Privacy Preservation And Integrity Maintenance Using Fined-Grained Attribute Based Key Encryption Scheme.

Sukanya Gorade,Kiran Langhe,Deepti Indure,Yogita Kamble

[1] *Sukanya Gorade BEIT, Information Technology, Dr.D.Y.Patil Institute of Engineering and Technology, Maharashtra,India.*
[2] *Kiran Langhe BEIT, Information Technology, Dr.D.Y.Patil Institute of Engineering and Technology, Maharashtra,India.*
[3]*Deepti Indure BEIT, Information Technology, Dr.D.Y.Patil Institute of Engineering and Technology, Maharashtra,India.*
[4]*Yogita Kamble BEIT, Information Technology, Dr.D.Y.Patil Institute of Engineering and Technology, Maharashtra,India.*

ABSTRACT

Cloud computing has become a significant computing model in the IT industry, but Security and privacy are very important issues in cloud computing. In existing system access control in clouds are centralized in nature However, storing sensitive data on untrusted servers is a challenging issue for this model.

To guarantee their confidentiality and proper access control of outsourced sensitive data, classical encryption techniques are used. However, such access control schemes are not feasible in cloud computing because of their lack of flexibility, scalability and in terms of access control definition.

Instead, Attribute-Based Encryption (ABE) techniques, A new fine-attribute based data control scheme for secure data storage in clouds that supports anonymous authentication. this mechanism is nested in between fine-grained and attribute based encryption mechanism, which uses for Creation of access policy, file accessing and file restoring process.

In this scheme key aggregation is also used with respect to usage of key mechanism. F-ABE (Fine-Attribute Based Encryption) schemes and creates a secure way for authenticity and integrity maintenance for user's sensitive data, stored on Cloud Server.

Keyword : - *Fine Grained, Searchable Encryption, Multi Keyword, Cloud Computing.*

1. Introduction

Cloud computing is one way of computing. Here the computing resources are shared by many users. The benefits of cloud can be extended from individual users to organizations. The data storage in cloud is one among them. The virtualization of hardware and software resources in cloud nullifies the financial investment for owning the data warehouse and its maintenance. The advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted for out sourcing, which obsoletes traditional data utilization based on plain text keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in cloud, it is crucial for the search service to allow multi-keyword query and provide result similarity ranking to meet the effective data retrieval need. Related works on search able encryption focus on single keyword search or Boolean keyword

search, and rarely differentiate the search results. None the less, these ways are not useful because of their high computational overhead for both the cloud sever and user. On the contrary, more functional precise intent solutions, such as search able encryption (SE) schemes have made particular contributions in terms of efficiency, functionality and protection. Searchable encryption schemes allow the consumer to store the encrypted data to the cloud and execute keyword search over cipher text domain. So some distance, considerable works have been proposed below exclusive chance items to achieve various search functionality, such as single keyword search, similarity search, multi- key phrase Boolean search, ranked search, multi-keyword ranked search etc. Among them, multi-keyword ranked search achieves more and more awareness for its functional applicability. Lately, some dynamic schemes have been proposed to aid inserting and deleting operations on report collection. These are giant works as its incredibly possible that the data owners need to replace their knowledge on the cloud server. However few of the dynamic schemes help effective multi-keyword ranked search.

This paper proposes a secure tree-structured search scheme over the encrypted cloud information, which helps multi-key phrase ranked search and dynamic operation on the file assortment. Specially, the vector are a model and the generally-used term frequency inverse file frequency mannequin are combined in the index construction and question generation to provide multi-keyword ranked search. As a way to receive excessive search efficiency, we assemble a tree-situated index constitution and propose a grasping Depth first Search algorithm headquartered on this index tree. Due to the unique structure of our tree-established index, the proposed search scheme can exibly achieve sub-linear search time and handle the deletion and insertion of files. The secure KNN algorithm is utilized to encrypt the index and query vectors, and mean while make sure accurate relevance ranking calculation between encrypted index and query vectors. To withstand extra ordinary at tacks in extra ordinary chance models, we construct two comfy search schemes: the elemental dynamic multi-keyword ranked search (BDMRS) scheme in the known cipher text mannequin, and the enhanced dynamic multi-key phrase ranked search (EDMRS) scheme within the known history mannequin.

