Secure and Efficient Data Retrieval Process based on Hilbert Space Filling Curve

N.S. Jeya karthikka
PG Scholar
Sri Ramakrishna Engg Collg
Coimbatore

S.Bhaggiaraj
Assistant Professor
Sri Ramakrishna Engg Collg
Coimbatore

V.Sumathy
Associate Professor
Govt Collg of Technology
Coimbatore

ABSTRACT

Cloud computing economically enables the paradigm of data service outsourcing. However, to protect the secret data, sensitive information in cloud first the data has to be encrypted before outsourced to the commercial public cloud. It is a very challenging task. Although traditional attribute based encryption technique allow users to securely search the data by using keywords, they support only Boolean search and are not yet sufficient to meet the effective data utilization need that is inherently demanded by large number of users and huge amount of data files in cloud. In this paper, the above mentioned problem can be solved by using Octree partition and the Hilbert space-filling curve.In this technique searching the keyword in search index is very simple. This scheme is suited for cloud storage systems with massive amount of data. This system improves the query results and privacy for search the data in cloud.

Keywords

cloud storage, octree partition, Hilbert space filling curve, search index

1. INTRODUCTION

As Cloud Computing becomes popular, more and more secret information are being centralized into the cloud, such as phone numbers, individual property records, company quotation data, and banking details etc. The data owner and cloud server are no longer in the same trusted domain may put the unencrypted data at risk the cloud server may leak secret information to unauthorized parties or even be hacked. The sensitive data has to be encrypted before outsourcing mainly for data privacy. However, data encryption makes effective data utilization .It is a very challenging task when large amount of data files can be outsourced into the cloud. In Cloud Computing, data owners may share their outsourced data with a large number of users, the receiver only retrieve particular data files they are interested in during a given session. One of the most standard ways to do so is through keyword-based search [1]. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios. Unfortunately, data encryption, which restricts user's ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data. Although traditional searchable encryption schemes, allow a user to securely search over encrypted data through keywords without first decrypting it, these techniques support only conventional Boolean keyword search without capturing any relevance of the files in the search result.

1. Our searchable encryption scheme provides rich expressiveness of index terms by exploiting ABE, thus providing data security.

2. Our scheme is more suitable for a cloud storage service which operates in one-upload-many-download manner due to its use of ABE, which naturally provides secure one-to-many communications.

1.1 Hilbert Space-Filling Curve

Space-filling curves (Sagan, 1994) map points in N-dimensional space into a 1-D linear order. The curve visits each point in space only one time in a certain order – usually points that are close on the curve are close in space.Space-filling curves preserve spatial proximity at local level to some extent; the closer two object in space, the higher possibility that they are close together in the linear order defined by space-filling curves [4]. A space-filling curve can be used with a space partition method. A high-dimensional space can be divided into different grid cells, which can be in turn further divided into smaller cells until the cell size or the number of interest objects in the cell is small enough. The level of such partition depends on the smallest cell size and the number of grid nodes in space that space-filling curve can pass through. Each cell is labelled by the unique number (called code) that defines cell's position in the order of space-filling curve. Both mathematical analysis and practical applications suggest that Hilbert space filling curve has best clustering ability and performance in data retrieval and response time among all kinds of space-filling curves.

2. RELATED WORK

2.1. Attribute-Based Encryption (ABE)

Attribute-based encryption[8] was firstly introduced by Sahai and Waters, in which they suggested that one's identity can be viewed as a combination of several attributes expressing the characteristics of the user in the form of access policy by using Boolean expressions such as AND, OR, or NOT. Later studies are broadly categorized into key-policy attribute-based encryption (KP-ABE) and cipher text-policy attribute-based encryption (CP-ABE) [5] studies. In KP-ABE, the access policy is associated with keys corresponding to attributes implying that an encryptor is not authorized to grant access to the encrypted content except of descriptive attributes for the data by the encryptor's choice. On the other hand, CP-ABE is complementary to KP-ABE [6] by enabling encryptor to specify access policy combined with the ciphertext. Both schemes allow secure one-to-many communications such as targeted broadcasts for a specific group and individual user according to their attributes, which are distinct from the traditional cryptographic approaches requiring the explicit identity of the intended receivers.

