


Research Review on the Application of Homomorphic Encryption in Database Privacy Protection

Yong Ma, School of Computer Information Engineering, Jiangxi Normal University, China

Jiale Zhao, School of Computer Information Engineering, Jiangxi Normal University, China

 <https://orcid.org/0000-0002-5895-9148>

Kangshun Li, College of Computer, Guangdong University of Science and Technology, China

Yuanlong Cao, School of Software, Jiangxi Normal University, China

Huyuan Chen, School of Mathematics and Statistics, Jiangxi Normal University, China

Youcheng Zhang, Nanjing Unary Information Technology Co. Ltd., China

ABSTRACT

With the advent and development of database applications such as big data and data mining, how to ensure the availability of data without revealing sensitive information has been a significant problem for database privacy protection. As a critical technology to solve this problem, homomorphic encryption has become a hot research area in information security at home and abroad in recent years. The paper sorted out, analyzed, and summarized the research progress of homomorphic encryption technology in database privacy protection. Moreover, the application of three different types of homomorphic encryption technology in database privacy protection were introduced respectively, and the rationale and characteristics of each technique were analyzed and explained. Ultimately, this research summarized the challenges and development trends of homomorphic encryption technology in the application of database privacy protection, which provides a reference for future research.

KEYWORDS

Artificial Intelligence, Ciphertext Calculation, Database, Fully Homomorphic Encryption, Information Security, Partial Homomorphic Encryption, Privacy Protection, Somewhat Homomorphic Encryption

1. INTRODUCTION

With the advent of the era of big data, as a convenient and efficient information carrier, the database has been widely used in various scenarios. However, some previously inconspicuous problems have become more and more prominent; among them, the security of private data storage is of the most significant concern. A security breach happened in the database of True Dialog in 2019, the American commercial SMS service provider, which led to the disclosure of millions of SMS messages¹; Besides, a cross-platform database company MongoDB was attacked by Hacker groups who infiltrated 22,900 insecure databases, which accounted for almost 47% of all MongoDB databases, with a total economic loss of US\$3.2 million in July 2020². Consequently, how to protect data privacy in various database applications has become a research hotspot in academia in recent years (Agrawal et al., 2001; Verykios et al., 2004; Agrawal et al., 2000; Zhao et al., 2020; Clifton et al., 2002; Zhang et

DOI: 10.4018/IJCINI.287600

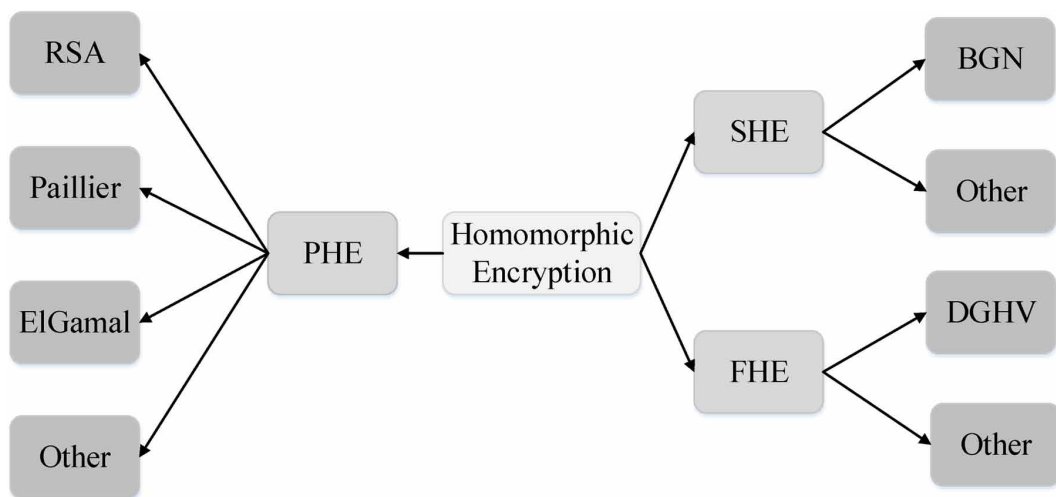
This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

al., 2005; Machanavajhala et al., 2007; Xiao et al., 2006; Xiao et al., 2007; Nayyar, 2019; Raj, R. et al., 2020; Li, J. et al., 2021).

In different protection schemes for database privacy data (Verykios et al., 2004; Agrawal et al., 2000; Zhao et al., 2020), encrypting plaintext privacy data is very useful. However, after the plaintext data is encrypted, the original keywords may become different ciphertext information, and the correlation among the keywords is also destroyed. In this way, it is difficult for the query algorithm to work appropriately based on these keywords. What's more, it is also a challenging task to query the ciphertext data without decrypting it. As a result, a homomorphic encryption scheme emerged. Homomorphic encryption is used to encrypt user privacy data, and a series of operations such as deletion, update, and retrieval can be directly performed on the encrypted data.

Homomorphic encryption is a type of encryption method with unique natural properties. And this concept was first proposed by Rivest et al (1978) in the 1970s. Compared with standard encryption algorithms, homomorphic encryption can not only achieve basic encryption operations but also it can realize multiple calculation functions between ciphertexts. In other words, calculating first and then decrypting can be similar to decoding first and calculating. Compared with traditional encryption methods, homomorphic encryption technology can calculate multiple ciphertexts and then decrypt them. By using this, there is no need to decrypt each ciphertext and spend high computational costs. This article divides the homomorphic encryption technology in database privacy protection into three categories according to the type and number of ciphertext operations supported: Partial Homomorphic Encryption (PHE); Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). Additionally, the representative algorithms of each type are shown in Figure 1.

Figure 1. Types of homomorphic encryption



This paper reviewed the research on the application of homomorphic encryption in database privacy protection based on analyzing related domestic and foreign research, which provides a reference for future research. The structure of this paper is as follows: Section 1 introduces the research problems and objectives of this paper; Section 2 introduced the research progress of database privacy protection and homomorphic encryption technology, providing a general overview of the relevant reviews; Sections 3-5 analyzed and summarize the application of three homomorphic encryption technologies in database privacy protection respectively; Section 6 summarized the applications of

various homomorphic encryption introduced in this article in database privacy protection, showing the direction to future research.

2. RESEARCH PROGRESS ON DATABASE PRIVACY PROTECTION AND HOMOMORPHIC ENCRYPTION

2.1 Database Privacy Protection

At present, the significant ways to ensure database privacy data security include firewalls, identity verification, and auditing. However, leakage of sensitive database information still occurs frequently, showing that these security measures are minimal to solve the problem. Ensuring that legitimate users can safely access their data, the most direct way is to encrypt the data before it is stored in the database. It is generally acknowledged that data encryption technology is the simplest and most capable technical method to protect users' data from illegal access, used in many fields widely such as the Internet, electronic communications, online shopping, and online banking. Although data encryption can effectively avoid unlawful access to data, it also destroys the underlying semantic structure of data, making it impossible to perform operations such as calculation and retrieval of ciphertext. Consequently, the results of ciphertext operations make little sense by using traditional encryption methods, making ciphertext retrieval become a new challenge (Chor et al., 1995; Boneh et al., 2007; Avni et al., 2015).

After the data is encrypted traditionally, it is necessary to decrypt the ciphertext first to perform operations such as retrieval, calculation, or analysis. It is well known that the amount of data in the database system is enormous, a lot of resources for decryption will be consumed even if only a small part of the data needs to be retrieved. This kind of problem also exists in other subject areas (Li et al., 2018; Li et al., 2018; Li et al., 2019; Li et al., 2019; Mora et al., 2019; Akbarnia et al., 2019; Kabir et al., 2020). Users prefer to utilize a server's computing power to process the ciphertext directly when information needs to be recovered or calculated to obtain valid results. In the whole process, the data processor cannot get any content related to the plaintext of the data.

Order-Preserving Encryption (OPE) is an encryption scheme that preserves the original plaintext sequence of the ciphertext. Although OPE can directly determine the order of ciphertext data, its security is not high, and it can be cracked by dichotomy, which cannot meet the needs of privacy data security. Therefore, a homomorphic encryption scheme is proposed. Homomorphic encryption technology can meet the above requirements well. The same as traditional encryption methods can protect the privacy of users, and homomorphic encryption technology also supports direct arithmetic operations, including addition and multiplication on the ciphertext. Most importantly, the operation process can guarantee that the relationship between ciphertext and plaintext remains consistent. Figure 2 is a more common database privacy protection model, reflecting how to implement secure queries among Data Owner, User, and Databases.

