Data Sci. Eng. (2018) 3:24–39
https://doi.org/10.1007/s41019-017-0046-0

Data Privacy Protection Mechanisms in Cloud

Niharika Singh[1] · **Ashutosh Kumar Singh**[2]

Received: 4 May 2017 / Revised: 4 September 2017 / Accepted: 24 September 2017 / Published online: 25 November 2017

Abstract In the cloud computing environment, the privacy of the electronic data is a serious issue that requires special considerations. We have presented a state-of-the-art review of the methodologies and approaches that are currently being used to cope with the significant issue of privacy. We have categorized the privacy-preserving approaches into four categories, i.e., privacy by cryptography, privacy by probability, privacy by anonymization and privacy by ranking. Moreover, we have developed taxonomy of the techniques that have been used to preserve the privacy of the governing data. We also presented a comprehensive comparison of the privacy-preserving approaches from the angle of the privacy-preserving requirements' satisfaction. Therefore, it is highly desirable that the mechanisms should be developed to deploy efficient auditing and accountability mechanisms that anonymously monitor the utilization of data records and track the provenance to ensure the confidentiality of the data.

Keywords Cloud computing · Data privacy · Data utilization · Privacy preserving

1 Introduction

In simple terms "the cloud" can be foreseen as a metaphor for the internet which is quite familiar cliché, but when it is combined to the term "computing," its meaning gets bigger and hazy. Cloud computing proposes the opportunity to organizations that would merely connect to the cloud and use the available resources on a *Pay Per use* basis that avoids the company's capital expenditure on supplementary of premises infrastructure resources. It promptly scales up and scales down rendering to business requirements. It consists of cloud client, services, application platform, storage and infrastructure measured services. Thus, the cloud computing is highly automated utility-based paradigm shift comprises of efficient and optimized framework that includes virtual desktops, servers and allocates services for computer network over the internet suggesting software applications and platform for easy and agile deployment of the secure data management.

The technology provides broad network access using resource pooling, on demand self-service with rapid elasticity. It results in continuous high availability, interoperability and standardized scalability for the hardware and software components providing data secrecy and ease for capital investment. Cloud storage is a system that allows users to store their sensitive and personal data in a secure way, making them able to access their data anywhere, at any time from any of the authorized devices. The cloud storage is usually dynamic in nature as it permits controlling the accessibility of the data by deleting/adding new users and devices. This idea is pursued to safeguard the users' important data, where cloud storage plays the role of an access control. Veracity depicts that the data are encrypted, but in many cases cloud servers has the decryption key and manages the rights for user accessibility. It is sensed as a critical problem in the case of private sensitive data records, such as administrative documents (e.g., bills or pay sheets, ID cards) [1]. Also, increasing amount of storage and computing requirements of users is outsourced to remote, but generally are not necessarily trusted servers. These blow off several privacy issues regarding accessing data on such servers that can be

✉ Niharika Singh
niharika.academics@gmail.com

[1] Universiti Teknology PETRONAS, Malaysia, Malaysia

[2] National Institute of Technology, Kurukshetra, India

defined as: sensitivity of (a) keywords sent in queries and (b) the retrieved data; both need to be hidden [2]. Hence, the security and privacy are entitled to be the most important issues. Particularly, the necessity and importance of privacy-preserving search techniques are still more pronounced in the cloud applications. The fact behind is the large companies that operate public clouds like Amazon Elastic Compute Cloud, Google Cloud Platform or Microsoft Live Mesh may access sensitive data such as access and search patterns. Also, hiding the query and the retrieved data has importance to ensure security and privacy for those using cloud services. Also, our research is based upon identifying privacy protection mechanisms processed over cloud standards, SaaS and DaaS, i.e., Software as a Service and Data as a Service, respectively, see Fig. 1. [There are mainly five standards represented by XaaS ("X" as a Service) where "X" can be varied, i.e., SaaS, PaaS, EaaS, DaaS, IaaS, depicted as Software, Platform, Education, Data and Infrastructure as a Service, respectively]. Furthermore, data are usually tackled by leveraging existing encryption cryptographic methods, such that only outsourced data are encrypted and are inaccessible by cloud servers that enable to protect the confidentiality of the data. These methods limit the flexibility of data retrieval and prevent the ciphertext holder from gaining access to the knowledge of the data [3].

In [4] it is claimed that storing huge volumes of data records in third-party cloud is susceptible to leakage, loss or theft. Traditional network security and privacy mechanisms are not quite sufficient for data storage and outsourcing. Thus, integrity and confidentiality of the stored data records are initiated as one of the major challenges elevated by external storages.

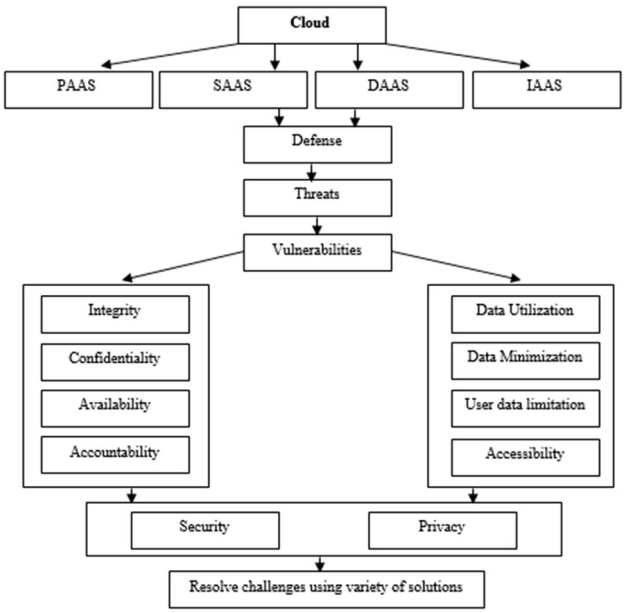

Fig. 1 Generalized view of privacy in the cloud

On the basis of Fig. 1, we can generalize the major factors responsible for comparing various approaches proposed to preserve privacy in the cloud. Conventionally, general privacy protection technologies are classified into three major categories, i.e., privacy by policy, privacy by statistical analysis, privacy by cryptography [4], but our study find that various architectural models should be divided into two major fields, i.e., cryptographic and non-cryptographic approaches. These are further segregated into subfields that can be referred from Fig. 2. Xiao et al. and Takabi et al. give overview about existing privacy and security issues in a cloud environment using contemporary privacy measures. Now, we are presenting a taxonomy of the approaches that have been used for preserving data privacy in the cloud.

2 Preliminaries

2.1 Basic Hybrid Architecture for Data Utilization

As shown in Fig. 3, there are four entities defined as the foundation for each following comparative system approaches, that is, data users/owners, attribute authority, public cloud and private cloud.

2.1.1 Attribute authority

It represents the key authority for attributes. Usually, it is in authority for generating private and public parameters for the system. Additionally, it is in charge of revoking, issuing and updating attribute private keys for various users.

2.1.2 Public cloud

This entity is responsible for a data outsourcing service. It assists in controlling the access from outside users to the stored data. It also works for the storage servers and providing corresponding content services via honestly executing the planned searching algorithm. In some cases, it is assumed that the public cloud is always online and has abundant computation power and storage capacity.

