VOL. 11, NO. 13, JULY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ATTRIBUTE BASED ENCRYPTION (ABE) ALGORITHM FOR SEARCHING AND SECURING ENCRYPTED DATA

A. Vinothkumar, M. Anand and S. Ravi
Department of Electronics and Communication Engineering, Dr. M.G.R. Educational and Research Institute University, Chennai, India
E-Mail: avinothme@gmail.com

ABSTRACT

Cloud servers are virtual servers that can be run on cloud computing environmental and it encrypts data by a common key. Due to this accessing client through server is easy. In this paper, client can encrypt, store and search data by their own key on server. If client sends encrypted queries to the server for searching, it returns the encrypted matching data without knowing about plain text. In this algorithm, attribute based searching mechanism is used for searching data on cloud in which server is only allowed to learn the set of encrypted documents and attribute of the documents and not the keyword data. A user's private key is associated with a set of attributes and ciphertext specifies an access policy over a defined universe of attributes within the system.

Keyword: encryption, decryption, data searching.

INTRODUCTION

In order to store large amount of data, client outsources their data files to a cloud server. The service provider is third party, thus the data must be encrypted as data can contain personal and private information. It is a type of public key encryption in which the secret key of a user and the ciphertext depend on attributes. The decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. Server allows clients to search the data. In searching process, all the data sent back to client for decryption may be searched. Thus, existing encryption methods are not suitable to search the encrypted data directly. Also searchable symmetric encryption is not efficient.

RELATED WORKS

[1] In this paper, novel computing paradigm introduced serious privacy challenges in that users' data are no longer locally possessed but stored on the remote server which belongs to a different trust domain compared with the data users This paper focuses on the privacy concerns in the secure search function performed over encrypted cloud data. In addition, the same search accuracy as the plaintext information retrieval can be realized using the state-of the-art similarity measure while search privacy is well protected.

[2] In this paper, a novel secure and efficient multi-keyword similarity searchable encryption (MK Sim) that returns the matching data items in a ranked ordered manner is proposed. Proposed scheme is proved to be secure against adaptive chosen-keyword attacks. Proposed scheme is adaptive semantically secure against adversaries and able to achieve optimal sub linear search time.

[3] Three application scenarios and identify the desirable security requirements are described. In this paper, two orthogonal categorizations and review the related security models for each category of SED schemes is provided. The practical issues related to SED (search in Encrypted data) schemes are analyzed. The brief analysis showed that there are a lot of potential securities issues facing SED schemes which are provably secure in their respective security models.

[4] In this paper, formalize and solve the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy is proposed. In proposed system, edit distance to quantify keywords similarity and develop two advanced techniques on constructing fuzzy keyword sets, which achieve optimized storage and representation overheads exploited.

[5] Proposed algorithm allows the users to query over the encrypted column directly without decrypting all the records. It's improves the performance of the system. The proposed algorithm works well in the case of range and fuzzy match queries. Proposed algorithm efficiently eliminates the limitations of the existing techniques for fuzzy match and range queries. This algorithm is efficient for searching of data whenever the retrieval of data is less than 40%of the total data.

[6] The proposed scheme, guarantees top-n multi keyword retrieval over encrypted cloud data with high privacy and practical efficiency using vector space model and TRSE, where in the majority of computing work is done on the server while the user takes part in ranking. The proposed system makes the system highly scalable and minimizes information leakage. Prevents overloads by ranking the files at the user side, reducing bandwidth and protects document frequency.

[7] An efficient encryption technique presented in this paper used for secure access to and storage of data on public cloud server, moving and searching encrypted data through communication channels while protecting data confidentiality. This method ensures data protection against both external and internal intruders. Data can be decrypted only with the provided by the data owner key, while public cloud server is unable to read encrypted data or queries. Answering a query does not depend on it size and done in a constant time. Data access is managed by the data owner. The proposed scheme allows unauthorized modifications detection.