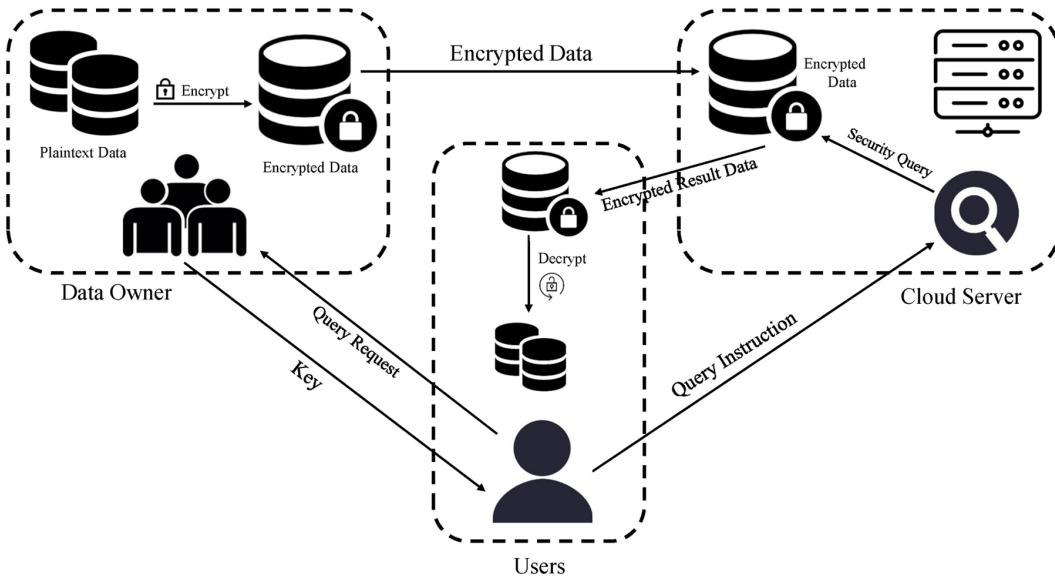
2.2 Research Progress Of Homomorphic Encryption

The research of homomorphic encryption can be traced back to the 1970s. It was not long after the RSA cryptosystem was put forward, Rivest et al (1978) proposed the concept of homomorphic encryption, also known as privacy homomorphism, which is defined as follows:

Let $E(K, x)$ denote that x is encrypted with encryption algorithm E and key K , and F represents an operation. If for encryption algorithm E and operation F , there is algorithm G such that:

$$E\left(K, F\left(x_1, \dots, x_n\right)\right) = G\left(K, F\left(E\left(x_1\right), \dots, E\left(x_n\right)\right)\right)$$

Figure 2. Database privacy protection model



It is said that encryption algorithm E is homomorphic for operation F .

Once homomorphic encryption was put forward, it became an open problem in cryptography. Homomorphic encryption is a form of encryption that allows users to perform specific algebraic operations directly on the ciphertext, and decrypt the result of the operation, which is similar to the result obtained by directly implementing the same procedure on the plaintext. Nevertheless, the advantage of homomorphic encryption is that users can still analyze and retrieve specific encrypted data when data is encrypted, improving the efficiency of data processing and guaranteeing the safe transmission of data. Ultimately, the correct encrypted data can still get the correct decryption result. Scholars at home and abroad have successively studied the PHE scheme that satisfies the multiplication or the addition, and further proposed SHE scheme that can meet the finite number of arithmetic and acquisition at the same time (Damgard et al., 2001; Goldwasser et al., 2005; Goldwasser et al., 2019; Kawachi et al., 2007; Naccache et al., 1998).

In 2009, IBM researcher Gentry (2009) proposed a new homomorphic encryption scheme based on the ideal lattice, which can perform an unlimited number of operations on encrypted data that can be performed on the plaintext without decryption, enabling in-depth and infinite analysis of encrypted information without affecting its confidentiality, which is called Fully Homomorphic Encryption (FHE). In 2010, Dijk et al (2010) came up with an integer-based FHE scheme DGHV based on Gentry's research. Although DGHV is more concise than Gentry et al.'s scheme, its practicality is still not high. In the same year, Smart et al (2010) and Gentry (2011) respectively further simplified the FHE scheme proposed by Gentry in 2009, reducing the length of the ciphertext and key, making the realization of the FHE scheme possible. In 2005, Regev et al (2005) proposed the FHE scheme based on the Learning with Error (LWE) problem, which is the most concise fully homomorphic encryption scheme by far. Intending to improve the efficiency of fully homomorphic encryption, cryptographers have successively proposed many FHE schemes (Brakerski et al., 2014; Gentry et al., 2011), which makes fully homomorphic encryption more practical.

PHE has the widest range of applications because of its simple design and relatively mature scheme. However, because its function is relatively single and can only perform one operation, it is not suitable for scenarios that deal with complex query services. FHE is the most ideal encryption scheme, which can handle complex query requests and can theoretically be applied in all scenarios.

Table 1. Main-Related terminologies

Terminology	Meaning
PHE	Partial Homomorphic Encryption
SHE	Somewhat Homomorphic Encryption
FHE	Fully Homomorphic Encryption
Key	Input parameters in plaintext and ciphertext conversion algorithms
Symmetric encryption	The encryption key is the same as the decryption key
Asymmetric encryption	Also known as public key encryption, the encryption key is different from the decryption key

But because of its high requirements for computing resources, so far there has not been a solution that can actually be deployed and run. SHE can be regarded as the product of the coordination of PHE and FHE, which can perform a limited number of different operations on the basis of reasonable requirements for computing resources.

The continuous development of homomorphic encryption technology has provided opportunities for researchers to apply partial homomorphic encryption technology and the latest fully homomorphic encryption technology to database privacy protection. It will also boost the continuous progress and development of this field. This article summarized and categorized the application methods and techniques of homomorphic encryption in databases. It also generalized the challenges faced by homomorphic encryption in database security, to provide a reference or help for future researchers better to solve the problem of database privacy data security. The Main-Related terminologies used in this article are shown in Table 1.

2.3 Review Of Related Literature

Literature (Kogos et al., 2017; Li et al., 2015; Li et al., 2018; Shundong et al., 2015; Mingjie et al., 2014) mainly introduces the research progress of homomorphic encryption technology and its application in various scenarios, which is targeting a summary of the research status, deficiencies, and core key issues that deserve further study. Moreover, this literature also illustrates the application of homomorphic encryption technology in cloud computing, electronic voting, and digital watermarking. Next, literature (Wang et al., 2002; Wu et al., 2006; Hui et al., 2014; Tian et al., 2017; Zhou et al., 2009) mainly summarizes the current research results in database privacy protection, expounds the basic principles and characteristics of various privacy protection technologies, and also introduces the typical applications of multiple technologies in detail. The literature mentioned above has made a relatively comprehensive overview of homomorphic encryption technology and database privacy protection technology. However, the major drawback is that they have not made a systematic summary of the application of homomorphic encryption technology in database privacy protection. Literature (Morampudi, M. et al., 2020; Devi, P. et al., 2020; Lamba, J. et al., 2020; Pang, H. et al., 2020; Wu, W. et al., 2020), the primary concentrate is on open key cryptographic algorithms in view of homomorphic encryption conspire for protecting security. The contextual investigation on different standards and properties of homomorphic encryption is given as the analyzing proof.

On the whole, Domestic and foreign research on the application of homomorphic encryption technology in database privacy protection is still in its infancy, and there is room for development in the future.

3. PHE-BASED DATABASE PRIVACY PROTECTION TECHNOLOGY

Since the concept of homomorphic encryption was first proposed by Rivest et al. in 1978, various encryption schemes have been continuously put forward. We call homomorphic operations that only support a single type (addition or multiplication) in the ciphertext domain as partial homomorphic encryption (PHE). PHE has top security and a relatively simple structure, where the RSA system, Paillier system, and ElGamal system are the most widely used. Unfortunately, they have considerable limitations in practical applications because of limited types of operations supported. The following is an introduction to the implementation of partially homomorphic encryption algorithms based on three encryption systems in database privacy protection.

3.1 Database Privacy Protection Scheme Based On Rsa

RSA cryptosystem proposed by Rivest et al (1978) in 1978 is the first practical public-key encryption scheme whose security is based on the difficulty of integer factorization. Besides, this system satisfies the multiplicative homomorphism.

