International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

ACCESS CONTROL BY SIGNATURE-KEYS TO PROVIDE PRIVACY FOR CLOUD AND BIG DATA

AISHWARYA S[1], ELAKKIYA M[2],NARMADHA K [2],PREETHI S[2],SHALINI A[3]

1UG Scholar, Department of CSE, Kingston Engineering College, Vellore-59
2UG Scholar, Department of CSE, Kingston Engineering College, Vellore-59
3Asst.Professor, Department of CSE, Kingston Engineering College, Vellore-59

Abstract-Privacy of data in subjects of cloud computing or big data is one of the most principal issues. The privacy methods studied in previous research showed that privacy infringement for cloud computing or big data happened because multi risks on data by external or internal attackers. An important risk to take into consideration when speaking of the privacy of the stored transactions is represented by the transactions' information which is not in the owner's control. Such a case is represented by the cloud servers that are administered by cloud providers which cannot be wholly trusted by the users with sensitive, private data such as business plans or private information. A simple method for protecting data privacy is by applying certain privacy techniques onto transactions' data, followed by the upload of the modified data into the cloud. In this paper, we are proposing a case study that is built on levels containing three models: cloud's architecture, transaction's manager and clients. Moreover, we consider that our case study is based on the premise of zero trust among the three models, therefore all the transactions take place with third-parties and the data movements are realized going through various levels of security.

Key Words: cloud computing,privacy

1.INTRODUCTION

Coud storage is an emerging model of storage to provide scalable, elastic and pay-as-you-use service to cloud computing users. For individual usage, the subscribers enjoy the freedom to access to their data anywhere, anytime with any device. When cloud storage is utilized by a group of users, it allows team members to synchronize and manage all shared documents. Moreover, it also saves the user a lot of capital investment of expensive storage equipments. Cloud delivers convenience to the customers and at the same time arouses many security and privacy problems . Since the data are physically stored on the multiple servers of the cloud service provider, the customers cannot fully in charge of their data. They worry about the privacy of the stored documents since the server may be intruded by hacker or the data could be misused by the internal staff for commercial purpose . The customers prefer to adopt the encryption technology to protect the data confidentiality, which meanwhile arouses another problem: how to execute data retrieval on the large volume of ciphertext. It is almost Y. Yang is with College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China; Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, China; Key Laboratory of Spatial Data Mining & Information Sharing, Ministry of Education, Fuzhou, China; University Key Laboratory of Information Security of Network Systems (Fuzhou University), Fujian Province, China; Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou, China. No customer could tolerate the huge transmission overhead and the waiting time for the data retrieval result. Searchable encryption technology not only exerts encryption protection of the data, but also supports efficient search function without undermining the data privacy. The data user generates a token of the content that he wants to search using his private key. Receiving the token, the cloud server searches on the encrypted data without decrypting the ciphertext. The most important point is that the server learns nothing about the plaintext of the encrypted data nor the searched content during the data retrieval procedure. However, most of the available searchable encryption schemes only support some basic search patterns, such as single keyword search, conjunctive keyword search and boolean search. Since the cloud computing is a fierce competition industry, it is of vital importance to provide good user experience. It is urgent to design novel searchable encryption schemes with expressive search pattern for cloud storage.

2. LITERATURE SURVEY

2.1 TITLE:Two-Factor Data Security Protection Mechanism for Cloud Storage System.
AUTHOR: Joseph K. Liu, Kaitai Liang, Willy Susilo;
YEAR : 2016
DESCRIPTION: In this paper, we propose a two-factor data security protection mechanism with factor revocability for cloud storage system. Our system allows a sender to send an encrypted message to a receiver

through a cloud storage server. The sender only needs to know the identity of the receiver but no other information (such as its public key or its certificate). The receiver needs to possess two things in order to decrypt the ciphertext. The first thing is his/her secret key stored in the computer. The second thing is a unique personal security device which connects to the computer. It is impossible to decrypt the ciphertext without either piece. More importantly, once the security device is stolen or lost, this device is revoked. It cannot be used to decrypt any ciphertext. This can be done by the cloud server which will immediately execute some algorithms to change the existing ciphertext to be un-decryptable by this device. This process is completely transparent to the sender. Furthermore, the cloud server cannot decrypt any ciphertext at any time. The security and efficiency analysis show that our system is not only secure but also practical.

