IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 28, 2020, accepted July 12, 2020, date of publication July 16, 2020, date of current version July 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009876

Data Security and Privacy Protection for Cloud Storage: A Survey

PAN YANG[1], NAIXUE XIONG[2], (Senior Member, IEEE), AND JINGLI REN[1]

[1]Henan Academy of Big Data, School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China
[2]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding author: Jingli Ren (renjl@zzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11771407, in part by the Chinese Academy of Engineering Advisory Project under Grant 2020-ZD-16, in part by the MOST Innovation Method Project under Grant 2019IM050400, and in part by the Key Discipline Construction Projects of Zhengzhou University under Grant XKZDQY202004.

ABSTRACT The new development trends including Internet of Things (IoT), smart city, enterprises digital transformation and world's digital economy are at the top of the tide. The continuous growth of data storage pressure drives the rapid development of the entire storage market on account of massive data generated. By providing data storage and management, cloud storage system becomes an indispensable part of the new era. Currently, the governments, enterprises and individual users are actively migrating their data to the cloud. Such a huge amount of data can create magnanimous wealth. However, this increases the possible risk, for instance, unauthorized access, data leakage, sensitive information disclosure and privacy disclosure. Although there are some studies on data security and privacy protection, there is still a lack of systematic surveys on the subject in cloud storage system. In this paper, we make a comprehensive review of the literatures on data security and privacy issues, data encryption technology, and applicable countermeasures in cloud storage system. Specifically, we first make an overview of cloud storage, including definition, classification, architecture and applications. Secondly, we give a detailed analysis on challenges and requirements of data security and privacy protection in cloud storage system. Thirdly, data encryption technologies and protection methods are summarized. Finally, we discuss several open research topics of data security for cloud storage.

INDEX TERMS Cloud storage, data security, cryptography, access control, privacy protection.

I. INTRODUCTION

With the rise of the Internet of Things (IoT), the number of information sensing devices connected to the Internet is increasing to realize the interconnection among people, devices and ''things''. A new forecast by IDC [80] estimates that there will be 41.6 billion internet of things devices or ''things'' in 2025, generating 79.4 zettabytes (ZB) of data. Not only that, people are still committed to improving the efficiency of data collection of devices in IoT, see, [59], [79]. The unprecedented amount of data is generated and hosted on the cloud service provider platform [78]. Due to the high performance, scalable and reliable datacenters of the cloud, many of the smart city applications and services will be hosted in the Cloud. Therefore, smart city residents and service providers can rely on cloud services to host, build and/or deploy their smart city services and applications [39].

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

Besides, the advantage of pay-as-you-go makes most traditional enterprises actively migrate data to the cloud. Cloud is not only the destination of workload, but also provides efficient operation practice, which makes enterprises have higher agility and flexibility. This has promoted both enterprises digital transformation and network modernization transformation [19]. In 2019, the Digital Economy Report released by the United Nations emphasizes that the digital economy is becoming an important driving force for economic development. According to incomplete statistics, the digital economy accounts for 4.5% to 15.5% of the world GDP [25]. Cloud computing is conducive to promoting the deep integration of Internet, big data, artificial intelligence and real economy, and is the core of accelerating the construction of modern economic system. According to Gartner, Inc. [34], the worldwide public cloud service market will grow by 17% in 2020, reaching $266.4 billion, up from $227.8 billion in 2019. Taken together, cloud application is still the mainstream.

Cloud storage is essentially a cloud computing system that allows users to store and share data on the Internet. The advantages of cloud storage include unlimited data storage space, convenient, safe and efficient file accessibility and offsite backup, and low cost of use. Cloud storage can be divided into five categories in practical applications, namely, public cloud storage, personal cloud storage, private cloud storage, hybrid cloud storage and community cloud storage. In public cloud, enterprises outsource data storage business to cloud storage providers (for instance AWS and Alibaba Cloud) without having to deploy infrastructures and maintain servers. The data can be accessed only by authorized user. The advantages of public cloud such as flexibility, scalability and cost saving attract plenty of small and medium enterprises. Personal cloud, also known as mobile cloud storage, is essentially a branch of public cloud, but differ from public cloud, it provides public cloud storage services for individual users. In private cloud, enterprises need to deploy cloud storage infrastructures and arrange professional staff to manage and maintain servers. This ensures that the private cloud has higher security than the public cloud and the control of data is in the hands of the enterprise itself. But the cost increases dramatically. This storage model is more suitable for large enterprises with large amount of expensive and sensitive data. Hybrid cloud is a combination of public cloud and private cloud, which inherits all the advantages of both. Enterprises can store expensive and sensitive data in private cloud and other data in public cloud. The appeal of this storage model continues to grow. As a new cloud storage mode in recent years, community cloud is very suitable for medical and financial industries. Community cloud provides cloud services for several businesses in a specific community. Usually these businesses have the same concerns or need to work together on some projects. Infrastructure construction and server management can be jointly undertaken by community Cloud members or outsourced to a third party.

From the perspective of storage architecture, the major cloud platforms typically offer three broad classes of storage: block storage, file storage and object storage [47]. 1) Cloud block storage, respected by Storage Area Networks (SAN), in essence provides a virtualized Storage Area Network with logical volume management provisioning via a simplified web services interface. 2) File storage, which is also referred to as file-level or file-based storage, is normally associated with Network Attached Storage (NAS) technology [73]. With the file system, file storage manages the sharing data and access to data stored on it more flexibly than block storage. Massive data brings a series of challenges to enterprises, such as storage expansion, data sharing, efficient transmission, cost and data security, when data storage reaches the PB level, the limitation of by NAS and SAN directly leads to the increase of equipment maintenance cost in the later period. They are unable to fully meet the enterprise's requirements for the reliability, availability, security and other indicators of mass storage data in that object storage is more critical. 3) Object storage, such as AWS S3, is optimized for storing large volumes of unstructured data.

Cloud storage is based on virtualization infrastructure and is similar to cloud computing in terms of accessible interfaces, scalability and measurement resources. It consists of four layers [116], which can be summarized as follows: 1) The storage layer, the basic part of cloud storage, is made up to storage devices and a unified storage device management comprise. 2) The primary management layer is the core part of cloud storage, and also the most challenging part of cloud storage. 3) The application interface layer is the most flexible part of cloud storage. 4) The last one is the access layer. From this point of view, cloud storage supplies data access services including data storage, data computation, authentication, and access control. Due to the characteristics of cloud storage, data security and privacy issues are inevitably generated in this process. The requirements of data security in cloud storage are mainly shown in the following aspects [8], [61], [93], [94], [108]:

- Data Confidentiality: Data confidentiality refers to prevent the active attack of unauthorized parties on users' data, and ensure that the information received by the data receiver is completely consistent with the information sent by the sender. That is to mean, only authorized people are entitled to access and obtain the data. Imagine your bank account. You should be able to access them, of course, and employees at the bank who are helping you with a transaction should be able to access them, but no one else should. Once accessed by others, data confidentiality is compromised, which is irreversible.
- Data Integrity: Data integrity is the reliability of the data, that is, the data can not be arbitrarily tampered with and replaced. For example, if you're shopping online on Amazon, someone can change the items in your cart without your authorization. The absence of data integrity can pose serious security issues.
- Data Availability: Data availability emphasizes that data can be accessed normally at any time, namely user can access, download, or do some modifications on data in the cloud as soon as they need it.
- Fine-Grained Access Control.
- Secure Data Sharing in Dynamic Group.
- Leakage-Resistant.
- Completely Data Deletion: When users no longer use cloud storage, they can completely delete the data outsourced to the cloud server and confirm that the data has been completely destroyed, instead of being cheated by malicious cloud service providers.
- Privacy Protection: While users enjoy the convenience of cloud storage, the cloud storage providers have captured their privacy information, such as personal identity, location, and sensitive data for the enterprise. Privacy security mechanisms are used to guarantee these data to be secret under curious adversaries and malicious employees of cloud service providers.

With the further centralization of data and the increase of data volume, it becomes problematic to secure data in cloud storage. Therefore, how to ensure that users and their information resources are not exposed will be a major concern of cloud service providers and scholars for a long time. However, the existing information security methods are no longer meet the information security requirements in the era of big data, and security threats will gradually become the bottleneck restricting the development of big data technology. In fact, data storage security includes static data security and dynamic data security in cloud storage. Static data security is to ensure the security of static data on the cloud storage system, while dynamic storage security is to ensure the integrity and confidentiality during data transmission. Data is transmitted through the IP network in the cloud storage, so security threats on the traditional network also exist in the cloud storage system, such as data destruction, data theft, data tampering, denial of service, etc., affecting the safe storage of data. In cloud storage system, users' data may be distributed across multiple servers, and each server may be shared by multiple users, which leads to the increasing risk of unauthorized access undesirably. Complex encryption algorithms are not friendly resources-limited users, so it is a practical problem to ensure that they can operate on their own devices. In addition, it should be high probability for users's devices to be under the side channel attack is very high. In summary, the data security and privacy-preserving in cloud storage system mainly faced with the following challenges:

- Fine-grained data access control.
- Malicious cloud service providers may return incorrect integrity audit results.
- Side channel attack.
- Malicious cloud service providers do not comply with customers' requests to completely delete data in the cloud.
- Privacy-preserving.

Although cloud storage has developed for many years, it is still very important in the Internet of Things, smart city and digital economy. Data security and privacy protection in cloud storage are still of great importance, which inspires us to present this review. we make a comprehensive review of the literature on data security and privacy issues, data encryption technology, and applicable countermeasures in cloud storage system. The main contributions of this paper are as follows

- We first make an overview of cloud storage, including definition, classification, architecture and applications.
- We give a detailed analysis of data security and privacy issues and mechanisms in cloud storage system.
- Data encryption technologies and protection methods are summarized. These correspond to the security requirements we mentioned earlier.
- We discuss several open research topics of data security for cloud storage.

The remainder of this article is organized as follows. Section II and Section III present the cryptography-based techniques and the state of art involved in data security and privacy-preserving, respectively. In Section IV, we discuss the clear research direction of cloud storage. Finally, we draw our conclusion in Section V.

II. DATA ENCRYPTION TECHNOLOGY

When data is outsourced to the cloud, its security is vulnerable. Encryption is an effective technique to protect data security. The essence of data encryption is to transform the original plaintext file or data into an string of unreadable code by some algorithms, which is usually called ciphertext. Even if someone intercepts the garbled code, he/she can't use the garbled code to get the original content, which effectively protects the confidentiality of the data and prevents the data from being tampered. Users who are authorized to access can decrypt the file with the corresponding private key, and then update, modify the ciphertext. Encryption is divided into symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key to encrypt and decrypt data. However, before using symmetric encryption, users need to determine a consensus key, which is very inconvenient for multi-user sharing files. By comparison, the asymmetric encryption, also known as public key encryption, is more convenient. Public key encryption contains a pair of keys. The public key that can be disclosed to others for encrypting files, while the private key is used for decrypting the ciphertext. In this section, we present some encryption technologies that are widely applied in cloud storage system.