Wang, L. et al (2020) established a safe and usable database system in a multi-cloud environment, whose security is guaranteed by RSA-based homomorphic encryption technology. In this system, data owners create a new table T , which has a list of N rows named A , where the row number of each row is r_i ($r_i > 0$), and it is stored as a single column named row_id . Column key $ck_A = \langle x, y \rangle$ is generated for column A , x , and y are random numbers ($x, y > n$), so table T has two columns (row_id, A). Before storing the data in the database, row key v_{key} is created by ck_A and row_id , and then use the row key to encrypt each row of data to generate v_e . The generating formula is $v_e = E(v, v_{key}) = v v_{key}^{-1} \bmod n$, where v_{key}^{-1} is the inverse of v_{key} , $v_{key} v_{key}^{-1} \bmod n = 1$. v_e will be stored in the cloud database. At this time, when the data owner adds a private data v to column A, v will be owned by the data owner and the database in two forms: v_{key} and v_e . When you need to decrypt the ciphertext, you only need to know v_e and v_{key} : $v = D(v_e, v_{key}) = v_e v_{key} \bmod n$.

In the solution of literature (Wang et al., 2020), whenever private data is added, a corresponding key will be stored locally, which will cause a storage burden for local users. Meanwhile, when the amount of information is large, a large number of local generations will be created. Consequently, decryption operations will consume a large number of local computing resources and impose a computational burden on local users. To solve this problem, Wei, z. z. et al (2013) reduced the pivotal acquisition time based on ensuring the security of the key by constructing a homomorphic fundamental agreement model, thereby improving the computational efficiency.

According to the solution in literature (Wei et al., 2013), it supposed that the RSA public-private key pair of the data owner is (PK_O, SK_O) , and the user is (PK_D, SK_D) , and RSA public-private of database S is (PK_S, SK_S) . When the user finds the required data, he/she sends a request to the data owner and the database respectively, and they respectively return the random number encrypted with the user's public key, namely R_O^{PKD} and R_S^{PKD} . Afterward, Data acquirer D conducts the following calculation by using the same multiplication of the RSA encryption system: $R_O^{PKD} R_S^{PKD} = (R_O R_S)^{PKD}$ to get $K = R_O R_S$, and ultimately, the computer uses the critical K to decrypt the plaintext data. In this scheme, the user only needs one multiplication operation and one decryption operation to get core key K .

3.2 Database Privacy Protection Scheme Based On Paillier

The Paillier system proposed by scholar Paillier in 1999 (Paillier, 1999) is the first additive homomorphic encryption cryptosystem based on the problem of judging composite residues, whose security is based on the issue of deciding composite residues. Moreover, this system supports arbitrary multiple additions of homomorphic operation.

Marwan, M. et al (2016) used Paillier's homomorphic cryptographic system to establish a secure cloud database. First, the data owner encrypts the data before transmitting it to the cloud database; in this process, the data owner has public-private key pair information. Next, it sends encrypted data to the cloud database. After that, the user uses the private key to obtain the final result during the decryption process. At the same time, the model uses a key management server (KMS) to ensure the security of the key exchange. This model is the most basic application of the Paillier system in the database field. Thanks to the homomorphism of the Paillier system, this model can improve the security of cloud data to a certain extent. Still, this model is not suitable for multi-user, high-concurrency, etc. Typical usage scenarios in the database field.

YU, Z. B. et al (2015) proposed the hPIRMR protocol. They used Paillier-based hash tools to construct a keyword-based retrieval scheme for cloud-encrypted files, which guarantees users access privacy and outsourced data security during query interaction. This solution requires two implementations of the PIR (Private Information Retrieval) protocol, resulting in more communication volume and query response time for both parties. Wang, S. et al (2014) proposed an actual private processing solution that implements homomorphic encryption to query public data databases. Paillier is used to provide complete query privacy, homomorphic encryption, and open data conversion. Compared with traditional PIR solutions, this method only involves one round of client-server interaction for processing one query, thereby saving bandwidth for the server.

Hingwe, K. K. et al (2016) proposed a homomorphic encryption access control scheme for database services based on hierarchical roles. By establishing a multi-level database access control model, queries are granted with the least access rights and session keys that are used for session management. If the query is to perform any operation on sensitive data, additional permissions are required to access sensitive data. Kim, K. I. et al (2019) proposed a KNN query model based on homomorphic encryption, which can hide data access patterns while ensuring data and query privacy, improving security significantly. Since data owners and users do not participate in query processing, this model also supports operations on outsourced databases. However, the query processing cost of this solution is high since it performs the required calculations on all encrypted data to achieve the purpose of hiding data access patterns.

Li. S. D. et al (2018) designed a new encoding method for the ciphertext sorting problem in the database, which combined Paillier additive homomorphic encryption algorithm, elliptic curve encryption system, secret segmentation, and threshold decryption algorithm. The secure multi-party computing protocol can resist collusion attacks of varying degrees. Yuan, X. P. et al (2011) tried to solve the problem that the string approximate matching query in the database cannot adequately protect the private information of the querying parties. In this sense, he established an approximate matching query protocol for the string data in the database. Using security technologies such as secure computing editorial distance protocol, homomorphic encryption, dazed transmission, the query for approximate matching of strings is realized in the case of effectively protecting the private information of the querying parties.

3.3 Database Privacy Protection Scheme Based On ElGamal

The ElGamal system is the first public-key encryption system based on discrete logarithms. Proposed by Egyptian cryptographer ElGamal in 1984 (ElGamal, 1985). The security of the ElGamal system is by the difficulty of discrete logarithms. And the ElGamal algorithm is based on operations on a finite field, whose feature is that the ciphertext is composed of two parts and meets the multiplicative homomorphic functions, suitable for secure storage of cloud database (Kalyani, P. et al., 2021).

Jia, Z. et al (2012) created privacy protection that can access control model based on secret sharing and ElGamal homomorphic attributes. In this model, the user's query is combined with the proxy server's access control strategy, and the user can get the query result with sufficient authority. Moreover, each proxy server has an independent access control matrix. First, a public-private key pair is generated in line with the ElGamal system, and the data owner sends the key to each proxy server and then encrypts the access control matrix. When the user submits a query request, and the query is completed, the access control matrix is rebuilt. Li, S. D. et al (2018) suggested an efficient string secret sorting scheme based on the ElGamal algorithm, which pre-processing the data before it was encrypted. It first calculates $R_i = (g^r \bmod p, h^r \bmod p)$, encryption is to perform several simple modular multiplication operations on R_i , trying to avoid the complex modular exponential operation in the encryption process of the original ElGamal algorithm.

Cominetti, E. et al (2020) presents two efficient partially homomorphic encryption schemes built upon the approximate common divisor problem, believed to be resistant to quantum computer attacks. Both proposals, named FAHE1 and FAHE2, are additively homomorphic and have a symmetric nature, meaning that they are useful in scenarios where encryption and decryption are performed by the same entity. This is the case, for example, of encrypted databases stored in a public cloud. experimental results show that both solutions provide considerable speed-ups when compared to Paillier. Namely, encryption and decryption with FAHE1 are, respectively, 120 and 25 times faster than Paillier's, while for FAHE2 both operations run more than 1000 times faster.

3.4 Other Phe Database Privacy Protection Schemes

Li, Z. C. et al (2018) argued a secure data query scheme based on homomorphic ciphers. It uses an adjustable onion encryption strategy (CryptDB), Paillier addition homomorphism, and ElGamal multiplication homomorphism to manipulate ciphertext directly. It avoids frequent client interaction with the server and encryption and decryption processing to resist selected plaintext attacks effectively. Tu, S. L. et al (2013) built the Monomi algorithm and improved CryptDB to handle more complex queries. Both MrCrypt (Tetali et al., 2013) and Crypsis (Stephen et al., 2014) put forward based on CryptDB, both of which can implement addition and multiplication homomorphic operations of ciphertext data. To support multiple types of query services, each piece of data and variable in CryptDB needs to be encrypted by using a different homomorphic encryption scheme, which leads to additional storage consumption. Furthermore, when outsourcing data is frequently transmitted, it will also increase communication overhead with third-party cloud service providers.

Combined with the characteristics of tree structure directory services, Zheng, Y. et al (2013) put forward a directory service data change capture method based on homomorphic hashing after looking at the research of commonly used data change capture strategies and technologies. Besides, this method combines the special shadow table method of the homomorphic hash function and memory database technology. Based on the homomorphism of the homomorphic hash algorithm, the change data of the directory service can be quickly obtained when the data change rate is low.