2.1.3 Data owners

This user node owns data files and bids to outsource them to the external storage server (i.e., the public or private cloud). It is responsible for enforcing and defining an access policy on its own files by encrypting them. Besides, the data owner also wants to delegate/finish the task of generating some searchable information for encrypted files

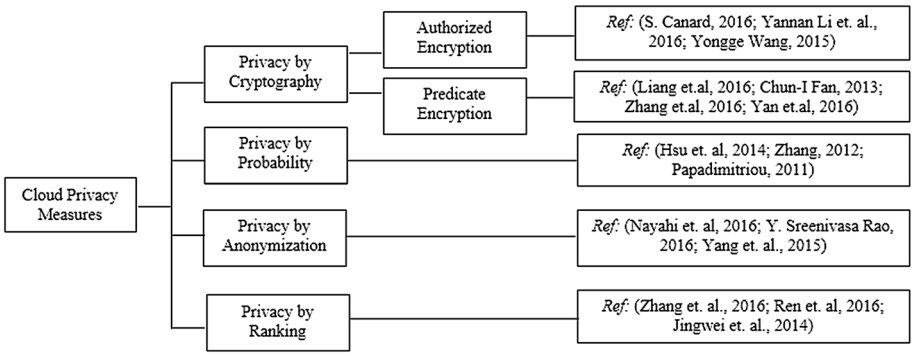

Fig. 2 Data privacy-preserving approaches

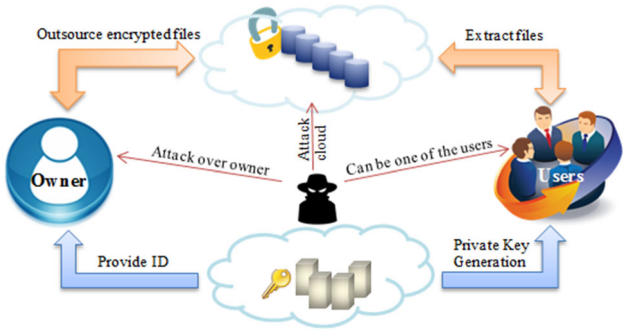

Fig. 3 Architecture for data utilization in cloud computing, including attack-prone areas

in order to make efficient utilization of the outsourced encrypted data files.

2.1.4 Data users

This entity is the one who wishes to access an outsourced file. Such users are able to delegate the task of generating "trapdoors" for searching keywords. For example, on receiving searching results, if the user is initially authorized to embed in the encrypted file but later, mistakenly it is not revoked, such user will be able to decrypt the ciphertext and retrieve the file.

2.1.5 Private cloud

This entity facilitates a user's secure usage of a cloud service. Specifically, the computing resources at the data user/owner side are restricted and the public cloud is less trusted in practice. A private cloud is capable to provide a data owner/user with an execution environment and infrastructure working as an interface between the public cloud and user. The interface offered by the private cloud allows users for submitting files and queries to be securely stored and computed, respectively.

Notice that Fig. 3 represents a novel architecture for the data utilization in cloud computing, which consists of the public cloud and a private cloud (i.e., twin cloud). Recently, such hybrid cloud setting has attracted lots of attention. On the other hand, the trusted private cloud could be a cluster of virtualized cryptographic co-processors. These might be offered as a service by a third party and offer the necessary hardware-based security types to implement a remote execution surroundings for privacy preserving by the users.

2.2 Adversary Model

Typically, it is assumed that the private cloud and public cloud are both "honest-but-curious" [5].

Precisely following approaches will follow the protocol, but try to find out as much secret information as possible based on their possessions that would help us detect the efficient approach. Fortunately, users would try to access data either within or out of the scope of their authorization. As foundation, one may assume that both keywords and files are sensitive. In this review paper, we suppose that all the files are sensitive and need to be fully protected against cloud environment, while keywords are semi-sensitive or quasi, and allowed to be known by the private cloud.

3 Cryptography-Oriented Privacy Measures

The cloud storage preserves data as if it is stacked in a locker, where the cloud storage acts the role of an access control to this locker. In reality, the data are encrypted, but usually the cloud server has the decryption key which manages the rights for each user to access the data. This is a critical problem in the case of private sensitive data such as administrative documents (e.g., bills, pay sheets or identity cards,) or, more generally, personal data. This is quite tricky in the case of confidential documents possessed by a business enterprise, i.e., shared between the collaborators or with trading partners. In fact, this problem can be effortlessly solved by merely encrypting the data before sending it to the cloud/safe. Different architectures were proposed to resolve this problem which work over two policies Trust-evaluation and predicate encryption, explained as follows.

3.1 Trust-evaluation/Authorized Encryption

The policy imposes to build users' trust in the evaluation of reliable systems, especially against market-accepted criteria. In concern to user trust, a contract dealing with the use of a key management system should point out the jurisdiction whose laws relate to that system.

3.1.1 Using Fine-Grain Management of the Rights

An advanced cryptographic tool called as "proxy re-encryption" scheme is used in [1]. Taking it as foundation, Canard et al. modified this scheme in a way that customers could manage their shared documents dynamically in a tree-based structure. Unfortunately, these changes were not that sharp to cope with existing systems. Then they presented the first true implementation of such system that includes smartphones to upload, download and share client's documents. It is focused on the fine-grain management of the rights, i.e., claiming and sharing the authorization of individuals on the basis of user priority.

Here, the problem is focused on the fine-grain management of the rights. Generally, the standard PRE scheme has a sharing property of "all or nothing." If the re-encryption key is generated by client A, then the proxy can re-encrypt for client B any document initially encrypted to client A. But, there is no way for client A to restrict what the proxy can re-encrypt or not, except by trusting it. In such case, if the storage space is structured as a tree, client A may want to only share a specific folder F_x, or any specific files $f_{x,y}$, but not all the files. This problem can be resolved using the conditional PRE scheme. In concern, let a unique condition $\omega_{f_{2,1,1}}$ that is defined during the encryption process is attached to each uploaded file $f_{2,1,1}$. In order to obtain ciphertext encryption using client A public key (pk_A) will be $(pk_A, f_{2,1,1}, \omega_{f_{2,1,1}})$.

In other case, if client A wishes to share its right to client B for folder F_2, then the re-encryption key is computed from A to B under a condition related to F_2. That generated key is certainly denoted as $rk_{A\to B,F_2}$, and further will be sent to the proxy that permits vertical transformation between users.

The third case needs more attention as there should be a particular path to traverse back to the root from the specific file location. To resolve this issue, after the vertical transformation a horizontal transformation is added inside the tree. It uses additional re-encryption keys such as $rk_{F_{2,1}\to F_2,A}$ or $rk_{f_{2,1,1}\to F_{2,1},A}$ that are modified ciphertext attached conditions, as shown in Fig. 4.

These steps look quite complicated, but precisely, it works upon a simple idea that is "for each couple (file, folder) or (folder, folder) in the path of the file to the root, client A needs to compute a re-encryption key (but only once for each link between folders)." This proxy re-encryption is resulted to be a good tool that has successfully maintained privacy by permitting the use of non-trusted platform for storing the sensitive documents. It can be further adapted to integrate additional features, e.g., de-duplication, indexation or some usual and unusual complex computations over encrypted data.

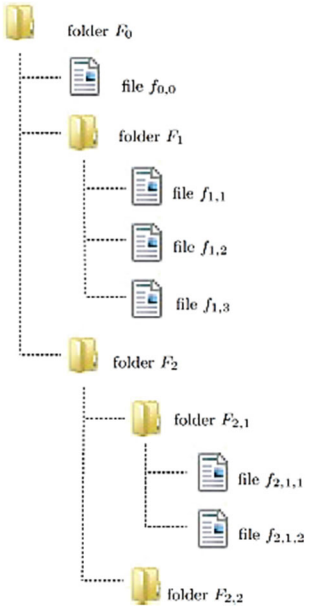

Fig. 4 Preferred tree-based structure to re-encrypt keys [1]

3.1.2 Using the Remote Data Auditing (RDA)

RDA technique falls in the cryptography category because it either provides a probabilistic or deterministic assurance for data intactness [6]. It embraces the properties like (a) *Efficiency*: data auditing with least possible computational complexity; (b) *Public Verifiability*: to allow data auditing process delegation to a trustworthy third-party auditor (TPA) and reducing the computational burden on the client's side; (c) *Detection Probability*: to check potential data corruption detection probability. In concern to maintain privacy certainly a critical issue related to data auditing is required to resolve. This is the case, when digital certificate expires in the PKI system, then it is needed to update cloud users and authenticators associated with the key. Yannan et al. proposed an authenticator-evolving and key-updating mechanism with the zero-knowledge privacy of the stored files for security. It unites zero-knowledge proof systems, homomorphic linear authenticators and proxy re-signatures. It is a combination of five different algorithms, i.e., *CrsGen*, *KeyGen*, *AuthGen*, *KeyUpdate* and *AuthUpdate* along with an interactive proof system to get a proof between a Prover and a Verifier.