[8] In this paper a security model for conjunctive keyword search over encrypted data and present the first schemes for conducting such searches securely. We propose a second scheme whose communication cost is on the order of the number of keyword fields and whose security relies on a new hardness assumption.

[9] In this paper, an efficient scheme for similarity search over encrypted data is proposed. To ensure the confidentiality of the sensitive data, a rigorous security definition and prove the security of the proposed scheme under the provided definition is provided. In addition, a real world application of the proposed scheme and verify the theoretical results with empirical observations on a real dataset is provided.

[10] In this paper, data encryption oncloud as well as corresponding security issues has been addressed. The proposed method incorporates two main phases: indexing and searching. Trap door and code word are the two security parameters applicable in this technique. Simulated results demonstrate that it provides fast and efficient ranking sentence search for unstructured data in original documents oncloud server. The proposed technique reduces the overhead of decryption thereby minimizing the search time to a considerable extent.

[11] In this paper, the problem of exact keyword match by providing searching with fuzzy keyword is solved. Two more techniques called gram based technique which is useful for reducing the time, providing fasts searching and increase the performance by considering substring from the given string is proposed. An experimental result demonstrates the efficiency of proposed solution.

[12] This paper proposes an efficient and more secure algorithm to solve the challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE). This paper introduced the concept of keyword buffer controller that allows for quick search of documents and establishes a set of strict privacy requirements for such a secure cloud data utilization system to become a reality.

[13] In this paper a proposed solution incorporate the hash table management and indexing techniques to keep track the actual data contents in terms of document features which may help for encrypting user data and identifying the user data and privacy. In this paper, a scheme for secure data accessing with maintaining its privacy by using strong cryptographic algorithm is introduced.

[14] In this paper, cryptographic schemes for the problem of searching on encrypted data and provideproofs of security for the resulting crypto systems is described. The proposed methods are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and blockcipher operations), and introduce almost no space and communication overhead.

[15] This paper first explains public key encryption with keyword Search (PEKS) algorithm and then proposes an improved secure searchable encryption algorithm based on Indistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA2). The proposed searchable encryption is mathematically proven secure and it has the ability to perform a search within the encrypted data.

SYSTEM MODEL

In this design, the client has a collection of n data to outsource to the cloud server in the encrypted form. The encrypted data be searchable by clients encrypt data using their own keys, and then outsources encrypted data to the server. To search over the document collection for a given keyword is sent to the cloud server. After receiving the keyword, the server is responsible to search the index and return the corresponding set of encrypted documents.

PROPOSED ALGORITHM DESCRIPTION

The Algorithms has developed in the following modules from A to Z along with the Keys.

Table-1. The proposed encryption method for searching the encrypted data directly.

Letter	Key
A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
N	0
K	+
M	-
P	*
Q	/
L	=
R	^
W	.
Z	i

Example: 1

As an illustration the arithmetic operation (162 + 138) is evaluated with the result value (162+138=300) and when the values in the expression are converted into floating value (162.0+138.0=300.0) encrypted expression is AFBWNKACHWNLCNNWN (by Table 1). At the decryption side, the searching is done by calculating edit distance.

a) Encrypted data searching

In encryption, client enters the numerical data with their private key (‘+’,’ –‘,’ *’,’ /’,’ =’,’ ^’, ‘.’,’i’). The

private keys entered by client should be arithmetic operator. The corresponding arithmetic operations are performed in encryption process. The result is appended with the client data. Data and result values are converted into floating point values. The resultant data is encrypted using the keyword table by searching the encrypted data directly. Each numeric value of the given example is mapped to the corresponding alphabets. (i.e., 1 mapped to A, + mapped to K etc,)
To illustrate, Let the
Plaint text with private key: 256-34
Executed operation with result: 256-34=222
Floating point conversion: 256.0-34.0=222.0

Plain text	2	5	6	.	0	-	3	4	.	0	=	2	2	2	.	0
Encrypted text	B	E	F	W	N	M	C	D	W	N	L	B	B	B	W	N

The pseudo code for the encry

256, 34 are the numerical value. '-' is the private key. The corresponding subtraction operation executed between 256 and 34. 256-34=222. The plain text "256-34" encrypted into "BEFWNMCDWNLBBBWN".

b) Modified encryption module

The steps for encrypting searched data are illustrated in the flow chart (Figure-1).