2.2 TITLE :PPHOPCM: Privacy-preserving High-order Possibilistic c-Means Algorithm for Big Data Clustering with Cloud Computing.

AUTHOR:Qingchen Zhang, Laurence T. Yang, Zhikui Chen, and Peng Li.

YEAR :2017.

DESCRIPTION: As one important technique of fuzzy clustering in data mining and pattern recognition, the possibilistic c-means algorithm (PCM) has been widely used in image analysis and knowledge discovery. However, it is difficult for PCM to produce a good result for clustering big data, especially for heterogenous data, since it is initially designed for only small structured dataset. To tackle this problem, the paper proposes a high-order PCM algorithm (HOPCM) for big data clustering by optimizing the objective function in the tensor space. Further, we design a distributed HOPCM method based on MapReduce for very large amounts of heterogeneous data. Finally, we devise a privacy-preserving HOPCM algorithm (PPHOPCM) to protect the private data on cloud by applying the BGV encryption scheme to HOPCM, In PPHOPCM, the functions for updating the membership matrix and clustering centers are approximated as polynomial functions to support the secure computing of the BGV scheme. Experimental results indicate that PPHOPCM can effectively cluster a large number of heterogeneous data using cloud computing without disclosure of private data..

2.3 TITLE:VABKS: Verifiable Attribute-based Keyword Search over Outsourced Encrypted Data.

AUTHOR:QingjiZheng ShouhuaiXu Giuseppe Ateniese .**YEAR :**2014.

DESCRIPTION: It is common nowadays for data owners to outsource their data to the cloud. Since the cloud cannot be fully trusted, the outsourced data should be encrypted. This however brings a range of problems, such as: How should a data owner grant search capabilities to the data users? How can the authorized data users search over a data owner's outsourced encrypted data? How can the data users be assured that the cloud faithfully executed the search operations on their behalf? Motivated by these questions, we propose a novel cryptographic solution, called verifiable attribute-based keyword search (VABKS). The solution allows a data user, whose credentials satisfy a data owner's access control policy, to (i) search over the data owner's outsourced encrypted data, (ii) outsource the tedious search operations to the cloud, and (iii) verify whether the cloud has faithfully executed the search operations. We formally define the security requirements of VABKS and describe a construction that satisfies them. Performance evaluation shows that the proposed schemes are practical and deployable.

2.4TITLE:Privacy-Preserving and Regular Language Search over Encrypted Cloud Data.

AUTHOR: Kaitai Liang, Xinyi Huang, FuchunGuo.

YEAR : 2016.

DESCRIPTION: Using cloud-based storage service, users can remotely store their data to clouds but also enjoy the high quality data retrieval services, without the tedious and cumbersome local data storage and maintenance. However, the sole storage service cannot satisfy all desirable requirements of users. Over the last decade, privacy-preserving search over encrypted cloud data has been a meaningful and practical research topic for outsourced data security. The fact of remote cloud storage service that users cannot have full physical possession of their data makes the privacy data search a formidable mission. A naive solution is to delegate a trusted party to access the stored data and fulfill a search task. This, nevertheless, does not scale well in practice as the fully data access may easily yield harm for user privacy. To securely introduce an effective solution, we should guarantee the privacy of search contents, i.e. what a user wants to search, and return results, i.e. what a server returns to the user. Furthermore, we also need to guarantee privacy for the outsourced data, and bring no additional local search burden to user. In this paper, we design a novel privacy-preserving functional encryption based search mechanism over encrypted cloud data. A major advantage of our new primitive compared to the existing public key based search systems is that it supports an extreme expressive search mode, regular language search. Our security and performance analysis show that the proposed system is provably secure and more efficient than some searchable systems with high expressiveness.

2.5 TITLE:Towards achieving Data Security with the Cloud Computing Adoption Framework.

AUTHOR:Victor Chang, MuthuRamachandran.