CrsGen(1^k): It takes a security parameter k as input and outputs *crs* as a common reference string, which is an implicit input to all algorithms described below.

KeyGen(crs): to input *crs*, it generates a public key *pk* and a secret key *sk* for the cloud user. The user publishes *pk* and keeps *sk* secret.

AuthGen(sk, F): as input it takes the secret key *sk* and a file $F = (m_1, m_2, \ldots, m_n)$ and outputs a set of authenticators $\{D_i\}$ for this file and a set of public verification parameter φ, which will be used for checking the data integrity in the proof phase.

KeyUpdate(sk, pk): On input the $(l-1)$ old key pairs (sk_{l-1}, pk_{l-1}), the algorithm outputs a new key pair (sk_l, pk_l).

AuthUpdate$(sk_l, pk_l, ft_{l-1}, \varphi)$: On input a new key pair (pk_l, sk_l) the original file tag ft_{l-1} and the public verification parameter φ, it outputs a new file tag ft_l and the new update key β_l that are valid under the new key pair.

Proof (P, V) This interactive protocol between the Prover (P) and the Verifier (V) takes a common input to (P, V) which is the public key *pk* and the public verification parameter φ. P has additional input the file $F = (m_1, m_2, \ldots, m_n)$ and a set of authenticators $\{D_i\}$ of this file. At the end of the protocol, V outputs a bit 1 or 0 to indicate whether the stored file is kept intact or not. For notational convenience, we use $P \Leftrightarrow V(pk, \varphi) = 1$ to indicate that V outputs 1 at the end of the interaction with P. We omit the parameters (pk, φ) when the context is clear.

For the privacy-preserving public auditing scheme, author feels soundness, completeness and data privacy are three security requirements. Completeness means that when interacting with the cloud server who keeps the data unchanged, the interactive protocol proof will always result in $P \Leftrightarrow V = 1$ when the cloud server and the TPA follow the protocol honestly. Thus, the concept of zero-knowledge data privacy helps to capture that the TPA learns no knowledge about the processed content except publically available information-based random file name. This is further strengthen on the basis of evaluation properties and proved the security including soundness is efficient and can be used in practice.

3.1.3 Using BP-XOR Gates

LT codes, LDPC codes and digital fountain techniques have received considerable attention from both industry and academics in the past few years. BP-XOR gates works as a productive approach to get Trust-evaluation in the field of cryptography [7].

In order to employ the underlying ideas of competent belief propagation (BP) decoding process in LDPC and LT codes, the paper plans the BP-XOR codes and uses them to project three classes of secret sharing schemes called pseudo-BP-XOR secret sharing schemes, LDPC secret sharing schemes and BP-XOR secret sharing schemes. By inducing the equivalence between the edge-colored graph model and degree-two BP-XOR secret sharing schemes, authors designed novel and ideal 2-out-of-n BP-XOR secret sharing schemes.

By employing techniques from array code design, it is also able to design other (n, k) threshold LDPC secret sharing schemes. In the efficient LDPC or BP-XOR secret sharing schemes that it builds, only linear number of XOR (exclusive-or) operations on binary strings are necessary for both secret reconstruction phase and secret distribution phase.

For a comparison, one should note that Shamir secret sharing schemes need $O(n^2)$ field operations for the secret reconstruction phase and $O(n \log n)$ field processes for the secret distribution phase. Additionally, author claims that such schemes attain the optimal update complexity for the secret sharing schemes. By update complexity for a secret sharing scheme, author meant that the average number of bits in the participant's shares that needs to be revised when certain bit of the master secret is changed. It is also requested the efficient secret sharing schemes discussed in this paper considerably used for massive data storage in cloud environments for achieving reliability and privacy without employing encryption techniques. On the basis, various similar approaches are compared in Table 1.

3.2 Predicate Encryption

This predicate encryption is considered as a novel cryptographic primitive which provides a fine-grained control over the encrypted data accesses. It is usually used for the biometric matching and secure cloud storage.

3.2.1 Achieving Regular Language Search

"Regular language search" is concerned to languages in automata. It first defines a new notion called searchable deterministic finite automata-based functional encryption. The notion is a general notion for PEKS [3]. Next, it designs a concrete construction satisfying the notion. In its construction, any system user can describe a data to be shared with "regular language" in an encrypted form, where the language description can be arbitrary length (e.g., an English sentence, or a paragraph). A valid data receiver can generate and deliver a search token represented as a deterministic finite automata (DFA) to a cloud server, such that the cloud server can locate the corresponding ciphertexts and return them to the data receiver. In the search phase, the server knows nothing about the

Table 1 Comparison of trust-based evaluation approaches

Architecture design	Approach	Implementation concept	Highlights/ problem domain	Advantages	Disadvantages
Canard [1]	Using advanced cryptographic tool, i.e., "proxy re-encryption"	Data sharing environment by modifying proxy re-encryption paradigm	Managed dynamically a tree structure for their shared document	Additional features like indexation, de-duplication can be integrated	From the user's point of view, the potential feeling slow is related to the size of the file
Yannan Li et al. [6]	Formalizing the model of zero-knowledge privacy for auditing with key update	Based on a concrete construction utilizing unidirectional proxy re-signature	Proof for better soundness and zero-knowledge privacy under the model	Reduces the communication and computation cost while maintaining the desirable security	Seems unpredictable in case of large datasets
Wang [7]	Employing the underlying ideas of efficient belief propagation (BP) decoding process in LDPC and LT codes	Designing the BP-XOR codes and use them to design three classes of secret sharing schemes called BP-XOR, pseudo-BP-XOR and LDPC secret sharing schemes	Achieving the optimal update complexity	When deployed, the users should make the assumption that cloud servers will not collude which is hard to achieve in some cases	It will be generally hard to prevent collusion attacks

search contents and the underlying data. They further present extensive evaluation for the system to show its security, and the efficiency compared to two most related works [8, 9].

Regular language search is productively effective for this system. This makes the system be the first of its type, to the best of our knowledge. It is undeniable that SSE (e.g., [10]) usually enjoys better efficiency in data searching compared to the public key-based searchable encryption. However, this novel system can support any arbitrary alphabet/regular language search, so that it is more human-friendly readable for search keyword design. Besides, the system provides verifiable (data integrity) check for system users (due to public-key-based feature). Moreover, the system does not need to require a data owner to pick up some special keywords before constructing keyword index structures, e.g., least frequent keyword, but also it only leverages a DFA structure to embed flexible search expressiveness, e.g., "AND, OR, NOT," unlike that of only limited in "a keyword AND (formula)" expression.

Author assumes, let *BSetup* be an algorithm that on input the security parameter n, outputs the parameters of a bilinear map as $(p, g, \hat{g}, G_1, G_2, G_T, e)$, where G_1, G_2 and G_T are multiplicative cyclic groups of prime order p, where $|p| = n$, and g is a random generator of G_1, $\hat{g}$ is a random generator of G_2.

