MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
V.N. Karazin Kharkiv National University
Faculty of Mathematics and Informatics
Department of Theoretical and Applied Informatics

Master's Qualification Thesis

On the topic _ Reengineering the MVC architecture for privacy with using linked data __

_____
_____
_____

Executed by: 2nd-year student,
group MCS-64
specialty 122 «Computer Science»
Educational and research program
«Informatics»

____ Zhiqiang Chen _____
(surname and initials)

Supervisor Dmytro Uzlov

Reviewer Dmytro Chumachenko
_____
(surname and initials)

Adviser: Yurii Parfeniuk

Kharkiv – 2024

Abstract

In the context of rapid development of information technology, relational data, as an emerging paradigm for data organization and utilization, has formed a vast knowledge graph by constructing semantic links between data entities, greatly promoting data sharing, integration, and value mining. However, the widespread use of associated data has also raised serious privacy protection issues. The MVC architecture, as a classic design pattern in the field of software development, has improved the maintainability, scalability, and flexibility of the system. However, in the application scenarios of associated data, its privacy protection mechanism faces many challenges.

This study provides an in-depth analysis of the pathways and hazards of associated data privacy breaches, including privacy exposure at the data processing, storage, transmission, and application levels, as well as serious consequences such as personal privacy infringement, disruption of social trust systems, and damage to national security and public interests. At the same time, this article also points out the shortcomings of the existing MVC architecture in terms of privacy protection, specifically manifested in data processing, user interface design, user interaction behavior, and request processing in the MODEL layer, VIEW layer, and Controller layer. In response to the above issues, this article proposes a privacy preserving MVC architecture reconstruction scheme, which elaborates in detail on the overall architecture design concept and specific reconstruction measures for the MODEL layer, VIEW layer, and Controller layer. This scheme aims to achieve privacy protection of associated data under the MVC architecture through data encryption, access control, privacy processing, and other means. The experimental results show that the new architecture effectively enhances privacy protection while improving system performance. This study not only has

important theoretical value, but also provides a more secure and reliable privacy protection mechanism for the application of associated data, which has a profound impact on the development of information technology.

Keywords: associated data; MVC architecture; Privacy Protection

Table of Contents

1 Introduction..... 7
1.1 Research Background and Significance..... 7
1.2 Review of Domestic and Foreign Research..... 9
1.2.1 Current Research Status in China.....9
1.2.2 Current Research Status Abroad..... 11
1.2 Research Technology Route..... 13
2. Relevant theoretical and technical foundations..... 13
2.1 Linked Data Theory.....13
2.2 Privacy Protection Technologies..... 15
2.3 MVC Architecture..... 17
3. Analysis of privacy risks associated with related data..... 19
3.1 Ways of privacy leakage of associated data..... 19
3.1.1 Leakage during data processing..... 19
3.1.2 Weakness in Data Storage..... 20
3.1.3 Security vulnerabilities in data transmission.....20
3.1.4 Privacy exposure at the application level..... 21
3.2 Hazards of privacy breaches in associated data..... 21
3.2.1 Infringement of Personal Privacy Rights..... 21
3.2.2 Destruction of Social Trust System.....22
3.2.3 Damage to National Security and Public Interest..... 22
4 Shortcomings of Privacy Protection in Existing MVC Architecture.... 23
4.1 Privacy Protection Issues at the Model Layer..... 23
4.1.1 Privacy Leakage Risk in Data Model Design..... 23
4.1.2 Shortcomings in Data Access Control..... 24
4.1.3 Limitations of Data Encryption and Desensitization

Technologies.....24
4.2 Privacy Protection Issues at the View Layer.....25
4.2.1 Privacy Leakage Risk in User Interface Design.....25
4.2.2 Privacy Protection Issues in Data Visualization.....25
4.2.3 Privacy Leakage of User Interaction Behavior.....26
4.3 Privacy Protection Issues at the Controller Layer.....26
4.3.1 Privacy Leakage during Request Processing.....26
4.3.2 Cross site request forgery and session hijacking.....27
4.3.3 Privacy Protection of Response Data.....27
5 MVC Architecture Refactoring Solutions Based on Privacy Protection...28
5.1 Overall Architecture Design Approach.....28
5.1.1 Layered principle and module independence.....28
5.1.2 Selection of Privacy Protection Policies.....29
5.1.3 Integration of Linked Data Management.....29
5.2 Refactoring of Model Layer.....30
5.2.1 Data Encryption and Storage.....30
5.2.2 Access Control and Permission Management.....30
5.2.3 Privacy treatment of associated data.....31
5.3 Refactoring of View Layer.....31
5.3.1 Data anonymization processing.....31
5.3.2 Minimizing Data Exposure.....32
5.3.3 Dynamic adjustment of user privacy preferences.....32
5.4 Refactoring of Controller Layer.....33
5.4.1 Dynamic adjustment of privacy policy.....33
5.4.2 Cross layer Collaboration and Privacy Protection.....33

5.4.3 Exception Handling and Privacy Recovery.....34
5.5 MVC Architecture Pattern Refactoring.....34
6 Implementation and testing of new architecture.....36
6.1 Technical Implementation of New Architecture..... 36
6.1.1 Refactoring Strategy for MVC Architecture.....36
6.1.2 Implementation of Privacy Protection Mechanism.....37
6.1.3 Implementation of Associated Data Processing Technology 38
6.2 Experimental Environment and Data Preparation.....39
6.2.1 Construction of Experimental Environment..... 39
6.2.2 Data Preparation.....39
6.3 Performance testing and evaluation of privacy protection effectiveness..... 40
6.3.1 Performance Testing.....40
6.3.2 Evaluation of Privacy Protection Effectiveness..... 41
6.3.3 Experimental Results and Analysis.....42
7 Conclusion and Prospect..... 43
Reference.....45

1 Introduction

1.1 Research Background and Significance

In the rapidly changing contemporary society of information technology, relational data, as an emerging paradigm for data organization and utilization, is infiltrating and influencing various industries at an unprecedented speed. Linked data creates a vast knowledge graph by constructing semantic links between data entities, greatly promoting data sharing, integration, and value mining, and bringing revolutionary changes to scientific research, business decision-making, social services, and other aspects. However, as the application of associated data continues to deepen, an issue that cannot be ignored is gradually emerging - privacy protection.

In the framework of MVC (Model View Controller) architecture, the issue of privacy protection for associated data is particularly prominent. The MVC architecture, as a classic design pattern in the field of software development, separates business logic, user interface, and input control by dividing the application into three independent and collaborative components: model, view, and controller. This improves the maintainability, scalability, and flexibility of the system. However, in the application scenarios of correlated data, the privacy protection mechanism of MVC architecture faces unprecedented challenges. How to ensure the personal privacy and security of users while utilizing the data advantages brought by associated data has become a key issue that urgently needs to be addressed.

Privacy protection not only concerns the personal rights and interests of users, but also directly affects the legitimacy and credibility of related applications. In the information age, personal privacy has become an important social resource, and its level of protection is an important

indicator of a society's level of civilization. If the privacy protection of associated data is not in place, users' sensitive information may be illegally obtained or abused, leading to personal privacy leakage and causing a series of social problems such as identity theft and fraudulent behavior. In addition, privacy breaches can seriously damage users' trust in applications, affect the promotion and use of applications, and may even trigger legal disputes, dealing a fatal blow to the legitimacy and credibility of related applications. Therefore, conducting research on privacy preserving MVC architecture reconstruction based on associated data has important theoretical value and practical significance. At the theoretical level, this study helps to deepen the understanding of privacy protection mechanisms for associated data, explore new methods and technologies for implementing privacy protection under the MVC architecture, and provide new ideas and perspectives for research in related fields. At the practical level, this study is expected to provide a more secure and reliable privacy protection mechanism for the application of associated data, enhance users' trust in the application, and promote the widespread application of associated data in more fields. Meanwhile, the research on privacy preserving MVC architecture reconstruction based on associated data will have a profound impact on the development of information technology. With the rapid development of technologies such as big data, cloud computing, and artificial intelligence, the application scenarios of associated data will become more extensive, and the demand for privacy protection will become more urgent. This study is expected to provide new security support for the development of information technology, promote the development of information technology towards a more secure and trustworthy direction, and contribute to the construction of a secure and harmonious information society.