2. Architectural Design

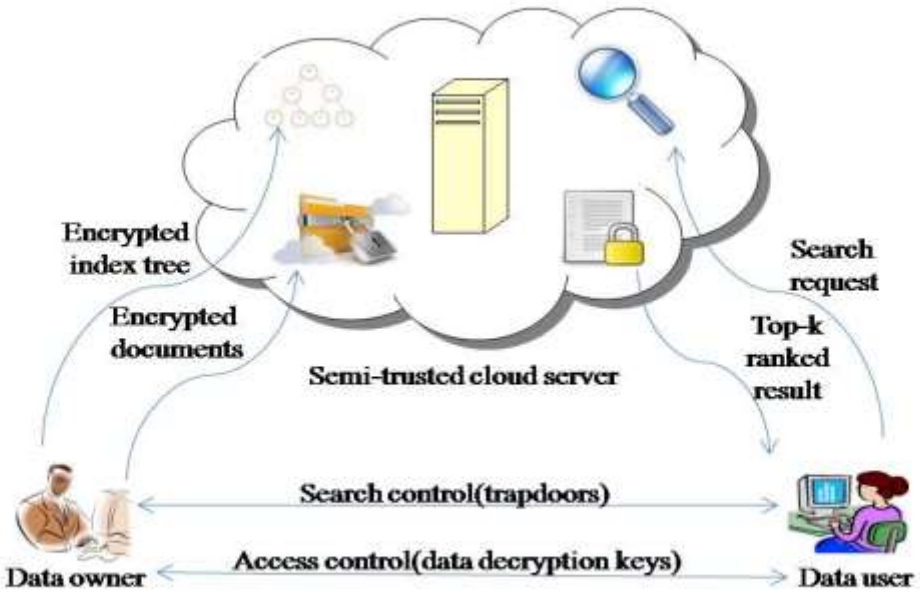

Fig.1: **System Model**

3. Algorithm:

3.1 RC6:

Notations for RC6

I)RC6-w/r/b parameters:
i) Word size in bits: w (32)(lg(w) = 5)
ii) Number of rounds: r (20)
iii)Number of key bytes: b (16, 24, or 32)

II)Key Expansion:
i)Produces array S[0 … 2r + 3] of w-bit round keys.

III)Encryption and Decryption:
i)Input/Output in 32-bit registers A,B,C

RC6 Encryption :

B = B + S[0]
D = D + S[1]
for i = 1 **to** r **do**

{
t = (B x (2B + 1)) <<<lg(w)
u = (D x (2D + 1)) <<<lg(w)
A = ((A Å t) <<< u) + S[2i]
C = ((C Å u) <<< t) + S[2i + 1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[2r + 2]
C = C + S[2r + 3]

B = B + S[0]
D = D + S[1]
for i = 1 **to** 20 **do**
{
t = (B x (2B + 1)) <<< 5
u = (D x (2D + 1)) <<< 5
A = ((A Å t) <<< u) + S[2i]
C = ((C Å u) <<< t) + S[2i + 1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[42]
C = C + S[43]

RC6 Decryption (for AES)

C = C - S[43]
A = A - S[42]

for i = 20 **downto**1 **do**
{
(A, B, C, D) = (D, A, B, C)
u = (D x (2D + 1)) <<< 5
t = (B x (2B + 1)) <<< 5
C = ((C - S[2i + 1]) >>> t) Å u
A = ((A - S[2i]) >>> u) Å t
}
D = D - S[1]
B = B - S[0]

3.2 K **Nearest Neighbour Algorithm :**

Assumptions

All the attribute values are numerical or real
Class attribute values are discrete integer values
For example: 0,1,2...

