International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

An Efficient Model Over Attribute Encrypted Outsourced Data Using Secure K-D Tree

J.PREMALATHA
M.Tech (VLSI)
Dept of E.C.E
Chaitanya Institute of Technology and Science,
Warangal, Telangana,India
Email: premaltha.jannu77@gmail.com

T. RATAN BABU
Assistant Professor
Dept of E.C.E
Chaitanya Institute of Technology and Science,
Warangal, Telangana,India
Email: ratantelusuri@gmail.com

ABSTRACT

In cloud era, it raises the intractability to keep the confidentiality and efficiency simultaneously while outsourcing encrypted data base. The secure issues are inevitable and urgent in range query (a pivotal component). *Secure range query* (SRQ), removing such concerns, is urgent to resolve. Concretely, a *secure k-d tree* (SKD tree) is constructed by invoking *comparable encryption* (CE) and AES, and SKD mechanism is proposed to resolve SRQ directly over the encrypted data. Until both the CE and AES are cracked, it is infeasible to reveal the query, data base or query result. In our proposal, only a single interaction is occurred between user and cloud, which promotes the searching efficiency dramatically. Experimental studies show that SKD mechanism is efficient over really geographical datasets.

***Keywords*-secure range query; k-d tree; comparable encryption**

I. INTRODUCTION

For individuals and organizations, subjected to various of limitations, there is a central requirement to outsource both the data base and workloads to cloud server. For example, by the end of 2013, Netflix has been 100% on Amazon Web Services1. However, cloud computing raises the secure concerns of user’s confidentiality.

A sound trade-off between security and efficiency are as follows. The private data owner (*owner*) encrypts the data base and builds a secure index. Following on that, *owner* subcontracts them to cloud server (*cloud*). The authorized user (*user*) submits the encrypted query to *cloud*, which locates the candidates of query directly over the encrypted data base afterwards. Finally, *user* complete the query by decrypting and pruning the returned encrypted candidates.

A. Motivation and Scenario

Consider a scenario that a national government organizes a large data base consisting of massive private resources with their specific locations (e.g., longitude, latitude, etc). The government (i.e., *owner*) allows paid companies (i.e., *user*) to query for private resources within a geographical region. Due to economize and security reasons, *owner* subcontracts both the encrypted data base and queries to *cloud*. Such a problem is named *secure range query* (SRQ).

SRQ is also applied extensively to *location-based services* (LBS), geo-referenced market research, etc. *Owner* and *user* are fully trusted. However, *cloud* is semi-honest by strictly following the mechanism but also being curious about the data base and workload.

During the whole process, *cloud* can analyze the workload without violating the mechanism. In this paper, we aim at proposing a mechanism to resolve SRQ problem with minimal communication cost and round trips between *user* and *cloud* dramatically, named *secure k-d* (SKD) mechanism. The number of round trip in SKD is only 1, which is the possible minimal one.

B. Contributions

For the sake of confidentiality, *cloud* is always considered to be a pure storage without any unilateral process capacity, which leads to multi round trips between *user* and *cloud*. At the mean time, existing schemes can not provide a sound filter capacity. Both put tremendous negative affect on the efficiency of existing schemes. The proposed scheme here resolves above concerns. Our contributions are as follows:

- We present a concrete SKD that minimize the communication cost and round trip.
- Experimental evaluations and comparisons over real datasets guarantees our scheme's superiority.

II. PRELIMINARIES AND RELATED WORK

We investigate CRT mechanism and show the drawbacks. Then, we review *comparable encryption* (CE) algorithm briefly and discuss the related works.

A. CRT mechanism

CRT, as a state-of-art in SRQ, employs R*-tree and AES to provide the confidential of query and data base. There exists a multi round trip between *user* and *cloud* in CRT. Recently, there are still many works related to data base outsourcing. Most relevant works, assume a pre-installed trusted component (e.g., software, tamper-proof device, etc) in *cloud*. Consequently, *cloud* must be opened completely to *owner* which is unacceptable. It is obvious that we make a sound tradeoffs in terms of confidentiality and efficiency.