This further defines the complexity assumptions that determine the base of the work. Considering the *(Asymmetric) l-Expanded BDHE Assumption:* It depicts that an algorithm A has advantage $Adv_A^{A-l-BDHE}$ in solving the asymmetric l-Expanded BDHE problem in (G_1, G_2) if $|\Pr\left[A\left(\hat{X}, e(g,\hat{g})^{a^{l+1}bs}\right) = 0\right] - \Pr[A(\hat{X}, T) = 0| \geq \in$, where the probability is over the random choice of generators $g \in G_1, \hat{g} \in G_2$, the random choice of exponents $a, b, c_0, \ldots, c_{l+1}, d \in Z_p^*, T \in_R G_T$, the random bits used by A, and $\hat{X}$ is a set of the following elements:

$$
\begin{aligned}
&g, \hat{g}, \hat{g}^a, g^a, \hat{g}^{ab/d}, g^{ab/d}, g^{b/d}, \hat{g}^{b/d}\\
&\quad \forall i \in [0, 2l+1], \quad i \neq l+1, \quad j \in [0, l+1] g^{a^i s}, g^{a^i bs/c_j},\\
&\quad \forall i \in [0, l+1] g^{\frac{a^i b}{c_i}}, \hat{g}^{\frac{a^i b}{c_i}}, \hat{g}^{c_i}, \hat{g}^{a^i d}, \hat{g}^{\frac{abc_i}{d}}, \hat{g}^{\frac{bc_i}{d}},\\
&\quad \forall i \in [0, 2l+1], \quad j \in [0, l+1] \hat{g}^{\frac{a^i bd}{C_j}}\\
&\quad \forall i, j \in [0, l+1], \quad i \neq j \hat{g}^{\frac{a^i bc_j}{c_i}}.
\end{aligned}
$$

It says the asymmetric l-Expanded BDHE assumption holds in (G_1, G_2) if no PPT algorithm has advantage $\in$ in solving the asymmetric l-Expanded BDHE problem in (G_1, G_2). It shows that the above extended complexity assumption still holds in the generic group model by employing the same proof technology introduced in [11]. Specifically, one can see from the set $\hat{X}$ that there are five elements in G_1 including $g, g^{b/d}, g^a, g^{ab/d}, g^{a^i s}$, $g^{a^i bs/C_j}, g^{a^i b/c_i}$.

Here, it shows that these elements cannot help an adversary to compute an exponent value $a^{l+1}bs$. For the element g, it is easy to see that there does not exist a $\hat{X}^{a^{l+1}bs}$ in $\hat{X}$. Similarly, it needs the G_2 elements $\hat{g}^{a^{l+1}ds}, \hat{g}^{a^l ds}, \hat{g}^{a^l bs}$, $\hat{g}^{a^z s}, \hat{g}^{a^z c_j}$ and $\hat{g}^{a^z s c_i}$ for the G_1 elements $g^{b/d}, g^a, g^{ab/d}$, $g^{a^i s}, g^{a^i bs/C_j}, g^{a^i b/c_i}$, respectively, where $i + z = l + 1$. It is

not difficult to see that the above G_2 elements cannot be provided by the set $\hat{X}$. However, the structure and the pattern is the premise of allowing users to achieve regular language search. Though, the premise sustains a shortage in the search token storage cost.

3.2.2 Using Key Encryption

Over encrypted cloud storage services such current privacy-preserving search schemes do not provide effective revocation for search privileges [12]. Targeting at symmetric predicate encryptions and cloud storage requirements, in this paper author proposed controllable privacy-preserving search in cloud storage. This scheme is based on Private-key hidden vector encryption with key confidentiality. Although, its efficiency is much better than approach present in [13]. In [13] author presented a symmetric-key predicate encryption scheme in support of inner product queries that considers predicate encryption in the symmetric-key setting.

In contrast, the controllable privacy-preserving search scheme has two new features. One is revocable delegated search that helps secret key owner to control the lifetime of the delegation. But, the other is undecryptable delegated search. Due to this feature, a delegated person cannot decrypt the returned matched ciphertexts even though one has the delegated privilege of search. Here, in Table 2, various predicate encryption-based approaches are compared to highlight the concepts.

Viewpoint: The category "privacy by cryptography" considers the original sensitive data and then is perturbed. This perturbation makes the data less sensitive which increases the integrity simultaneously. It surely helps in storing and sharing sensitive data in the cloud environment safely where storage-based computing resources are provided by a third-party service provider. But, this

Table 2 Comparison of predicate-based evaluation approaches

Architecture design	Approach	Implementation concept	Highlights/problem domain	Advantages	Disadvantages
Liang et al. [3]	Supports an extreme expressive search mode, regular language search in comparison with other similar schemes	System user describes data to be shared with regular language in encrypted form, where the language description can be arbitrary length	DFA associated with the token provides a fine-grained character (which is from a given alphabet) match pattern	The premise incurs a shortage in search token storage cost	System suffers from the largest storage cost for search token, while other systems have similar space cost
Chun-I Fan [12]	To propose a variant of symmetric predicate encryption, which provides controllable privacy-preserving search functionalities	Includes revocable delegated search and undecryptable delegated search	Functionalities help the owner of a cloud storage can easily control the lifetimes and search privileges of cloud data	Suitable for delegation-based business applications in cloud computing	Does not support complex access control and search privileges
Zhang et al. [14]	System complies with legal requirements and overcomes internal attacks	Designing trust management system that controls unwanted traffic with privacy protection	Evaluation on extra costs introduced by privacy protection shows system practicality	Semantically secure against chosen plaintext (IND-CPA) attacks if computational composite residuosity assumption (CCRA) holds	The proposed system introduces more computation time, incurs more communication overhead and consumes more storage, compared with the original GTM system
Yan et al. [15]	To propose two practical schemes to guard privacy of trust evidence providers based on additive homomorphic encryption	Use of Trust-evaluation algorithms cooperating with the PPTE schemes to resist internal attacks	Overcome attacks raised by internal malicious evidence providers to some extent even though the Trust-evaluation is partially performed in an encrypted form	First scheme achieves better computational efficiency, while the second one provides greater security	Compromise with expense of a higher computational cost

communication raises serious concern for the adoption of advanced computing technologies to cope with individual privacy. These issues can be resolved by the following privacy categories.

4 Probability-Oriented Privacy Measures

Traditionally, watermarking was preferred to preserve the data privacy. It works on serving to the agents after performing data modifications. If the data are discovered in the hands of an unauthorized agent (who is a malicious recipient), then the watermarks can be destroyed. Thus, in support of privacy by probability some best approaches are presented here, as follows:

4.1 Effective Solution for Data Leakage

In [16] author deliberates the following problem: "A data distributor gives sensitive data to a set of allegedly trusted agents (third parties). Some of the leaked data are found in an unauthorized place (e.g., on somebody's laptop or on the web). The distributor must weigh the outlook that the leaked data came from one or more agents, as opposed to having been independently gathered by other unauthorized means." In concern to agents, author has proposed data allocation strategies that improve the probability of identifying leakages. Such methods do not depend on alterations of the released data (e.g., watermarks). In some cases, author also injects "realistic but fake" data records to further improve the chances of detecting leakage and identifying the guilty agent or party.

It considers basic entities and agents to build guilty agent detection system. Following the impression that has a distributor who owns a set $T = \{t_1, t_2, \ldots, t_p\}$ of valuable data objects.

The distributor desires to share some of the objects with a set of agents $U, U_2, \ldots, U_q$, which do not wish the objects be leaked to other third parties. The objects in T might be of any type and size, for example, it could be tuples in a relation, or relations in a database.

An agent U_i receives a subset of objects $R_i \subseteq T$, that is, determined either by an explicit request or sample request:

- Sample request $R_i = \text{SAMPLE}(T, m_i)$: Any subset of m_i records from T can be given to U_i.
- Explicit request $R_i = \text{EXPLICIT}(T, \text{cond}_i)$: Agent U_i receives all T objects that satisfy cond_i.

This recruits a procedure which presumes that after sending objects to agents, the distributor notices a set $S \subseteq T$ has leaked. This means that some third party, called the target, is caught in possession of S. For example, the target might be displaying S on its website, or perhaps as part of a legal discovery process, the target turned over S to the distributor.