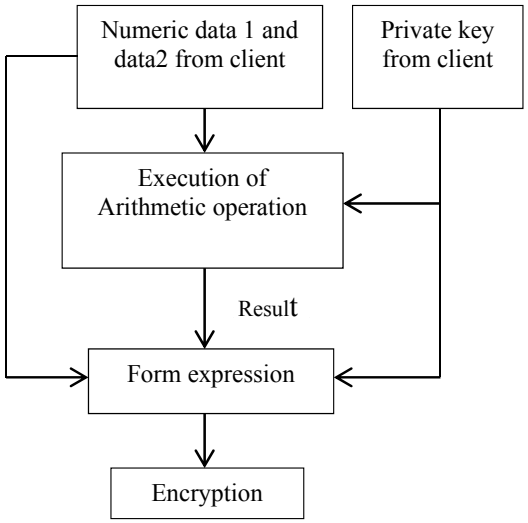

Figure-1. Encryption flow chart.

The implementation of proposed encryption algorithm is essential for secure communication. It takes numeric data, plain text and key as parameters and can be straight forwardly adapted to various implementation contexts/security requirements.

c) Pseudo code for encryption

The pseudo code for the encryption is as follows:

```
set index i=0 for formed expression
for( length of formed expression)
switch (expression(i))
case (i)
expression(i)==numeric data
expression (i)=(A or B or C or D or E or F or G or H or I)
case (ii)
expression(i)==arithmetic operator
expression(i)=(K or M or Q or P)
case (iii)
expression(i)==Power
expression(i)=R
case (iv)
expression(i)==floating point
expression(i)=W
end switch statement
end for statement
return expression
```

Searching

The attribute of encrypted data is stored in server. If server searches query, the query attribute is compared with encrypted data attribute. The matched files return to the client.

PROPOSED METHOD ON SEARCHING QUERIES OVER INCOMPLETE DATA

The attributes of every database is stored in server as an information system. A method of searching query over an incomplete data is proposed in this paper. Search query over incomplete data (S) is achieved by extracting attributes of the data from information system which has closest attribute to S. In server, there is a possibility for attribute missing in one of these databases. If client searches over the incomplete data, answering will not be possible because of mismatch between query and information system attributes. Information system of incomplete data does not contain the attribute value. By proposed algorithm, for answering query unknown attribute is determined from the similar information system. Consider S1 and S2 are the incomplete information systems, same set of information system stored with same set of attribute used to describe them. Values of attributes (a_{S1} and a_{S2}) for both systems (S1 and S2) are same. The relationship (z) between S1 and S2 is calculated by a_{S2}- a_{S1}. S1 was transferred to S2 by mapping z. This can be represented as z(S1)=S2 or $z(a_{S1})=z(a_{S2})$.

Table-2. Information system for S1.

X	a	b	C	d	e
x1	{(a_1,0.33), (a_2,0.66)}	{(b_1,0.66), (b_2,0.33)}	c_1	d_1	{(e_1,0.5), (e_2,0.5)}
x2	{(a_2,0.25), (a_3,0.75)}	{(b_1,0.33), (b_2,0.66)}		d_2	e_1
x3		b_2	{(c_1,0.5), (c_3,0.5)}	d_3	e_3
x4	a_3		c_2	d_1	{(e_1,0.66), (e_2,0.33)}
x5	{(a_1,0.66), (a_2,0.33)}	b_1	c_2		e_1
x6	a_2	b_2	c_3	d_2	{(e_2,0.33), (e_3,0.66)}

Table-3. Information system for S2.