2.2. Searchable encryption

The original goal of searchable encryption is to provide privacy-preserving keyword searches of encrypted data against an intermediate gateway, which involves a message exchange process between the sender and the receiver. The first searchable encryption scheme was the Public-key Encryption with Keyword Search (PEKS) scheme based on Identity-Based Encryption (IBE),[3] originally proposed by Boneh et al.. Since PEKS is devised to forward the encrypted contents to a designated receiver with its unique identity.To provide better expressiveness, other searchable encryption schemes based on ABE are introduced. One of representative works is Hidden Vector Encryption (HVE). HVE is more advanced work compared to PEKS, as it provides conjunctive and range queries. Despite their advantages, the previous PEKS and HVE schemes are not suitable for one-upload-many-download cloud storage systems because they mainly focus on one-time message-delivery scenarios. Our goal is to construct an efficient searchable encryption scheme based on ABE to support a one-upload-many-download policy[7].

3. ARCHITECTURE DESCRIPTION AND REQUIREMENTS

3.1. System description

In this section, we describe the architecture of the cloud storage system. As depicted in Fig. 1, the system consists of the following four entities:

(1) Trusted authority: This is the key generation centre, which is fully trusted by all other participants of the system. It generates public parameters and the master secret key. It also generates user-specific private keys to the set of attributes for data access, cipher text decryption for receivers, and anonymous keys for data owners.

(2) Cloud service provider: This is an entity that provides data storage and retrieval service to subscribing users. It stores the data content outsourced by the data owner. This content is searchable and downloadable to intended receivers who have sufficient credentials.

(3) Data owner: This is the cloud storage subscriber who wants to upload its data content anonymously to the cloud storage system after encryption. The encrypted content can be shared with intended receivers who have sufficient credentials as specified by the data owner.

(4) Retriever: This is another cloud storage subscriber which queries the CSP for encrypted data in the cloud storage system by using a pseudonym of the data owner. Only retrievers who have legal rights satisfying the access policy specified by the data owner can access the encrypted content and restore the original message from it.

3.2. System initialization and Key generation

The Setup algorithm, performed by a trusted authority, will choose a bilinear group G of prime order p with generator g. Next it chooses two random exponents $\alpha, \beta \in \mathbb{Z}_p$, and a cryptographic hash function H : $\{0,1\}^* \rightarrow \mathbb{G}$, which will be modelled as a random oracle. Then public parameter PK and master secret MK are computed as:

$$PK = (\mathbb{G}, g, h = g^{\beta}, \omega = e(g,g)^{\alpha}), MK = (\beta, g^{\alpha})$$

PK can be accessed by all participants in the system, but master secret MK is kept secret.

In this phase, the trusted authority generates anonymous key A_o for data owner and private key SK for receiver. For the data owner with ID_o, the trusted authority runs anonymous key generation algorithm, $KeyGen_o$, and returns the result, $A_0 = H(ID_0)^{\beta}$ to the data owner. For a receiver $u_i \in U$ with identity ID_i, the trusted authority runs KeyGeni algorithm by choosing a random $r \in Z_p$ for individual user and $r_j \in Z_p$ for each attribute $\lambda_j \in \Lambda_i$ where Λ_i denotes the attribute set belonging to user u_i. Then the private key SK is computed as:

$$SK = (D = g^{\frac{\alpha+r}{\beta}}, \{D_j = g^r H(\lambda_j)^{r_j}, D_j' = g^{r_j}, D_j'' = H(\lambda_j)^{\beta}\}_{\lambda_j \in \Lambda_i})$$

3.3 Data outsourcing to cloud

Prior to outsourcing of data content which a data owner with its identity as ID_o holds, the data owner generates its pseudonym by running PseudoGen (PK, ID_o). In PseudoGen algorithm, the data owner chooses a random $t \in Zp$ and generates its pseudonym $P_o = H(ID_o)$ t by itself and publicizes this pseudonym. Both of A_o and P_o are used to make an agreed session key with intended receivers, which will be used to scramble attributes exposed in an access structure in non-interaction manner. Then the data owner encrypts data M by running Encrypt algorithm, which will take as input the public parameter PK, its pseudonym P_o, a message M to be encrypted under the access tree T, and output the cipher text CT_0. After then, attribute scrambling procedure, AttrScm, is applied to the cipher text CT_0 generating new cipher text CT to be located in the cloud storage.