A. IBE: IDENTITY-BASED ENCRYPTION

In the traditional PKI (Public Key Infrastructure), in order to confirm that the identity information is consistent with the public key used for encryption, the sender needs to authenticate the identity information of the receiver through a trusted third-party Certificate Authority (CA) before encrypting a file with the public key. This process may lead to the sender's workload significantly increased when he wants to share data with multiple receivers. In order to solve this problem, the concept of identity based cryptography was proposed by Shamir [68] in 1984. The idea is to associate the user's identity information with the public key, so that there is no need to verify the receiver's certificate before encryption. In 2001, Boneh and Franklin [12] formally gave the definition and security model of Identity-Based Encryption IBE, and applied bilinear map to construct a secure IBE scheme in their seminal paper. In such a system, Alice is a sender wants to send an encrypted message to Bob. Private Key Generator (PKG), a trusted third party, is required to generate the corresponding public key and private key. First, in order to encrypt the message, Alice utilizes the receiver's unique identity information (Bob's e-mail: Bob@g.com) to generate the public key from PKG. Then Alice sends the encrypted message to Bob. The receiver Bob contacts the PKG and authenticates to obtain the corresponding private key. The Fig. 1 shows how the identity-based encryption works. Soon afterwards many scholars improved the IBE. Boneh and

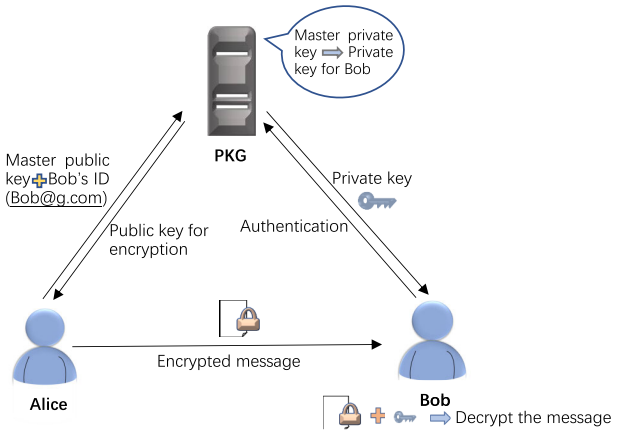

FIGURE 1. Identity-based encryption.

Boyen [10] got the chosen security of IBE system under the standard model, and the full security IBE scheme was studied by [11], [37], [85].

The revocable IBE revocation algorithm usually takes the public parameter PP, user ID, revocation list RL, revocation time t and state st as input, and the updated revocation list as output. See Algorithm 1.

Algorithm 1 Revoke

Input: PP, ID, RL, t, st
Output: The updated RL

1: $RL \leftarrow RL \cup \{(ID, t)\}$
2: **return** RL

Reference [12] proposed the first IBE scheme with revocation of public keys. By defining the public key as "ID + validity period", the receiver is allowed to use the private key to decrypt in a certain period. After the validity period is exceeded, the receiver needs to apply to PKG for updating the private key to obtain the decryption permission again. Once the public key of someone is revoked, PKG will not update the private key for him or her. No matter how many times the private key is updated, only the receiver needs to interact with PKG, while the sender does not. This scheme greatly improves the practicality of identity-based encryption. In 2015, Li *et al.* [51] improved the result of [12] with introducing outsourced computation into IBE revocation and showed the security definition of outsourcing revocable IBE for the first time. In this scheme, PKG no longer undertakes the task of key update except to send a private key for decryption to the user at the beginning. This private key contains identity component $IK[ID]$ and time component $TK[ID]_{T_i}$, where T_i means that $TK[ID]_{T_i}$ is valid during the period T_i. The Key Update Public Cloud Service Provider (KU-CSP) is responsible for updating time components for users who are not revoked. KU-CSP terminates updating T_i for revoked user as soon as he/she submits revocation application to PKG. Later, Boldyreva *et al.* [9] used binary trees to manage identities for effective revocation.

When a user revokes his/her identity, the data owner usually update the ciphertext to ensure that the user can no longer access the previously available data and the subsequently shared data. This period involves a decryption–re-encryption–upload process. This process not only increases the exposure of private key, but also increases the computing cost and time cost of data owner. To solve this problem, Wei *et al.* [90] defined a searchable storage IBE that can protect "forward security" + "backward security", which can also resist private key exposure. In this scheme, each ID is randomly assigned to a leaf node. Unrevoked user has a node $\theta \in Path(\eta) \cap KUNodes(BT, Rl, T)$ in a certain period T, which allows the user to obtain the decryption key by re-randomizing private key $(\theta, SK_{ID,\theta})$ and update key $(\theta, KU_{T,\theta})$, while for the revoked user, the decryption key cannot be obtained without θ. Lee [50] found that when a ciphertext is updated from periodic T to periodic $T+1$, its plaintext is not available by the decryption key at time $T+1$. They improved the scheme with the method in [49].

B. ABE: ATTRIBUTE-BASED ENCRYPTION

In identity based encryption scheme, identity is a meaningful string, which is different from each other. However, the flexibility of IBE scheme runs into bottlenecks when the ciphertext is to be legally accessed by multiple users. In 2005, Sahai and Waters [67] proposed the fuzzy identity-based encryption in the first time, which is the origin of attribute based encryption (ABE). Different from identity based encryption, identity is replaced by a set of attributes in the attribute based encryption, and only users whose attribute set matches the access policy can access the encrypted data. Generally, ABE algorithm consists of four parts:

1) Setup phase, also known as the system initialization phase, in which pertinent security parameters are input and corresponding public parameters (PK) and master key (MK) are generated;
2) KeyGen stage, namely the key generation stage, data owner submit their own attributes to the system to obtain the private key associated with the attributes;
3) Encryption phase, the data owner encrypts the data by his/her public key and get the ciphertext (CT) and sends it to the receiver or to the public cloud.
4) Decryption phase, decryption users get ciphertext, decryption with their own private key SK.

ABE is promising to provide fine-grained access control over encrypted files in the data sharing applications, in that the data owner can specify who can access the encrypted data. It is mainly divided into two categories: Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE).

In 2006, Goyal and Pandey [40] developed KP-ABE. In the KP-ABE system, each ciphertext is associated with a set of attributes, while the use's private key is related to an access policy for the attributes. For instance, C1 is a ciphertext encrypted by a set of attributes ("Student", "Applied Mathematics") (see Fig. 2). The access policy of

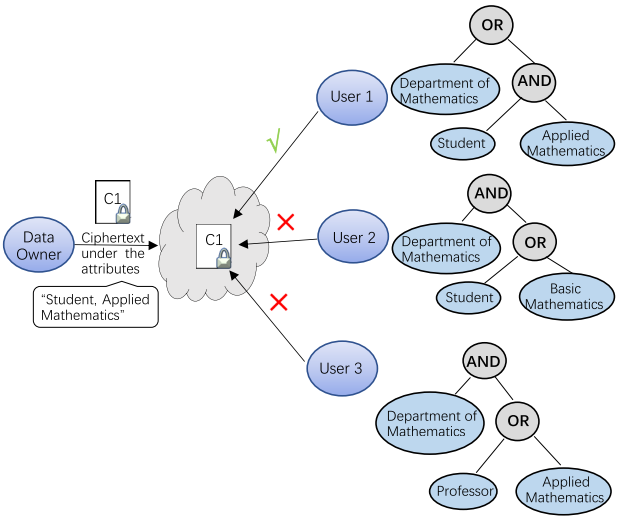

FIGURE 2. KP-ABE in cloud.

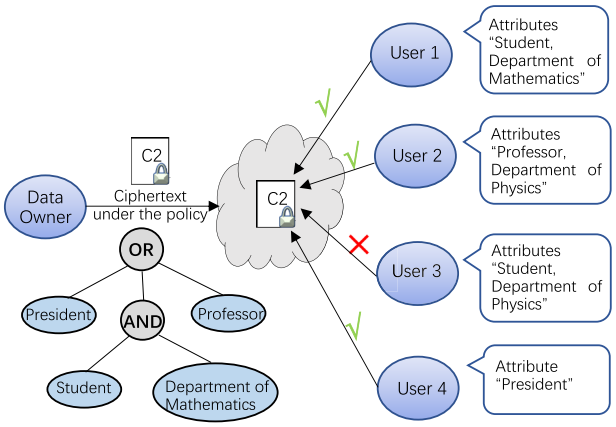

FIGURE 3. CP-ABE in Cloud.

user 1 is ''('Department of Mathematics') OR ('Student' AND 'Applied Mathematics')''. Obviously, the attributes contained in the ciphertext C1 satisfy the access policy of user 1, so he has the privilege to decrypt C1. While user 2 can decrypt the ciphertext with attributes (''Department of Mathematics'', ''Student'') OR (''Department of Mathematics'', ''Basic Mathematics''), but not C1. In the same way, user 3 can't decrypt C1, either.

In 2007, Bethencourt *et al.* [7] provided the first construction of CP-ABE. In CP-ABE, the policy is embedded in the ciphertext, and data owner can define the access policy to determine which attributes the person with can access the ciphertext. User's private key is related to the set of corresponding attributes. From a mathematical point of view, access structures can be seen as a monotonic '' access tree'', and its nodes consist of threshold gates and the leaves describe attributes. For example, a sensitive file is encrypted by an access policy ''('President') OR ('Student' AND 'Department of mathematics') OR ('Professor')'', which implies that only someone with attributes (''President'') or (''Student'', ''Department of Mathematics'') or (''President'') can access the file (see Fig. 3). Cheung and Newport [21] presented an improved scheme based on [7], which is proved to be CPA secure and CCA secure under the Decisional Bilinear Diffie-Hellman (DBDH) assumption.