Jiang, Y. J. et al (2011) proposed a data set outsourcing scheme that is targeting Mignotte secret sharing for the privacy matching problem in the outsourcing database system. In this system, the user interacts with the third-party service provider, constructs the discriminant through additive homomorphic encryption and secret reconstruction, and judges, whether the element of the user's data set belongs to the data set of the data owner by judging the value of the discriminant, is zero, and finally realizes privacy match. Yang, Y. B. et al (2007) do not encrypt the primary and foreign keys on the table. To begin with, for encrypted numeric fields, if constant arithmetic operations are required, the secret homomorphic encryption technology is used. Moreover, if the retrieval function is also required, then the entry address index table is established for the ciphertext, and the binary search method is used to realize the retrieval. When it comes to numeric fields with a large retrieval volume and few arithmetic operations, the ciphertext entry address index or ciphertext index is used.

For encrypted date-type areas, ciphertext entry address index or ciphertext index technology can be used to achieve. As for other types of encrypted fields, we need to convert different types into one of the previous three classes and then perform encryption processing. Combined with the advantages of RSA public-key encryption and homomorphic encryption, Gui, Q. et al (2009) adopted a cryptographic hierarchical management mechanism based on the ARBSM algorithm. They proposed a simple and effective privacy protection distributed association rule mining algorithm (PPDM-ARBSM). Introducing a password management server (CMS) and data mining server (DMS) into the algorithm not only effectively protects the security of sensitive data but also uses the transaction similarity matrix to quickly and centrally generate a global K-item frequent set.

Cao, N. et al (2013) proposed a multi-keyword sorting search technology. The idea of the scheme is to add false keywords when encrypting a vector with a key to perform division or multiplication (the key is composed of a vector and two matrices), accordingly, the client will also apply the same operation (with a few changes) to encrypt the query vector with the same key and send it to the cloud. After the cloud receives it, the encrypted vector (query and index) will be processed to generate a similar vector. Li, J. et al (2010) invented technology for fuzzy keyword retrieval of encrypted data. By employing it, the data owner constructs an index by building a set of fuzzy keywords and then shares it between the data owner and authorized users. The key calculates the trap set of the key, and the data owner sends this index to the cloud. When retrieving the data set, the authorized user uses the shared key with the data owner to calculate the trap set of the query key, and then sends it to the cloud, after receiving the query, the cloud compares with the index table. It returns all possible encrypted file identifiers based on fuzzy keywords. Jain, R. et al (2020) propose a registration authentication storage data assess (RASD) structure and homomorphic-private-practical uniformity testing-based plan structured for giving security to authoritative information put away on cloud catalogs utilizing a novel security plot. The essential favorable position of the proposed structure is its simple usage, high security, and overall less computational expense. Findings–Experimentation-based outcomes demonstrated that the RASD structure not only gives a high-security layer for delicate information but also enables a decrease in computational expense and performs better when compared with existing conventions for distributed computing. The performance analysis of each PHE-based database privacy protection technology is shown in Table 2.

4. SHE-BASED DATABASE PRIVACY PROTECTION TECHNOLOGY

A homomorphic encryption scheme can support the homomorphic operations of addition and multiplication in the ciphertext domain at the same time. Still, considering the reduction of the noise when the ciphertext is generated, the number of operations of a certain type of operation has to be limited to complete the decryption process. In other words, homomorphic encryption schemes can only perform ciphertext calculations on specific data sets, and they are only suitable for particular application scenarios in reality. To ensure that SHE can serve the calculation of custom data sets, such as databases, many studies have begun to promote existing homomorphic encryption schemes.

4.1 Database Privacy Protection Scheme Based On Bgn

The BGN system is the first scheme that can simultaneously support arbitrary multiple additions and one-time multiplication homomorphic operations, that is, it can calculate the quadratic function of the ciphertext. What's more, the encryption process has no ciphertext length expansion and has semantic security. The BGN system is only suitable for quadratic expressions, but this system is the closest to the fully homomorphic encryption scheme.

Kerschbaum, F. et al (2012) put forward an outsourced private data query model based on homomorphic encryption, which introduces the service provider P into the system. Both the data owner and the server send their data encrypted by the BGN algorithm to the service provider. Still, the latter does not know any information about the input or output. This model has linear complexity,

Table 2. Performance evaluation of database privacy protection technology based on PHE

Partial Homomorphic Encryption		Safety Degree	Computational overhead	Communication overhead	Data Integrity	Data Dependency
RSA Scheme	Wang, L. et al.	high	high	medium	high	high
	Wei, Z. Z. et al.	high	medium	medium	medium	high
Paillier Scheme	Marwan, M. et al.	medium	high	high	high	low
	YU, Z. B. et al.	high	high	high	medium	high
	Wang, S. et al.	medium	medium	medium	high	low
	Hingwe, K. et al.	high	high	high	high	high
	Kim, H. I. et al.	high	high	medium	medium	low
	Li, S. D. et al.	high	high	medium	medium	high
	Yuan, X. P. et al.	high	medium	high	low	high
ElGamal Scheme	Jia, Z. et al.	high	high	high	medium	high
	Li, S. D. et al.	high	low	medium	high	low
	Cominetti, E. et al.	high	medium	high	low	medium
Other Scheme	Li, Z. C. et al.	high	high	high	high	medium
	Tu, S. L. et al.	high	medium	medium	high	medium
	Tetali, S. D. et al.	high	high	high	high	medium
	Stephen, J. J. et al.	high	high	high	high	medium
	Yu, Z. et al.	medium	medium	low	medium	high
	Jiang, Y. J. et al.	medium	medium	high	medium	low
	Yang, Y. B. et al.	high	medium	medium	medium	high
	Gui, Q. et al.	high	high	medium	high	high
	Cao, N. et al.	high	high	medium	low	low
	Li, J. et al.	high	high	high	low	low
	Jain, R. et al.	medium	high	high	medium	high

and it is safe in the malicious model. Moreover, it has nothing to do with the size of the client set, but it cannot achieve common functions such as secure session management, role-level maintenance, and multi-level access control.

4.2 Other She Database Privacy Protection Schemes

Palamakumbura, S. et al (2015) created a private database query scheme based on homomorphic encryption, which uses secondary homomorphic encryption and batches homomorphic encryption to improve the security and query speed of private data. Compared with the traditional method, this scheme uses SHE technology to improve the security and query speed of private data greatly, but its secondary homomorphism and batch homomorphism is based on the inverted index pre-predicted by the model, which cannot support the dynamic update of data in the database very well. Yihua, Z. et al (2019) proposed an f-mOPE database ciphertext retrieval scheme. To improve the overall security of the scheme, the subtree number generation adopts an improved SHE technology, and the ciphertext index of the data is determined by comparing the binary tree number.

Table 3. Performance evaluation of database privacy protection technology based on SHE

Somewhat Homomorphic Encryption		Safety Degree	Computational overhead	Communication overhead	Data Integrity	Data Dependency
BGN Scheme	Kerschbaum, F. et al.	medium	high	medium	high	low
Other Scheme	Palamakumbura. et al.	high	high	high	high	high
	Yihua, Z. et al.	low	medium	medium	high	high
	Bai, J. et al.	high	medium	high	medium	low
	Boneh, D. et al.	medium	medium	low	medium	high

Bai, J. et al (2018) designed a new type of private database secret homomorphic retrieval protocol. By using an encryption algorithm based on the Learning With Errors over Ring (LWER) problem, it solves the privacy leakage problem involved in the threat model after performing a limited number of homomorphic operations. Boneh, D. et al (2013) constructed a private database query system that supports a rich query set. The basic idea is to encode the database into one or more polynomials and use the user’s query to process these polynomials to obtain a new polynomial whose root is the index of the matching record. In this system, the user holds the key, and the server has the corresponding public key, and ultimately the server encodes the database as a binary polynomial $D(x, y)$. The user constructs a univariate query polynomial $Q(y)$ and sends the encryption coefficient of the polynomial Q to the server. Once the database polynomial $D(x, y)$ and the encrypted query polynomial $Q(y)$ is given, the server uses the additive homomorphism of the cryptographic system to calculate the encrypted polynomial $A(x, y) = D(x, y) - Q(y)$. Then the server uses the plaintext value A_i to calculate the encrypted polynomial $A_i(x) = A(x, A_i), i = 1, 2, \dots, t$. We can say that the root of the polynomial A_i is the query of the index of the matching record. The performance analysis of each SHE-based database privacy protection technology is shown in Table 3.