X	a	b	c	d	e
x1	{(a_1,0.33), (a_2,0.66)}	{(b_1,0.66), (b_2,0.33)}	c_1	d_1	{(e_1,0.5), (e_2,0.5)}
x2	{(a_2,0.25), (a_3,0.75)}	b_1	{(c_1,0.33), (c_3,0.66)}	d_2	e_1
x3	a_1	b_2	{(c_1,0.5), (c_3,0.5)}	d_2	e_3
x4	a_3		c_2	d_1	{(e_1,0.66), (e_2,0.33)}
x5	{(a_1,0.66), (a_2,0.33)}	b_1	c_2	d_2	e_1
x6	a_2	b_2	c_3	d_2	{(e_2,0.33), (e_3,0.66)}

From Table-2 and Table-3 of S1 and S2 it is evident that unknown attribute values are determined by mapping. The mapping is subject to the constraints that the original data is integer. Attribute value of S2 is lesser than attribute value of S1. This is given by z(S1)=S2.

A. Decryption

In decryption, encrypted data with their encrypted private key ('K', 'M',' P',' D',' L',' R', 'W','Z') is decrypted. The encrypted data is decrypted using the keyword table for numeric values of each letter, A mapped to 1, K mapped to +.

Encrypted data: BEFWNMCDWNLBBBWN

Delete W (floating point) and zero (0) from encrypted data: BEFMCDLBBB

Separate encrypted data into two sequences by L (equal): BEFMCD, LBBB

First sequence consider as data: BEFMCD

Separate first sequence by encrypted private key of client: BEF, CD

Mapped data: 256, 34

Encrypted text	B	E	F	W	N	M	C	D	W	N	L	B	B	B	W	N
Decrypted text	2	5	6	.	0	-	3	4	.	0	=	2	2	2	.	0

256, 34 are the numerical value. '-' is the private key. The result of subtraction operation is 222. The decrypted plain text is "256-34"

B. Flow chart for decryption

Figure-2 shows the flow chart for decrypting the encrypted data using the keyword Table.

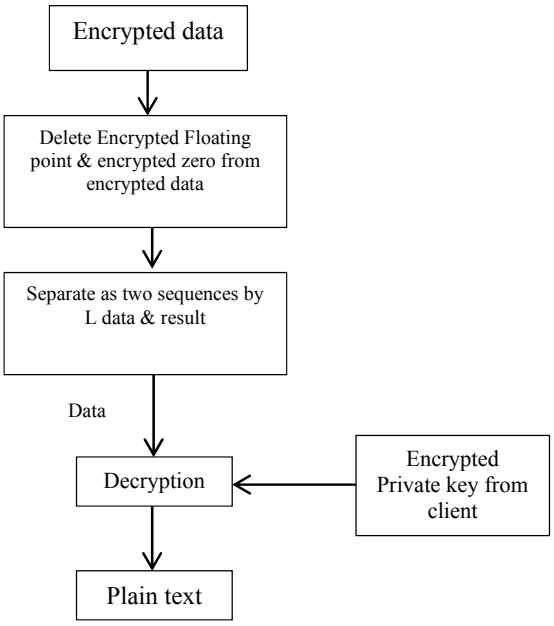

Figure-2. Flow chart of decryption.

C. Pseudo code for decryption

set index i=0 for encrypted data sequences
for length of encrypted data sequences
switch(expression(i))
case (i)
expression(i)== (A or B or C or D or E or F or G or H or I)
expression (i)= (0 to 9)
case (ii)
expression(i)== (K or M or Q or P)
expression(i)= arithmetic operator
case (iii)
expression(i)==R
expression(i)=Power
case (iv)
expression(i)==W
expression(i)= floating point
end switch statement
end for statement
return plain text

IMPLEMENTATION DETAILS

The encryption scheme according to Table-1 is implemented in Java language and the output is shown in Figure-3 for the sample input provided. The input includes numeric data, arithmetic expression (addition, subtraction, multiplication and division), power, exponentiation and also complex representation.