Algorithm

1. Read the training data from a file <x, f(x)>
2. Read the testing data from a file <x, f(x)>
3. Set K to some value
4. Set the learning rate LR
5. Set the value of N for number of folds in the cross validation
6. Normalize the attribute values by standard deviation

Assign random weight wi to each instance xi in the training set
Divide the number of training examples into N sets
Train the weights by cross validation
For every set Nk in N, do
Set Nk = Validation Set
For every example xi in N such that xi does not belong to Nk do
Find the K nearest neighbors based on the Euclidean distance
Calculate the class value as
Summation(wk X xj,k) where j is the class attribute
If actual class != predicted class then apply gradient descent
Error = Actual Class-Predicted Class
For every Wk

- Wk = Wk + LR X Error

Calculate the accuracy as

- Accuracy = (# of correctly classified examples / # of examples in Nk) X 100

Train the weights on the whole training data set
For every training example xi
Find the K nearest neighbors based on the Euclidean distance
Calculate the class value as

- Summation(wk X xj,k) where j is the class attribute

If actual class != predicted class then apply gradient descent
Error = Actual Class-Predicted Class
For every Wk

- Wk = Wk + LR X Error

Calculate the accuracy as

- Accuracy = (# of correctly classified examples / # of training examples) X 100

Repeat the process till desired accuracy is reached
For each testing example in the testing set
Find the K nearest neighbors based on the Euclidean distance
Calculate the class value as

- Summation(wk X xj,k) where j is the class attribute

Calculate the accuracy as
Accuracy = (# of correctly classified examples / # of testing examples) X 100

Example with Gradient Descent:

Consider K = 3, $\alpha = 0.2$, and the 3 nearest neighbors to xq are x1,x2,x3

K nearest neighbors	Euclidean Distance	Class	Random Weights
X1	12	1	W1 = 0.2
X2	14	2	W2 = 0.1
X3	16	2	W3 = 0.005

- Class of xq = 0.2 * 1 + 0.1 * 2 + 0.005 * 2 = 0.41 => 0
- Correct Class of xq = 1
- Applying Gradient Descent
- W1 = 0.2 + 0.2 * (1 - 0) = 0.4
- W2 = 0.1 + 0.2 * (1 - 0) = 0.3
- W3 = 0.005 + 0.2 * (1 - 0) = 0.205
- Class of xq = 0.4 * 1 + 0.3 * 2 + 0.205 * 2 = 1.41
- Class of xq => 1
- Simple K-NN would have predicted the class as 2

3.3 SPEKE Algorithm:

STEPS FOR SPEKE ARE AS BELOW:

1. Alice and Bob agree to use an appropriately large and randomly selected safe prime p, as well as a hash function H().
2. Alice and Bob agree on a shared password π.
3. Alice and Bob both construct g = H(π)2 mod p. (Squaring makes g a generator of the prime order subgroup of the multiplicative group of integers modulo p.)
4. Alice chooses a secret random integer a, then sends Bob ga mod p.
5. Bob chooses a secret random integer b, then sends Alice gb mod p.
6. Alice and Bob each abort if their received values are not in the range [2,p-2], to prevent small subgroup confinement attack.
7. Alice computes K = (gb mod p)a mod p.
8. Bob computes K = (ga mod p)b mod p.

Both Alice and Bob will arrive at the same value for K if and only if they use the same value for π. Once Alice and Bob compute the shared secret K they can use it in a key confirmation protocol to prove to each other that they know the same password π, and to derive a shared secret encryption key for sending secure and authenticated messages to each other. Unlike unauthenticated Diffie-Hellman, SPEKE prevents man in the middle attack by the incorporation of the password. An attacker who is able to read and modify all messages between Alice and Bob cannot learn the shared key K and cannot make more than one guess for the password in each interaction with a party that knows it. In general, SPEKE can use any prime order group that is suitable for public key cryptography, including elliptic curve cryptography.