The attributes of user may change for various reasons. For instance, one transfers from one job to another. Attributes changes mean that one may not be unqualified for accessing data that were previously authorized. In addition, the malicious behavior (such as collude with hackers) of some authorized users may disclose the confidentiality and privacy of the data, which makes data owner suffer losses. Therefore, a secure revocation in ABE is necessary. Existing revocation schemes can be divided into indirect revocation (see [3], [9], [58], [98]) and direct revocation (see [71], [107]). In indirect schemes, trusted authority periodically interacts with non-revoked users and updates the decryption key for them, while revoked user's decryption key is invalid. This implies an indirect revocation. Xu *et al.* [98] drew on the idea of revocation in [9], [67]. Namely, the decryption key consists of two parts, long-term secret key and update key, and the update key needs to be updated regularly. The difference is that the attribute set will be divided into two disjoint sets, each one combines with the master key to generate a secret key, respectively. The two secret keys are different and have the property of re-randomization, so that decryption key exposure resistance can be achieved. Besides, the tree-based data structure is introduced to reduce computational burden for key generation centre.

On the other hand, in direct revocation schemes, trusted authority generates a revocation list including all revoked users, which is public for every user. Data owner specifies the revoked users directly in ciphertext so that all contained revoked users cannot decrypt this ciphertext, even if their attributes (or access policies) match the access policy (or attribute set) embedded in ciphertext. Shi *et al.* [71] presented a KP-ABE scheme with direct revocation and verifiable ciphertext delegation. In their scheme, trusted authority revokes users via updating revocation list and any interaction with non-revoked users at the same time. After receiving the new revocation list, the third party (such as cloud service provider) updates the ciphertext using public information, and this ensure the new ciphertext cannot be decrypted by revoked users. Finally, any authorized auditor has the privilege to verify if the third party has updated the ciphertext correctly. This scheme not only forbids revoked users to decrypt the new ciphertext, but also provides verifiable function for data owners to ensure that ciphertext has been updated under the new revocation list. In 2016, Ma *et al.* [60] improved [71]. With the technology from [64], they achieve large universe construction, where the size of attributes is not limited and can be exponentially large, and new attributes can be added into the system. Xiong *et al.* [96] proposed a CP-ABE scheme gathering properties on direct revocation, partially hidden policy and outsourced decryption.

In general, only key revocation does not prevent users from using the old private key to decrypt the previously accessible ciphertext. In order to restrict the illegal access of revoked users, the data owner will update the access policy or re encrypt the ciphertext. When it comes to the dynamic sharing of many people, this scheme is obviously inefficient. To solve this problem, the concept of revocable storage is proposed, which support both key revocation and ciphertext update. In 2012, Sahai *et al.* [66] presented a practical revocable storage attribute based encryption, where the database will regularly update the stored ciphertext with the available public information, and any revoked user will lose access privileges after the ciphertext is updated. Recently, Wei *et al.* [89] considered secure sharing and dynamic access revocation of the EHR data in public cloud. Both forward security and backward security [90] are obtained simultaneously.

In the existing ABE schemes, a great deal of attributes lead to a large scale of access policy, and the ciphertext size of most ABE schemes increase with the complexity of access policies. As a result, ciphertext redundancy has increased significantly, which not only cause expensive computation when user have to decrypt the ciphertext by local device, but also increases users' workload. This is especially unfriendly for resource-constrained users. To solve this problem, Many Abe schemes are proposed to reduce the burden of resource-constrained users. For example, outsourcing computing to cloud service providers [45], [53], designing ciphertext of constant size, compacting policy [83] and improving policy management [87]. More concretly, Li *et al.* [53] presented an outsourcing KP-ABE scheme with efficient query processing, which implements outsourcing key-issuing and outsourcing decryption. The data owner uploads the ciphertext with a keyword set to the storage cloud service provider. Users submit a trap door for a keyword such as ''book'' to the cloud service providers to request keyword search. After receiving the client's request, cloud service provider immediately performs partial decryption and keyword search on the ciphertext, and returns the matching results to the user. Outsourcing decryption enables users to save a lot of computing resources on the premise of maintaining confidentiality of data. Using trapdoor instead of keyword plaintext to perform query processing avoids cloud service provider using cookie records to pry into users' privacy and preferences. Wang *et al.* [84] compact the scale of access policy through greedy compacting algorithm, so that the ciphertext redundancy can be reduced due to the decreased policy scale. Multiple users share the public policy nodes. By introducing flexible factor and overlap factor, the policy-computing efficiency and compact ratio are analyzed. Policy-compacting fundamentally solves the problem of ciphertext redundancy caused by the large scale of policy, which is of great significance to improve the performance of Abe scheme. In order to improve the scalability of CP-ABE scheme, Wang *et al.* [83] designed an scalable access policy based on the idea of blocked linear secret sharing scheme (BLSSS), which has lower storage costs, computation and communication overhead. A comparison of ABE schemes mentioned above is showed in Table 1.

C. HOMOMORPHIC ENCRYPTION

Although the identity based encryption and attribute based encryption introduced earlier can guarantee the confidentiality of data in the cloud to a certain extent, they have some drawbacks. If a user needs to update his encrypted files stored in the cloud, he has two methods. One is to modify the ciphertext in the cloud. However, after the modified ciphertext is decrypted, it will usually become meaningless garbled code and cause data damage. The other is to update the decrypted file, and send the encrypted new file to the cloud. This is very complex and cumbersome. If his file contains a large amount of data, the process of downloading, decrypting and encrypting will not only take a lot of time, but also have a high demand for the computing power of the user's local device. In addition, the transmission process from local to cloud also brings the risk of data leakage. To solve this problem, homomorphic encryption shows great superiority. Homomorphic encryption is a kind of public key encryption, which allows users to perform certain algebraic operations on ciphertext and still get the encrypted text, and the result after the ciphertext is decrypted is consistent with the result of the same operation on plaintext. With Fig. 4 and table it's easier for us to understand how homomorphic encryption works in cloud. Data owner encrypt the file by homomorphic encryption and send it to the cloud server. The authorized users can decrypt the ciphertext with the corresponding private keys. If user 2 wants to perform some specific operations on ciphertext, the only thing he needs to do is send the functions corresponding to the operations to the cloud server. The server get operand and perform the operation without decrypt the ciphertext and return the encrypted result to user 2. Homomorphic encryption effectively protects the security of outsourced data.

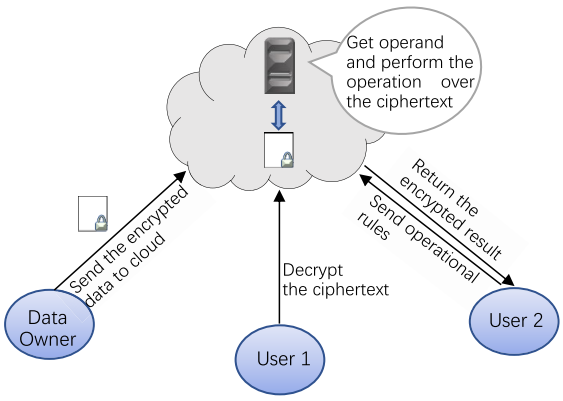

FIGURE 4. Homomorphic encryption in cloud.

From the point of view of mathematics, homomorphic encryption embodies the concept of homomorphism [32]. Given a homomorphism $f : A \rightarrow A^*$ is a structure-preserving

TABLE 1. Comparison of ABE schemes.

Categories	Categories	Proposed Schemes	Technical Methods	Advantages	Reference
Revocation	Direct Revocation	Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation	• KP-ABE • Linear secret sharing scheme	• Direct revocation • Secure ciphertext delegation • Verifiable update • Selective security	[71]
		Directly revocable and verifiable key-policy ABE scheme for large universe	• KP-ABE • Linear secret-sharing scheme • Subset difference method	• Direct revocation • Verifiable update • Large universe construction • Selective security	[60]
		Partially policy-hidden attribute-based broadcast encryption with secure delegation	• CP-ABE • Linear secret-sharing scheme • Broadcast encryption scheme • Outsourcing approach	• Direct revocation • Partially hidden policy • Verifiable outsourced decryption	[96]
	Indirect Revocation	Revocable attribute-based encryption with decryption key exposure resistanpce and ciphertext delegation	• CP-ABE • Tree-Based Revocation Approach • ABE scheme in [65]	• Indirect revocation • Randomizable piecewise key generationp • Ciphertext delegation • Decryption key exposure resistance	[98]
Revocable Storage		Dynamic credentials and ciphertext delegation for attribute-based encryption	• KP-ABE & CP-ABE • Linear secret-sharing scheme	• Revocable storage • Ciphertext delegation • Fully security	[66]
Reduction in Computation Overhead	Outsourcing Computation	Outsourced attribute-based encryption with keyword search function	• KP-ABE • Secret sharing scheme based on Lagrangep interpolation method	• Lower computation overhead for client • Efficient keyword search processing • CPA-secure	[53]
	Compacting Policy	Compact ciphertext-policy attribute-based encryption	• CP-ABE • Greedy compacting algorithm • Policy compacting method	• Reduction inp ciphertext redundancy • Lower storage and computation overhead	[83]
	Improving Policy Management	Scalable ciphertext-policy attribute-basedp encryption	• CP-ABE • Blocked linear secret sharing scheme	• Lower storage cost • Lower computation and communication overhead • Collision-resilient	[84]

map between sets A and A^* with the composition operations $\circ$ and $\bullet$, respectively. Let $a,\ b,\ c \in A$, with $c = a \circ b$ and $a^* = f(a),\ b^* = f(b),\ c^* = f(c) \in A^*$. Based on the above assumptions, we can get $f(a \circ b) = f(a) \bullet f(b)$. Consider that the homomorphism $f(\cdot)$ is a one-to-one mapping and represents the encryption procedure and A is the data set consists of our data stored in the cloud; f^{-1}, the inverse of f with $a = f^{-1}(a^*),\ b = f^{-1}(b^*),\ c = f^{-1}(c^*)$, is the decryption procedure and the composition operations are the specific types of computations carried out with ciphertext. The work principle of homomorphic encryption is show in Table 2.

According to the computing power of ciphertext, homomorphic encryption can be divided into three categories: Partial Homomorphic Encryption (PHE, also known as semi homomorphic encryption), Somewhat Homomorphic Encryption (SHE) and Full Homomorphic Encryption (FHE).

PHE refers that one operation is allowed to be performed on ciphertext, addition homomorphism or multiplication homomorphism, not both. To support the additive homomorphism on ciphertext, a classical scheme of additive homomorphic encryption was proposed by Paillier [63]. Fast decryption scheme based on Paillier homomorphic was present by El Makkaoui *et al.* [30]. The unique feature of

TABLE 2. Mapping representation of homomorphic encryption.