5. FHE-BASED DATABASE PRIVACY PROTECTION TECHNOLOGY

Fully homomorphic encryption is an encryption scheme proposed by Rivest et al. in 1978, shortly after the RSA algorithm was proposed. Since the cryptography community proposed the fully homomorphic encryption scheme, it has been hailed as the holy grail of cryptography. From 2009 to the present, a series of fully homomorphic encryption schemes have been implemented and optimized. The first generation of fully homomorphic encryption schemes is all based on the construction method of Gentry (Smart et al., 2010; Stehlé et al., 2010; Brakerski et al., 2011; Coron et al., 2011; Coron et al., 2012; Brakerski et al., 2014), and the second generation of fully homomorphic encryption schemes is based on LWE’s (Brakerski, 2012; López-Alt et al., 2012; Gentry et al., 2013). The significance of the fully homomorphic algorithm is to fundamentally solve the confidentiality problem when the data information operation is entrusted to a third party so that it can play a huge role in database privacy protection.

5.1 Database Privacy Protection Scheme Based On Dghv

Dijk et al. proposed the DGHV scheme in 2009, which is a fully homomorphic encryption scheme based on integers. The encryption scheme only uses addition and multiplication on the integer range,

and it also employs basic modular arithmetic while ignoring the concept of the ideal lattice. Moreover, its safety is based on the difficulty of approximating the greatest common factor problem. All in all, the scheme still has the operation and efficiency of SHE.

Yi, X. et al (2012) constructed a universal single-database PIR protocol based on FHE and then extended it to a comprehensive single-database PBR protocol. First, the user generates a public key and private key pair (pk, sk) for the fully homomorphic encryption scheme; after that, the user sends pk to the database server but remains sk secret. Then, the user selects an index i where i is one and uses the public key pk to encrypt i . Finally, the ciphertext is sent as a query to the database server. According to the response generation circuit and homomorphic characteristics, the server calculates the third-digit encryption with the database, query, and pk as the response, and returns the answer to the server. Finally, the user decrypts the reaction to obtain the i -th bit.

Xiong, T. et al (2016) put forward a new parallel homomorphic encryption scheme that aims at the long time-consuming disadvantage of big data encryption in cloud computing network databases. In the cloud computing network environment, DGHV homomorphic encryption algorithm is used for preliminary data encryption, and the noise generated during the encryption process is denoised. Introducing the parallel characteristics of the MapReduce computing framework, DGHV uses the block algorithm to segment the big data in the cloud environment, and the homomorphic algorithm is employed to encrypt the data. Finally, the ciphertext in the entire database is obtained.

5.2 Other Fhe Database Privacy Protection Schemes

Zhao, F. et al (2014) proposed a cloud computing security solution based on fully homomorphic encryption of Craig Gentry's ideal lattice. However, this scheme is not optimized for proprietary scenarios, but directly applies Craig Gentry's fully homomorphic encryption algorithm. Meanwhile, it fails to optimize the problem of high resource consumption, which leads to the low portability of the solution and cannot be promoted and widely used. Gupta, C. P. et al (2013) proposed a completely homomorphic encryption scheme based on matrix operations, which uses a small size symmetric key, so it is suitable for many data-centric applications. In this scheme, if two entities possessing the decryption key and encrypted data decide to collude, this arrangement is vulnerable to collusion attacks. But as seen here, these keys are different in this scheme, so it is collusion-resistant. Besides, the role of the delegator can be borne by the data owner itself, thereby increasing the security of the key generator.

For the first time, Liu, J. Q. et al (2019) proposed an anonymization scheme based on fully homomorphic encryption and global generalization. The advantage of this scheme is that it can support multiple applications, multiple protection principles, and technical parameters, and guarantee the availability of the proposed model in terms of efficiency. Gopal, G. N. et al (2012) designed a scheme that is based on Gentry's FHE scheme, which uses the key to encrypt each keyword in the file for the query, so that the cloud can only operate on the ciphertext data and return the ciphertext without knowing the key. As a result, this method is highly dependent on the huge resources of cloud computing to overcome the performance problems of fully homomorphic encryption. Gahi, Y. et al (2015) proposed a fully homomorphic encryption scheme that is based on a relational database to protect the integrity and confidentiality of data. Still, its performance hindered the actual implementation of the system.

Mani, M. et al (2013) proposed a fully homomorphic encryption scheme that can perform algebraic processing. Among them, Murali Mani et al. divided the decryption step into two parts. First, the service provider calculates the sum of the p column values in the encrypted result table and sends it to the client. Then, the client decrypts the sum into value, such as n . After that, the client requests n' lines from the service provider (where $n' > n$, the service provider, can see n'). The service provider sorts the rows in the result table according to the encrypted p value, while maintaining any other sorting (specified in the query) and sending the top n' rows. In the end, the

client can verify that the sum of the p column values is equal to n (the service provider never knows n). In this way, the two-step process provides compact and accurate results returned to the client.

Chechulina, D. et al (2015) designed a completely homomorphic encryption scheme that is not available with practically acceptable keys and output data. This scheme uses congruence relations to avoid the increase in data size. Akavia, A. et al (2018) proposed a new secure, searchable encryption scheme. It not only allows the client to provide a search token to the server but also permits the server to search for the token to cover all documents encrypted with different keys that he/she can access. What's more, this scheme uses multikey, fully homomorphic encryption (MFHE), which is capable of operating multiple unrelated keys for input encryption.

Ying-Hua, L. et al (2014) put forward and implemented a distributed privacy protection FHE-DBIRCH model that is also based on fully homomorphic encryption. The basic idea of the algorithm is to calculate the CF_i value from the data set, and then encrypt it and transmit it to the main data set. After the data set performs certain calculations on the ciphertext, it is decrypted to construct a joint data set CF (Clustering Feature) tree. Then, the PAM (Partitioning Around Medoid) algorithm is used to cluster the leaf nodes of the main CF tree, remove outliers (i.e., sparse clusters), and merge dense clusters. Finally, broadcast the set of K groups calculated by the master site to each slave site S_i .

Tan, B. et al. (2020) design an efficient private database query (PDQ) protocol which supports compound conditions with equality and order comparisons. To this end, they first present a private comparison algorithm on encrypted integers using FHE, which scales efficiently for the length of input integers, by applying techniques from finite field theory. Then, they consider a scenario for PDQ protocols, querying for values based on a conjunction of one order and four equality conditions on key columns. The proposed algorithm and protocol are implemented and tested to determine their performance in practice. The proposed comparison algorithm takes about 25:259 seconds to compare 697 pairs of 64-bit integers using Brakerski-Gentry- Vaikuntanathan's leveled FHE scheme with single instruction multiple data (SIMD) techniques at more than 138 bits of security. This yields an amortized rate of just 36 milliseconds per comparison.

The performance analysis of each FHE-based database privacy protection technology is shown in Table 4.

6. SUMMARY AND OUTLOOK

In recent years, with the continuous development of homomorphic encryption technology, its application in database privacy protection has gradually become a hot research topic in the information security industry. This article focused on database applications and reviewed the current research status of privacy protection technology based on homomorphic encryption. At present, although there are still new homomorphic encryption schemes being proposed, the construction method has not made a breakthrough in essence. The reason is that this encryption scheme solving difficult problems based on traditional mathematics has relatively low computational efficiency. Besides, its ciphertext expansion rate and the amount of calculation are very high.

Existing database privacy protection schemes based on homomorphic encryption technology often require data owners to do a lot of preparation work in the process of data security storage, such as establishing and maintaining corresponding index tables, or implementing ciphertext operations through third-party agents, etc. The former will consume a lot of storage space and have a great impact on the user's experience, while the latter increasing the risk of data leakage due to the increase of third-party participants.

Database technology has become one of the most widely used computer technologies. Ensuring the security of data in the database is an eternal goal pursued by data owners and users. The introduction

Table 4. Performance evaluation of database privacy protection technology based on FHE

Fully Homomorphic Encryption		Safety Degree	Computational overhead	Communication overhead	Data Integrity	Data Dependency
DGHV Scheme	Yi, X. et al.	high	high	high	high	medium
	Xiong, T. et al.	medium	low	medium	low	high
Other Scheme	Zhao, F. et al.	high	high	medium	medium	high
	Gupta, C. P. et al.	medium	low	medium	high	low
	Liu, J. Q. et al.	medium	medium	low	low	high
	Gopal, G. N. et al.	medium	low	low	medium	medium
	Gahi, Y. et al.	high	high	medium	high	low
	Mani, M. et al.	high	low	low	high	high
	Chechulina. et al.	medium	medium	medium	high	medium
	Akavia, A. et al.	medium	high	high	high	medium
	Ying-Hua. et al.	high	high	medium	medium	high
	Tan, B. et al.	medium	high	high	medium	high

of homomorphic encryption methods has made people one step closer to this goal. However, since database privacy protection technologies such as homomorphic encryption have developed for a short time, there are still many problems. This paper introduces the latest research results of homomorphic encryption technology in database privacy protection, focusing on the advantages and disadvantages of related technologies, and provides a certain reference for subsequent research. Through the analysis of relevant theories and techniques, this paper believes that in future research, researchers should focus on specific usage scenarios and combine with artificial intelligence technologies that have been widely used in various fields to provide users with more secure and intelligent data services.