B. Comparable Encryption

Furukawa presented an optimized short CE in 2014. CE is composed of Gen, Enc, Der and Cmp. Here, Gen, system parameter *param* and secret key *mkey* are omitted for the lack of space. Further information is recommended.

Enc: *ciph* = Enc(*num*).

Der: *tok* = Der(*num*).

Cmp: *res* = Cmp(*ciph*, *ciph_*, *tok*).

C. Related work

***Cryptography based approaches*.**

Providing *database as a service* (DaaS) was firstly introduced. Agrawal et al. presented an insecure orderpreserving encryption (OPE) scheme that allows any comparison directly over encrypted data. Popa et al. presented the first ideal secure OPE. However, multi round trips are inevitable and lead to low efficiency. In *owner* either builds a B+-tree on one-dimensional data or an R*-tree for two-dimensional data. At the meantime, conventional cryptographic techniques (e.g., AES) are employed to encrypt nodes in the tree. However, in CRT, The multi round trips are intolerable. In SKD, *cloud* burdens most of workloads, and thereby further the quality of service is guaranteed.

***Anonymity based approaches*.**

To protect privacy, *user*'s location is generalized by a trusted query (i.e., location) anonymizer. The anonymous spatial region is instead of exact location. It hides the location of *user*, but does not protect the data base. Following on that, *t*-closeness and *l*-diversity are proposed. In these methods, the generalization process alters the dataset only for statistical purposes. This goes against our original intention.

III. CONSTRUCTION

This section introduces the problem of SRQ. Following on that, a concrete SKD mechanism is proposed.

A. Problem Definition

The architecture is illustrated in Fig. 1. First *owner* builds an appropriate secure index and encrypts the data base. Both them are subcontracted to *cloud*. The secret keys are shared between *owner* and *user*. When issuing a new query, *user* encrypts the query and submits it to *cloud*. Following on that, *cloud* conducts the query over encrypted data base and index, and then returns the encrypted candidate. Finally, *user* decrypts and prunes the candidates to get the accurate result.

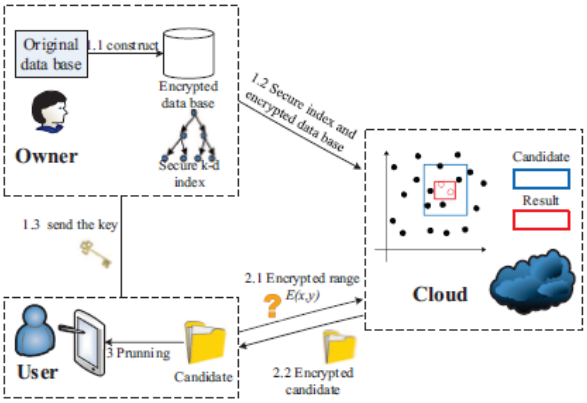


Figure 1: The architecture for secure range query.

B. Secure k-d tree

SKD tree employs short CE to achieve confidentiality of nodes in standard k-d tree. It provides a security guarantee resisting any type of location-based attack (e.g., the general attack). During the whole query processing of encrypted query $E(q)$, there is only single communication with *cloud* while locating candidate. The algorithm will be executed until the original (remainder) dataset D is empty (Lines 1-3). First of all, determine the scopes of each dimension separately and encrypt the elements in nodes under CE. Then, we calculate the split dimension and generate the root of subtree. We encrypt the data of all the leaves. Then, repeat all the procedures to allocate all data.

The SKD tree can be constructed in which each node stores the ciphers of all the data *aes ciph* encrypted by AES and the ciphers of all the data *ce ciph* encrypted by CE.

C. Secure Range Query

Given a new query q, *user* generates encrypted query $E(q) = \{tok0, ciph0, tok1, ciph1\}$ by encrypting each dimension with *ce*. Then, *user* sends $E(q)$ to *cloud*, to locate candidate. It is a typical recursive access, and we let the detail ignored. Since CE can achieve cipher comparison, *overlap* can be instantiated in a trivial way and is omitted here. The encrypted candidate generated is sent to *user*. Then, he decrypts and prunes the false candidates. Such a process is trivial and omitted here too. Note that, S is global, but *root* is local.