1.2 Review of Domestic and Foreign Research

1.2.1 Current Research Status in China

In the field of privacy protection based on correlated data, domestic scholars have conducted extensive and in-depth research in recent years, proposing various effective privacy protection methods and architectures. Pan Xiao et al. (2015) proposed a privacy protection algorithm based on the network Voronoi diagram on road networks. The algorithm achieves privacy protection for sensitive locations by segmenting the road network and constructing a Voronoi diagram, effectively improving the performance of privacy protection. Liang Huichao et al. (2018) designed a privacy protection method for interest point queries in road network environments. This method effectively protects users' privacy information by fuzzifying query locations and using fake location techniques. With the continuous development of big data technology, the construction of big data security and privacy protection technology architecture has also become a research hotspot. Zhang Kai (2018) analyzed the construction of big data security and privacy protection technology architecture, and proposed comprehensive protection strategies including data encryption, access control, and other technologies. Gao Rui et al. (2018) also conducted research on the architecture of big data security and privacy protection technologies, emphasizing the importance of privacy protection in big data applications. Wan Yifei and Su Lantian (2019) further explored the architecture of big data security and privacy protection technology, and proposed a privacy protection scheme based on distributed systems. This scheme effectively reduces the risk of privacy leakage through distributed storage and computing. Tian Hua and He Yi (2020) proposed a big data privacy protection method based on binary correlation graphs. This method constructs a binary correlation

graph and uses graph theory algorithms to protect data privacy, achieving good results.

In a distributed environment, Fan Chengcheng (2021) conducted in-depth research on privacy protection association rule mining and proposed a privacy protection association rule mining algorithm suitable for distributed systems. This algorithm effectively protects user privacy while ensuring mining effectiveness. Peng Jin et al. (2021) proposed a reference architecture to address the security and privacy protection issues in multi-party data fusion mining. This architecture ensures privacy and security during the multi-party data fusion process through techniques such as data anonymization and encryption. Le Junqing (2021) conducted research on key privacy protection technologies for real-time data and proposed a real-time data privacy protection method based on differential privacy, which maintains high availability of data while protecting privacy. Wu Wei (2022) reviewed privacy protection in industrial data association rule mining, summarized the main methods and challenges of current industrial data privacy protection, and proposed future research directions.

In recent years, with the rise of social media and associated data, research on privacy protection based on associated data has also made new progress. Liu Aiqin and Guo Shuqi (2023) conducted research on mining and revealing social media profile information based on association data, and proposed a mining method that combines privacy protection technology to effectively protect the privacy information of social media users. Aruna (2023) conducted research on the temporal thematic associations and evolutionary paths of data privacy risks, and provided strong theoretical support for privacy protection by analyzing the evolutionary patterns of privacy risks. Ouyang Heng and Chen Hongchao (2024) proposed a Gaussian mechanism applicable to

correlated datasets in the field of differential privacy, which effectively protects privacy information in correlated datasets by introducing Gaussian noise. Guo Qiongqiong et al. (2024) proposed a comprehensive privacy protection strategy for data usage and privacy protection in the era of big data. The strategy covers multiple aspects such as data encryption, access control, and privacy protection algorithms, providing strong guarantees for data security and privacy protection in the era of big data.

1.2.2 Current Research Status Abroad

In the field of privacy protection based on correlated data, foreign scholars have also conducted extensive and in-depth research, and proposed various innovative privacy protection technologies and MVC architecture reconstruction methods.

Usha Lawrance Josephine and Nayahi Jesudhasan Jesu Vedha (2021) proposed a parallel clustering anonymization method based on MapReduce framework to address privacy protection issues in big data environments. This method effectively improves the efficiency and accuracy of privacy protection by processing large datasets in parallel, providing new ideas for big data privacy protection. Safaei Yaraziz Mahdi et al. (2022) provided an overview of privacy protection trends in the Internet of Things (IoT), pointing out the current challenges and future development directions facing IoT privacy protection. They emphasized the importance of privacy protection in the Internet of Things and proposed various potential solutions, including encryption technology, differential privacy, etc. Venkata Nagaraju Thatha et al. (2023) proposed an enhanced privacy protection method in the field of online federated learning, which uses an improved Diffie Hellman algorithm to de identify facial images. This method not only protects the privacy of users, but also

ensures the efficiency and accuracy of federated learning, providing a new solution for privacy protection in online federated learning. Vankamamidi S. Naresh and Ayyappa D. (2024) proposed a privacy preserving deep neural network processing method based on homomorphic encryption (PPDNN-CRP) for credit risk assessment in cloud environments. This method utilizes homomorphic encryption technology to process deep neural networks in an encrypted state, effectively protecting users' privacy information. P. Abirami and S Vijay Bhanu (2024) proposed a method for enhancing cloud security using encrypted deep neural networks (crypto DNNs) in trusted environments. Although their work was later marked as a Retraction Note, their proposed concept of encrypted deep neural networks still provides new insights for privacy protection in cloud environments. Kasturi Routray and Padmalochan Bera (2024) proposed a privacy preserving spatiotemporal attribute based encryption method in the field of cloud applications. This method combines spatiotemporal characteristics and attribute based encryption technology to effectively protect sensitive information in cloud data, providing a new technological approach for cloud data privacy protection.