YEAR : 2015.

DESCRIPTION: Offering real-time data security for petabytes of data is important for Cloud Computing. A recent survey on cloud security states that the security of users' data has the highest priority as well as concern.

We believe this can only be able to achieve with an approach that is systematic, adoptable and well-structured. Therefore, this paper has developed a framework known as Cloud Computing Adoption Framework (CCAF) which has been customized for securing cloud data. This paper explains the overview, rationale and components in the CCAF to protect data security. CCAF is illustrated by the system design based on the requirements and the implementation demonstrated by the CCAF multi-layered security. Since our Data Center has 10 petabytes of data, there is a huge task to provide real-time protection and quarantine. We use Business Process Modeling Notation (BPMN) to simulate how data is in use. The use of BPMN simulation allows us to evaluate the chosen security performances before actual implementation. Results show that the time to take control of security breach can take between 50 and 125 hours. This means that additional security is required to ensure all data is well-protected in the crucial 125 hours. This paper has also demonstrated that CCAF multi-layered security can protect data in real-time and it has three layers of security: 1) firewall and access control; 2) identity management and intrusion prevention and 3) convergent encryption. To validate CCAF, this paper has undertaken two sets of ethical-hacking experiments involved with penetration testing with 10,000 trojans and viruses. The CCAF multi-layered security can block 9,919 viruses and trojans which can be destroyed in seconds and the remaining ones can be quarantined or isolated. The experiments show although the percentage of blocking can decrease for continuous injection of viruses and trojans, 97.43% of them can be quarantined. Our CCAF multi-layered security has an average of 20% better performance than the single-layered approach which could only block 7,438 viruses and trojans. CCAF can be more effective when combined with BPMN simulation to evaluate security process and penetrating testing results.

2.6 TITLE :Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage.
AUTHOR:G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner.
YEAR : 2014.
DESCRIPTION:Remote data integrity checking is of crucial importance in cloud storage. It can make the clients verify whether their outsourced data is kept intact without downloading the whole data. In some application scenarios, the clients have to store their data on multi-cloud servers. At the same time, the integrity checking protocol must be efficient in order to save the verifier's cost. From the two points, we propose a novel remote data integrity checking model: ID-DPDP (identity-based distributed provable data possession) in multi-cloud storage. The formal system model and security model are given. Based on the bilinear pairings, a concrete ID-DPDP protocol is designed. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (computational Diffie-Hellman) problem. In addition to the structural advantage of elimination of certificate management, our ID-DPDP protocol is also efficient and flexible. Based on the client's authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.
2.7 TITLE :OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices.
AUTHOR: Jin Li, Xiao Tan, Xiaofeng Chen, Duncan S. Wong.
YEAR : 2014.
DESCRIPTION: Cloud Computing moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. In this work, we study the problem of ensuring the integrity of data storage in Cloud Computing. To reduce the computational cost at user side during the integrity verification of their data, the notion of public verifiability has been proposed. However, the challenge is that the computational burden is too huge for the users with resource-constrained devices to compute the public authentication tags of file blocks. To tackle the challenge, we propose OPoR, a new cloud storage scheme involving a cloud storage server and a cloud audit server, where the latter is assumed to be semi-honest. In particular, we consider the task of allowing the cloud audit server, on behalf of the cloud users, to pre-process the data before uploading to the cloud storage server and later verifying the data integrity. OPoR outsources the heavy computation of the tag generation to the cloud audit server and eliminates the involvement of user in the auditing and in the preprocessing phases. Furthermore, we strengthen the Proof of Retrievabiliy (PoR) model to support dynamic data operations, as well as ensure security against reset attacks launched by the cloud storage server in the upload phaseCloud Computing moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. In this work, we study the problem of ensuring the integrity of data storage in Cloud Computing. To reduce the computational cost at user side during the integrity verification of their data, the notion of public verifiability has been proposed. However, the challenge is that the computational burden is too huge for the users with resource-constrained devices to compute the public authentication tags of file blocks. To tackle the challenge, we propose OPoR, a new cloud storage scheme involving a cloud storage server and a cloud audit server, where the latter is assumed to be semi-honest. In particular, we consider the task of allowing the cloud audit server, on behalf of the cloud users, to pre-process the data before uploading

to the cloud storage server and later verifying the data integrity. OPoR outsources the heavy computation of the tag generation to the cloud audit server and eliminates the involvement of user in the auditing and in the preprocessing phases. Furthermore, we strengthen the Proof of Retrievabiliy (PoR) model to support dynamic data operations, as well as ensure security against reset attacks launched by the cloud storage server in the upload phase.