(1) Research on database privacy protection based on artificial intelligence

With the advent of the third wave of artificial intelligence, artificial intelligence technology has gradually been widely used in various fields, and the field of database privacy protection is no exception. In 2016, Microsoft introduced artificial intelligence technology to homomorphic encryption. It demonstrated CryptoNets where database providers can use artificial pre-feedback neural networks to perform predictive analysis based on ciphertext data, which proves that database privacy protection based on artificial intelligence technology is indeed feasible. It is nothing but the first attempt of artificial intelligence technology in database privacy protection. How other artificial intelligence techniques can be applied to the field of database privacy protection to improve the security and computing efficiency of private data is a topic worthy of study.

(2) Research on privacy protection of heterogeneous distributed databases

With the rapid development of network technology today, it often happens for large enterprises that branches in different regions use different database systems, which makes heterogeneous distributed databases more and more popular. However, because of the relatively independent and heterogeneous characteristics of each site in a distributed environment, other operations such as communication and data collaboration will become more frequent. These operations, no matter intentionally or unintentionally, pose a threat to sensitive data and private information. Therefore, the

privacy protection problem of heterogeneous distributed databases is unavoidable. In 2019, Huawei released the native database GaussDB, and the distributed storage FusionStorage 8.0, both of which are mainly optimized for computing efficiency. The heterogeneous computing innovation framework makes full use of the advantages of X86, ARM, GPU, and NPU. However, it has failed to optimize its privacy protection. Therefore, the research of database privacy protection based on heterogeneous distributed systems is also a potential topic.

(3) Research on database privacy protection based on specific application scenarios

Although databases are widely used in all fields, the specific application scenarios in each area are very different. Not only are the manifestations, storage methods, quantities, and update frequencies of data separate, but also the manifestations and amounts of private information are often different. Take the application in the location services as an example. When the user moves frequently, the location application must regularly update the data in the database to ensure data consistency. In this process, how to provide users with fast, accurate, and transparent services based on ensuring that users' private data is not leaked is a very important research topic.

ACKNOWLEDGMENT

This research was supported by the Key Research and Development Plan of Jiangxi Province [grant numbers 20181ACE50029], the National Science and technology award Reserve Cultivation Project of Jiangxi Province [grant number 20192AEI91005], and Natural Science Foundation of Guangdong Province of China [grant numbers 2020A1515010784].

REFERENCES

- Agrawal, D., & Aggarwal, C. C. (2001, May). On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 247-255). doi:10.1145/375551.375602
- Agrawal, R., & Srikant, R. (2000, May). Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (pp. 439-450). doi:10.1145/342009.335438
- Akavia, A., Feldman, D., & Shaul, H. (2018). Secure Search via Multi-Ring Fully Homomorphic Encryption. *IACR Cryptol. ePrint Arch.*, 2018, 245.
- Akbarnia, A., & Rashvand, M. (2019). Effect of temperature, water content and velocity on the quality of virgin olive oil extracted through three-phase centrifuge. *AIMS Agriculture and Food*, 4(1), 165. doi:10.3934/agrfood.2019.1.165
- Avni, H., Dolev, S., Gilboa, N., & Li, X. (2015, September). SSSDB: database with private information search. In *International Workshop on Algorithmic Aspects of Cloud Computing* (pp. 49-61). Springer. doi: doi:10.1007/978-3-319-29919-8_4
- Bai, J. (2018). New homomorphic retrieval protocol for privacy database. *Application Research of Computers*, 35(317), 892-894. doi: 10.3969/j.issn.1001-3695.2018.03.053
- Boneh, D., Gentry, C., Halevi, S., Wang, F., & Wu, D. J. (2013, June). Private database queries using somewhat homomorphic encryption. In *International Conference on Applied Cryptography and Network Security* (pp. 102-118). Springer. doi: doi:10.1007/978-3-642-38980-1_7
- Boneh, D., Kushilevitz, E., Ostrovsky, R., & Skeith, W. E. (2007, August). Public key encryption that allows PIR queries. In *Annual International Cryptology Conference* (pp. 50-67). Springer. doi: doi:10.1007/978-3-540-74143-5_4
- Brakerski, Z. (2012, August). Fully homomorphic encryption without modulus switching from classical GapSVP. In *Annual Cryptology Conference* (pp. 868-886). Springer. doi: doi:10.1007/978-3-642-32009-5_50
- Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory*, 6(3), 1-36. doi:10.1145/2633600
- Brakerski, Z., & Vaikuntanathan, V. (2011, August). Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Annual cryptology conference* (pp. 505-524). Springer. doi: doi:10.1007/978-3-642-22792-9_29
- Brakerski, Z., & Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, 43(2), 831-871. doi:10.1137/120868669
- Cao, N., Wang, C., Li, M., Ren, K., & Lou, W. (2013). Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, 25(1), 222-233. doi:10.1109/TPDS.2013.45
- Chechulina, D., Shatilov, K. A., & Krendelev, S. (2015). Fully Homomorphic Encryption for Secure Computations in Protected Database. In *FedCSIS* (pp. 125-131). Position Papers. doi:10.15439/2015f140
- Chor, B., Goldreich, O., Kushilevitz, E., & Sudan, M. (1995, October). Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science* (pp. 41-50). IEEE.
- Clifton, C., Kantarcioglu, M., & Vaidya, J. (2002, November). Defining privacy for data mining. In *National science foundation workshop on next generation data mining* (Vol. 1, No. 26, p. 1). Academic Press.
- Cominetti, E. L., & Simplicio, M. A. (2020). Fast additive partially homomorphic encryption from the approximate common divisor problem. *IEEE Transactions on Information Forensics and Security*, 15, 2988-2998.
- Coron, J. S., Mandal, A., Naccache, D., & Tibouchi, M. (2011, August). Fully homomorphic encryption over the integers with shorter public keys. In *Annual Cryptology Conference* (pp. 487-504). Springer. doi: doi:10.1007/978-3-642-22792-9_28

- Coron, J. S., Naccache, D., & Tibouchi, M. (2012, April). Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 446-464). Springer. doi: doi:10.1007/978-3-642-29011-4_27
- Damgard, I., & Jurik, M. (2001). A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. *Proceedings of Public Key Cryptography 2001*, 13-15. doi: doi:10.7146/brics.v7i45.20212
- Devi, P., Sathyalakshmi, S., & Subramanian, D. V. (2020). A comparative study on homomorphic encryption algorithms for data security in cloud environment. *International Journal of Electrical Engineering & Technology*, 11(2), 129–138.
- ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), 469–472. doi:10.1109/TIT.1985.1057074
- Gahi, Y., Guennoun, M., & El-Khatib, K. (2015). *A secure database system using homomorphic encryption schemes*. arXiv preprint arXiv:1512.03498.
- Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing* (pp. 169-178). doi: doi:10.1145/1536414.1536440
- Gentry, C., & Halevi, S. (2011, May). Implementing gentry's fully-homomorphic encryption scheme. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 129-148). Springer. doi: doi:10.1007/978-3-642-20465-4_9
- Gentry, C., & Halevi, S. (2011, October). Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science* (pp. 107-109). IEEE. doi: doi:10.1109/focs.2011.94
- Gentry, C., Sahai, A., & Waters, B. (2013, August). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference* (pp. 75-92). Springer. doi: doi:10.1007/978-3-642-40041-4_5
- Goldwasser, S., & Kharchenko, D. (2005, February). Proof of plaintext knowledge for the Ajtai-Dwork cryptosystem. In *Theory of Cryptography Conference* (pp. 529-555). Springer. doi: doi:10.1007/978-3-540-30576-7_29
- Goldwasser, S., & Micali, S. (2019). Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Providing Sound Foundations for Cryptography* (pp. 173–201). doi:10.1145/3335741.3335749
- Gopal, G. N., & Singh, M. P. (2012, July). Secure similarity based document retrieval system in cloud. In *2012 International Conference on Data Science & Engineering (ICDSE)* (pp. 154-159). IEEE. doi: doi:10.1109/icdse.2012.6282318
- Gui, Q., & Cheng, X. H. (2009). A distributed association rule mining method for privacy protection. *Microelectronics & Computer*, (9), 57-60. doi: 10.19304/j.cnki.issn1000-7180.2009.09.017
- Gupta, C. P., & Sharma, I. (2013, October). A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds. In *2013 Fourth International Conference on the Network of the Future (NoF)* (pp. 1-4). IEEE. doi: doi:10.1109/nof.2013.6724526
- Hingwe, K. K., & Bhanu, S. M. S. (2016). Hierarchical role-based access control with homomorphic encryption for database as a service. In *Proceedings of International Conference on ICT for Sustainable Development* (pp. 437-448). Springer. doi: doi:10.1007/978-981-10-0135-2_43
- Hui, L., Wenhai, S., Fenghua, L., & Boyang, W. (2014). Secure and privacy-preserving data storage service in public cloud. *Journal of Computer Research and Development*, 51(7), 1397. doi:10.7544/issn1000-1239.2014.20140115
- Jain, R., & Nayyar, A. (2020). A novel homomorphic rasd framework for secured data access and storage in cloud computing. *Open Computer Science*, 10(1), 431–443.
- Jia, Z., Pang, L., Luo, S. S., Zhang, J. Y., & Xin, Y. (2012, August). A privacy-preserving access control protocol for database as a service. In *2012 International Conference on Computer Science and Service System* (pp. 849-854). IEEE. doi: doi:10.1109/csss.2012.217