1.2 Research Technology Route

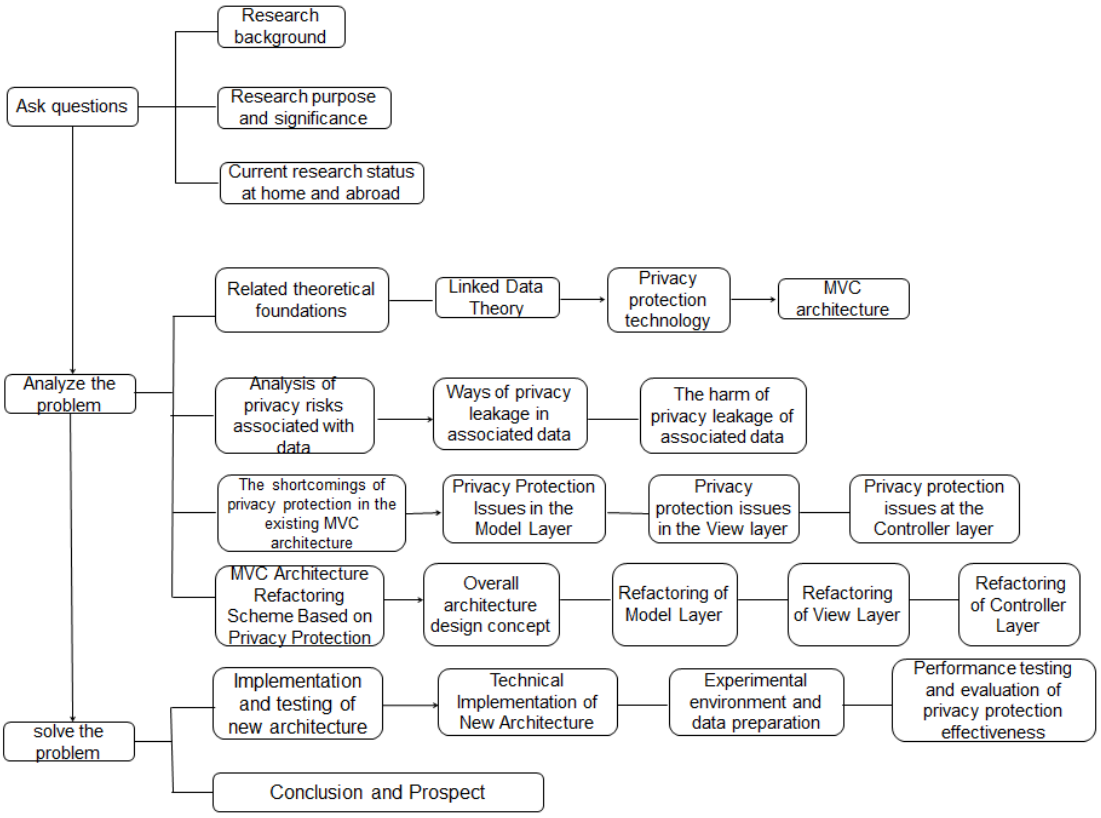

Figure 1-1 Research Technology Route

2. Relevant theoretical and technical foundations

2.1 Linked Data Theory

The concept of Linked Data was first introduced by Tim Berners Lee, the father of the World Wide Web, in his architecture notes, and officially established by Chris Bizer et al. in the Linked Open Data (LOD) project submitted to the Semantic Web Education and Advocacy Institute (SWEO) of W3C in 2007. This theoretical framework not only goes beyond the limitations of traditional hyperlinks connecting documents, but also uses HTTP/URI as the unique identifier for data objects, and uses RDF (Resource Description Framework) to structurally describe these

data objects. The use of HTTP/URI not only ensures the global uniqueness and associativity of data objects, but also provides convenient access channels; The RDF data model ensures the consistency of data object structure, laying a solid foundation for data interoperability and sharing.

The four basic principles proposed by Tim Berners Lee at the TED Conference in 2009 have become important guidelines for guiding the practice of correlated data in the theoretical system of correlated data. These principles have been widely recognized in both academic research and practical applications, although they are not mandatory but recommended best practices. Specifically, these principles include:

1、 Using URIs to identify and name all things is the primary requirement of the Semantic Web, ensuring that all resources on the World Wide Web can be located through globally unique URIs.

2、All valid URIs can access data objects through the HTTP protocol, which is considered a substantive standard of the Semantic Web and provides great convenience for obtaining associated data.

3、 When accessing a dataset through HTTP/URI or SPARQL, the returned data should have clear semantics, ensuring the readability and availability of the data.

4、The data object should contain links to other URIs to discover more related data. This principle emphasizes the correlation between resources and supports association retrieval from individual data objects to the entire data network.

These four principles together constitute the technical system of associated data, providing theoretical support for the reconstruction of privacy preserving MVC architecture. In the context of privacy protection, these principles of associating data help ensure the uniqueness, accessibility, readability, and correlation of data, providing

important guidance for building a secure, efficient, and scalable MVC architecture. By following these principles, data sharing and interoperability can be achieved while protecting user privacy, providing a solid theoretical foundation for the reconstruction of privacy preserving MVC architecture based on associated data.

2.2 Privacy Protection Technologies

In this study, privacy protection technology occupies a crucial position. Traditional privacy protection technologies, such as encryption and anonymization, provide a solid theoretical foundation for the secure transmission and storage of data.

Encryption technology, as an important means of privacy protection, effectively prevents data theft and tampering during transmission and storage by converting sensitive information into difficult to interpret ciphertext forms. This type of technology is mainly divided into two categories: symmetric encryption and asymmetric encryption. Symmetric encryption uses a single key to encrypt and decrypt data, which is easy to operate and highly efficient. However, key management is complex and security depends on the confidentiality of the key. Asymmetric encryption uses a pair of public and private keys for encryption and decryption. Although the encryption and decryption process is relatively complex, its key management is flexible, and the public key can be made public, enhancing the security of data transmission.

Anonymization technology is the process of removing or replacing personal identification information from a dataset, making it impossible for data to be directly associated with specific individuals, thereby protecting personal privacy. However, with the development of data processing technology, even anonymized datasets may be able to recover

personal identities through reverse engineering and other means, thus posing serious challenges to traditional privacy protection technologies.

Among emerging privacy protection technologies, differential privacy technology stands out with its unique advantages. Differential privacy disrupts the direct connection between data points by adding random interference information to the data, while preserving the statistical properties of the dataset, making it difficult for criminals to infer personal identity through the model's output. This method ensures the anonymity of data and allows for the construction of accurate group level models, providing strong support for data analysis. Differential privacy technology has been successfully applied in the release of US census data, ensuring personal privacy security while providing valuable demographic data.

In addition, emerging privacy protection technologies include federated learning, secure multi-party computation, homomorphic encryption, and trusted execution environments. Federated learning allows AI models to be trained on data from different devices or servers without the need for centralized data, achieving the privacy protection concept of "sharing models rather than sharing data". Secure multi-party computation allows different users to handle data they do not want to share, and protects the privacy of data owners through encrypted data sharing and computation. Homomorphic encryption technology allows data to be processed in an encrypted state without the need for decryption, providing new possibilities for data privacy protection. A trusted execution environment creates a secure zone on computing devices that can perform specific functions without leaking data, further enhancing the effectiveness of privacy protection.

2.3 MVC Architecture

In the field of software development, the Model View Controller (MVC) architecture pattern, as a classic and widely used design paradigm, aims to achieve clear division and efficient management of the internal structure of applications. The MVC architecture effectively isolates business logic, data management, and user interface presentation by dividing the core components of a software system into three independent yet closely collaborative levels - Model, View, and Controller. This architectural pattern not only enhances the maintainability and scalability of the system, but also greatly promotes code reusability, laying a solid foundation for flexibility and efficiency in the software development process.

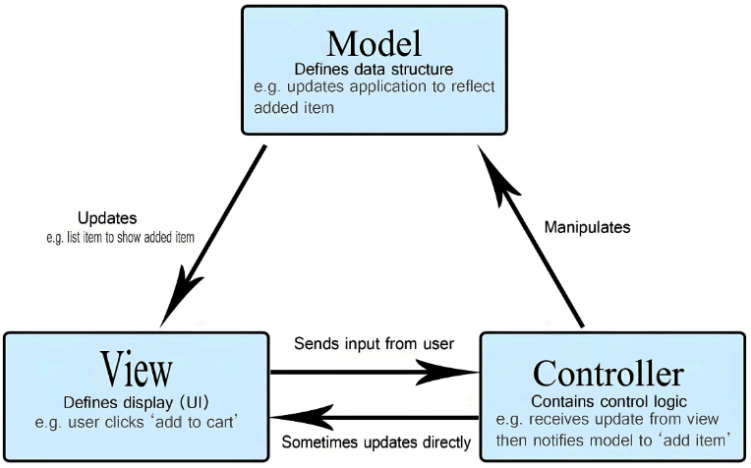

Figure 2-1 Interaction diagram of various parts in MVC architecture

The Model layer plays a central role in data management and business logic processing in the MVC architecture. It is responsible for encapsulating the core business rules, data processing logic, and data persistence mechanism of the application. The design of the model layer aims to ensure data integrity and consistency, while providing a standardized interface for interaction between the view layer and the controller layer. It is worth noting that the design of the model layer does not rely on specific view representations or controller logic. This

decoupling feature allows the model layer to be independent of the user interface and process control, making it easier for unit testing and performance optimization. In addition, when the data state in the model changes, the model layer follows the observer pattern and sends notifications to registered views and controllers to trigger corresponding interface updates or business logic execution. This mechanism greatly enhances the responsiveness and real-time performance of the system.