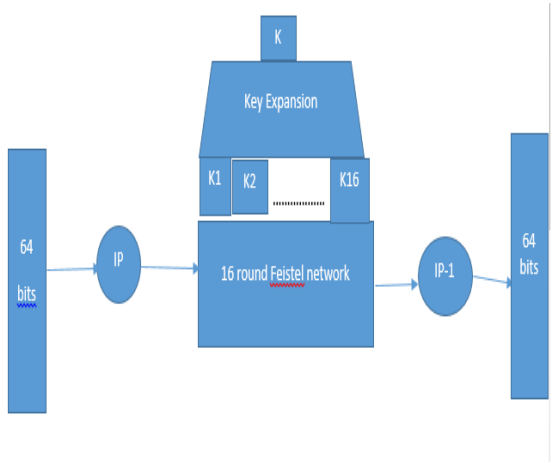
3.PROPOSED SYSTEM

In this paper, a vector space model is utilized and each report is addressed by a vector, which implies each archive can be viewed as a point in a high dimensional space. Because of the connection between various reports, all the records can be isolated into a few classifications.

Instead of utilizing the conventional arrangement search technique, a backtracking calculation is created to look through the objective archives. Cloud worker will initially look through the classifications and get the base wanted sub-class. At that point the cloud worker will choose the ideal k reports from the base wanted sub-class. The estimation of k is recently settled by the client and shipped off the cloud worker. On the off chance that current sub-classification can not fulfill the k records, cloud worker will follow back to its parent and select the ideal reports from its sibling classifications. This cycle will be executed recursively until the ideal k records are fulfilled or the root is reached.

To confirm the honesty of the query item, an evident design dependent on hash work is built.

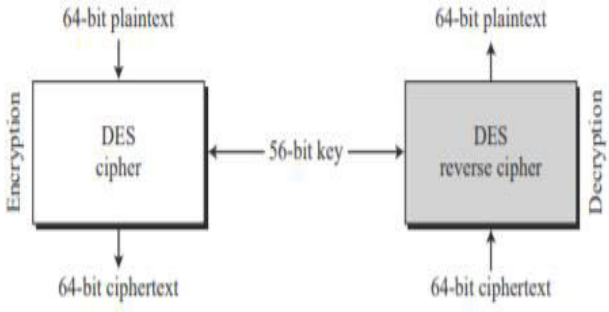
3.1 ALGORITHM USED
DATA ENCRYPTION SCHEME



- DES has an initial permutation and final permutation after 16 rounds
- thesepermautations are inverse of each other and operate on 64 bits.
- They have no cryptographic significance.
- The designers did not disclose their purpose
- There are several properties
- We highlight some:

-The rows are permutations
-The outputs are a non-linear combination of the inputs
-Change one bit of the input,and half of the output bits change(Avanlanche Effect)
-Each output bit is dependent on all the input bits

Let us concentrate on encryption; later we will discuss decryption. The encryption process is made of two permutations (P-boxes), which we call initial and fi nal permutations, and sixteen Feistel rounds. Each round uses a different 48-bit round key generated from the cipher key according to a predefined algorithm described later in the chapter. Figure 6.2 shows the elements of DES cipher at the encryption site.