Since the agents $U, U_2, \ldots, U_q$ have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent and that the S data were obtained by the target through other means. For example, assume that one of the objects in S represents a customer X. Perhaps X is also a customer referred to some other company which provides the data to the target agent. It is also possible that X can be reconstructed from several publicly available sources on the web. The goal is to guess the precise likelihood that the leaked data came from the agents as opposed to other sources. Instinctively, the more data in S, the harder it is for the agents to argue they did not leak anything.

The process of consideration and evaluation can compute the probability $\Pr\{G_i|S\}$ that agent U_i is guilty

$$\Pr\{G_i|S\} = 1 - \prod_{t \in S \cap R_i} \left(1 - \frac{(1-p)}{|V_t|}\right)$$

Here, p is the probability of the event occurred and V_t stands for the set of agents.

Authors [16] proposed an algorithm that finds agents which are eligible to receiving fake objects in $O(n)$ time. Then, in the main loop the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop takes $O(B)$ time, where B refers to Fake object creation. Hence, the running time of the algorithm is determined to be $O(n + B)$.

In spite of any difficulty, authors have worked on predicting that it is possible to assess the likelihood that an agent is responsible for a leak, established on the data overlapping with the leaked data and the data of other agents. It is based on the probability that objects can be "guessed" by other means. The model is moderately simple, but it is believed that it seizures the essential trade-offs. The algorithms described here implement a diversity of data distribution strategies that can improve the distributor's chances to identify a leaker. Authors represented that distributing objects cautiously makes a significant difference in identifying guilty agents, particularly in the case where agents must receive large overlap in the data.

The future work includes the investigation of agent guilt models that capture leakage scenarios, for example, determining the suitable model for cases where various agents can collude and identify fake tuples. An initial discussion of such a model is available in [17]. Another open problem is the extension of the allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance).

5 Using Probabilistic Hybrid Logics

In [18], Hsu et al. proposed a combination of basic hybrid logic and quantitative uncertainty logic with a satisfaction operator. This technique is highlighted individually in comparison with the other approach due to its special features.

The logic is expressive and flexible enough to represent many existing privacy criteria, such as *k*-anonymity, logical safety, *l*-diversity, *t*-closeness, and δ-disclosure. The main contribution of the logic is twofold. On the one hand, the uniformity of the framework explicates the common principle behind a variety of privacy requirements and highlights their differences. For example, the difference between syntactic and semantic privacy criteria is easily observed by using the logical specifications. On the other hand, the generality of the framework extends the scope of privacy specifications.

In particular, one can specify heterogeneous requirements between different individuals, so it is possible to achieve personalized privacy specification. For example, one can use $@_i\neg[a_1]\emptyset \wedge @_j\neg[a_1]\psi$ to express different privacy requirements of individuals *i* and *j*. Moreover, the logic allows arbitrary combinations of existing privacy requirements, so we can express compound privacy criteria. For example, we can use $@_i\neg[a_1]\emptyset \wedge l_{a_1}(i) \leq \frac{1}{k}$ to express that both logical safety and *k*-anonymity are required for the individual *i*. Since unexpected attacks may occur occasionally, existing criteria may be inadequate; hence, it may be necessary to specify new criteria. For example, the logical safety criterion may be combined with δ-disclosure to require formulas in Sec(*i*), instead of simply *f*-atoms, to satisfy the δ-disclosure privacy criterion. In addition, it is possible to consider the weight of a secret in order to measures the seriousness of revealing the secret. Thus, $W\ \mathrm{sec} : U \times \Gamma \rightarrow [0,1]$ is defined as the weight function for each individual and secret. Then, we can combine the weight with existing privacy criteria to obtain new privacy protection models. This may facilitate a more effective trade-off between privacy protection and data utility. The logic language provides a uniform framework to meet the specification needs of such new criteria as well as existing ones. See Table 3 for various similar approaches proposed earlier.

6 Anonymization-Oriented Privacy Measures

This category of privacy perseverance comprises subset generations of well-known method, "anonymization." Based on secured multiparty computation, collaborative privacy-preserving data mining sustains high computational cost and communication. Data anonymization is an encouraging technique in the field of privacy-preserving data mining castoff to protect the data against identity disclosure. Common attacks and information loss possible on the anonymized data are intense challenges of anonymization. Lately, data anonymization using data mining techniques has showed noteworthy enhancement in data utility. Still the prevailing techniques lack in effective handling of attacks. Out of numerous proposed methods, the following privacy approach based on anonymization sounds appealing in a few comparative terms with other categories.

7 For Utility Preserving Data Clustering

In this paper [20], an anonymization algorithm established on clustering and resilient to similarity attack and probabilistic inference attack is proposed. The anonymized data are dispersed on hadoop distributed file system. The method attains a better trade-off between privacy and utility. In this work, the data utility is measured in terms of accuracy and *F* measure with respect to different classifiers.

Nayahi and Kavitha [20] proposed a KNN(G,S) clustering algorithm to achieve anonymized clusters each with uniform distribution of sensitive values. The (G,S) clustering algorithm [21] governs the single best neighbor of each cluster and allocates one instance at a time to the existing cluster. The authors have modified the algorithm to overcome the skew in the sensitive value distribution of the resultant clusters using the K-nearest neighbor technique. The KNN-(G,S) clustering algorithm determines the KN nearest neighbors using the following equation from each sensitive value group and adds the *KN* records to the clusters at a time.

$$\mathrm{KN} = |D_i|/\mathrm{NOB}$$

where $|D_i|$ is the number of instances in each sensitive value subgroup (D_i); and NOB is the number of clusters. This formula divides the records equally among all the clusters. Hence, there will be an evenly spread distribution of sensitive values in the formed clusters with original data set. The given data set D is sorted with respect to the sensitive attribute value *SA*. After sorting, the input data set is divided into D_1, D_2,…, D_n subgroups. Each subgroup would contain identical values for *SA*. All the remaining subgroups other than $D_{\min}$ (i.e., the cardinality of the least frequent value in SA or smallest group) are considered as D_{rem}. Based on the values of *k* and *S,* the algorithm works in two cases. In Case 1 the *k* value is less than or equal to *S* $(k \leq S)$, and in Case 2 the *k* value is greater than

Table 3 Comparison of probability-based evaluation approaches

Architecture design	Approach	Implementation concept	Highlights/problem domain	Advantages	Disadvantages
Hsu et al. [18]	The logic allows arbitrary combinations of existing privacy requirements, so compound privacy criteria can be expressed	Designing a probabilistic hybrid logic for the specification of data privacy requirements	Facilitate a more effective trade-off between privacy protection and data utility	Logic is expressive and flexible enough to represent many existing privacy criteria, such as *k*-anonymity, logical safety, l-diversity, t-closeness, and §-disclosure	More robust approach can be expressed using the concept as it lacks in representation
Papadimitriou [16]	To detect data leakage using probability when sensitive data are shared between trusted agents	Proposed data allocation strategies (across the agents) that improve the probability of identifying leakages	Approach do not rely on alterations of the released data (e.g., watermarks)	During the large overlap in the data distributing objects judiciously can make a significant difference in identifying guilty agents	Opens many loopholes for different kind of problems which requires to be solved by extending this method graciously
Zhang [19]	Generation of noise injections to manipulate recording strategies of malicious service providers	HPNGS generates noise requests based on their historical probability	For effective customer privacy protection, the number of noise requests should be kept as few as possible	Approach can significantly reduce the number of noise requests over its random counterpart by over 90%	Environment considers and tackles with individual malicious service providers but not in parallel

S ($k > S$). It claims that it has reduced the complexity also as compared to other similar approaches.