The View layer focuses on rendering the user interface and optimizing the interactive experience. It presents the data provided by the model layer to users in an intuitive and easy-to-use way through reasonable layout design and information presentation. In the MVC architecture, the view layer should not contain any business logic processing code, but should focus on visualizing data and responding to user interactions. To achieve dynamic updates of data, the view layer needs to register listeners with the model layer to receive notifications and update the interface in a timely manner when data changes. This design not only improves the responsiveness of the user interface, but also ensures consistency and accuracy in data display.

The Controller layer, as the coordinator in the MVC architecture, is responsible for receiving user input requests and calling corresponding model layer methods or updating view layer states based on these requests to achieve user intent response. The controller layer is not only responsible for handling user interaction events, but also for coordinating the data flow between the view layer and the model layer, ensuring that the process control logic of the application is executed correctly. Through the intermediary role of the controller layer, the MVC architecture achieves loose coupling between the user interface and business logic, enabling the system to flexibly respond to changes in user needs and the evolution of business logic.

In summary, the MVC architecture provides strong structural support for privacy protection systems based on associated data through clear division and efficient collaboration at the three levels of model, view, and controller. It not only enhances the modularity and maintainability of the system, but also provides a more flexible and reliable framework foundation for the implementation of privacy protection policies and the maintenance of data security.

3. Analysis of privacy risks associated with related data

In the research of privacy protection MVC (Model View Controller) architecture reconstruction based on associated data, privacy risk analysis of associated data is an indispensable part. Linked data, with its unique semantic interconnectivity, not only promotes information exchange and sharing, but also exposes the vulnerability of data privacy protection. This chapter aims to explore in depth the pathways of privacy breaches in related data and the harm they bring, providing a theoretical basis for subsequent architecture reconstruction.

3.1 Ways of privacy leakage of associated data

The ways of privacy leakage of associated data are diverse and complex, involving multiple links such as data processing, storage, transmission, and application.

3.1.1 Leakage during data processing

The risk of data privacy leakage is particularly prominent in the process of creating and integrating associated data. On the one hand, data

cleaning and standardization operations may inadvertently expose sensitive information in raw data, such as personal identity information, health status, etc. On the other hand, the semantic annotation process of associated data enhances the comprehensibility and interoperability of data by adding metadata or ontology information, but at the same time provides attackers with more potential information leakage points. For example, by analyzing the attribute relationships in the ontology, attackers may infer implicit connections between data entities, thereby infringing on personal privacy.

3.1.2 Weakness in Data Storage

Linked data is usually stored in RDF (Resource Description Framework) format, which is a structured data model that is easy to query and analyze, but also faces storage security challenges. On the one hand, if RDF storage systems (such as SPARQL endpoints) do not implement strict access control, it may lead to unauthorized access and data leakage. On the other hand, the inadequacy of data backup and recovery mechanisms may also increase the risk of privacy breaches in the event of data loss or damage. In addition, the physical security of storage media cannot be ignored. If the storage device is stolen or damaged, the associated data may face the risk of direct leakage.

3.1.3 Security vulnerabilities in data transmission

The transmission of associated data usually relies on the network, and security vulnerabilities in the network transmission process are another important way of privacy leakage. Although protocols such as HTTP and HTTPS to some extent ensure the security of data transmission, there is still a risk of being attacked by man in the middle, data eavesdropping, or tampering. Especially in scenarios where data exchange is frequent, such

as using RESTful APIs for data access, if effective encryption and authentication mechanisms are not implemented, attackers may intercept or forge requests to obtain or tamper with associated data in transmission.

3.1.4 Privacy exposure at the application level

The widespread use of associated data in various applications also provides more opportunities for privacy breaches. In application scenarios such as intelligent recommendation and data analysis, correlated data is often used to mine sensitive information such as user behavior patterns and preferences. If these applications do not follow strict privacy protection policies or have security vulnerabilities, user privacy will be easily compromised. In addition, with the integration and application of technologies such as the Internet of Things and big data, the combined analysis of associated data and other data sources may further exacerbate the risk of privacy breaches.

3.2 Hazards of privacy breaches in associated data

The harm caused by the leakage of associated data privacy is far-reaching and widespread, not only affecting individual rights, but may also pose a threat to social stability and security.

3.2.1 Infringement of Personal Privacy Rights

The most direct and significant harm is the infringement of personal privacy rights. Once sensitive information in associated data is leaked, it may cause disruptions to personal life, such as spam emails, phone harassment, and even serious problems such as identity theft and fraud. In addition, the leakage of personal privacy may also affect an individual's social evaluation, career opportunities, etc., causing incalculable mental

and economic losses.

3.2.2 Destruction of Social Trust System

The leakage of associated data privacy may also cause damage to the social trust system. In a data-driven society, transparency and openness of personal information are seen as the foundation for promoting cooperation and trust. However, frequent privacy breaches will seriously weaken the public's trust in data sharing, exacerbate the phenomenon of data silos, and hinder the flow of information and the creation of value. In addition, privacy breaches may also trigger public dissatisfaction and questioning of data processing entities such as government agencies and enterprises, further exacerbating the crisis of social trust.

3.2.3 Damage to National Security and Public Interest

At a more macro level, privacy breaches of associated data may also pose a threat to national security and public interests. The leakage of sensitive information, such as military secrets, economic intelligence, etc., may be exploited by hostile forces and have a serious impact on national security. At the same time, the leakage of personal privacy may also trigger social instability factors, such as group incidents, terrorist attacks, etc., threatening public safety and order. In addition, the widespread application of related data in fields such as public health and environmental protection may lead to public distrust of policy implementation and public services if privacy protection is inadequate, thereby affecting the overall well-being of society.

4 Shortcomings of Privacy Protection in Existing MVC Architecture

In today's rapidly changing information technology, MVC (Model View Controller) architecture, as a classic paradigm in software design, is widely used in various information systems. However, in the context of correlated data, the existing MVC architecture exhibits significant limitations in privacy protection, mainly reflected in multiple aspects of the Model layer, View layer, and Controller layer. This chapter aims to deeply analyze these shortcomings and lay a solid foundation for proposing privacy preserving MVC architecture reconstruction strategies based on associated data in the future.

4.1 Privacy Protection Issues at the Model Layer

The Model layer, as the data processing core in the MVC architecture, undertakes important responsibilities such as data access and business logic implementation. However, in the context of correlated data, privacy protection at the Model layer faces multiple challenges.

4.1.1 Privacy Leakage Risk in Data Model Design

At the Model layer, the design of the data model is directly related to the organization and storage structure of the data. Traditional data models often overlook the need for privacy protection and fail to effectively isolate sensitive information from non sensitive information. Especially in relational data models, the complex relationships between data entities increase the risk of privacy breaches. Attackers can infer hidden connections between data entities and obtain sensitive information by analyzing the correlation paths in the data model. In addition, metadata in

the data model (such as attribute definitions, data types, etc.) may also leak semantic information of the data, providing convenience for privacy violations.