Each of these permutations takes a 64-bit input and permutes them according to a predefined rule. We have shown only a few input ports and the corresponding output ports. These permutations are keyless straight permutations that are the inverse of each other. For example, in the initial permutation, the 58th bit in the input becomes the fi rst bit in the output. Similarly, in the fi nal permutation, the fi rst bit in the input becomes the 58th bit in the output. In other words, if the rounds between these two permutations do not exist, the 58th bit entering the initial permutation is the same as the 58th bit leaving the fi nalpermutation

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
    {
        mixer (leftBlock, rightBlock, RoundKeys[round])
        if (round!=16) swapper (leftBlock, rightBlock)
    }
    combine (32, 64, leftBlock, rightBlock, outBlock)
    permute (64, 64, outBlock, cipherBlock, FinalPermutationTable)
}
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
    copy (32, rightBlock, T1)
    function (T1, RoundKey, T2)
    exclusiveOr (32, leftBlock, T2, T3)
    copy (32, T3, rightBlock)
}
swapper (leftBlock[32], rigthBlock[32])
{
    copy (32, leftBlock, T)
    copy (32, rightBlock, leftBlock)
    copy (32, T, rightBlock)
}
function (inBlock[32], RoundKey[48], outBlock[32])
{
    permute (32, 48, inBlock, T1, ExpansionPermutationTable)
    exclusiveOr (48, T1, RoundKey, T2)
    substitute (T2, T3, SubstituteTables)
    permute (32, 32, T3, outBlock, StraightPermutationTable)
}
substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
    for (i = 1 to 8)
    {
      row ← 2 × inBlock[i × 6 + 1] + inBlock [i × 6 + 6]
      col ← 8 × inBlock[i × 6 + 2] + 4 × inBlock[i × 6 + 3] +
            2 × inBlock[i × 6 + 4] + inBlock[i × 6 + 5]
```

Key Generation The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process

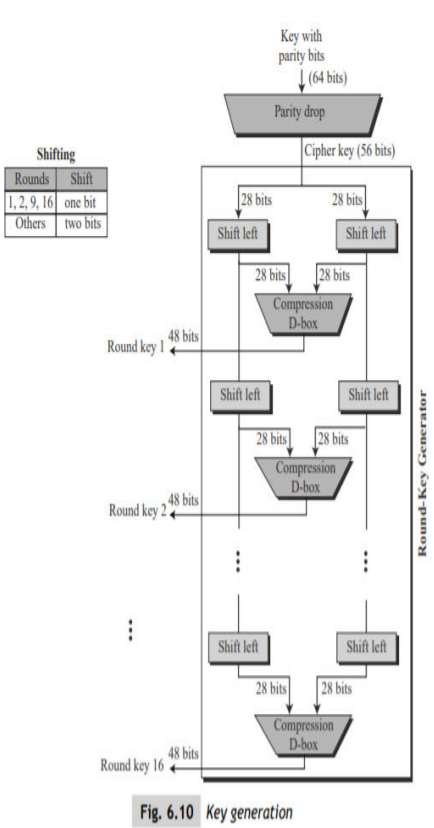


Fig. 6.10 Key generation

D-Boxes Between two rows of S-boxes (in two subsequent rounds), there are one straight D-box (32 to 32) and one expansion D-box (32 to 48). These two D-boxes together provide diffusion of bits. We have discussed the general design principle of D-boxes in Chapter 5. Here we discuss only the ones applied to the D-boxes used inside the DES function. The following criteria were implemented in the design of D-boxes to achieve this goal:

1. Each S-box input comes from the output of a different S-box (in the previous round).
2. No input to a given S-box comes from the output from the same box (in the previous round).
3. The four outputs from each S-box go to six different S-boxes (in the next round).
4. No two output bits from an S-box go to the same S-box (in the next round).
5. If we number the eight S-boxes, S1, S2, …, S8, a. An output of Sj−2 goes to one of the fi rst two bits of Sj (in the next round). b. An output bit from Sj −1 goes to one of the last two bits of Sj (in the next round). c. An output of Sj +1 goes to one of the two middle bits of Sj (in the next round).
6. For each S-box, the two output bits go to the fi rst or last two bits of an S-box in the next round. The other two output bits go to the middle bits of an S-box in the next round.
7. If an output bit from Sj goes to one of the middle bits in Sk (in the next round), then an output bit from Sk cannot go to the middle bit of Sj . If we let j = k, this implies that none of the middle bits of an S-box can go to one of the middle bits of the same S-box in the next round.
S-boxes At least three weaknesses are mentioned in the literature for

1. In S-box 4, the last three output bits can be derived in the same way as the fi rst output bit by complementing some of the input bits.
2. Two specifically chosen inputs to an S-box array can create the same output.