Though, the worst case analysis on the computational cost of the KNN-(G, S) clustering algorithm is figured and represented using big O notation. On the other side, the worst case storage cost of proposed algorithm is given by $S_1(n)$. The given data set is divided into sub-groups and stored separately incurring an additional storage cost shown in $S_3(n)$. The storage cost of the clusters formed in either Case 1 or Case 2 of the algorithm is given by $S_{5,6}(n)$. The total size of all the clusters would be equal to the size of the original data set. The total storage cost of algorithm is given by $S(n)$. After simplifying the storage cost of the KNN-(G,S) clustering algorithm is $O(n)$

$$S_1(n) = O(\log_2 n)$$

$$S_3(n) = O(|n_1|+|n_2|+\cdots+|n_S|) = O(n)$$

$$S_{5,6}(n) = O(|C_1|+|C_2|+\cdots+|C_S|) = O(n)$$

$$S(n) = S_1(n) + S_3(n) + S_{5,6}(n)$$

$$S(n) = O(\log_2 n) + O(n) + O(n)$$

$$S(n) = O(\log_2 n) + O(2n)$$

$$S(n) \in O(n).$$

One of the best features of this approach is that it overcomes the possibility of probabilistic inference attack easily. The possibility is compared to the other techniques based on *t*-closeness [22, 23].

The existing techniques on data anonymization do not show such comparison. Specifically, most of the existing techniques assess the performance using traditional metrics only. Unlike those, the information loss is measured in terms of traditional metrics such as global certainty penalty [4, 24], non-uniform entropy metric [25], normalized information loss [26, 27], normalized certainty penalty [4], query error [11, 28], sum of squared errors [29]. But, in proposed approach, Nayahi and Kavitha worked best to succeed in the anonymization using centroid-based replacement of *QID* values that is computationally superior to suppression in terms of information loss and less expensive than generalization.

After studying this work, we could judge the effectively summarized facts and the need to include in our review work especially in this particular privacy category. Hence, Table 4 gives an overview of similar comparative approaches.

8 Ranking-Oriented Privacy Measures

For privacy apprehensions, secure searches over encrypted cloud data have encouraged numerous research works under the single-owner model. Basics of "Ranking" is precisely explained by [14, 31] On the contrary, most cloud servers in cutting-edge practice not only serve one owner; instead, they support multiple owners to segment the profits brought by cloud computing. Thus, we classify several

Table 4 Comparison of anonymization-based evaluation approaches

Architecture design	Approach	Implementation concept	Highlights/problem domain	Advantages	Disadvantages
Nayahi et al. [21]	Designing an anonymization algorithm based on clustering and resilient to similarity attack and probabilistic inference attack	The anonymized data are distributed on hadoop distributed file system	The data utility is measured in terms of accuracy and *F* measure with respect to different classifiers	The method achieves a better trade-off between privacy and utility	No comparative results are shown to prove betterment with respect to other methods
Sreenivasa Rao [30]	Proposing a provable secure CP-ABSC scheme for cloud-based PHR sharing system	Construction exhibits short ciphertext size and requires less number of pairings	Scheme provides fine-grained access control, confidentiality, authenticity, signer privacy and public verifiability	Scheme exploits monotone Boolean function predicates and realizes security in standard model	None
Yang et al. [4]	A hybrid solution for privacy-preserving data sharing in cloud environment	Different methods are innovatively combined to support multiple paradigms of medical data sharing with different privacy strengths	Supports multiple data accessing paradigms with different privacy strengths	It shows that the synergy of different privacy-preserving technologies can provide a better balance between the information utilization and privacy protection	Needs to express how performance affects multiple clients accessing the cloud service simultaneously

approaches in two, i.e., (a) for single owner, (b) for the multi-owner cloud environment. Out of various classified approaches, we are reviewing best one for the individual sub-categories.

9 Common Considerations

In order to present the variation for both of the review approaches on a common platform, following terms are considered.

The policy is based upon bilinear map which assumes two cyclic groups G_1 and G_2 of prime order p_1 and G_1 generated by g_1 (known as generator). Thus, bilinear mapping $e : G_1 \times G_1 \rightarrow G_2$ must satisfy the following:

- Bi-linearity property: for all $y, z \in G_1$ and $a, b \in Z_p$, where $Z_{p_1} = \{0, 1, 2, \ldots, p_1 - 1\}$, we are having $e(y^a, z^b) = e(y, z)^{ab}$.
- Computability property: for any, $y, z \in G_1$ there is a polynomial time algorithm to compute the mapping $e(y, z) \in G_2$.
- Non-degeneracy property, $e(g1, g1) \neq 1$.

10 Based on Single-Owner Cloud Environment

In the paper [5], author has aimed at efficiently solving the problem of fine-grained access control on searchable encrypted data for the single-owner cloud system. It considers a hybrid architecture in which a private cloud is introduced as an access interface between the public cloud and user. Under the hybrid architecture, it has considered a practical keyword search scheme which concurrently supports fine-grained access control over encrypted data. Further, the exact keyword search, it grants an advanced scheme under this new architecture to capably achieve fuzzy keyword search. Lastly, based on the scrutiny in both of the schemes that the main computational cost at user side is the ABE scheme, such that author discusses the issue of outsourcing ABE to private cloud to further relieve the computational cost at user side.

In concern, the work examines the problem of privacy-preserving data utilization in cloud computing and intends the data utilization system supporting for the following three:

- *Fine-Grained Access Control* The data owner is permitted to impose an access policy on each file to be uploaded that precisely entitles the set of data users allowed to access. The public cloud is also prohibited from learning the plaintexts of data files.
- *Authorization/Revocation* Each authorized user is capable to get their individual private key to execute search and decryption. When a user's key has been retracted, the user will no longer be able to search and read the outsourced files.
- *Keyword-Based Query* An authorized user is capable to use individual private key for generating a query for assured keywords. Though the public cloud implements a "search" directly on the encrypted data and proceeds the matched files.

11 Using Multi-owner Cloud Environment

The paper [14] offers schemes to deal with privacy-preserving ranked multi-keyword search in a multi-owner model (PRMSM). To empower cloud servers to achieve secure search without knowing the actual data of both keywords and trapdoors, one methodically constructs a novel secure search protocol. To rank the explored results and preserve the privacy of relevance scores between files and keywords, author has proposed a novel additive instruction and privacy-preserving function family. To prevent the attackers from pretending to be legal data users submitting searches and from eavesdropping secret keys, author proposed a novel dynamic secret key generation protocol. PRMSM also works as a data user authentication protocol which supports efficient data user revocation. All-encompassing experiments on real-world datasets endorse the effectiveness and efficiency of PRMSM (Fig. 5).

Individually this multi-owner environment works on the foundation of these two considerations.

Consideration-1 Given a probabilistic polynomial time adversary A, he asks the challenger B for the cipher-text of his submitted keywords for polynomial times. Then A sends two keywords w_0 and w_1 which are not challenged before, to B. B randomly sets $\mu \in \{0, 1\}$ and returns an encrypted keyword $\hat{w}_\mu$ to A. A continues to ask B for the cipher-text of keyword w, the only restriction is that w is not w_0 or w_1. Finally, A outputs its guess μ' for μ. We define the advantage that A breaks PRMSM as $\text{Adv}_A = |\Pr|\mu = \mu'| - \frac{1}{2}|$. If Adv_A is negligible, we say that PRMSM is semantically secure against the chosen keyword attack.

Consideration-2 Given a probabilistic polynomial time adversary A, he asks the challenger B for the cipher-text of his queried keywords for t times. Then B randomly chooses a keyword w^*, encrypts it to $\hat{w}^*$ and sends $\hat{w}^*$ to A. A outputs its guess w' for w^*, and wins if $w' = w^*$. We define the probability that A breaks keyword secrecy as $\text{Adv}_A = \Pr|w' = w^*|$. We say that PRMSM achieves keyword secrecy if $\text{Adv}_A = \frac{1}{u-t} + \in$, where $\in$ is a negligible parameter, t denotes the number of keywords that A has known, and u denotes the size of keyword dictionary.

This work preserves privacy as well as security that means it first requires to manipulate the data to preserve data privacy. Then, it needs to lock the data using key-based encryption to provide security. In progress, it starts with the user authentication process that follows the following format.

