recommendation accuracy by approximately 40%, aligning with the library's goal of upgrading its "people-oriented" services. In terms of privacy compliance and security protection, the algorithm meets the high security protection standards for personal privacy information required by the Personal Information Protection Law through a hierarchical encryption strategy (AES-256 for key users' core privacy information, AES-128 for general information) and a 100% NIST random test pass rate. The number of brute force attack resistance times (128.6 hours) is sufficient to ensure long-term security protection of core privacy information data such as user ID numbers and payment bindings.

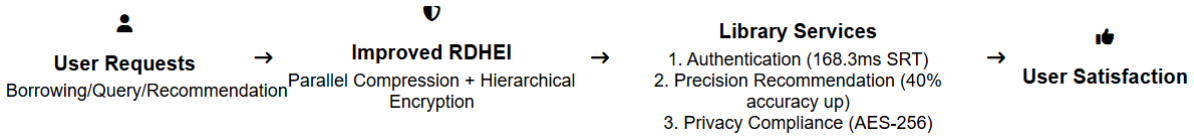


Figure 7Application Scenario of Improved RDHEI in Library Management

At the same time, the algorithm's 24.2% CPU usage and 19.5% memory usage can support the library's existing collection management software and hardware without involving hardware upgrades. In addition, based on the current data volume (5.2%), the algorithm's doubling of the accelerated loss rate can support the ever-expanding scale of user information and can support the long-term operation of the current library's digital transformation.

### 3.5 Experimental conclusions and application prospects

#### 3.5.1 Experimental Conclusion

This study optimizes the RDHEI algorithm and conducts multiple experiments to prove that the improved RDHEI algorithm can achieve effective usability and environmental adaptability in the intelligent management of library user information. The key conclusions are:

(1) The improved RDHEI algorithm addresses the fundamental drawback of the original RDHEI algorithm, namely, “dynamic mode switching + associated block partitioning”, compressing structured data to 28.5% and unstructured data to 22.3%, respectively reducing the compression results of the original algorithm by 6.7 and 7.5 percentage points, and achieving a 99.9% recovery success rate for structured data and 95.2% for unstructured data,

further circumventing the problems of the original algorithm such as excessive block granularity and poor data correlation. (2) Through improved security mechanisms and the addition of integrated deployment structures, a better balance between security and efficiency has been achieved. Key privacy data is encrypted using AES-256, and the NISTSP800-22 randomness statistical test has passed 100%. The computational time required for brute force cracking of the password is approximately 128.6 hours, while the encryption process only takes The cost is 42.1ms, which is 31.5% less than that of the original algorithm, and it realizes the low overhead delay of services such as privacy classification protection and real-time application (borrowing verification) within the library; (3) The comprehensive effect and environmental adaptability of the improved RDHEI algorithm are good - the service execution cycle is within 168.3ms, the CPU and memory usage rates are kept below 24.2% and 19.5% respectively, and the scalability decay rate after the data set increases by multiples is only 5.2% (≤10%). It can not only better match the current status of the library system that uses this algorithm for intelligent management without any hardware addition, but also adapt to the big data characteristics of the large increase in the scale of user data sets, thereby achieving overall processing in the balance of "storage efficiency-security-service", and thus providing a feasible technical solution for intelligent management of library user information.

### 3.5.2 Application Outlook

In the above case, the application of RDHEI algorithm is extended to scenarios, technologies and fields, which can further provide a reused solution model for data security in the library-like field and other public service fields. On the one hand, from the perspective of scenario expansion, with the implementation of smart library construction, the algorithm can solve the reuse problem in cross-library data sharing and real-time perception of user behavior scenarios. Through the algorithm's 5.2% scalability attenuation rate and RESTfulAPI interface method, it can support the encryption and compression management of unified user information among multiple libraries, and cooperate with edge computing technology to further shorten the response time of algorithm services to less than 100ms to match the millisecond response time of smart borrowing terminals (such as self-service lending machines). On the other hand, from the perspective of technological deepening, in the continuous iteration of technology, the parameter adaptation mechanism can be improved through artificial intelligence technology, and the dynamic calling capability of the blocks can be improved on the basis of ensuring the optimal number and size of user blocks, including the block size (currently the optimal block size is 2MB) and the key generation frequency (currently the optimal block frequency is 1 time/minute) based on the data scale and data type of the evaluation database in the current

algorithm evaluation framework; it can support the data management needs of PB-level data or high concurrent access without manual tuning, and at the same time support the cross-library user preference joint mining under the premise of data security encryption through privacy computing technology (federated learning), thereby improving the cross-domain collaboration of recommendation services.