PLAN 1	PLAN 2
Choose one operation (here is $\bullet$) on the ciphertext and then decrypt	Carry out the same operation on the plaintext
$a^* \bullet b^* = f(a) \bullet f(b) = f(a \circ b)$ $f^{-1}(a^* \bullet b^*) = f^{-1}(f(a \circ b)) = a \circ b$	$f^{-1}(a^*) \circ f^{-1}(b^*) = a \circ b$

this scheme is that the private key is used for encrypting and decrypting files, and the evaluation key is used for performing computation (additive homomorphism) on the encrypted files. For multiplicative homomorphism, [31] gives a ElGamal homomorphic model. An additive homomorphic encryption model based on elliptic curve encryption with ElGamal. The interesting thing about this model is that it does not encrypt the plaintext directly. Instead, the plaintext is first converted to an integer, then by a encoding function mapped points on an elliptic curve, and finally encrypt the points. When decrypting, first convert the encryption points to an integer, and then calculate the corresponding plaintext.

SHE scheme supports both addition and multiplication, although the times of multiplication that can be performed are limited. Most SHE schemes can do the mixed operation of addition and multiplication on the data encrypted by the same public key. Zhang [112] presented a SHE scheme applicable for multi-user to cooperation on data encrypted with their public keys, respectively. Since different user encrypt their data with different public key, it is not feasible to directly perform operations on ciphertext. Therefore, re-encrypt the ciphertext in the same way is necessary. Addition and multiplication can be performed on the re-encrypted ciphertext, and each user involved can decrypted the computed result using their own private key, which is corresponding to the public key used for the first level encryption. Quantum cryptography was introduced in the SHE scheme to obtain unconditional security and efficient query on ciphertext in [75], and the proposed scheme belongs to symmetric encryption. Multi-user training machine learning model on encrypted data is also studied in recent years. In this case, the functions used to learn the model are generally continuous functions, which need to be approximated by polynomial functions. Generally speaking, the higher the degree of polynomials is, the smaller the error of approximation is, but this will cause the greater the noise and the more time it takes to calculate the encrypted data. To solve this problem, the degree of approximate polynomials is set in an appropriate interval, and the resulting noise is controlled within a threshold value in [77]. When the noise reaches the threshold value, the server reports the calculated results (ciphertext) to the customer. The advantage of this model is that the client only needs to decrypt and view the returned results, and the server processes the whole calculation process.

The data encrypted by homomorphism can be performed by mixed operation of addition and multiplication simultaneously, and the number of times is unlimited. FHE is on the right track since the first FHE scheme based on ideal lattice was proposed in [36]. In order to weaken the hypothesis, Brakerski and Vaikuntanathan [16] proposed a FHE scheme based on learning with errors (LWE). First, relinearization was introduced to achieve SHE, which does not involve ideals. Then in order to obtain FHE from SHE, the dimension-modulus reduction technique is creatively proposed to cancel the hardness hypothesis in [36]. Brakerski *et al.* [15] Constructed a more efficient layered homomorphic encryption scheme, and bootstrappiing procedure exists only to optimize performance. Inspired by the knowledge of scale, [14] reduce the noise of ciphertext multiplication in LWE-based FHE scheme without modulus switching. In order to make multiplication natural for ciphertext, Gentry *et al.* [38] introduced approximate eigenvector method to make ciphertext be the matrix. In addition, they also gained identity-based FHE and attribute-based FHE. Cheon *et al.* [20] proposed a RLWE full encryption scheme to support floating-point calculation, where rescaling is the core technology. By rescaling, if the plaintext is divided by an integer, the corresponding ciphertext and the preinserted errors are divided by the same integer, where the errors are bounded. This ensures that the ciphertext modulus increases linearly rather than exponentially. Although decryption is approximate to the original plaintext, its accuracy can be predicted by rounding, which is similar to the approximate calculation for floating-point. Although this scheme implements a lot of primary operations on the representation of encrypted floating-point real values, it does not support the size comparison operation for given floating-point values. In order to solve this problem, Moon and Lee [62] introduced TFHE [22] algorithm on the basis of the [20], and obtained higher performance comparison operation.

D. SEARCHABLE ENCRYPTION

Most people choose to store data in the cloud due to the unlimited space of cloud storage and the flexible service. To ensure data security, users typically encrypt data before uploading it to the cloud. As mentioned earlier, this ensures the confidentiality of the data. But if someone wants to search for an encrypted file uploaded in the cloud, he/she will encounter some trouble. Since the data is encrypted in the cloud, users cannot search the encrypted files directly. There are two solutions for this problem. One is that the user downloads the encrypted files to local, decrypts the ciphertext, and then searches the keyword over the plaintext. This method is secure but inefficient. If the retrieved file contains massive data, it will consume a lot of computing resources and time. Another solution is to decrypt the ciphertext in

cloud and retrieve plaintext on cloud server. However, this solution will expose the context of these files, which seriously threatens data security and users' privacy. Therefore, how to enable users to search for specific keywords on encrypted files securely in the cloud has become the concern of many scholars [5], [46], [48], [52], [76]. Searchable encryption is a cryptography primitive that allows authorized users to retrieve ciphertext in the cloud by some means (such as keyword query). Its feature is to ensure that the cloud server returns encrypted data files of interest to users without knowing the ciphertext content. In terms of the way of encryption, searchable encryption can be divided into Searchable Symmetric Encryption (SSE) and Public Key Encryption with Keyword Search (PEKS).

SSE is a kind of searchable encryption based on symmetric cryptography. Recently, there are many literatures focus on designing of mechanisms for searching over encrypted data. Specifically, in 2000, Song *et al.* [72] designed a practical searchable encryption technique, which implements keyword based query for whole document depending on XOR operation. In this scheme, each word w_i in the whole document is encrypted with the same secret key, where the encrypted w_i is written as W_i. The ciphertext C_i is obtained by XORing W_i with the pseudo-random term generated by the data owner. To search for word w_i, the cloud server will XOR W_i with all $C_j s$ and return the correct C_i to data owner. Obviously, the search time increases linearly with whole encrypted document. In order to improve the efficiency of searchable encryption and make the files matched by keywords more satisfy the interests of users, Wang *et al.* [82] proposed the ranked searchable symmetric encryption scheme, where documents retrieved by single-keyword search will be ranked via relevance. In this scheme, order-preserving symmetric encryption was introduced to obtain higher efficiency. With the popularity and increase of outsourced data, it is necessary to allow multiple keywords in search requests. Cao *et al.* [18] proposed a secure multi-keyword ranked search over encrypted data. They use coordinate matching to retrieve as many documents as possible, and measure the relevance between documents and keywords by using inner product similarity. In order to reduce the retrieval failure caused by misspelling, Fu *et al.* [33] improved multi-keyword searchable encryption by adding fuzz search functionality. Their core technology is that each keyword is represented by uni-gram vector. With this, the misspelled word can be represented by the word highly similar to the correct one through computing their Euclidean distance. Recently, researches [104], [111] on multi-keyword search in mult-owner model enriches searchable symmetric encryption. In Yin *et al.*'s scheme, a group of data owners secretly share two l-bit primes $q_1, q_2 \in Z_q$ with $q = q_1 \cdot q_2$, where q_1 is used to encrypt the security index by data owners, and q_2 is kept by the authorized data user to encrypt the query keywords. They predefine the keyword dictionary $KD = \{w_1, w_2, \cdots, w_n\}$, in which each keyword has its own fixed position. Data owner D_i extracts keywords $W_{i,j} = \{w_1, w_4\}$ from data file $F_{i,j}$ and calculates security index $I_{i,j} = (g^{h(w_1)+q_1 \cdot sk}, R_2, R_3, g^{h(w_4)+q_1 \cdot sk}, R_5, \cdots, R_n)$. This design avoids the risk that the number of keywords in each file is leaked. In addition, Du *et al.* [28] proposed a multi-client SSE supporting boolean queries. Their solution not only supports the data owner to dynamically update someone's query permission without affecting others' normal use of data, but also reduces the interaction between users and owners.

The searchable encryption based on public key cryptography is PEKS. In 2004, Boneh *et al.* [13] designed a Public Key Encryption with keyword Search (PEKS) algorithm, which is used to implement searchable encryption on the email encrypted by public key. In this scheme (see Fig. 5), Bob sends encrypted message $E(M)$ and PEKS value (related to the keywords in the message M) $PEKS(pk, w_i), i = 1, 2, \cdots, n$ to the email server. Alice sends the trapdoor T_w of the specified keyword (such as "urgent") to the server, so that the server checks if there is an $i \in \{1, 2, \cdots, n\}$ to make $w_i = w$. During the whole process, PEKS value will not reveal any email content except the specified keywords. After that, Baek *et al.* [4] improved Boneh et al's scheme, and constructed an effective PEKS scheme with a safe channel removed. However, their solutions only address the searchable encryption issue with fewer keywords. There is a lack of practicability for the huge amount of data in the cloud with many keywords. Most of the existing searchable encryption schemes implement selectively retrieves encrypted files by using keyword search over the ciphertext of data as well as ensure security protection and retrieve privilege over the encrypted files for both data owners and users. However, sometimes users need to store a lot number of keys to decrypt the ciphertext files and generate trapdoors, and they have to submit massive trapdoors to search the keyword over a large number of file. Verifiable searchable encryption has been designed [74] to ensure the privacy of keyword and handle the threat from a semi-honest but curious server. Generally, users have to store a lot number of keys to generate trapdoors

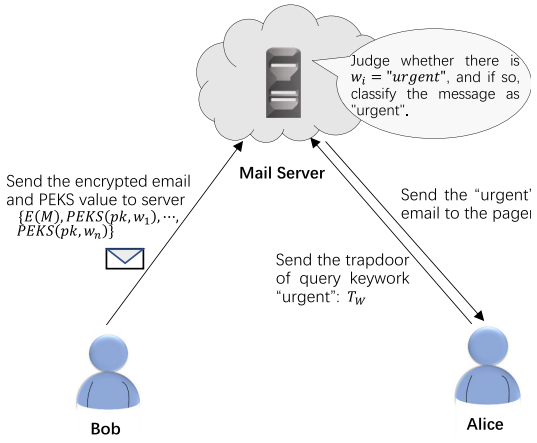

FIGURE 5. PEKS in [13].

and decrypt the ciphertext. It is a big challenge for users to manage their keys. The key-aggregate searchable encryption schemes [23] have been proposed to reduce the number of keys for users. Recently, Wang *et al.* [88] proposed an efficient verifiable key-aggregate keyword searchable encryption (EVKAKSE) system model. In this scheme, data owner uploads encrypted files and related encrypted potential keywords to the cloud server. And then, data owner send users an aggregate key, which allows users to retrieve files over the decrypted files by using keywords directly, decrypt ciphertext and verify the safety and practicality of retrieved result. Next, to perform keyword search over sharing files, users have to generate an aggregate trapdoor using the mentioned aggregate key. With the aggregate key, users can perform keyword search over the authorized files. Furthermore, this scheme is able to protect the keyword and its ciphertext and the submitted trapdoor from being determined by the semi-honest but curious cloud server and malicious cloud server. Any insider attacker cannot calculate a valid users' aggregate key from the trapdoor.