D. Complexity analysis

Hash is the cost of computing hash function which is considered as the dominant cost. The cost to encrypt query is 2(4*nb*+1)*d* ・Hash, which integrates the time of generating ciphers and tokens. Furthermore, the query phase is almost completely conducted in *cloud*, in which the cost is *O*(*d* ・*n*1−1*d*) ・ 2(*nb*−*l*+2) ・ Hash. *User* needs to decrypt *nc* candidates. Hence, the cost mainly comes from the decryption and is *d* ・ *nc* ・ *td*, in which *td* is the encryption time for a single dimension.

E. Security analysis

Security concern of our scheme mainly depends on CE, since AES is deeply probed and widely adopted. We consider the possible security breach from an attacker who can not change the SKD mechanism itself. Assume that *A* (including *cloud*) is an attacker and monitors all the processing of queries, aiming to grab useful information. The confidentiality of the query and data is protected due to the indistinguishability property of CE. Assume there is an adversary that can obtain non-trivial information from CE ciphers with a nonnegligible probability. It is easy to distinguish the differences between those ciphers. They are contradictory.

Additionally, it is intractable for attacker *A* to obtain non-trivial information from returned encrypted candidate. Suppose that *A* obtains a set of candidates within the scope of the hyper-rectangle bounded by query *q*. All the information *A* can inferred is that ***p*** falls in the hyper-rectangle. That is to say, for any two points in the hyper-rectangle, *A* cannot distinguish them, since they follow the same path along SKD tree. Therefore, the whole query processing only reveals trivial information to an external attacker *A*.

IV.RESULTS

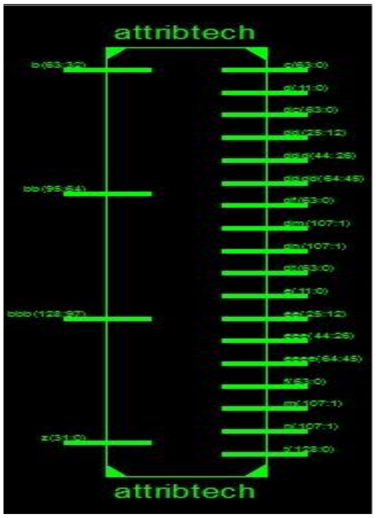
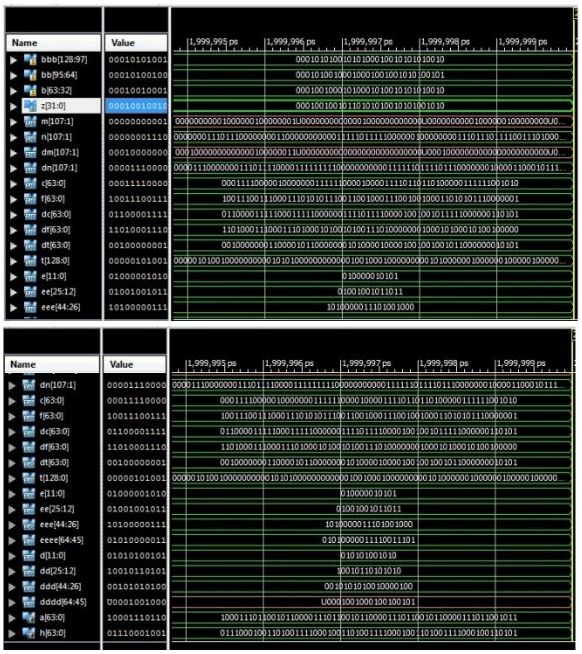