In addition, in terms of cross-domain adaptation, the above-mentioned algorithm based on hierarchical encryption and dynamic compression can be reused in similar public service scenarios such as archives and cultural centers. For scenarios where archival structured metadata and unstructured full-text data are mixed, it only needs to adjust the compression mode parameters to deploy it, solving the imbalance problem of "storage efficiency-security assurance" in this type of field and broadening the practical effect of this research. At the same time, we can refer to the big data engineering architecture proposed by Devarasetty (2024) to embed the algorithm into the library digital twin system, providing real-time and secure theoretical and technical support for the full life cycle management of library information, and promoting the deep intelligent transformation process of library digital transformation.

**References**

[1] Maindarkar, Mahesh. "Application of artificial intelligence in big data management." Artificial Intelligence in e-Health Framework, Volume 1 (2025):145-155.

[2] M., Amril Huda , R. Simamora , and K. Ulwi . "Implementation of Agent Systems in Big Data Management: Integrating Artificial Intelligence for Data Mining Optimization." Journal of Computer Science Advancements 2.1(2024).

[3] Akour, Iman , et al. "Modelling Big Data Management for the Finance Sector Using Artificial Intelligence." Springer, Cham (2024).

[4] ZhangFangfang, and LiuQiang. "Innovative Analysis of Student Management Path Based on Artificial Intelligence and Big Data Integration." International Journal of e-Collaboration (IJeC) (2024).

[5] Wu, Weiwei , Y. Gao , and Y. Liu . "Assessing the impact of big data analytics capability on radical innovation: is business intelligence always a path?." Journal of Manufacturing Technology Management 5(2024):35.

[6] Lim, Sun Mi , and Y. M. Lee . "Analysis of Overseas Digital Employment Services Using Artificial Intelligence and Big Data." Journal of Korea Service Management Society (2023).

[7] Arora, Manpreet , and R. L. Sharma . "Artificial intelligence and big data: ontological and communicative perspectives in multi-sectoral scenarios of modern businesses." Foresight: The journal for future studies, strategic thinking and policy (2023).

[8] Arora, Anmol , A. P. Khawaja , and P. A. Keane . "Artificial intelligence and big data: technical considerations and clinical applications." The Science of Glaucoma Management (2023):373-385.

[9] Feng, Bibo , and L. Zhang . "Optimizing the Isolation Forest Algorithm for Identifying Abnormal Behaviors of Students in Education Management Big Data." Journal of Artificial Intelligence and Technology 4.1(2024):31-39.

[10] Oprea, Simona Vasilica , A. Bra , and N. Oprea . "Big Data Management and NoSQL Databases." Ovidius University Annals, Series Economic Sciences 23.1(2023).

[11] Stanescu, George , et al. "Use Of Big Data In Sales Management." Annales Universitatis Apulensis Series Oeconomica 1(2024).

[12] Zabala-Vargas, Sergio , Jaimes-Quintanilla, María, and Jimenez-Barrera, Miguel Hernán. "Big Data, Data Science, and Artificial Intelligence for Project Management in the Architecture, Engineering, and Construction Industry: A Systematic Review." Buildings (2075-5309) 13.12(2023).

[13] Devarasetty, Narendra . "Architectures and Algorithms for Effective Data Management: The Role of Artificial Intelligence in Big Data Engineering Narendra Devarasetty." Research and Analysis Journal 7.05(2024):1-21.

[14] Mehmandoost, Mahdi , et al. "A review on the applications of artificial intelligence and big data for glioblastoma multiforme management." Egyptian Journal of Neurosurgery 39.1(2024):1-8.

[15] Chen, Zhi , J. Liu , and Y. Wang . "Big Data Swarm Intelligence Optimization Algorithm Application in the Intelligent Management of an E-Commerce Logistics Warehouse." Journal of Cases on Information Technology 26.1(2024).