In this section, we summarize four encryption technologies commonly used in cloud storage, which ensure the confidentiality of data in the cloud. From the perspective of access control, IBE embeds ''identity'' into public key and private key, which makes IBE have great advantages in protecting the private data of a single or a small number of users, such as encrypting e-mail. In addition, IBE is also applied to proxy re-encryption (such as [24]) to obtain lightweight encryption schemes, which makes users with limited resources no longer be bothered by the complex computation when decrypting. Compared with IBE, ABE, as a fuzzy identity encryption, has higher scalability. ABE allows the data owner to use the user's attributes as a medium to specify the legitimate users, and obtains high-efficiency fine-grained access control functionality. Because the length of ciphertext increases with the amount of user attributes, the decryption might requires heavy computing. In order to solve this problem, the combination of ABE and IBE (for example [35]). can not only obtain fine-grained access control, but also reduce the computation and communication cost during decrypting prase. In addition to access control, homomorphic encryption realizes the ability to perform predefined operations on ciphertext, searchable encryption realizes the ability to retrieve ciphertext, which increases the user's control over data and attracts more potential users.

III. PRESENT RESEARCH FOCUS

In the following part, we provide a introduction for state of the art researches on data security and privacy protection in cloud storage system.

A. ONE-TO-MANY ENCRYPTION

The high scalability and unlimited expansion of cloud storage attract more and more users and organizations to share their data in the cloud. Some data owners upload data to the cloud for their own use through the Internet at any time, regardless of location and time constraints, such as private cloud storage. If the data is only for personal use, encryption can largely ensure the confidentiality of private data. When it comes to sharing data to multiple parties (such as organizations or groups), one-to-many data sharing mode (one data owner, multiple data users) is more suitable for them. The data owner gives access to a specific group by designing a fine-grained access control scheme. In this case, collusion-resistant and tamper-resistant are worthy of deep consideration. In this section, we have investigated the literature in one to many encryption, and reviewed one-to-many encryption from three aspects: the preset cooperative access control of designated multi-user, the fuzzy multi-party shared access control to deal with emergencies and the security access control to dynamic multi-group.

There is a common sense that the security of a lock that can only be opened by many different keys is much higher than that of a lock that can only be opened by one key. For enterprises or organizations, the data confidentiality of some encrypted files can highly be guaranteed, if the access policy requires multiple employees with different attribute sets to obtain the access permission through cooperate, where access request should be denied even if one of them is absent. Xue *et al.* [100] proposed a controlled collaboration access control scheme, which improved the model of [7]. In their scheme, a set of translation nodes are inserted in the policy tree by data owner, translation value is added into ciphertext via cloud server and translation key is embedded into the secret key in PKG, and all of there are designed to make multi-user collaboration access feasible. The data owner can remove the translation nodes to cancel the privilege for cooperation access. Their scheme can effectively avoid malicious deletion and modification of important files by single enterprise employees. Collusion-resistant also avoids the illegal access to confidential data by unqualified users.

In order to realize temporary access authorization in the process of cross domain data sharing, Yang *et al.* [101] presented a self-adaptive access control system with secure deduplication. They considered how to enable the unqualified doctors to access and decrypt the electronic medical records of the patient in an emergency (such as coma of patient), so as to provide more accurate treatment plans for the patient. In such a scheme, the electronic medical records and physiological parameters detected by wearable devices in real time are encrypted and transmitted to the public cloud server by data owner (usually patient), which pre-sets a break-glass key to decrypt the data mentioned above, a password for generating the key, and a list of people who knows the password. Person on the list interacts with the cloud servers with the password to generate the break-glass key, which temporarily allow unauthorized medical workers to access the patient's electronic medical records. The traditional access control system only allows qualified users to access encrypted data legally, which is fatal for patients who need emergency treatment, in that not all doctors are qualified to access. Their system solves the problem of temporary access authorization

TABLE 3. Comparison of relevant schemes on data confidentiality in cloud storage.

Reference	Objective	Proposed Schemes	Technical Methods	Security Features
[100]	Collaborative access control	Attribute-based controlled collaborative access control scheme	• CP-ABE • Translation nodes insertion	• Data confidentiality • Collusion-resistant • Controlled collaborative • Secure data sharing • Secure revocation
[101]	Self-adaptive access control	Self-adaptive access control with smart deduplication	• CP-ABE • Break-glass access • Secure deduplicetion	• Data confidentiality • Secure deduplication • Secure cross-domain data sharing
[95]	Group-oriented access control	Attribute-based privacy-preserving data sharing for dynamic groups	• CP-ABE • Broadcast encryption • Re-encryption algorithms	• Data confidentiality • Fine-grained access control • Dynamic groups data sharing • Collusion-resistant

in electronic medical record sharing. It can not only ensure the confidentiality of the patient's data, but also make the original unauthorized doctors can access the patient's data legally.

Personal health data is collected by intelligent wearable devices or by hospitals, which can help doctors get a comprehensive understanding of patients' conditions. In order to protect privacy, data owners will choose to encrypt the data and upload it to the cloud. Many data owners, hospitals, health institutions, etc. form a cloud data sharing system with multiple groups. Each participant in the system will be divided into a specific group. Only when users satisfy two conditions can they access the shared data: 1) they belong to the specified dynamic group; 2) their attributes meet the predefined access policy. To achieve secure data sharing in the above, Xiong *et al.* [95] considered data sharing involving multiple dynamic groups. They put forward a secure attribute-based broadcast encryption scheme, which realizes data sharing among multiple groups and supports offline and online computing functionalities. In addition, attributes in the access policy are anonymous to protect users' privacy.

B. DATA INTEGRITY

With cloud storage services, more and more users outsource their data to the cloud and realize the data sharing with others. Ensuring data integrity remains a top priority for data security. Since outsourced data is often kept in unknown places, how to detect whether the data remains integrity without downloading the data has become a concern. In order to check the integrity, existing solutions include provable data possession (PDP) model proposed by Ateniese *et al.* [2] and proof of retrievability (POR) model presented by Shacham and Waters [69]. Furthermore, outsourced data integrity auditing schemes have been proposed to guarantee the integrity of the data stored in the cloud. Generally speaking, data integrity auditing can be broadly divided into two categories [86], namely private auditing and public auditing. In the former, only the data owner can audit the integrity of the outsourced data. Although privacy auditing schemes are secure and efficient, they require high computing resources and networks for auditing. Once data owners are unavailable due to network failures or limited computing resources, privacy audits cannot be performed. In public auditing, the data owner can delegate the audit to an independent third party auditor (TPA), so both data owner and third party auditor can verify the integrity of outsourced data. Compared with the privacy audit, the public audit scheme is not affected by the owner's network and resources. Even if the owner cannot confirm the correctness of the data, the third-party audit can still perform the auditing task. Because of the fault tolerance of public auditing, public auditing schemes have been presented in a lot of literature [42], [43], [56], [70], [86], [109].

In 2017, Shen *et al.* [70] proposed an efficient public auditing protocol based on conventional public key infrastructure (PKI)-based cryptography. In their model, global and sampling verification is proposed to address the issue that data owner may distrust the cloud has stored their data securely and the cloud service provider may become anxious owing to their users' wrongly accusation during their cooperation; Data dynamics is more efficient by the novel dynamic structure consisting of doubly linked info table and location array, where data update and batch auditing are easier to implement; Furthermore, to improve the practicability of their model, they established public auditing, blockless verification, which support public verifiability and prevent data leakage from cloud service providers and auditors various auditing.

Since the key management in PKI-based scheme is more complex than those in ID-based cryptosystem, source-constrained users are more likely the later one. In 2016, an Identity (ID)-based public auditing based on homomorphic ID-based signature was designed by Zhang and Dong [109] for cloud storage system, which implement batch auditing in the multi-user setting and prevent forge attack, replace attack and replay attack from an untrusted cloud

server. The ID-based protocol simplify the key management and the public auditing schemes with batch auditing lighten the auditors' and users' load. Their model made a great contribution to save communication and computation overhead. In 2019, Li *et al.* [56] formalize data integrity auditing based on the Fuzz Identity-based cryptography. It's very interesting that they addressed the key management issues by brought in biometric-based identities in traditional public verifiable RDIC protocols, which allows TPA or users to verify the data integrity without retrieving the entire dataset.

In 2018, He *et al.* [43] presented a certificateless provable data possession (CL-PDP) scheme. This scheme implements remote data integrity auditing for cloud-based smart grid data management systems. Specifically, the data owner can delegate a third-party auditor to verify the integrity and detect modification of the data. The verifier is allowed to audit the integrity of a large number of data belonging to different users simultaneously. Furthermore, during this period, curious auditors can not get the content of verified data, namely data confidentiality is ensured. Other references for certificateless public auditing schemes see He *et al.* [42] and [43] and Wang *et al.* [81].

Wang *et al.* [86] provided a lightweight certificate-based public/private auditing scheme in 2020. It is a certificate-based PDP scheme that was based on asymmetric pairing for the sake of minimizing storage space and communication cost, and is secure under both the public key replacement adversary and the malicious certifier adversary. In their scheme, the audit phase is divided into PrivateVerify and PublicVerify, which correspond to private auditing and public auditing, respectively. Since data owners have more information than auditors, the former executes PrivateVerify more efficiently when data integrity auditing is required. If data owner is not available, the auditor can execute PublicVerify directly.

C. DATA DELETION

Users' data is typically distributed across multiple cloud servers, which may be Shared by users who do not know each other. If one user wants to delete a file in local storage, the safest way is to burn or shred it, but this is obviously not feasible for files in the cloud. In the cloud, users need to entrust cloud service providers to delete unnecessary files. Usually the cloud service deletes the file in the form of a logical deletion. Logical deletion essentially hides the corresponding data rather than the real deletion. This may result in the user's privacy being exposed to others. On the other hand, cloud service providers may also falsely delete data and cheat users due to business interests. Therefore, how to verify that the data has been deleted safely is an important part of protecting the data security in the data life cycle. Hash function is a one-way function that maps data to fixed length values, known as hash values. Generally, the definition domain of hash function is larger than the hash value domain, so it is difficult to get the inverse of hash value. Hash is mainly used for authentication and public audit. In recent years, due to the characteristics of hash function, hash algorithm has also been used to prove whether cloud service providers can delete data irrecoverably according to user requirements, among them, Merkle Hash Tree is very popular.