3.4 Data user Access outsourced data

Data query (Query) At the initial round, a retriever can first acquire a pseudonym list of data owners from the CSP. The CSP can extract and update the list of pseudonyms corresponding data owners according to the stored encrypted contents, it would deliver the proper list to the receiver. Once the retriever determines to retrieve a data of some data owner with $C'' = P_o$ in the cloud storage and wants to access it, it can generate cryptographic index terms for corresponding attributes as the agreed session key.

Data retrieval (Retrieve) On request of an access to encrypted contents stored in the cloud storage with scrambled index terms, the CSP determines whether the requested item is stored in the storage and which one is satisfied with the requested index terms. In this phase, the CSP can easily retrieve a list of cipher texts where in T from the cloud storage using a common DBMS mechanism.

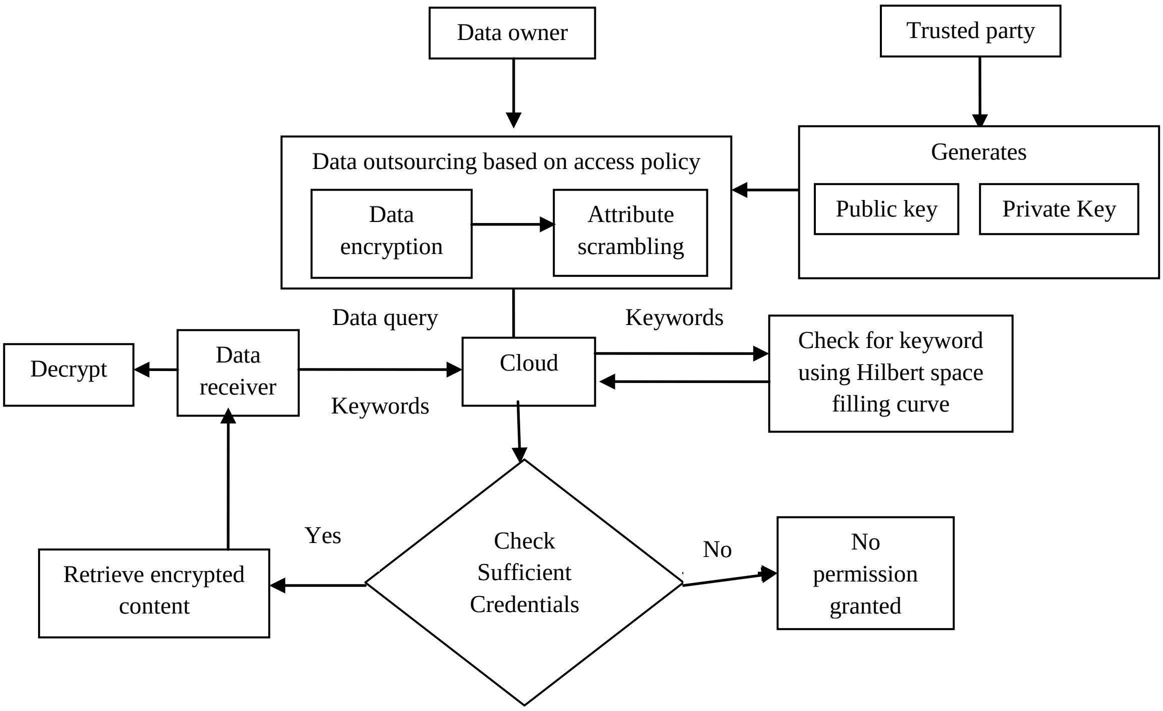


Fig:1 Architecture diagram

Decrypt: Upon receiving the requested content in encrypted form, the receiver obtains the plaintext by using decryption algorithm Decrypt node which can be described as a recursive algorithm. For ease of exposition, we present the simplest form of the decryption algorithm.