- Jiang, Y. J., Yang, B., Zhang, M. W., & Chen, X. R. (2011). Secure Computation Protocol for Private Matching and Inclusion Relation against Outsourced Database System. *Computer Science*, 38(3), 120. doi:10.3969/j.issn.1002-137X.2011.03.026
- Kabir, M. S. N., Rasool, K., Lee, W. H., Cho, S. I., & Chung, S. O. (2020). Influence of delayed cooling on the quality of tomatoes (*Solanum lycopersicum* L.) stored in a controlled chamber. *AIMS Agriculture and Food*, 5(2), 272. doi:10.3934/agrfood.2020.2.272
- Kalyani, P., Masooda, M., & Namrata, P. (2021). Preserving Privacy of Data in Distributed Systems Using Homomorphic Encryption. In *Progress in Advanced Computing and Intelligent Engineering* (pp. 306-313). Springer.
- Kawachi, A., Tanaka, K., & Xagawa, K. (2007, April). Multi-bit cryptosystems based on lattice problems. In *International Workshop on Public Key Cryptography* (pp. 315-329). Springer. doi: doi:10.1007/978-3-540-71677-8_21
- Kerschbaum, F. (2012, May). Outsourced private set intersection using homomorphic encryption. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security* (pp. 85-86). doi: doi:10.1145/2414456.2414506
- Kim, H. I., Kim, H. J., & Chang, J. W. (2019). A secure kNN query processing algorithm using homomorphic encryption on outsourced database. *Data & Knowledge Engineering*, 123, 101602. doi:10.1016/j.datak.2017.07.005
- Kogos, K. G., Filippova, K. S., & Epishkina, A. V. (2017, February). Fully homomorphic encryption schemes: The state of the art. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (pp. 463-466). IEEE. doi: doi:10.1109/eiconrus.2017.7910591
- Lamba, J., & Venkaiah, V. C. (2020). Privacy-preserving frequent itemset mining in vertically partitioned database using symmetric homomorphic encryption scheme. *International Journal of Information Privacy, Security and Integrity*, 4(3), 203–225.
- Li, J., Liu, Y., & Wu, S. (2021, May). Pipa: Privacy-preserving Password Checkup via Homomorphic Encryption. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (pp. 242-251). ACM.
- Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., & Lou, W. (2010, March). Fuzzy keyword search over encrypted data in cloud computing. In *2010 Proceedings IEEE INFOCOM*. IEEE. doi:10.1109/infcom.2010.5462196
- Li, K., Chen, Y., Li, W., He, J., & Xue, Y. (2018). Improved gene expression programming to solve the inverse problem for ordinary differential equations. *Swarm and Evolutionary Computation*, 38, 231–239. doi:10.1016/j.swevo.2017.07.005
- Li, K., Liang, Z., Yang, S., Chen, Z., Wang, H., & Lin, Z. (2019). Performance analyses of differential evolution algorithm based on dynamic fitness landscape. *International Journal of Cognitive Informatics and Natural Intelligence*, 13(1), 36–61. doi:10.4018/IJCINI.2019010104
- Li, K., Wang, H., & Li, S. (2018). A mobile node localization algorithm based on an overlapping self-adjustment mechanism. *Information Sciences*. Advance online publication. doi:10.1016/j.ins.2018.12.006
- Li, L., Yu, X. Z., Yang, Y. Q., & Zheng, L. L. (2015). Survey on homomorphic encryption technology. *Jisuanji Yingyong Yanjiu*, 32(11), 3209–3214. doi:10.3969/j.issn.1001-3695.2015.11.002
- Li, R., Qiu, L., & Zhang, D. (2019). Research on an improved coordinating method based on genetic algorithms and particle swarm optimization. *International Journal of Cognitive Informatics and Natural Intelligence*, 13(2), 18–29. doi:10.4018/IJCINI.2019040102
- Li, S. D., Kang, J., Yang, X. Y., & Dou, J. W. (2018, July). String Sorting Based Efficient Secure Database Query. *Journal of Software*, 1893–1908. doi:10.13328/j.cnki.jos.005358
- Li, S. D., Kang, J., Yang, X. Y., Dou, J. W., & Liu, X. (2018). Secure Multiparty Characters Sorting. *Chinese Journal of Computers*, 41(5), 1172–1188. doi:10.11897/SPJ.1016.2018.01172
- Li, Z. C., Yang, W., Yang Y. T., Sun, Y. F., & Liang, L. (2018). Design of Homomorphic Cloud Platform Based on Onion Encryption Model. *Computer Engineering*, (8), 5. doi: 10.19678/j.issn.1000-3428.0047251

- Li, Z. Y., Gui, X. L., Gu, Y. J., Li, X. S., Dai, H. J., & Zhang, X. J. (2018). Survey on homomorphic encryption algorithm and its application in the privacy-preserving for cloud computing. *Journal of Software*, 29(7), 1830–1851. doi:10.13328/j.cnki.jos.005354
- Liu, J. Q., Chen, F. H., Xu, C. F., Guo, H., & Li, T. (2019). Full-Domain Anonymization Algorithm Based on Fully Homomorphic Encryption in the Cloud. *Chinese Journal of Computers*, 42(4), 837–850. doi:10.11897/SP.J.1016.2019.00837
- López-Alt, A., Tromer, E., & Vaikuntanathan, V. (2012, May). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing* (pp. 1219-1234). doi: doi:10.1145/2213977.2214086
- Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 3-es. 10.1145/1217299.1217302
- Mani, M., Shah, K., & Gunda, M. (2013). *Enabling secure database as a service using fully homomorphic encryption: Challenges and opportunities*. arXiv preprint arXiv:1302.2654.
- Marwan, M., Kartit, A., & Ouahmane, H. (2016). Towards a secure cloud database using Paillier's homomorphic cryptosystem. In *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)* (pp. 360-365). IEEE. doi: doi:10.1109/icmcs.2016.7905648
- Mingjie, L., & An, W. (2014). Fully Homomorphic Encryption and Its Applications. *Journal of Computer Research and Development*, 51(12), 2593. doi:10.7544/issn1000-1239.2014.20131168
- Mora, A., Sufi, U., Roach, J. I., Thompson, J. F., & Donis-Gonzalez, I. R. (2019). *Evaluation of a small-scale desiccant-based drying system to control corn dryness during storage*. doi: 10.3934/agrfood.2019.1.136
- Morampudi, M. K., Prasad, M. V., & Raju, U. S. N. (2020). Privacy-preserving iris authentication using fully homomorphic encryption. *Multimedia Tools and Applications*, 79.
- Naccache, D., & Stern, J. (1998, November). A new public key cryptosystem based on higher residues. In *Proceedings of the 5th ACM conference on Computer and communications security* (pp. 59-66). doi: doi:10.1145/288090.288106
- Nayyar, A. (2019). *Handbook of Cloud Computing: Basic to Advance research on the concepts and design of Cloud Computing*. BPB Publications.
- Paillier, P. (1999, May). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques* (pp. 223-238). Springer. doi: doi:10.1007/3-540-48910-x_16
- Palamakumbura, S., & Usefi, H. (2015, July). Database query privacy using homomorphic encryptions. In *2015 IEEE 14th Canadian Workshop on Information Theory (CWIT)* (pp. 71-74). IEEE. doi: doi:10.1109/cwit.2015.7255155
- Pang, H., & Wang, B. (2020). Privacy-preserving association rule mining using homomorphic encryption in a multikey environment. *IEEE Systems Journal*, 15(2), 3131–3141.
- Raj, R. J. S., Prakash, M. V., Prince, T., Shankar, K., Varadarajan, V., & Nonyelu, F. (2020). Web based database security in internet of things using fully homomorphic encryption and discrete bee colony optimization. *Malaysian Journal of Computer Science*, 1–14.
- Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the Association for Computing Machinery*, 56(6), 1–40. doi:10.1145/1060590.1060603
- Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 4(11), 169-180.
- Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126. doi:10.1145/359340.359342
- Shundong, L., Jiawei, D., & Daoshun, W. (2015). Survey on homomorphic encryption and its applications to cloud security. *Journal of Computer Research and Development*, 52(6), 1378. doi:10.7544/issn1000-1239.2015.20131494