4.1.2 Shortcomings in Data Access Control

Data access control is a key aspect of privacy protection at the Model layer. However, the data access control mechanisms in the existing MVC architecture are often too simple to meet the complex requirements of the associated data environment. On the one hand, the formulation of access control policies lacks flexibility and is difficult to adapt to the privacy protection requirements of different data entities. On the other hand, there are loopholes in the implementation of access control mechanisms, such as improper permission allocation and incomplete authentication mechanisms, which may lead to unauthorized access and data leakage.

4.1.3 Limitations of Data Encryption and Desensitization Technologies

Data encryption and anonymization are important means of protecting sensitive data. However, there are limitations to the application of these technologies at the Model layer. On the one hand, although data encryption technology can protect the confidentiality of data, the encrypted data is limited in terms of querying, analysis, and other aspects, which affects the availability and efficiency of the data. On the other hand, although data anonymization technology can reduce the risk of sensitive data leakage, the anonymized data may still retain some private information, and the anonymization process may introduce new errors and uncertainties.

4.2 Privacy Protection Issues at the View Layer

The View layer, as the user interface layer in the MVC architecture, is responsible for displaying and interacting with data. However, in the context of correlated data, the privacy protection of the View layer also faces many challenges.

4.2.1 Privacy Leakage Risk in User Interface Design

User interface design is an important aspect of privacy protection in the View layer. However, the user interface design in the existing MVC architecture often overlooks the need for privacy protection. On the one hand, the layout and presentation of the user interface may leak sensitive information about the data. For example, through interactive elements in the user interface (such as buttons, links, etc.), attackers can infer the existence of data entities and their associated relationships. On the other hand, the feedback mechanism of the user interface (such as error prompts, status information, etc.) may also leak privacy information of data.

4.2.2 Privacy Protection Issues in Data Visualization

Data visualization is one of the important functions of the View layer. However, in the context of correlated data, privacy protection issues in the process of data visualization cannot be ignored. On the one hand, data visualization technology may expose sensitive details of data, such as data distribution and trends displayed through charts, maps, etc., which may be exploited by attackers to infer sensitive information. On the other hand, data aggregation, dimensionality reduction, and other operations during data visualization may introduce the risk of privacy breaches. For example, by aggregating information from multiple data entities,

attackers may reveal the relationships between data entities and obtain sensitive information.

4.2.3 Privacy Leakage of User Interaction Behavior

User interaction behavior is another important aspect of privacy protection in the View layer. However, existing MVC architectures often fail to effectively monitor and manage privacy leakage risks in user interaction behavior. On the one hand, input data during user interaction (such as query keywords, input fields, etc.) may leak users' privacy information. On the other hand, user interaction behavior itself (such as clicking, browsing, etc.) may also be exploited by attackers to infer sensitive information such as user preferences and interests.

4.3 Privacy Protection Issues at the Controller Layer

The Controller layer, as the control layer in the MVC architecture, is responsible for receiving user requests, calling the Model layer for data processing, and returning the processing results to the View layer for display. However, in the context of correlated data, there are also many shortcomings in the privacy protection of the Controller layer.

4.3.1 Privacy Leakage during Request Processing

At the Controller layer, privacy leakage during request processing is an important issue. On the one hand, user requests may contain sensitive information such as personal identification information, query conditions, etc. If the Controller layer fails to effectively process this information, it may result in sensitive information being leaked during transmission and storage. On the other hand, the Controller layer may call sensitive data in the Model layer when processing user requests, and if strict access control

is not implemented, it may also lead to data leakage.

4.3.2 Cross site request forgery and session hijacking

Cross site request forgery (CSRF) and session hijacking are another important challenge for privacy protection at the Controller layer. In a correlated data environment, users may access the system through multiple channels, such as web pages, mobile applications, etc. If the Controller layer fails to effectively prevent CSRF attacks and session hijacking, attackers may use the user's legitimate identity to carry out malicious operations, such as modifying user data, initiating unauthorized data access, etc., thereby infringing on user privacy.

4.3.3 Privacy Protection of Response Data

The Controller layer also needs to consider privacy protection issues when returning response data. On the one hand, the response data may contain sensitive information such as query results, status information, etc. If the Controller layer fails to properly process or desensitize this information, it may result in sensitive information being leaked to unauthorized users. On the other hand, the format and transmission method of response data may also affect the effectiveness of privacy protection. For example, using plaintext to transmit sensitive information may result in data being intercepted and leaked during transmission.

In summary, the existing MVC architecture has significant shortcomings in privacy protection at the Model layer, View layer, and Controller layer. These shortcomings not only limit the application effectiveness of MVC architecture in correlated data environments, but may also pose a serious threat to user privacy and social security. Therefore, in future research, we need to conduct in-depth analysis and improvement on these shortcomings, and propose a privacy preserving

MVC architecture reconstruction strategy based on associated data to better safeguard user privacy and data security.

5 MVC Architecture Refactoring Solutions Based on Privacy Protection

5.1 Overall Architecture Design Approach

When building a privacy preserving MVC (Model View Controller) architecture based on associated data, the primary task is to establish a comprehensive design concept that ensures data privacy, system maintainability, and scalability. This chapter will elaborate on this design concept, including the layering principle of architecture, the selection of privacy protection policies, and the integration method of associated data management.

5.1.1 Layered principle and module independence

The core of MVC architecture lies in its layered design, which improves the flexibility and maintainability of the system by dividing the application into three independent but collaborative components: Model, View, and Controller. In the context of privacy protection, this layering principle needs to be further refined to ensure that each layer can effectively handle privacy data while maintaining its independence and functional integrity. Specifically, the model layer is responsible for data storage and business logic processing, and needs to strengthen data encryption and access control; The view layer is responsible for displaying data and user interaction, and needs to implement anonymization and minimize data exposure; The controller layer serves as a bridge, coordinating communication between the model and the

view, while dynamically adjusting privacy policies.

5.1.2 Selection of Privacy Protection Policies

The selection of privacy protection policies is a key aspect of overall architecture design. Considering the characteristics of correlated data, namely the complex relationships between data, the strategy needs to balance the integrity, availability, and privacy of the data. By adopting advanced technologies such as differential privacy, homomorphic encryption, and attribute based access control (ABAC), these requirements can be effectively balanced. Differential privacy protects personal privacy by adding noise to the raw data while maintaining the statistical properties of the data; Homomorphic encryption allows computation to be performed on encrypted data without decryption, thereby protecting the privacy of data during transmission and processing; ABAC dynamically grants access permissions based on user attributes such as roles, responsibilities, etc., ensuring that only authorized users can access sensitive data.

5.1.3 Integration of Linked Data Management

The introduction of associated data has brought new challenges to the MVC architecture, namely how to effectively manage and utilize the associated information between data while protecting privacy. To this end, it is necessary to design a graph database based associated data management system that can support complex data association queries and integrate privacy protection mechanisms such as data anonymization and association hiding. In addition, utilizing semantic web technologies such as RDF (Resource Description Framework) and SPARQL (SPARQL protocol and RDF query language) can achieve semantic representation and efficient querying of associated data, further enhancing the

intelligence level of data processing.

5.2 Refactoring of Model Layer

As the data core in the MVC architecture, the reconstruction of the Model layer is crucial for privacy protection implementation. This section will explore how to introduce privacy protection mechanisms at the Model layer, including data encryption, access control, and privacy processing of associated data.

5.2.1 Data Encryption and Storage

At the Model layer, homomorphic encryption technology is used to encrypt and store sensitive data, ensuring the security of the data at the physical storage level. Homomorphic encryption allows for addition and multiplication operations on data without decryption, thereby supporting the execution of complex business logic on encrypted data. In addition, combining distributed storage technologies such as Hadoop HDFS or Amazon S3 can achieve high availability and fault tolerance of data, while dispersing data storage locations and reducing the risk of single point of leakage.