3.5. Space filling curve base query search

A space-filling curve can be used with a space partition method. A high-dimensional space can be divided into different grid cells, which can be in turn further divided into smaller cells until the cell size or the number of interest objects in the cell is small enough. The level of such partition depends on the smallest cell size and the number of grid nodes in space that space-filling curve can pass through. Each cell is labeled by the unique number (called code) that defines cell's position in the order of space-filling curve. The way of labeling determines the order in which the cells are stored in 1-D media.

3.6 Generation of Hilbert curves

Due to the self-similarity property of the Hilbert curve s, higher level Hilbert curve can be generated recursively from the lower level curve. The generation can be done with a set of recursive rules.

The 3-D Hilbert curve maps 3-D points into a 1-D linear order. The sequential number (position) of a point on the curve is called Hilbert code. Each Hilbert code has the corresponding enclosing cell in 3-D space. Data points are ordered according to the sequence in which the curve visits the cells that enclose the data points. Each cell can be assigned to base-4 digit ([0-3]) in 2-D and base-8 digit ([0~7]) in 3-D to represent its position relative to the parent (next lower level) cell. The 3-D Hilbert code is a string of base-8 digits, the length of string equals to the coding level.

In the encoding step, all the data points are encoded into the initial encoding level. It should be noted that it is not necessary to encode each index point by exactly following the procedure.

3.6.1 Encoding procedure

1. Given a point (x,y,z) and the 0-level cube;

2. Find the closest vertex *i* among all 8 vertices of the cube;

3. Set *i* as Hilbert code for this level;

4. Apply *i*th permutation rule to get 8 vertices of the sub-cell which contains the point;

5. Repeat step 2-4 until desired coding level;

6. Return Hilbert code string.

3.6.2 Decoding procedure

1. Get the 0-level cube;

2. *i* = 1;

3. Pick up the *i*th digit from Hilbert code string;

4. Apply *i*th permutation rule to get the vertices of the sub-cell;

5. *i* = *i* + 1;

6. Repeat step 3-5 until *i* > length (code);

7. Return the vertices of current cell.

3.6.3 Indexing procedure

1. Choose initial encoding level;

2. Encode data points with initial level;

3. Count number of data points with the same Hilbert code;

4. Generate the list of Hilbert code for refining process;

5. Generate the list of Hilbert code for combining process;

6. Run refining procedure until list from step 4 is empty;

7. Run combining process until list from step 5 is empty;

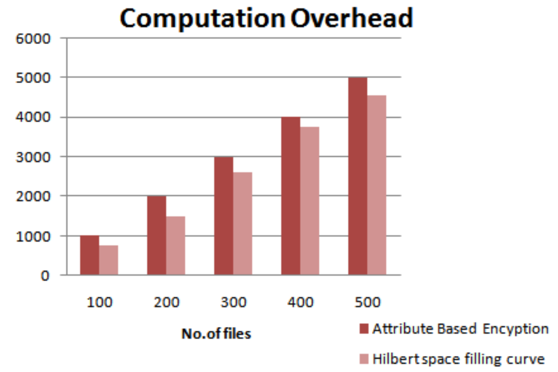
8. Group data by Hilbert code;

9. Insert grouped data points into server.

3.6.4 Querying

Among different types of spatial queries, this paper implements the most frequently used window (range) query. It is the process to find all the data points inside a given 3-D cube. We first encode the 8 vertices of the query region until all of them have different Hilbert codes. This means the query region has been divided into 8 cells. Only data cells inside these 8 cells need to be considered. The database is then scanned to retrieve the subset of data cells. If the cell from the subset is inside the query window, all the data points in the cell are selected; if the cell overlaps with the query window, each data point in the cell is scanned to decide if it is inside the query window. Since the data points in each cell are no more than the target size, it is not a time consuming process to perform this scanning process.