Request counter	Last request time	Personally identifiable data	Random number	CRC

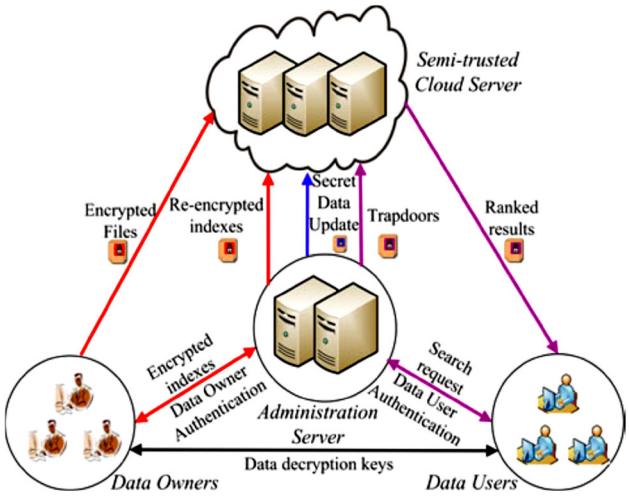

Fig. 5 Architecture of privacy-preserving keyword search in a multi-owner and multi-user cloud model

Here, the significant fact of a successful authentication is to offer both the dynamically changing secret keys and the historical data of the equivalent data user.

An example has been illustrated to get the main idea of the keywords matching protocol (the detailed protocol is elaborated in the following sections). Assume Alice wants to use the cloud to store her file F; she first encrypts her file F and gets the ciphertext C. To enable other users to perform secure searches on C, Alice extracts a keyword $w_{i,h}$ and sends the encrypted keyword $\hat{w}_{i,h} = (E_{a'}, E_o)$ to the administration server. The administration server further re-encrypts $E_{a'}$ to E_a and submits $\hat{w}_{i,h} = (E_a, E_o)$ to the cloud server. Now Bob wants to search a keyword $w_{h'}$; he first generates the trapdoor $T'_{w_{h'}}$ and submits it to the administration server. The administration server re-encrypts $T'_{w_{h'}}$ to $T_{w_{h'}} = (T_1, T_2, T_3)$, generates a secret data S_a, and submits $T_{w_{h'}}$, S_a to the cloud server. The cloud server will judge whether Bob's search request matches Alice's encrypted keyword by checking whether $\hat{e}(E_a, T_3) = \hat{e}(E_a, T_1) \cdot \hat{e}(S_a, T_2)$ holds.

The keywords encryption benefits in two ways; First, losing the key of one data owner would not lead to the disclosure of other owners' data. Second, the cloud server cannot see any relationship among encrypted keywords. This follows the proposal of Trapdoor construction formula and is constructed in a ways that the privacy of trapdoor is protected as long as the discrete logarithm problem is hard.

And it is constructed in a way that the privacy of trapdoor is protected as long as the discrete logarithm problem is hard.

Different from prior works, this approach supports authenticated data users to attain efficient, secure and convenient searches over multiple data owners' data. To

Table 5 Comparison of ranking-based evaluation approaches

Architecture design	Approach	Implementation concept	Highlights/problem domain	Advantages	Disadvantages
Zhang et al. [14]	To propose schemes to deal with privacy-preserving ranked multi-keyword search in a multi-owner model (PRMSM)	Novel search protocol to perform secure search without knowing the actual data of both keywords and trapdoors	PRMSM supports efficient data user revocation	Scheme enables authenticated data users to achieve secure, convenient, and efficient searches over multiple data owners' data	Work has some practical distractions that are supposed to be resolved in the extended version
Ren et al. [31]	A searchable encryption is presented against both data and access pattern leakage	A homomorphic exclusive-or (XOR) function is defined to enable the evaluation key to be calculated instead of storing	An effective and feasible approach performs with a query of less than 60 ms among 100,000 entries	Searching time of solution is faster by using hash-based indexing than the default clear text utility grep, which speedup searching process and protects access pattern leakage against eves-dropping	Comparison and results can be represented better
Jingwei et al. [5]	Hybrid architecture for privacy-preserving data utilization	Data utilization system is provided to achieve both exact keyword search and fine-grained access control over encrypted data	System for exact keyword search and access control over encrypted data	Scheme under this new architecture efficiently achieves fuzzy keyword search	The issue of outsourcing ABE to further relieve computational cost at user side could be better
Cengiz [32]	An efficient multi-keyword search scheme that ensures users' privacy against both external adversaries	Uses cryptographic techniques as well as query and response randomization	Both search terms in queries and returned responses are protected against privacy violations	Incorporates an effective ranking capability in the scheme that enables user to retrieve only the top-matching results	Less efficient than other similar preferred schemes

proficiently authenticate data users and detect attackers who steal the secret key and execute illegal searches, author has proposed a novel dynamic secret key generation protocol and a new data user authentication protocol. To enable the cloud server to perform secure search among multiple owners' data encrypted with different secret keys, they systematically construct a novel secure search protocol. Moreover, the work shows that the approach is computationally efficient, even for large data and keyword sets. See Table 5 for similar approaches based on Ranking.

12 Evaluation and Analysis

A detailed comparison of the presented approaches in terms of the privacy requirements is provided in Table 6. In Table 6, "×" and "√" symbols indicate whether a specific privacy-preserving check is fulfilled or not, respectively, whereas "–" represents that a particular requirement is not discussed. As can be observed that majority of the presented techniques fulfill the privacy-preserving requirements, such as integrity, accessibility and confidentiality. However, the requirements, such as data utility and data minimization, are met by only few techniques. There appears an important relationship between the data utility and data minimization and most of the presented approaches maintain unlinkability through data minimization. Another important observation is particular to the probability and multi-keyword ranking approaches. These approaches may propagate the keys to the unwanted users having attributes similar to the legitimate users. Nonetheless, the probability and multi-keyword ranking approaches have been quite effectively utilized to achieve a desired level of privacy. We also observe that most of the presented cryptographic techniques have successfully been able to minimize the key management overheads despite of their inherent complexities. For instance, the anonymization is considered as less efficient in terms of computation, whereas the multi-keyword ranking has a standing of costly privacy primitive because of bilinear computations. However, the presented schemes in this survey based on the aforementioned probability and cryptographic schemes sufficiently minimized the privacy preserving overhead. The higher data utility-based approaches can never be truly safe in public clouds because they are susceptible to information disclosure by some insiders or the other hackers.

However, higher data utility-based approaches when used only in private clouds preserve the privacy to a desired level because the infrastructure in such cases is trusted. Therefore, for systems operating in public or hybrid clouds, using reasonably strong privacy-preserving idea is highly important that can be depicted in Table 7.

Table 6 Comparison of various privacy-preserving approaches

Work	Technique	Adversary assumption	Data utility	Data minimization	Accessibility	Confidentiality	Integrity
Canard [1]	Proxy re-encryption	Trusted servers	√	×	×	√	√
Yannan Li et al. [6]	Key-based auditing	Trusted servers	×	√	√	×	×
Wang [7]	Key-based updating	Untrusted servers	√	×	√	√	×
Liang et al. [3]	Encryption using regular language	Semi-trusted servers	√	×	√	×	√
Chun-I Fan [12]	Predicate encryption	Untrusted servers	×	√	√	√	×
Zhang et al. [14]	Encryption against plaintext (IND-CPA) attacks	Semi-trusted servers	×	√	×	√	√
Yan et al. [15]	PPTE schemes	Semi-trusted servers	√	×	√	×	×
Hsu et al. [18]	Probabilistic hybrid logic	Trusted servers	×	√	×	√	×
Papadimitriou [16]	Guilty agent detection using probability	–	√	×	×	√	√
Zhang [19]	Historical probability	Semi-trusted servers	×	√	√	√	√
Nayahi et al. [21]	Cluster-based anonymization	Trusted servers	√	√	√	×	√
Sreenivasa Rao [30]	CP-ABE scheme	Untrusted servers	×	√	×	√	√
Yang et al. [27]	Anonymization-based hybrid logic	–	√	×	×	√	√
Zhang et al. [14]	Multi-keyword search	Semi-trusted servers	√	√	√	×	√
Ren et al. [31]	Homomorphic exclusive-or (XOR) function	Untrusted servers	√	√	√	×	√
Jingwei et al. [5]	Fine-grained access control	Trusted servers	√	×	√	√	√
Cengiz [32]	Multi-keyword cryptography	Semi-trusted servers	×	√	√	√	√
Cao [33]	Optimal anonymization	Trusted servers	√	√	√	√	×
Elger [34]	Reasonable anonymization	Trusted servers	√	×	√	√	√
Ardagna [35]	Exception-based access control solution	Untrusted servers	√	√	×	√	√
Ni [36]	P-RBAC	Semi-trusted servers	√	√	×	√	√
Jin [37]	Unified access control scheme	Untrusted servers	√	×	√	√	√