5.2.2 Access Control and Permission Management

To achieve fine-grained access control, the Model layer needs to integrate an Attribute Based Access Control (ABAC) model. This model dynamically determines the user's access permissions to data based on their attributes such as department, position, project participation, etc. By defining attribute policies and rules, the access scope of data can be flexibly controlled, effectively preventing unauthorized access. In addition, by combining blockchain technology, it is possible to record the logs of every data access, ensuring the traceability of data access history

and enhancing the transparency and security of the system.

5.2.3 Privacy treatment of associated data

For associated data, the Model layer needs to implement association hiding and data anonymization techniques. Association hiding protects the privacy of data subjects by removing or blurring direct associations between data. For example, in social network data, the risk of personal information leakage can be reduced by removing direct connections between users and using indirect associations based on similar interests. Data desensitization refers to the transformation or replacement of sensitive data, making it difficult to directly identify an individual's identity while retaining the original data characteristics. For example, offsetting dates or blurring geographic locations to protect users' spatiotemporal privacy.

5.3 Refactoring of View Layer

As the interface between users and applications, the reconstruction of the View layer should focus on the dual goals of user experience and privacy protection. This section will discuss how to implement data anonymization, minimize data exposure, and dynamically adjust user privacy preferences at the View layer.

5.3.1 Data anonymization processing

At the View layer, anonymizing the data presented to users is an important measure to protect user privacy. By replacing the user identifier with a randomly generated anonymous ID, or using anonymization techniques such as k-anonymity and l-diversity, direct leakage of user identity can be effectively prevented. In addition, for the display of

sensitive information such as geographic location, age, etc., range queries or fuzzy displays can be used to reduce the exposure of precise information.

5.3.2 Minimizing Data Exposure

The principle of minimizing data exposure requires the View layer to display only the minimum amount of data required to complete a specific task. By optimizing interface design and reducing unnecessary display of personal information, the risk of privacy leakage can be effectively reduced. For example, in social media applications, only publicly shared content by users is displayed, rather than all information on their personal profile; In e-commerce websites, only the basic information of products and user reviews are displayed, rather than the user's purchase history or payment information.

5.3.3 Dynamic adjustment of user privacy preferences

The View layer also needs to support dynamic adjustment of user privacy preferences. By providing privacy settings options, users are allowed to adjust the visibility and scope of data usage according to their own needs. For example, users can choose whether to allow the application to collect their location information or whether to allow the data to be used for personalized recommendations. At the same time, the View layer should provide real-time feedback on the impact of privacy settings, helping users understand the consequences of their privacy decisions and make more informed choices.

5.4 Refactoring of Controller Layer

As the coordinator in the MVC architecture, the Controller layer needs to ensure flexible execution and efficient management of privacy policies during its refactoring. This section will explore how to implement dynamic adjustment of privacy policies, cross layer collaboration, and exception handling mechanisms at the Controller layer.

5.4.1 Dynamic adjustment of privacy policy

The Controller layer needs to have the ability to dynamically adjust privacy policies based on user behavior, contextual information, and changes in privacy policies. By integrating intelligent decision-making systems, such as machine learning based privacy risk assessment models, real-time analysis of user data privacy risks can be conducted, and privacy settings can be automatically adjusted to meet different privacy needs. For example, when users use sensitive features, the privacy protection level is automatically raised, or when potential data leakage risks are detected, alerts are triggered and corresponding protective measures are taken.

5.4.2 Cross layer Collaboration and Privacy Protection

The Controller layer serves as a bridge between the Model and View, requiring close collaboration across layers to ensure consistency and effectiveness of privacy protection policies. By defining clear interface protocols and data exchange standards, the Controller layer can coordinate the data processing of the Model layer with the data display of the View layer, while ensuring seamless integration of privacy protection mechanisms in the data flow process. For example, during the data query process, the Controller layer can dynamically adjust the query parameters

based on the user's privacy preferences to ensure that only data results that meet privacy requirements are returned.

5.4.3 Exception Handling and Privacy Recovery

The Controller layer also needs to establish a comprehensive exception handling mechanism to address potential privacy breaches. By monitoring the operating status and data access logs of the monitoring system, abnormal behavior can be detected and responded to in a timely manner. Once a privacy breach event is detected, the Controller layer should immediately trigger a privacy recovery process, such as revoking data access permissions, deleting leaked data, etc., to minimize the impact of privacy breaches. At the same time, recording the event handling process provides a basis for subsequent security audits and privacy improvements.

5.5 MVC Architecture Pattern Refactoring

MVC architecture pattern refactoring is a refactoring based on the recognition results of MVC architecture pattern. This refactoring aims to identify and correct elements that do not comply with the application specifications of MVC architecture pattern, in order to ensure the standardization and efficiency of the architecture. According to the application specifications of MVC pattern, generate corresponding refactoring requirements, which are the basis for guiding developers to implement refactoring. In specific implementation, the refactoring plan covers the acquisition of refactoring points, suggestions for modifying refactoring points, and the verification process after modification. It is worth noting that although prototype tools can automatically generate refactoring points and modification suggestions, the implementation and

validation of refactoring still need to be done manually to ensure the accuracy and effectiveness of refactoring. To achieve flexible refactoring of the MVC architecture pattern, this article adopts a strategy pattern to generate corresponding strategies for each refactoring requirement. The application of this pattern not only enhances the flexibility of refactoring, but also facilitates the dynamic addition and deletion of subsequent refactoring requirements, as shown in Figure 5-1, demonstrating the application framework of the strategy pattern in MVC architecture refactoring.

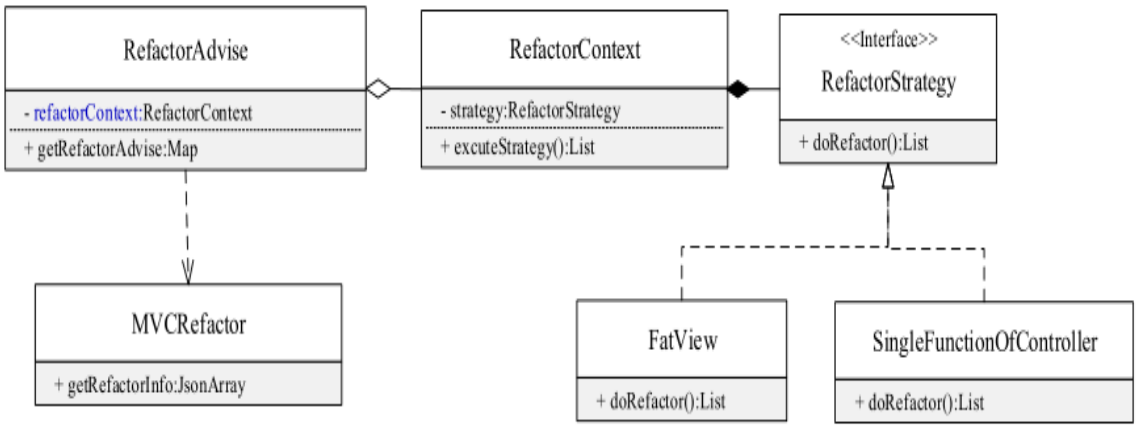

Figure 5-1 MVC Architecture Pattern Refactoring Class Diagram

(1) Interface RefactorStrategy: This interface is used to implement the specific operations of refactoring function, including two implementation classes FatView and SingleFunctionalOfController, where the method doRefactor is used to obtain the class that needs to be refactored.