3.4.Diffie-Hellman Key Exchange

Step	Alice	Bob
1	**Parameters: p,g**	
2	**A = random()** **$a = g^A \pmod p$**	**Random() = B** **$g^B \pmod p = b$**
3	**a** →	← **b**
4	**$K = g^{BA} \pmod p = b^A \pmod p$**	**$a^B \pmod p = g^{AB} \pmod p = K$**
5	← **E_K (data)** →	

3.5. Simple Password Exponential Key Exchange (SPEKE)

Step	Alice	Bob
1	**Parameters: p**	
2	**$G = H(password)^2$**	**$H(password)^2 = G$**
3	**A = random()**	**Random() = B**

	$a = G^A \pmod p$	$G^B \pmod p = b$
4	$a \longrightarrow$ $\longleftarrow b$	
5	$K = G^{BA} \pmod p = b^A \pmod p$	$a^B \pmod p = G^{AB} \pmod p = K$
6	$\longleftarrow E_K(\text{data}) \longrightarrow$	

3.5 Diffie-Hellman Encrypted Key Exchange

Step	Alice	Bob
1	Shared Secret: S = H(password)	
2	Parameters: p, g	
3	A = random() $a = g^A \pmod p$	Random() = B $g^B \pmod p = b$
4a	$E_S(a) \longrightarrow$ $\longleftarrow E_S(b)$	
4b	$a \longrightarrow$ $\longleftarrow E_S(b)$	
4c	$E_S(a) \longrightarrow$ $\longleftarrow b$	
5	$K = g^{BA} \pmod p = b^A \pmod p$	$a^B \pmod p = g^{AB} \pmod p = K$
6	$\longleftarrow E_K(\text{data}) \longrightarrow$	

4. CONCLUSIONS

In this paper, we have investigated on the fine-grained multikeyword search (FMS) issue over encrypted cloud data, and further intend to propose two FMS schemes. The FMS I includes both the relevance scores and the preference factors of keywords to enhance more precise search and better user's experience, respectively. The FMS II achieves secure and efficient search with practical functionality, i.e., "AND", "OR" and "NO" operations of keywords. Furthermore, we intent to propose the enhanced schemes supporting classified sub-dictionaries (FMSCS) to improve efficiency.

5. ACKNOWLEDGEMENT

In the end, We would like to take this opportunity to special thanks to Dr. Pramod Patil, Principal of College and Prof. Santosh Chobe, Head of Department for their kind support and providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.

6.REFERENCES

[1] Zhihua Xia, Member, IEEE, Xinhui Wang, Xingming Sun, Senior Member, IEEE, and QianWang, Member, IEEE, A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 2015.

[2] A break in the clouds: towards a cloud definition, ACM SIGCOMM Com-put.

Commun. Rev., vol. 39, no. 1,pp. 5055, 2009.

[3] B. S. Kamara and K. Lauter, Cryptographic cloud storage, in RLCPS, Jan-uary 2010, LNCS. Springer, Heidelberg.

[4] M. Chuah, W. Hu, Privacy-aware BedTree Based Solution for Fuzzy Multikeyword Search over Encrypted Data, 31st International Conference on Distributed Computing Systems Workshops, 2011.

[5] A. Singhal, Modern information retrieval: A brief overview, IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 3543, 2001.R. Nicole, Title of paper with only first word capitalized, J. Name Stand. Abbrev. in press.

[6] Chang Liu, Liehuang Zhu, Longyijia Li, Yuan Tan, Fuzzy Keyword Search On Encrypted Cloud Storage DataWith Small Index, 2011, Proceedings of IEEE CCIS2011.M. Young, The Technical Writers Handbook. Mill Valley, CA: University Science, 1989.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, Searchable sym-metric encryption: improved denitions and efcient constructions, in Pro-ceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 7988.

[8] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, Fuzzy keyword search over encrypted data in cloud computing, in INFOCOM, 2010 Pro-ceedings IEEE.

[9] D. Song, D. Wagner, and A. Perrig, Practical techniques for searches on encrypted data, in IEEE Proceedings of S&P, 2000.

[10]Wenjun Luo, Jianming Tan, Public Key EncryptionWith Keyword Search Based On Factoring, 2012, Proceedings of IEEE CCIS2012.

[11] Wang Jie, Yu Xiao, Zhao Ming, Wang Yon, A Novel Dynamic Ranked Fuzzy Keyword Search Over Cloud Encrypted Data, 2014, IEEE 12th In-ternational Conference on Dependable, Autonomic and Secure Computing.