13 Conclusion

In this age of cloud computing and big data, privacy protection is becoming an unavoidable stumbling block in front of us. Encouraged by this, we plotted the major milestones in privacy study and kept up to date from different perspectives. It aims to pave a consistent ground for concerned readers to explore this promising and emerging field. We summarized the privacy study upshots in different research principles and communities. Specifically, we presented the mathematical efforts announced by the related privacy frameworks and models.

The methodologies are well classified into four classifications and prepare section-wise tables governing their characteristics. Furthermore, the optimum solution for a type of cloud privacy-preserving model is highlighted section-wise as well as when all taken together for better understanding and clarification to new researchers in this area. It is believed that such effort in privacy is essential and highly demanded in problem solving in the cloud

Table 7 Representing information utility and complexity on the basis of nature of various privacy access

Privacy access	Information utility	Complexity	Approach	Advantage	Disadvantage
Probability and ranking-based privacy using authenticated data structures	High	Low	Policy-oriented over low privacy strength	Reduction in communication overhead providing ease during the implementation and computations	Vulnerable to unauthorized access
Cryptographic privacy policy with defined data accessibility level	Low	High	User-centric unlinkability over strong privacy strength	Providing record-level protection prevents eves-dropper across multiple data environment	Theoretically doable uses forward index technologies
Anonymized and Linkable data privacy with specific data accessibility	Moderate	High	Linkable policy with limited information access	Addresses issues of policy anomalies mechanisms also reduces overhead on both communication and computation	Comparatively high query processing time due to rejection of extra data

environment, and it is undeniably worthwhile to invest our energy and passion to look forward to the best approach on the basis of data utility and privacy accessibility.

References

1. Canard S, Devigne J (2016) Highly privacy-protecting data sharing in a tree structure. Future Gener Comput Syst 62:119–127
2. Örencik C, Savaş E (2014) An efficient privacy-preserving multi-keyword search over encrypted cloud data with ranking. Distrib Parallel Database 32(1):119–160
3. Liang K, Huang X, Guo F, Liu JK (2016) Privacy-preserving and regular language search over encrypted cloud data. IEEE Trans Inf Forensics Secur 11(10):2365–2376
4. Yang JJ et al (2015) A hybrid solution for privacy preserving medical data sharing in the cloud environment. Future Gener Comput Syst 43–44:74–86
5. Li J et al (2014) Privacy-preserving data utilization in hybrid clouds. Future Gener Comput Syst 30(1):98–106
6. Li Y et al (2016) Privacy preserving cloud data auditing with efficient key update. Future Gener Comput Syst 78:789–798
7. Wang Y (2015) Privacy-preserving data storage in cloud using array BP-XOR codes. IEEE Trans Cloud Comput 3(4):425–436
8. Waters B et al (2007) Conjunctive, subset, and range queries on encrypted data. Theory Cryptogr 4392:535–545
9. Zheng Q et al (2014) VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In: IEEE conference computer communication (INFOCOM)
10. Cash D et al (2013) Highly-scalable searchable symmetric encryption with support for Boolean queries. In: Advances in cryptology—CRYPTO, Berlin, Germany
11. Komishani EG et al (2016) PPTD: preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. Knowl Based Syst 94:43–59
12. Chun-I Fan S-YH (2013) Controllable privacy preserving search based on symmetric predicate encryption in cloud storage. Future Gener Comput Syst 29:1716–1724
13. Shen E et al (2009) Predicate privacy in encryption systems. In: The 6th theory of cryptography conference, TCC, Verlag
14. Zhang W et al (2016) Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing. IEEE Trans Comput 65(5):1566–1578
15. Yan Z et al (2016) Two schemes of privacy-preserving trust evaluation. Future Gener Comput Syst 62:175–189
16. Garcia-Molina H et al (2011) Data leakage detection. IEEE Trans Knowl Data Eng 23(1):51–63
17. Garcia-Molina H, Papadimitriou P (2010) Data leakage detection. IEEE Trans Knowl Data Eng 51–63. https://doi.org/10.1109/TKDE.2010.100
18. Hsu T-S, Liau C-J, Wang D-W (2012) Logic, probability, and privacy: a framework. In: Turing-100, pp 157–167
19. Zhang G et al (2012) A historical probability based noise generation strategy for privacy protection in cloud computing. J Comput Syst Sci 78:1374–1381
20. Jesu Vedha Nayahi J, Kavitha V (2016) Privacy and utility preserving data clustering for data anonymization and distribution on Hadoop. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2016.10.022
21. Jesu Vedha Nayahi J, Kavitha V (2015) An efficient clustering for anonymizing data and protecting sensitive labels. Int J Uncert Fuzziness Knowl Based Syst 23:685–714
22. Li N et al (2007) T-closeness: privacy beyond k-anonymity and l-diversity. In: IEEE international conference on data engineering
23. Rebollo-Monedero D et al (2010) From t-closeness-like privacy to postrandomization via information theory. IEEE Trans Knowl Data Eng 22:1623–1636
24. Amiri F et al (2015) Hierarchical anonymization algorithms against background knowledge attack in data releasing. Knowl-Based Syst 101:71–89
25. Kohlmayer F et al (2014) A flexible approach to distributed data anonymization. J Biomed Inform 50:62–76
26. Zhang X et al (2015) Proximity-aware local recoding anonymization with mapreduce for scalable big data privacy preservation in cloud. IEEE Trans Comput 64(8):2293–2307
27. Wen-Yang L et al (2015) Privacy preserving data anonymization of spontaneous ADE reporting system dataset. BMC Med Inform Decis Mak 16:58
28. Goryczka S et al. (2014) m-Privacy for collaborative data publishing. IEEE Trans Knowl Data Eng 26(10)

29. Soria-Comas J et al (2015) t-Closeness through microaggregation: strict privacy with enhanced utility preservation. IEEE Trans Knowl Data Eng 27(11):3098–3110
30. Rao YS (2017) A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing. Future Gener Comput Syst 67:133–151
31. Rena SQ et al (2016) Secure searching on cloud storage enhanced by homographic indexing. Future Gener Comput Syst 65:102–110
32. Örencik C et al (2014) An efficient privacy-preserving multi-keyword search over encrypted cloud data with ranking. Distrib Parallel Datab 32:119–160
33. Cao N et al (2011) Privacy-preserving multi-keyword ranked search over encrypted cloud data. In: INFOCOM
34. Elger BS et al (2010) Strategies for health data exchange for secondary, cross-institutional clinical research. Comput Methods Prog Biomed 99:1–21
35. Ardagna CA et al (2010) Access control for smarter healthcare using policy spaces. J Comput Secur 29(8):848–858
36. Ni Q et al (2009) Privacy-aware role-based access control. IEEE Secur Privacy 7(4):35–43
37. Jin J et al (2011) Patient-centric authorization framework for electronic healthcare services. Comput Secur 30(2–3):116–127