Output screen shots

Input data

162 + 138
256 - 34
28 * 2
43 / 34
11 ^ 2
23.58 ~ 1
4 i 43

Encrypted data

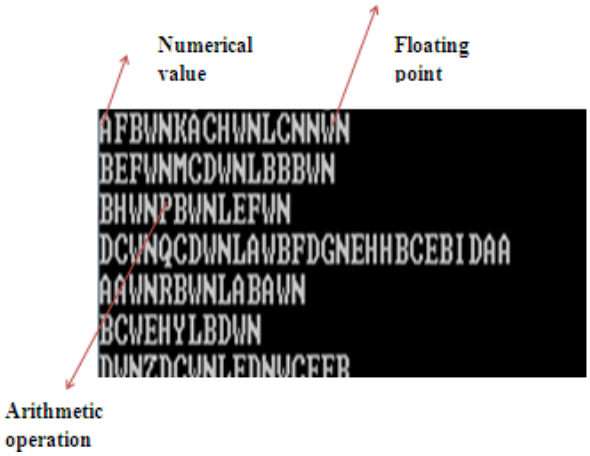

Figure-3. Encrypted output.

In Figure-4 the encrypted and decrypted speed details has mentioned in implemented reported and proposed in table 3 and Table-4.

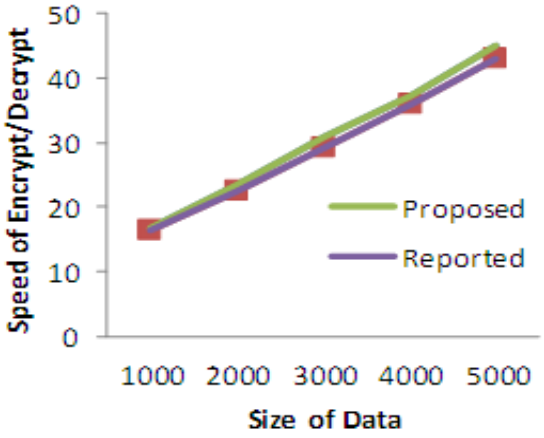

Figure-4. Speed of encryption and decryption.

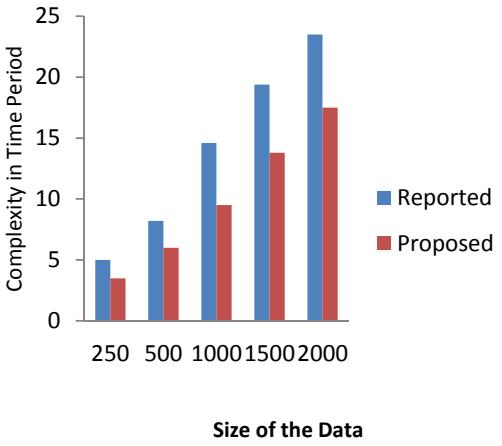

Figure-5. Comparison result of complexity.

Table-4. Comparison of retrieved documents.

Percentage of retrieved documents	Time in seconds	
	Proposed	Reported
2	0.25	1.40
4	0.30	1.45
6	0.35	1.46
8	0.40	1.47
10	0.45	1.48
12	0.50	1.49
14	0.55	1.50
16	0.60	1.51
18	0.65	1.52
20	0.70	1.53

Table-5. Comparison of search time.

Search time in seconds	Data size	
	Proposed	Reported
0.5	1MB	0.5MB
1.0	2MB	1.2MB
1.5	3MB	2.1MB
2.0	4MB	3.2MB
2.5	5MB	4.1MB

CONCLUSIONS

In this paper the encryption and decryption of data is done by user private key. The query searching is based on the attribute of the data. Query search over the incomplete data is achieved by extracting missing attributes from comparably relative files. The encrypted scheme is implemented in java threads. By this algorithm, client securely outsources their data on cloud server and searching data of encrypted to decrypt the data easy by proposed algorithm. Encryption involves processor overhead many cloud providers will only offer basic encryption on a few database fields, such as passwords and account numbers. To keep low costs and high performance. The method alternative to encryption is provided that does not require much processing power. As a future scope, redacting techniques are used that needs to remain confidential or the use of proprietary encryption algorithms created by the vendor.