To assured data deletion, Xue *et al.* [99] proposed a efficient attribute revocation scheme based on Merkle Hash Tree. Once the cloud server receives the deletion request from a user, it will re-encrypt the corresponding files using the re-encryption key generated by the trust authority. At the same time, according to attribute revocation, a new root of the Merkle Hash Tree will be sent to data owner so that he can verify the data has been deleted successfully. In addition to data deletion validation, other users can still use cloud services normally during the process of deleting one user's data.

In 2019, Yang *et al.* [102] presented a fine-grained data deletion scheme in order to prevent malicious tampering with data from cloud servers and hackers as well as the incomplete data deletion of cloud service providers. Rank-based Merkle Hash Tree chain is introduced to check whether the data block is altered or deleted on the behalf of user.

D. LEAKAGE-RESILIENT

Side channel attack allows adversary to destroy cryptography technology by collecting information leaked by encryption algorithm. The user downloads and decrypts the ciphertext on the local device under normal circumstances. The attacker uses the side channel attack (for example, monitoring the electromagnetic radiation emitted by the computer screen, monitoring the power consumption of electronic devices or recording the sound of the user's keystroke) to grab part of the information of the user's decryption key. In order to handle this situation, the concept of leakage-resilient is introduced into the cryptography scheme (for instance, [6], [65]). Among them, the study of memory leakage is the most extensive. Memory leakage is a strong leakage model including secret key leakage. Once the private key is revealed, the encryption scheme will be invalid. Although the side channel attack is affected by physical distance, with the development of unmanned aerial vehicle (UAV) and intelligent mobile devices, the side channel attack will become more easier and cheaper.

Existing leakage models usually can be divided into three categories: 1) The bounded retrieval model [29]. In this model, f is arbitrary polynomial-time computable leakage function with a bounded output value. Leakage-resilient can be obtained by designing secret key whose size is longer than the output of f; 2) The bounded leakage model [1]. In this model, f is a polynomial-time computable leakage function with a given bounded output value, which is generally regarded as the minimum entropy of secret key; 3) The auxiliary input memory model [27]. There is a premise in this model, namely, it's hard to recover the secret key no matter how much information is leaked. With it, unbounded output length is allowed for leakage function f; 4) The continuous leakage model [17]. Different from the previous three models, the leakage function here can have continuous output,

and the output is bounded in each bounded period of time, while the amount of output can be unbounded. In such a situation, the security of the encryption scheme is guaranteed by updating the private key periodically, while updating the public key is not required. In fact, given the initial public key pk and the private key sk_1. After the attacker continuously obtains the bounded information of sk_1, sk_1 is updated to sk_2. At this time, sk_1 is invalid for the decrypted ciphertext, so the information collected by the attacker is invalid. So according to that, even if attacker collected boundless information, which comes from different parts of sk_1, sk_2, sk_3, $\cdots$, it is still hard to recover the decryption key.

Hu *et al.* [44] proposed a CCA secure public-key encryption scheme, which can resilient continuous leakage and tampering attacks by updating the private key. In fact, they did not get the expected results directly. They first achieve the CCA security in continuous memory leakage (CML) model. After that, one-time lossy-filter is introduced to obtain CCA security in continuous key-leakage and tampering (CLT) model.

In bounded leakage model, the amount of leakage may be bounded in a certain period. For example the information is intercepted by an attacker using the bounded side-channel attack. Sometimes, a continuous leakage incurs in each invocation of the cryptosystem. The amount of leakage of private key is limited between two consecutive private key updates, while the whole leakage amount may be arbitrary large. Zhang *et al.* [115] presented a continuous leakage-resilient identity-based encryption scheme (CLR-IBE) to protect data security from partial secret key leakage in the continuous leakage model. It is a big data storage system in cloud computing. In this scheme, the secrete keys are uploaded periodically in a big data storage system. By defining a leakage ratio: $\frac{l}{|sk|}$, where l denotes the size of leakage, and sk means the size of private key, they proved that their scheme allows a high leakage ratio 1/3. Recently, Li *et al.* [55] proposed a hierarchical attribute-based encryption scheme, which can continuously resilient the leakage of master key and private key. In this scheme, when the leakage length of the master key and the private key is bounded, the proposed scheme is secure under the standard model. When the attribute universe is consistent with the attribute set of depth K, the master key should be re randomized. At this time, the key update algorithm is started. Considering that leakage is tolerable during the update process, and the amount of leakage is logarithmically related to the safety parameters. As long as the key is updated regularly and the key secret information is not leaked in the process, the continuous leakage elasticity can be obtained. This scheme has the same leakage ratio to [115].

E. PRIVACY-PRESERVING

The convenience and scalability of cloud storage system attract more and more individual and enterprise users to outsource their data to cloud service providers. However, there is a risk of privacy disclosure. For instance, the Electronic Health Records (EHRs) including patient's medical records are stored in the cloud, which not only facilitate the patient to seek medical advice in different hospitals, but also facilitate the doctor to provide more accurate treatment plan for the patient according to the records. Once the sensitive information, such as identity information and home address, is leaked or tampered with, irreparable harm would be caused to patients. Besides, identity and attribute leakage issues are also threatening the privacy of data owners and authorized users. Due to the diversity of cloud data, conventional privacy-preserving mechanisms are unable to provide comprehensive privacy protection in the cloud. Therefore, protection schemes [114], [115], [117] about sensitive information privacy, identity privacy and attributes privacy etc. are developed to achieve more specified privacy protection.

CP-ABE schemes plays a pivotal role in implementing data sharing and fine-grained access control. Only the private key generated by attributes of user's matches the access policy embedding in the ciphertext, the ciphertext could be decrypted. In the general CP-ABE scheme, access policies are stored in the cloud in the form of plaintext. Nevertheless, access policies and attribute sets sometimes contain sensitive information of data owners and users authorized to share data, and the attribute privacy of data owners and users is easily exposed by the predefined access policies. Zhang *et al.* [114] designed an anonymous CP-ABE access control system with collusion-resistance for resource-limited user. In order to protect attribute privacy, the access policy is hidden in the ciphertext by encrypting an symmetric key. In such a system, the authorized users should not know anything about the access policy determined by the data owner, even if they can access and decrypt the ciphertext by using their distributed attribute private key successfully. Xiong *et al.* [95] proposed a group-oriented ABE model to satisfy the requirement for one-to-many data sharing. In this scheme, data owner first need to send the encrypted files, hidden access policy and the set of authorized users' identities to the cloud. They protect attribute of the authorized receiver from being exposed by hiding the access policy fully before uploading the encrypted data to the cloud.

To verify the correction of data stored in cloud storage with low computing resources and communication costs, public auditing schemes are proposed so that both the third public auditor (TPA) and data owner have privilege to perform the auditing task. However, when the TPAs are checking the integrity of data, they may be very curious about identity of audited user and some other sensitive information. This may cause the identity privacy of users to be disclosed to hackers or sold to illegal organizations. Therefore, the protection of identity privacy is of great significance. When TPAs are auditing the correctness of remote data, the joining, exiting and revocation of members in a dynamic group and TPAs' curiosity will lead to the disclosure of member's identity information. For this problem, Yu *et al.* [105] developed an identity privacy preservation for public auditing protocol. In this protocol, multiple users in a dynamic group talk things over to share a public-secret key pair so

TABLE 4. Comparison of representative schemes on leakage-resilience.

Reference	Leaked Objects	Proposed Scheme	Assumption	Technical Methods
[55]	• Master key • Private key	Continuous leakage-resilience hierarchical attribute based encryption (CLR-HABE)	Composite order bilinear group	• Dual system methodology • HABE • CP-ABE
[44]	Memory leakage	CCA-secure public-key encryption with continuous leakage and tampering resilience	• Symmetric external Diffie-Hellman • d-Rank hiding assumption • Naor-Yung double encryption paradigm	One-time lossy filter
[115]	Private key	Continuous leakage-resilient identity-based encryption scheme (CLR-IBE)	Static assumptions	IBE
[106]	Memory leakage	Full-secure leakage-resilience function encryption scheme	Subgroup decision	• Dual system methodology • Functional encryption • Leakage-resilient pair encoding scheme

that TPAs could perform data auditing without any knowledge about users' identities. Furthermore, since the target group secret key is generated by a hash function, any user who is joining the group can only know the information after he joined but not the previous information, and anyone who leaves the group will no longer be able to obtain the information after he leaves. Therefore, the privacy of private key is also protected. In the framework designed by Yang *et al.* [103], the more members of data sharing group, the less probability the identity privacy will be obtained by the auditor. Besides, group manager can trace and disclose dishonest members to reduce the tempered threat of shared data.

In response to malicious attacks from untrusted cloud service providers, Zhang and Zhao [110] drawn support from the idea of chameleon hash algorithm to hide the real public keys of data owner by generating dynamic public keys. This idea preserves the identity privacy of data owner from being obtained or calculated by cloud server.

To against both threats form malicious cloud server and TPA, Zhang *et al.* [113] put forward a conditional identity privacy protection mechanism. This scheme is mainly used to protect the identity privacy and sensitive information of patients in EHRs. They used public auditing to ensure that the data integrity of patients and prevents malicious cloud service providers from returning error audit reports. The PKG generate an anonymous identity with valid period T by patient's real identity and the computing well-defined. Based on the hardness assumption, any adversary will not be able to learn the patient's authentic identity information.

IV. OPEN ISSUES AND THE POSSIBLE DEVELOPMENT

A. PRIVACY-PRESERVING MACHINE LEARNING IN CLOUD STORAGE

Machine learning is very popular and widely used, such as data mining, medical diagnosis, DNA sequencing, image recognition and so on. Recently, more and more government departments (such as the Ministry of transport and the Department of Public Security) and medical institutions have migrated massive valuable data to the cloud. Taking the Ministry of transport as an example, if these data can be fully mined, it will be helpful to reduce road traffic congestion, traffic accidents and predict the 24-hour speed of a road section in the future. Furthermore, the joint data analysis of the Department of transportation and the Department of public security is also conducive to reducing the occurrence of criminal incidents in public places. Therefore,the combination of machine learning and cloud has become a new focus.

But now there are two problems: 1) departments that do not trust each other may refuse to share data in order to protect their own data security. 2) In the face of massive cloud data, users with limited resources may not be able to carry out effective data mining and model training because of the high cost of computing and communication. Outsourced the model training calculation to the cloud will increase the risk of leakage of key parameters of its own model. Although there are some researches on cloud based machine learning, for example, machine learning with public auditing [41], machine learning training and classification scheme based on homomorphic encryption [54], and homomorphic deep learning [57]. But the efficiency and security of these programs are not satisfactory.