3. It is possible to obtain the same output in a single round by changing bits in only three neighborin S-boxes
D-boxes One mystery and one weakness were found in the design of D-boxes:

1. It is not clear why the designers of DES used the initial and fi nal permutations; these have no security benefits.
2. In the expansion permutation (inside the function), the fi rst and fourth bits of every 4-bit series are repeated.

Key Scheduling Algorithm

The subkeys are computed using the following method:
1. The P-array and then the four S-Boxes are initialized with a fi xed string. The string is the hexadecimal digits of π.
2. P1 is XOR-ed with 32 bits of the key, P2 is XOR-ed with the next 32 bits of the key, and so on for all the bits of the key. If needed the key bits are cycled to ensure that all the P-array elements are XOR-ed.
3. An all-zero string is encrypted with the Blowfish algorithm, with the subkeys P1 to P18 obtained so far in steps 1 and 2. 4. P1 and P2 are replaced by the 64 bit output of step 3. 5. The output of step 3 is now encrypted with the updated subkeys to replace P3 and P4 with the ciphertext of step 4. 6. This process is continued to replace all the P-arrays and the S-Boxes in order.
This complex key-scheduling implies that for faster operations the subkeys should be precomputed and stored in the cache for faster access. Security analysis by Serge Vaudenay shows that for a Blowfish algorithm implemented with known S-Boxes (note that in the original cipher the S-Boxes are generated during the encryption process) and with r-rounds, a differential attack can recover the P-array with 28r+1 chosen plaintexts.

3.2 CONCLUSION

In this paper, we introduce a large universe searchable encryption scheme to protect the security of cloud storage system, which realizes regular language encryption and DFA search function. The cloud service provider could test whether the encrypted regular language in the encrypted cipher text is acceptable by the DFA embedded in the submitted search token. In the test procedure, no plaintext of the regular language or the DFA will be leaked to the cloud server. We also put forth a concrete construction with lightweight encryption and token generation algorithms. An example is given to show how the system works. The proposed scheme is privacy-preserving and indistinguishable against KGA, which are proved in standard model. The comparison and experiment result confirm the low transmission and computation overhead of the scheme.

REFERENCES

[1] Erl T, Cope R, Naserpour A. Cloud computing design patterns.Prentice Hall Press, 2015.
[2] Li Z, Dai Y, Chen G, et al. Toward network-level efficiency for cloud storage services[M]//Content Distribution for Mobile Internet: A Cloud-based Approach. Springer Singapore, 2016: 167-196.
[3] Sookhak M, Gani A, KhanMK, et al. Dynamic remote data auditing for securing big data storage in cloud computing[J]. Information Sciences, 2017, 380: 101-116.
[4] Zhang Q, Yang L T, Chen Z, Li P. Privacy-preserving doubleprojection deep computation model with crowdsourcing on cloud for big data feature learning[J]. IEEE Internet of Things Journal, 2017,
DOI: 10.1109/JIOT.2017.2732735.
[5] Zhang Q, Yang L T, Chen Z, Li P. PPHOPCM: Privacy-preserving High-order Possibilistic c-Means Algorithm for Big Data Clustering with Cloud Computing[J]. IEEE Transactions on Big Data, 2017, DOI: 10.1109/TBDATA.2017.2701816.
[6] Liu J K, Liang K, Susilo W, et al. Two-factor data security protection mechanism for cloud storage system[J]. IEEE Transactions on Computers, 2016, 65(6): 1992-2004.
[7] Boneh D, Waters B. Conjunctive, subset, and range queries on encrypted data[C]//Theory of Cryptography Conference. Springer Berlin Heidelberg, 2007: 535-554.