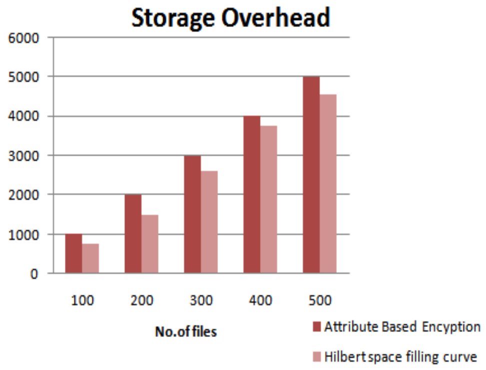
4. RESULT AND ANALYSIS



This graph compares the performance analysis between the attribute based encryption and Hilbert space filling curve. X axis represents the number of file size. Y axis represents the computation overhead. Hsfc reduces the computation overhead over the ABE. In existing system when the number of files increased computation overhead is increased compared to the proposed system.

Table 1 :Comparison

Computation overhead		
No .of files	Attribute based encryption	Hilbert space filling curve
100	1000	750
200	2000	1500
300	3000	2600
400	4000	3750
500	5000	4550



This graph compares the performance analysis between the attribute based encryption and Hilbert space filling curve. X axis represents the number of file size. Y axis represents the storage overhead. Hsfc reduces the storage overhead over the ABE. In existing system when the number of files increased storage overhead is increased compared to the proposed system.

Table 2: Comparison

Storage overhead		
No .of files	Attribute based encryption	Hilbert space filling curve
100	1000	750
200	2000	1500
300	3000	2600
400	4000	3750
500	5000	4550

5. CONCLUSION

Traditional attribute based encryption techniques allow users to securely search over encrypted data through keywords, they support only Boolean search and are not yet sufficient to meet the effective data utilization need that is inherently demanded by large number of users and huge amount of data files in cloud. In this paper, a new searchable cryptosystem is proposed by exploiting Hilbert Space filling curve with proper obfuscation of attributes. It provides enhanced quality of the retrieval service via simple comparisons for data retrieval. Comparatively, complicated retrieval operations for a specific encrypted data in existing approaches are not appropriate for an environment where numerous receivers request frequently for huge amount of data. The proposed scheme provides richer expressiveness of access policy than previous studies. It means that our scheme describes more fine-grained access control than previous ones and might reduce the number of searching index terms by using Boolean operators rather than simple concatenation. This implies our cryptosystem is more suitable to cloud storage services.

6. REFERENCE

[1] Philippe Golle, Jessica Staddon, Brent Waters. " Secure Conjunctive Keyword Search Over Encrypted Data" . Applied Cryptography and Network, Springer- 2004.

[2] Boneh.D and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. of TCC, 2007, pp. 535554.

[3] Sahai A, Waters B. Fuzzy identity-based encryption. In: Cramer R, editor. Advances in cryptology, EUROCRYPT 2005, vol. 3494. Berlin/ Heidelberg: Springer; 2005. p. 557–57.

[4] Jin, G. and J. Mellor-Crummey (2005). SFCGen: An Framework for Efficient Generation of Multi-dimensional Space-filling Curves. ACM Transactions on Mathematical Software 31(1): 120-148.

[5] Bethencourt J, Sahai A, Waters B. Ciphertext- policy attribute-based encryption. In: IEEE symposium on security and privacy; 2007. p. 321–34.

[6] Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search. In: Cachin C, Camenisch J, editors. Advances in Cryptology, EUROCRYPT 2004, vol. 3027. Berlin/Heidelberg: Springer; 2004. p. 506–22.

[7] Frikken K, Atallah M, Li J. Attribute-based access control with hidden policies and hidden credentials. IEEE Trans Comput 2006;55:1259–70.

[8] Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin S, Gennaro R, Keromytis A, Yung M, editors. Applied cryptography and network security, vol. 5037. Berlin/Heidelberg: Springer; 2008. p. 111–29.