For the above challenges, we think there are two research directions in the future.

1) Design a more secure privacy protection scheme to ensure that sensitive information in shared data is hidden, especially data involving highly sensitive information such as government data and medical data.

2) Design efficient and secure outsourced privacy protection scheme to support more machine learning algorithms (such as incremental learning).

B. POST-QUANTUM ENCRYPTION

In recent years, with the rapid development of blockchain, Internet of things and quantum computing, the world's attention to data security and privacy has increased to an unprecedented level, which all put forward more and higher requirements for data security and data privacy protection. At present, the security of public key cryptography depends on some mathematical problems (such as discrete logarithm problem and factorization of large integers) which are difficult to solve in traditional computers and classical algorithms. In 1994, the proposed short algorithm directly threatened the RSA and a related algorithms. Recently, the research and development of quantum computer has become the focus of many commercial companies. Although it is not clear when a practical quantum computer will be implemented, some quantum computers have been designed, such as Honeywell recently announced the construction of a 64 bit quantum computer.

Post quantum cryptography is a new generation of cryptography that can resist the attack of quantum computer on existing cryptography. The following is the present researches and existing open issues about main post quantum encryption algorithms.

1) The authentication mechanism of hash-based signature algorithm is Merkel hash tree, whose security relies on the collision resistance of hash function. Merkel hash tree is applied to integrity auditing, data deletion [99], [102] etc. Due to the use of tree structure in hash based construction scheme, there are only digital signature construction at present, and there are very few public key encryption systems.

2) The lattice-based algorithm can realize cryptography construction such as encryption, digital signature, attribute encryption and homomorphic encryption, whose security depends on the difficulty of solving the problems in lattice. Under the same security, the lattice based algorithm has smaller public key size, faster computing speed and higher security compared with the hash-based one. Recently, lattice cryptography construction based on LWE (learning with errors) [14], [16], [26] and RLWE (ring-LWE) [20] develops rapidly. For instance, it is noted that Wei *et al.*'s research on the revocable storage IBE [90] is based on bilinear pairing. Their scheme has good performance but can't resist quantum attack. Lattice based revocable storage still needs further exploration.

V. CONCLUSION

In this paper, we give a detail survey on data security and privacy preservation in cloud storage system. First of all, from the outstanding performance of cloud in the digital economy, enterprise digital transformation, Internet of things and other fields, we confirm that cloud computing and cloud storage will still be the mainstream. We first analyze eight elements of data security in cloud storage system: data confidentiality, data integrity, data availability, fine-grained access control, secure data sharing in dynamic group, leakage-resistant, complete data deletion and privacy protection. Next, we introduce the encryption principles of IBE, ABE, homomorphic encryption, searchable encryption and the research direction of new encryption models. Data encryption technologies and protection methods are summarized. These correspond to the mentioned security requirements. Finally, we put forward some several open research topics of data security for cloud storage.

REFERENCES

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proc. TCC*, San Francisco, CA, USA, 2009, pp. 474–495.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. ACM-CSS*, New York, NY, USA, 2007, pp. 598–609.

[3] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Proc. IMACC*, Cirencester, U.K., Dec. 2009, pp. 278–330.

[4] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. ICCSA*, Perugia, Italy, 2008, pp. 1249–1259.

[5] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2007, pp. 535–552.

[6] F. Berti, O. Pereira, T. Peters, and F. X. Standaert, "On leakage-resilient authenticated encryption with decryption leakages," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 3, pp. 271–293, 2017.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2007, pp. 321–334.

[8] T. Bhatia and A. K. Verma, "Data security in mobile cloud computing paradigm: A survey, taxonomy and open research issues," *J. Supercomput.*, vol. 73, no. 6, pp. 2558–2631, Jun. 2017.

[9] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. ACM CCS*, Alexandria, VA, USA, 2008, pp. 417–426.

[10] D. Boneh and X. Boyen, "Efficient selective-ID secure identity based encryption without random oracles," in *Proc. Adv. Cryptol. (Eurocrypt)*, Interlaken, Switzerland, vol. 3027, 2004, pp. 223–238.

[11] D. Boneh and X. Boyen, "Secure identity-based encryption without random oracles," in *Proc. CRYPTO*, vol. 3152. Berlin, Germany: Springer, 2004, pp. 443–459.

[12] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, vol. 2139. Berlin, Germany: Springer, 2001, pp. 213–229.

[13] D. Boneh, G. Di Crescenz, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT*, vol. 3027. Berlin, Germany: Springer, 2004, pp. 506–522.

[14] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2012, pp. 868–886.

[15] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, pp. 1–36, Jul. 2014.

[16] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, Jan. 2014.

[17] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan, "Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci. (FOCS)*, Las Vegas, NV, USA, Oct. 2010, pp. 501–510.

[18] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.

[19] B. Casemore, "Network modernization: Essential for digital transformation and multicloud," IDC, Framingham, MA, USA, White Paper US45603019, Nov. 2019.

[20] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Hong Kong, 2017, pp. 409–437.

[21] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th Proc. ACM CCS*, Alexandria, VA, USA, 2007, pp. 456–465.
[22] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *J. Cryptol.*, vol. 33, no. 1, pp. 34–91, Jan. 2020.
[23] B. Cui, Z. Liu, and L. Wang, "Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2374–2385, Aug. 2016.
[24] H. Deng, Z. Qin, Q. Wu, Z. Guan, and Y. Zhou, "Flexible attribute-based proxy re-encryption for efficient data sharing," *Inf. Sci.*, vol. 511, pp. 94–113, Feb. 2020.
[25] *Digital Economy Report*, document UN Symbol: UNCTAD/DER/2019, United Nations Conference on Trade and Development, Geneva, Switzerland, 2019.
[26] C. Dong, K. Yang, J. Qiu, and Y. Chen, "Outsourced revocable identity–based encryption from lattices," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 11, p. e3529, Nov. 2019.
[27] Y. Dodis, Y. T. Kalai, and S. Lovett, "On cryptography with auxiliary input," in *Proc. 41st Annu. ACM Symp. Symp. Theory Comput.*, 2009, pp. 621–630.
[28] L. Du, K. Li, Q. Liu, Z. Wu, and S. Zhang, "Dynamic multi-client searchable symmetric encryption with support for Boolean queries," *Inf. Sci.*, vol. 506, pp. 234–257, Jan. 2020.
[29] S. Dziembowski, "Intrusion-resilience via the bounded-storage model," in *Proc. TCC*, vol. 3876. Berlin, Germany: Springer, 2006, pp. 207–224.
[30] K. El Makkaoui, A. Ezzati, A. Beni-Hssane, and S. Ouhmad, "Fast Cloud–Paillier homomorphic schemes for protecting confidentiality of sensitive data in cloud computing," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 6, pp. 2205–2214, Jun. 2020, doi: 10.1007/s12652-019-01366-3.
[31] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[32] *Fully Homomorphic Encryption: Cloud Security*. Accessed: Feb. 2020. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/fully-homomorphic-encryption
[33] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2706–2716, Dec. 2016.
[34] *Gartner: Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020*. Accessed: Feb. 2020. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020
[35] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes Cryptogr.*, vol. 86, no. 11, pp. 2587–2603, Nov. 2018.
[36] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, Bethesda, MD, USA, May 2009, pp. 169–178.
[37] C. Gentry, "Practical identity-based encryption without random oracles," in *Proc. EUROCRYPT*, vol. 4004. Berlin, Germany: Springer, 2006, pp. 445–464.
[38] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2013, pp. 75–92.
[39] A. Gharaibeh, M. A. Salahuddin, S. Jahed Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2456–2501, 4th Quart., 2017.
[40] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Alexandria, VA, USA, 2006, pp. 89–98.
[41] A. Hassan, R. Hamza, H. Yan, and P. Li, "An efficient outsourced privacy preserving machine learning scheme with public verifiability," *IEEE Access*, vol. 7, pp. 146322–146330, Oct. 2019.
[42] D. He, N. Kumar, S. Zeadally, and H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1232–1241, Mar. 2018.
[43] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 64–73, Mar. 2018. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.584.1010&rep=rep1&type=pdf
[44] C. Hu, R. Yang, P. Liu, T. Li, and F. Kong, "A countermeasure against cryptographic key leakage in cloud: Public-key encryption with continuous leakage and tampering resilience," *J. Supercomput.*, vol. 75, no. 6, pp. 3099–3122, Jun. 2019.
[45] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.
[46] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multiuser system," in *Proc. Int. Conf. Pairing-Based Cryptogr.* Berlin, Germany: Springer, 2007, pp. 2–22.
[47] IBM. *Block*. Accessed: Feb. 2020. [Online]. Available: https://www.ibm.com/cloud/learn/block-storage
[48] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proc. ACM CCS*, Raleigh, NC, USA, 2012, pp. 965–976.
[49] K. Lee, S. G. Choi, D. H. Lee, J. H. Park, and M. Yung, "Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency," in *Proc. Adv. Cryptol.-ASIACRYPT*. Berlin, Germany: Springer, 2013, pp. 235–254.
[50] K. Lee, "Comments on 'Secure data sharing in cloud computing using revocable-storage identity-based encryption,'" *IEEE Trans. Cloud Comput.*, early access, Feb. 13, 2020, doi: 10.1109/TCC.2020.2973623.
[51] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 425–437, Feb. 2015.
[52] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–5.
[53] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep. 2017.
[54] J. Li, X. Kuang, S. Lin, X. Ma, and Y. Tang, "Privacy preservation for machine learning training and classification based on homomorphic encryption schemes," *Inf. Sci.*, vol. 526, pp. 166–179, Jul. 2020.
[55] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.
[56] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K.-R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 72–83, Jan. 2019.
[57] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.
[58] X. Liang, R. Lu, X. Lin, and X. Shen, "Ciphertext policy attribute based encryption with efficient revocation," Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep., 2010, vol. 2, p. 8. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.584.1010&rep=rep1&type=pdf
[59] X. Liu, S. Zhao, A. Liu, N. Xiong, and A. V. Vasilakos, "Knowledge-aware proactive nodes selection approach for energy management in Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 1142–1156, Mar. 2019.
[60] H. Ma, T. Peng, and Z. Liu, "Directly revocable and verifiable key-policy attribute-based encryption for large universe," *Int. J. Netw. Secur.*, vol. 19, no. 2, pp. 272–284, Mar. 2017.
[61] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *J. Netw. Comput. Appl.*, vol. 84, pp. 38–54, Apr. 2017.
[62] S. Moon and Y. Lee, "An efficient encrypted floating-point representation using HEAAN and TFHE," *Secur. Commun. Netw.*, vol. 2020, pp. 1–18, 2020, Art. no. 1250295, doi: 10.1155/2020/1250295.
[63] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT*, vol. 1592. Berlin, Germany: Springer, 1999, pp. 223–238.
[64] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Secur.* Berlin, Germany, 2013, pp. 463–474.
[65] O. Ruan, Y. Zhang, M. Zhang, J. Zhou, and L. Harn, "After-the-fact leakage-resilient identity-based authenticated key exchange," *IEEE Syst. J.*, vol. 12, no. 2, pp. 2017–2026, Jun. 2018.
[66] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Proc. CRYPTO*, Berlin, Germany: Springer, 2012, pp. 199–217.