REFERENCES

[1] Wenhai Sun, Wenjing Lou, Y. Thomas Hou, and Hui Li. 2014. Privacy-Preserving Keyword Search over Encrypted Data in Cloud Computing. Springer Science, New York, USA. pp. 189-212.

[2] Mikhail Strizhov and Indrajit Ray. 2014. Multi-keyword Similarity Search over Encrypted Cloud Data. International conference on ICT systems security and privacy protection. 428: 52-65.

[3] Qiang Tang. 2012. Search in Encrypted Data: Theoretical Models and Practical Applications. IGI Global.

[4] P. Naga Aswani and K. Chandra Shekar. 2012. Fuzzy Keyword Search over Encrypted Data using Symbol-BasedTrie-traverse Search Scheme in Cloud Computing. c s chronicle.

[5] Manish Sharma, Atul Chaudhary and Santosh Kumar. 2013. Query Processing Performance and Searching over Encrypted Data by using an Efficient Algorithm. International Journal of Computer Applications (0975-8887). 62(10).

[6] D. Pratiba, Dr. G. Shobha and Vijaya Lakshmi.P.S. 2015. Efficient Data Retrieval from Cloud Storage using Data mining technique. International Journal on Cybernetics and Informatics (IJCI). 4(2).

[7] Hasan Omar Al-Sakran. 2015. Accessing Secured data in Cloud Computing Environment. International Journal of Network Security & Its Applications"(IJNSA). 7(1).

[8] Philippe Golle, Jessica Staddon and Brent Waters. 2004. Secure Conjunctive Keyword Search Over Encrypted Data. Second International Conference, ACNS.

[9] Mehmet Kuzu, Mohammad Saiful Islam and Murat Kantarcioglu. 2012. Efficient Similarity Search over Encrypted Data. (ICDE), 2012 IEEE 28th International

Conference on Data Engineering, ISSN 1063-6382, pp.1156-1167.

[10] Muhammad Sajid Khan, Chengliang Wang, Ayesha Kulsoom and ZabeehUllah. 2013. Searching Encrypted Data on Cloud. IJCSI International Journal of Computer Science Issues. 10(6) (1).

[11] P.Niranjan Reddy and Y.Swetha. 2013. Techniques for Efficient Keyword Search in Cloud Computing. (IJCSIT) International Journal of Computer Science and Information Technologies. 4(1): 66-68.

[12] Suman M and B. Chempavathy. 2014. An Approach for Efficient and Secure Retrieval of Encrypted Cloud Data Based on Top-K Multi keywords. (IJCSIT) International Journal of Computer Science and Information Technologies. 5(3): 3239-3241.

[13] Krati Mehtoand Rahul Moriwal. 2015. A Secured and Searchable Encryption Algorithm for Cloud Storage. International Journal of Computer Applications (0975 - 8887). 120(5).

[14] Dawn Xiaodong, Song David and Wagner Adrian Perrig. 2000. Practical Techniques for Searches on Encrypted Data. IEEE Symposium on Security and Privacy.

[15] Majid Nategizad, Majid Bakhtiari and MohdAizaini Maarof. 2014. Secure Searchable Based Asymmetric Encryption in Cloud Computing. Int. J. Advance. Soft Comput. Appl. 6(1).

[16] Cong Wang, Ning Cao, Kui Ren and Wenjing Lou. 2010. Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data. 30th International Conference on Distributed Computing Systems (ICDCS'10), ISSN.1045-9219, pp. 1467-1479.