- Smart, N. P., & Vercauteren, F. (2010, May). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography* (pp. 420-443). Springer. doi: doi:10.1007/978-3-642-13013-7_25
- Smart, N. P., & Vercauteren, F. (2010, May). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography* (pp. 420-443). Springer. doi: doi:10.1007/978-3-642-13013-7_25
- Stehlé, D., & Steinfeld, R. (2010, December). Faster fully homomorphic encryption. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 377-394). Springer. doi: doi:10.1007/978-3-642-17373-8_22
- Stephen, J. J., Savvides, S., Seidel, R., & Eugster, P. (2014). Practical Confidentiality Preserving Big Data Analysis. *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*.
- Tan, B. H. M., Lee, H. T., Wang, H., Ren, S. Q., & Khin, A. M. M. (2020). Efficient private comparison queries over encrypted databases using fully homomorphic encryption with finite fields. *IEEE Transactions on Dependable and Secure Computing*.
- Tetali, S. D., Lesani, M., Majumdar, R., & Millstein, T. (2013, October). MrCrypt: Static analysis for secure cloud computations. In *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications* (pp. 271-286). doi: doi:10.1145/2544173.2509554
- Tian, H., Zhang, Y., Li, C., & Xing, C. X. (2017). A Survey of Confidentiality Protection for Cloud Databases. *Chinese Journal of Computers*, 10, 2245–2270. doi:10.11897/SP.J.1016.2017.02245
- Tu, S. L., Kaashoek, M. F., Madden, S. R., & Zeldovich, N. (2013). *Processing analytical queries over encrypted data*. doi: 10.14778/2535573.2488336
- Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010, May). Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 24-43). Springer.
- Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1), 50–57. doi:10.1145/974121.974131
- Wang, L., Yang, Z., & Song, X. (2020). SHAMC: A Secure and highly available database system in multi-cloud environment. *Future Generation Computer Systems*, 105, 873–883. doi:10.1016/j.future.2017.07.011
- Wang, S., Agrawal, D., & El Abbadi, A. (2014). Towards practical private processing of database queries over public data. *Distributed and Parallel Databases*, 32(1), 65–89. doi:10.1007/s10619-012-7118-y
- Wang, X. F., & Wang, S. P. (2002). A study of database encryption methods. *Journal of XAUT*, 18(3), 263–268. doi:10.3969/j.issn.1006-4710.2002.03.011
- Wei, Z. Z., Yang, Y. T., & Chen, Z. W. (2013). Ciphertext retrieval in database based on RSA's multiplicative homomorphism. *Journal of Harbin Engineering University*, 5, 641–645. doi:10.3969/j.issn.1006-7043.201206052
- Wu, P., & Zhang, Y. (2006). An overview of Database security. *Computer Engineering*, 32(12), 85–88. doi:10.3969/j.issn.1000-3428.2006.12.033
- Wu, W., Liu, J., Wang, H., Hao, J., & Xian, M. (2020). Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique. *IEEE Transactions on Knowledge and Data Engineering*.
- Xiao, X., & Tao, Y. (2006, June). Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (pp. 229-240). doi:10.1145/1142473.1142500
- Xiao, X., & Tao, Y. (2007, June). M-invariance: towards privacy preserving re-publication of dynamic datasets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 689-700). doi:10.1145/1247480.1247556
- Xiong, T., Wang, Y., & Mei, Y. (2016). Network database encryption optimization model simulation analysis of cloud computing. *Kexue Jishu Yu Gongcheng*, 16(21), 299–302. doi:10.3969/j.issn.1671-1815.2016.21.052

- Yang, Y. B. (2007). A universal and efficient database data encryption design. *Journal of Computer Application*, 27(B06), 242-244. doi: 1001-9081(2007)S1-0242-03
- Yi, X., Kaosar, M. G., Paulet, R., & Bertino, E. (2012). Single-database private information retrieval from fully homomorphic encryption. *IEEE Transactions on Knowledge and Data Engineering*, 25(5), 1125–1134. doi:10.3969/j.issn.1000-3428.2012.24.001
- Yihua, Z., Wen, J., & Yuguang, Y. (2019). Database Ciphertext Retrieval Scheme Based on f-mOPE. *Dianzi Yu Xinxi Xuebao*, 41, 8. doi:10.11999/JEIT180805
- Ying-Hua, L. I. U. (2014). Research on privacy preserving FHE-DBIRCH model. *Computer Engineering & Science*, (7), 32. doi: .10.3969/j.issn.1007-130X.2014.07.029
- Yu, Z., & Bao, G. (2013). Method based on homomorphic hash for data changes capture of LDAP directory. *Jisuanji Yingyong Yanjiu*, 30(7), 2007–2009. doi:10.3969/j.issn.1001-3695.2013.07.022
- Yu, Z. B., & Zhou, Y. H. (2015). Keyword based privacy-preserving retrieval over cloud encrypted data. *Computer Science*, (S1), 365-369. doi: CNKI:SUN:JSJA.0.2015-S1-091
- Yuan, X. P., Zhong, H., Huang, H. S., & Yi, L. (2011). Secure Query Protocol for String Approximate Matching. *Computer Engineering*, 37(20), 142–144. doi:10.3969/j.issn.1000-3428.2011.20.049
- Zhang, N., & Zhao, W. (2005, August). Distributed privacy preserving information sharing. In *Proceedings of the 31st international conference on Very large data bases* (pp. 889-900). Academic Press.
- Zhao, F., Li, C., & Liu, C. F. (2014, February). A cloud computing security solution based on fully homomorphic encryption. In *16th international conference on advanced communication technology* (pp. 485-488). IEEE. doi: doi:10.1109/icact.2014.6779008
- Zhao, J., Ma, Y., Cui, J., Peng, Y., Li, K., & Wang, T. (2021). SecSky: A Secure Dynamic Skyline Query Scheme With Data Privacy. *IEEE Access: Practical Innovations, Open Solutions*, 9, 5690–5703. doi:10.1109/ACCESS.2020.3047950
- Zhou, S. G., Li, F., Tao, Y. F., & Xiao, X. K. (2009). Privacy preservation in database applications: A survey. *Chinese Journal of Computers*, 32(5), 847–861. doi:10.3724/SP.J.1016.2009.00847

ENDNOTES

¹ <https://chuangyi.chuangyety.com/jxdz/20191202/120220555.html>

² https://www.sohu.com/a/406840310_588599

Kangshun Li is currently a Full Professor from the South China Agricultural University, Guangzhou, China. He received his BSc in Computational Mathematics from the Nanchang University, Nanchang, China, and PhD in Computer Software and Theory from the Wuhan University, Wuhan, China, in 1983 and 2006, respectively. His current research interests are intelligent computation, image identification, data mining, soft engineering, embedded system, evolvable hardware, evolutionary modelling, parallel computation, and neural network.

Yuanlong Cao received the B.S. degree in the Computer Science and Technology from Nanchang University, China, in 2006, the M.S. degree in the Software Engineering from the Beijing University of Posts and Telecommunications (BUPT), in 2008, and the Ph.D. degree from the Institute of Network Technology, BUPT, in 2014. He was an Intern/Software Engineer with BEA TTC, IBM CDL, and DT Research, Beijing, from 2007 to 2011. He is currently an Associate Professor with the School of Software, Jiangxi Normal University, China. His research interests include multimedia communications and the next-generation Internet technology. He has served as the Technical Reviewer for several journals, including the IEEE Transactions on Industrial Informatics, IEEE Transactions on Cognitive Communications and Networking, IEEE Access, Computer Communications, Journal of Network and Computer Applications.

Zhang Youcheng was awarded the Masters, Nanjing University of Aeronautics and Astronautics. Chairman of Nanjing Unary Information Technology Co. Ltd.