[67] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2005, pp. 457–473.

[68] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1984, pp. 47–53.

[69] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. ASIACRYPT*, Melbourne, VIC, Australia, 2008, pp. 90–107.

[70] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.

[71] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Inf. Sci.*, vol. 295, pp. 221–231, Feb. 2015.

[72] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE SP*, Berkeley, CA, USA, May 2000, pp. 44–55.

[73] Spectralogic. *Comparing File (NAS) and Block (SAN) Storage*. Accessed: Mar. 1, 2020. [Online]. Available: https://edge.spectralogic.com/?&fuseaction=home.displayFile&DocID=4630

[74] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 3025–3035, Nov. 2014.

[75] X. Sun, T. Wang, Z. Sun, P. Wang, J. Yu, and W. Xie, "An efficient quantum somewhat homomorphic symmetric searchable encryption," *Int. J. Theor. Phys. Volume*, vol. 56, no. 4, pp. 1335—1345, Apr. 2017.

[76] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan, and C. Glackin, "A new secure and lightweight searchable encryption scheme over encrypted cloud data," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 4, pp. 530–544, Oct. 2019.

[77] H. Takabi, E. Hesamifard, and M. Ghasemi, "Privacy preserving multi-party machine learning with homomorphic encryption," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 1–5.

[78] Y.-Y. Teing, A. Dehghantanha, K.-K.-R. Choo, and L. T. Yang, "Forensic investigation of P2P cloud storage services and backbone for IoT networks: BitTorrent sync as a case study," *Comput. Electr. Eng.*, vol. 58, pp. 350–363, Feb. 2017.

[79] H. Teng, Y. Liu, A. Liu, N. N. Xiong, Z. Cai, T. Wang, and X. Liu, "A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities," *Future Gener. Comput. Syst.*, vol. 94, pp. 351–367, May 2019.

[80] According to a New IDC Forecast. *The Growth in Connected IoT Devices Is Expected to Generate 79.4 ZB of Data in 2025*. Accessed: Mar. 1, 2020. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

[81] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," in *Proc. IEEE CNS*, National Harbor, MD, USA, Oct. 2013, pp. 136–144.

[82] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. ICDCS*, Genova, Italy, 2010, pp. 253–262.

[83] J. Wang, C. Huang, N. N. Xiong, and J. Wang, "Blocked linear secret sharing scheme for scalable attribute based encryption in manageable cloud storage system," *Inf. Sci.*, vol. 424, pp. 1–26, Jan. 2018.

[84] J. Wang, N. N. Xiong, J. Wang, and W.-C. Yeh, "A compact ciphertext-policy attribute-based encryption scheme for the information-centric Internet of Things," *IEEE Access*, vol. 6, pp. 63513–63526, 2018.

[85] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology—EUROCRYPT*, vol. 3494. Berlin, Germany: Springer, 2005, pp. 114–127.

[86] F. Wang, L. Xu, K.-K.-R. Choo, Y. Zhang, H. Wang, and J. Li, "Lightweight certificate-based public/private auditing scheme based on bilinear pairing for cloud storage," *IEEE Access*, vol. 8, pp. 2258–2271, 2020.

[87] X. Wang, T. Luo, and J. Li, "A more efficient fully homomorphic encryption scheme based on GSW and DM schemes," *Secur. Commun. Netw.*, vol. 2018, pp. 1–14, Dec. 2018, doi: 10.1155/2018/8706940.

[88] X. Wang, X. Cheng, and Y. Xie, "Efficient verifiable key-aggregate keyword searchable encryption for data sharing in outsourcing storage," *IEEE Access*, vol. 8, pp. 11732–11742, 2020.

[89] J. Wei, X. Chen, X. Huang, X. Hu, and W. Susilo, "RS-HABE: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud," *IEEE Trans. Dependable Secure Comput.*, early access, Oct. 21, 2019, doi: 10.1109/TDSC.2019.2947920.

[90] J. Wei, W. Liu, and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption," *IEEE Trans. Cloud Comput.*, vol. 6, no. 4, pp. 1136–1148, Oct./Dec. 2018.

[91] L. Wu, B. Liu, and W. Lin, "A dynamic data fault-tolerance mechanism for cloud storage," in *Proc. EIDWT*, Xi'an, China, Sep. 2013, pp. 95–99.

[92] Z. Wu, N. Xiong, W. Han, Y. N. Huang, C. Y. Hu, Q. Gu, and B. Hang, "A fault-tolerant method for enhancing reliability of services composition application in WSNs based on BPEL," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 3, Mar. 2013, Art. no. 493678.

[93] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Inf. Sci.*, vol. 387, pp. 195–204, May 2017.

[94] Z. Xia, T. Shi, N. N. Xiong, X. Sun, and B. Jeon, "A privacy-preserving handwritten signature verification method using combinational features and secure KNN," *IEEE Access*, vol. 6, pp. 46695–46705, Aug. 2018.

[95] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2739–2750, Sep. 2019.

[96] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 453–461, Aug. 2019.

[97] S. Xu, G. Yang, Y. Mu, and X. Liu, "A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance," *Future Gener. Comput. Syst.*, vol. 97, pp. 284–294, Aug. 2019.

[98] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, Apr. 2019.

[99] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Inf. Sci.*, vol. 479, pp. 640–650, Apr. 2019.

[100] Y. Xue, K. Xue, N. Gai, J. Hong, D. S. L. Wei, and P. Hong, "An attribute-based controlled collaborative access control scheme for public cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2927–2942, Nov. 2019.

[101] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Inf. Sci.*, vol. 479, pp. 567–592, Apr. 2019.

[102] C. Yang, Q. Chen, and Y. Liu, "Fine-grained outsourced data deletion scheme in cloud computing," *Int. J. Electron. Inf. Eng.*, vol. 11, no. 2, pp. 81–98, Dec. 2019.

[103] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *J. Syst. Softw.*, vol. 113, pp. 130–139, Mar. 2016.

[104] H. Yin, Z. Qin, J. Zhang, L. Ou, F. Li, and K. Li, "Secure conjunctive multi-keyword ranked search over encrypted cloud data for multiple data owners," *Future Gener. Comput. Syst.*, vol. 100, pp. 689–700, Nov. 2019.

[105] Y. Yu, Y. Mu, J. Ni, J. Deng, and K. Huang, "Identity privacy-preserving public auditing with dynamic group for secure mobile cloud storage," in *Proc. NSS*, New York, NY, USA, 2015, pp. 28–40.

[106] Z. Yu, M. H. Au, Q. Xu, R. Yang, and J. Han, "Towards leakage-resilient fine-grained access control in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 77–763, Jan. 2018.

[107] F. Zhang, Q. Li, and H. Xiong, "Efficient revocable key-policy attribute based encryption with full security," in *Proc. IEEE CIS*, Guangzhou, China, Nov. 2012, pp. 477–481.

[108] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, Mar. 2018.

[109] J. Zhang and Q. Dong, "Efficient ID-based public auditing for the outsourced data in cloud storage," *Inf. Sci.*, vols. 343–344, pp. 1–14, May 2016.

[110] J. Zhang and X. Zhao, "Efficient chameleon hashing-based privacy-preserving auditing in cloud storage," *Cluster Comput.*, vol. 19, no. 1, pp. 47–56, Mar. 2016.

[111] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1566–1577, May 2016.
[112] Z. Wei, "A pairing-based homomorphic encryption scheme for multi-user settings," *Int. J. Technol. Hum. Interact.*, vol. 12, no. 2, pp. 72–82, Apr. 2016.
[113] X. Zhang, J. Zhao, C. Xu, H. Li, H. Wang, and Y. Zhang, "CIPPPA: Conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors," *IEEE Trans. Cloud Comput.*, early access, Jul. 10, 2019, doi: 10.1109/TCC.2019.2927219.
[114] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.
[115] Y. Zhang, M. Yang, D. Zheng, P. Lang, A. Wu, and C. Chen, "Efficient and secure big data storage system with leakage resilience in cloud computing," *Soft Comput.*, vol. 22, no. 23, pp. 7763–7772, Aug. 2018.
[116] D. Zhe, W. Qinghong, S. Naizheng, and Z. Yuhan, "Study on data security policy based on cloud storage," in *Proc. IEEE IEEE 3rd Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, Beijing, China, May 2017, pp. 145–149.
[117] F. Yundong, W. Xiaoping, and W. Jiasheng, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *Proc. DSC*, Shenzhen, China, Jun. 2017, pp. 205–212.

PAN YANG received the B.S. and M.S. degrees in pure mathematics from Zhengzhou University, Zhengzhou, China, in 2017 and 2019, respectively, where she is currently pursuing the Ph.D. degree in applied mathematics with the School of Mathematics and Statistics. Her current research interests include the data security, differential equation, and data science.

NAIXUE XIONG (Senior Member, IEEE) received the Ph.D. degree from Wuhan University and the Ph.D. degree from the Japan Advanced Institute of Science and Technology. He is currently an Associate Professor (5th year) with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. He has published over 200 international journal articles and over 100 international conference papers. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Information Science*, and an Editorial Member for over ten international journals.

JINGLI REN received the Ph.D. degree in applied mathematics from the Beijing Institute of Technology, in 2004.

She became a Professor with the School of Mathematics and Statistics, Zhengzhou University, in 2006, where she is currently the Deputy Dean of the Henan Academy of Big Data. She is also a Humboldt Scholar of Germany and a Distinguished Professor of Henan Province. She has published over 80 international journal articles. Her research interests include applied mathematics, applied statistics, and data science. She conducted four Projects of National Nature Science Foundation of China, one Alexander von Humboldt Fellowship for Experienced Researcher, and five Provincial Projects.