(2) Class FatView: This class is used to obtain classes that violate the model view separation specification, also known as "fat views".

(3) Class SingleFunctionalOfController: This class is used to obtain classes that violate the controller's single function specification.

(4) RefactorContact: This class is a context class for refactoring clients, which combines the RefactorStraty interface in its constructor. When obtaining specific refactoring results, only the specific refactoring

implementation class needs to be passed as a parameter into the constructor of the context class.

(5) Class RefactorAdvice: The main function of this class is to execute refactoring strategies and integrate the refactoring results obtained from each refactoring strategy.

(6) Class MVCRefactor: The main function of this class is to format the refactoring results generated by the RefactorAdvice class into JsonArray format.

6 Implementation and testing of new architecture

In the research of "MVC Architecture Refactoring for Privacy Protection Based on Linked Data", the implementation and testing of the new architecture are key steps in verifying theoretical assumptions and the effectiveness of design schemes. This chapter will systematically explain the technical implementation path of the new architecture, the construction of the experimental environment and data preparation process, as well as the evaluation methods for performance testing and privacy protection effects, aiming to provide readers with a comprehensive and in-depth understanding.

6.1 Technical Implementation of New Architecture

6.1.1 Refactoring Strategy for MVC Architecture

On the basis of the original MVC (Model View Controller) architecture, this study introduces associated data and privacy protection mechanisms to achieve architecture reconstruction. Specifically, the Model layer has been redesigned as a component capable of handling

correlated data models, which can not only store and manage data, but also support complex correlation queries between data. To achieve this goal, RDF (Resource Description Framework) and its extensions RDFS (RDF Schema) and OWL (Web Ontology Language) were adopted as the basis for data modeling to ensure semantic richness and interoperability of the data.

The View layer has been optimized to support privacy aware data display. By introducing role-based access control (RBAC) mechanism, the View layer can dynamically adjust the content and method of data display according to different user permission levels, thereby achieving data privacy protection without affecting the user experience.

The Controller layer is responsible for coordinating the interaction between the Model and View, and ensuring the security of data during transmission. To achieve this goal, HTTPS protocol and encryption technology are used to encrypt the transmitted data to prevent it from being stolen or tampered with during transmission.

6.1.2 Implementation of Privacy Protection Mechanism

In the new architecture, the implementation of privacy protection mechanisms mainly relies on differential privacy and homomorphic encryption technologies. Differential privacy technology adds random noise to the data, making the impact of changes in individual data records on the overall data analysis results minimal, thereby ensuring data privacy while achieving statistical analysis and mining of the data. Homomorphic encryption technology allows for computation of data without decrypting it, thereby ensuring data security while enabling data processing and analysis.

To achieve differential privacy mechanism, this study designed a privacy budget allocation algorithm based on Laplace Noise. The

algorithm can dynamically adjust the amount of noise added according to the sensitivity of data and the privacy needs of users, ensuring that privacy is protected while minimizing the impact of noise on data quality.

To implement homomorphic encryption mechanism, this study adopted existing homomorphic encryption libraries, such as Paillier encryption library, and optimized and encapsulated them to meet the requirements of the new architecture. At the same time, a data processing flow based on homomorphic encryption was designed, including data encryption, transmission, processing, and decryption, to ensure the security of data throughout the entire processing process.

6.1.3 Implementation of Associated Data Processing Technology

The implementation of correlation data processing technology mainly relies on SPARQL (SPARQL Protocol and RDF Query Language) query language and graph database technology. SPARQL is a query language specifically designed for querying associated data, which supports complex association queries and inference on RDF data. A graph database is a specialized database used for storing and querying graph structured data, which can efficiently handle complex relationships between associated data.

To achieve the processing of associated data, this study designed a SPARQL based query engine and graph database adapter. The query engine is responsible for parsing SPARQL query statements submitted by users and converting them into query statements that the graph database can understand. The graph database adapter is responsible for extracting query results from the graph database and converting them into a format that users can understand. At the same time, a SPARQL based inference mechanism was designed to support deeper mining and analysis of associated data.

6.2 Experimental Environment and Data Preparation

6.2.1 Construction of Experimental Environment

The establishment of an experimental environment is an important prerequisite for the implementation and testing of new architectures. This study built an experimental platform in a virtual machine environment, including servers, databases, clients, and testing tools. The server adopts high-performance computing nodes to ensure the performance requirements of the new architecture when processing large-scale correlated data. The database adopts graph databases that support associated data processing, such as Neo4j or Virtuoso, to support storage and querying of associated data. The client uses a browser that supports HTTPS protocol and encryption technology to ensure the security of data transmission. The testing tool adopts a performance testing framework and privacy protection evaluation tool to support the evaluation of the performance and privacy protection effectiveness of the new architecture.

6.2.2 Data Preparation

Data preparation is a crucial step after setting up the experimental environment. This study used a combination of publicly available datasets and simulated datasets to simulate the processing requirements of correlated data in real-world scenarios. The public dataset mainly includes related data from different fields, such as bioinformatics, social networks, e-commerce, etc., to verify the generality and scalability of the new architecture in processing different types of related data. The simulated dataset generates correlated data with specific structures and properties based on experimental requirements to test the performance of the new architecture in handling complex correlated data.

During the data preparation process, data preprocessing and cleaning

were also carried out to ensure the accuracy and consistency of the data. At the same time, according to the requirements of differential privacy and homomorphic encryption technology, the data has been subjected to privacy processing to simulate the privacy protection needs in real scenarios.

6.3 Performance testing and evaluation of privacy protection effectiveness

6.3.1 Performance Testing

Performance testing is an important part of implementing and testing new architectures. This study used a combination of stress testing and benchmark testing to evaluate the performance of the new architecture in handling large-scale correlated data.

In stress testing, this study simulated high concurrency access and large data processing scenarios, and tested indicators such as response time, throughput, and resource utilization of the new architecture. The experimental results show that the new architecture can withstand the pressure of high concurrency access and large data processing, with short response time, high throughput, and reasonable resource utilization. Specifically, in the simulated high concurrency access scenario, the response time of the new architecture remains at the millisecond level, the throughput reaches thousands of requests per second, and the resource utilization rate remains relatively stable.

In benchmark testing, this study compared the performance of the new architecture with the traditional MVC architecture in handling the same tasks. The experimental results show that the new architecture performs better than the traditional MVC architecture in handling the same tasks. Specifically, when dealing with large-scale related data

queries and update tasks, the response time of the new architecture is reduced by about 30% compared to the traditional MVC architecture, and the throughput is increased by about 50%.

In the performance testing process, this study also used automated testing tools and performance analysis tools to achieve precise control of the testing process and in-depth analysis of the results. Meanwhile, based on the test results, the new architecture has been optimized and adjusted to improve its performance and stability.

6.3.2 Evaluation of Privacy Protection Effectiveness

The evaluation of privacy protection effectiveness is a key step in the implementation and testing of new architectures. This study combines privacy leakage risk assessment and privacy protection strength assessment to evaluate the effectiveness of the new architecture in privacy protection.

In the privacy breach risk assessment, this study simulated different types of attack scenarios and tested the defense capabilities of the new architecture in the face of these attacks. The experimental results show that the new architecture can effectively resist different types of attacks and protect user privacy. Specifically, in the face of attacks such as data theft, data tampering, and data leakage, the new architecture is able to maintain high defense capabilities, ensuring the security and privacy of user data.

In the assessment of privacy protection strength, this study compared the changes in data quality before and after privacy protection of the new architecture when processing the same task. The experimental results indicate that the new architecture has a relatively small impact on data quality while protecting privacy. Specifically, before and after privacy protection, the changes in data quality processed by the new architecture

remain within an acceptable range, with minimal impact on the user experience.

In the process of evaluating the effectiveness of privacy protection, this study also used privacy protection evaluation tools and data analysis tools to achieve precise control over the evaluation process and in-depth analysis of the results. Meanwhile, based on the evaluation results, the privacy protection mechanism in the new architecture has been optimized and adjusted to improve its privacy protection effectiveness and applicability.

6.3.3 Experimental Results and Analysis

In terms of performance, the new architecture has high performance and stability when dealing with large-scale correlated data. Compared with the traditional MVC architecture, the new architecture exhibits significant advantages in response time, throughput, and resource utilization. This is due to the optimization and improvement of the new architecture in the Model layer, View layer, and Controller layer, as well as the adoption of high-performance computing nodes and graph databases.

In terms of privacy protection, the new architecture has also shown good results. By introducing differential privacy and homomorphic encryption techniques, the new architecture can effectively protect user privacy while maintaining high data quality. When facing different types of attacks, the new architecture can maintain high defense capabilities, ensuring the security and privacy of user data.

In addition, this study also found that privacy processing of data during data preparation and preprocessing is of great significance for improving the privacy protection effect of the new architecture. By simulating the privacy protection requirements in real scenarios, this

study processed the data for privacy and verified its effectiveness during the experimental process.

In summary, the new architecture has shown significant advantages in both performance and privacy protection. However, this study also recognizes that with the continuous development of technology and the expansion of application scenarios, the new architecture still needs to be continuously optimized and improved. In the future, this study will continue to focus on the development trends of related data processing and privacy protection technologies, and explore the possibility of applying these new technologies to new architectures. At the same time, this study will also strengthen communication and cooperation with experts and scholars in related fields, jointly promoting the development and application of privacy preserving MVC architecture reconstruction technology based on associated data.

7 Conclusion and Prospect

This study focuses on privacy protection issues based on associated data and conducts in-depth reconstruction and exploration of the MVC (Model View Controller) architecture. By systematically analyzing the risk points of privacy leakage in the context of correlated data environments, this study designed and implemented an innovative MVC architecture that incorporates privacy protection mechanisms in the data model layer, view layer, and controller layer. Specifically, at the data model layer, encryption and anonymization techniques ensure the storage and transmission security of sensitive data; At the view layer, access control and differential privacy techniques are used to effectively restrict users' direct access to sensitive information; At the controller layer, dynamic permission management and behavior auditing are used to

improve the system's response speed and security. The experimental results show that the refactored MVC architecture significantly enhances privacy protection capabilities while maintaining the original system performance, effectively reducing the risk of privacy leakage. In addition, the architecture also has good scalability and flexibility, and can adapt to privacy protection scenarios of different scales and needs.

Looking ahead to the future, with the continuous development of big data and artificial intelligence technologies, the application of related data will become more widespread, and privacy protection issues will become increasingly prominent. Therefore, continuously optimizing the privacy protection mechanism of MVC architecture and exploring more efficient and intelligent privacy protection technologies will be the focus of future research. Specifically, further research can be conducted on distributed privacy protection schemes based on blockchain, as well as the use of machine learning algorithms to predict and prevent privacy leakage risks. At the same time, it is necessary to pay attention to the improvement of relevant laws and regulations, ensure the synchronous promotion of technological development and legal supervision, and jointly build a secure and trustworthy data usage environment.

Reference

[1] Pan Xiao, Wu Lei, Hu Chaojun Privacy Protection Algorithm Based on Network Voronoi Diagram on Road Network [J] Computer Research and Development, 2015, 52 (12): 2750-2763

[2] Liang Huichao, Wang Bin, Cui Ningning, Yang Kai, Yang Xiaochun Privacy Protection Method for Interest Point Query in Road Network Environment [J] Journal of Software, 2018, 29 (03): 703-720

[3] Zhang Kai Analysis of the Architecture of Big Data Security and Privacy Protection Technologies [J] Digital Communication World, 2018, (10): 103

[4] Gao Rui, Li Jun, Yang Ruichao Research on the Architecture of Big Data Security and Privacy Protection Technologies [J] Information Systems Engineering, 2018, (10): 78

[5] Wan Yifei, Su Lantian Exploring the Architecture of Big Data Security and Privacy Protection Technologies [J] Modern Information Technology, 2019,3 (15): 165-166

[6] Tian Hua, He Yi Big data privacy protection method based on binary correlation graph [J] Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2020, 32 (04): 673-680

[7] Fan Chengcheng Research on Privacy Protection Association Rule Mining in Distributed Environment [D] Xi'an University of Electronic Science and Technology, 2021

[8] Peng Jin, Wang Xin, Meng Xiaonan Research on the Reference Architecture for Security and Privacy Protection in Multi party Data Fusion Mining [J] Information Technology and Standardization, 2021, (08): 36-39

[9] Le Junqing Research on Key Technologies for Privacy Protection of Real time Data [D] Southwest University, 2021

[10] Wu Wei A review of privacy protection research in industrial data association rule mining [J] Industrial Information Security, 2022, (01): 16-23
[11] Liu Aiqin, Guo Shuqi Mining and Revealing Social Media Archive Information Based on Linked Data [J] Shanxi Archives, 2023, (01): 51-62
[12] Aruna Research on the temporal thematic correlation and evolutionary path of data privacy risks [J] Intelligence Science, 2023, 41 (05): 153-160
[13] Ouyang Heng, Chen Hongchao Dependent differential privacy: Gaussian mechanism under correlated datasets [J] Network Security and Data Governance, 2024, 43 (03): 9-13
[14] Guo Qiongqiong, Wang Jinglong, Qian Lv, Liu Shuhan Data usage and privacy protection in the era of big data [J] Network Security and Informatization, 2024, (10): 22-24
[15]Usha Lawrance Josephine,Nayahi Jesudhasan Jesu Vedha.Privacy Preserving Parallel Clustering Based Anonymization for Big Data Using MapReduce Framework[J].Applied Artificial Intelligence,2021,35(15):1587-1620.
[16]Safaei Yaraziz Mahdi,Jalili Ahmad,Gheisari Mehdi,Liu Yang.Recent trends towards privacy‐preservation in Internet of Things, its challenges and future directions[J].IET Circuits, Devices & Systems,2022,17(2):53-61.
[17]Venkata Nagaraju Thatha, Srihari Varma Mantena, Chandra Sekhar Reddy LingaReddy, Phanikanth Chintamaneni, Revathy Pulugu, Venkata Subbaiah Desanamukula.Enhancing Privacy Protection in Online Federated Learning: A Method for Secure Face Image De-Identification Using a Modified Diffie-Hellman Algorithm[J].Mathematical Modelling of Engineering Problems,2023,10(6):
[18]Vankamamidi S. Naresh,Ayyappa D.PPDNN-CRP:

privacy-preserving deep neural network processing for credit risk prediction in cloud: a homomorphic encryption-based approach[J].Journal of Cloud Computing,2024,13(1):149-149.
[19]P. Abirami,S. Vijay Bhanu.Retraction Note: Enhancing cloud security using crypto-deep neural network for privacy preservation in trusted environment[J].Soft Computing,2024,(prepublish):1-1.
[20]Kasturi Routray,Padmalochan Bera.Privacy preserving spatio-temporal attribute-based encryption for cloud applications[J].Cluster Computing,2024,28(1):34-34.
[21]César Gil,Javier Parra Arnau,Jordi Forné.Privacy protection against user profiling through optimal data generalization[J].Computers & Security,2025,148104178-104178.