Figure 2: RTL SCHEMATIC



```
Name          Value         |1,999,995 ps |1,999,996 ps |1,999,997 ps |1,999,998 ps |1,999,999 ps
bbb[128:97]   00010101001   00010101001010100010010101010010
bb[95:64]     00010100100   00010100100010001001001010100101
b[63:32]      00010010001   00010010001010001010010101010010
z[31:0]       0001001001C   00010010010110101010010101001010
m[107:1]      00000000001   000000000010000001000000001U00000000000000100000000000000U00000000000100000010000000U0...
n[107:1]      00000001110   00000001110111000000000110000000000001111101111100000010000000011101110111100111010000...
dm[107:1]     00010000000   00010000000000000010000001 1U000000000000000000000000000U0001000000000000000000000000U0...
dn[107:1]     00001110000   00001110000000111011110000111111111000000000001111110111101110000000100001100010111...
c[63:0]       00011110000   0001111000001000000011111100001000011110110110100000111111001010
f[63:0]       10011100111   1001110011100011101010111001100100011100100100011010101110000001
dc[63:0]      01100001111   0110000111111000111110000001111011110000100100101111100000011 0101
df[63:0]      11010001110   1101000111000111010001010010100111010000000100010100010100100000
dt[63:0]      00100000001   0010000000110000101100000000101000010000100100100101100000000 10101
t[128:0]      00000101001   00000101001000000000001010100000000000001001000100000000010100000001000000100000100000...
e[11:0]       01000001010   010000010101
ee[25:12]     01001001011   01001001011011
eee[44:26]    10100000111   10100000011101001000

Name          Value         |1,999,995 ps |1,999,996 ps |1,999,997 ps |1,999,998 ps |1,999,999 ps
dn[107:1]     00001110000   00001110000000111011110000111111111000000000001111110111101110000000100001100010111...
c[63:0]       00011110000   0001111000001000000011111100001000011110110110100000111111001010
f[63:0]       10011100111   1001110011100011101010111001100100011100100100011010101110000001
dc[63:0]      01100001111   0110000111111000111110000001111011110000100100101111100000011 0101
df[63:0]      11010001110   1101000111000111010001010010100111010000000100010100010100100000
dt[63:0]      00100000001   0010000000110000101100000000101000010000100100100101100000000 10101
t[128:0]      00000101001   00000101001000000000001010100000000000001001000100000000010100000001000000100000100000...
e[11:0]       01000001010   010000010101
ee[25:12]     01001001011   01001001011011
eee[44:26]    10100000111   10100000011101001000
eeee[64:45]   01010000011   01010000011110011101
d[11:0]       01010100101   010101001010
dd[25:12]     10010110101   10010110101010
ddd[44:26]    00101010100   0010101010010000100
dddd[64:45]   U0001001000   U0001001000100100101
a[63:0]       10001110110   1000111011001011000011101100101100001110110010110000111011001011
h[63:0]       01110001001   0111000100110100111100010011010011110001001101001111000100110101
```

Figure 3: OUTPUT WAVEFORM

V. CONCLUSION

We coin a mechanism for constructing an SKD tree for efficient range query over encrypted multidimensional data, which provides the confidential of query and data base, and range search ability at the same time. In our mechanism, it is intractable for an attacker (including *cloud*) to recover the original dataset precisely from the SKD tree, encrypted query or encrypted candidate. The computation cost and round trip between *cloud* and *user* are extremely low compared to the previous techniques (i.e., CRT). Experimental studies over four real private spatial datasets shows that our scheme is light enough specific to the resource constrained users.

VI.REFERENCES

[1] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Enabling search services on outsourced private spatial data," *The VLDB Journal*, vol. 19, no. 3, pp. 363–384, 2010.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *SIGMOD*, 2004, pp. 563–574.

[3] E. Damiani, S. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Balancing confidentiality and efficiency in untrusted relational dbmss," in *CCS*, 2003, pp. 93–102.

[4] J. Furukawa, "Short comparable encryption," in *Cryptology and Network Security*, 2014, vol. 8813, pp. 337–352.

[5] H. Hacig¨um¨us, B. Iyer, and S. Mehrotra, "Providing database as a service," in *ICDE*. IEEE, 2002, pp. 29–38.

[6] R. A. Popa, F. H. Li, and N. Zeldovich, "An idealsecurity protocol for order-preserving encoding," in *S&P*, 2013, pp. 463–477.

[7] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *ICDCS*. IEEE, 2005, pp. 620–629.

[8] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *SIGMOD*, 2008, pp. 121–132.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *TKDE*, vol. 19, no. 12, pp. 1719–1733, 2007.

[10] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *VLDB*, 2006, pp. 763–774.

[11] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *ICDE*, 2007, pp. 106–115.

[12] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *TKDD*, vol. 1, no. 1, p. 3, 2007.

Authors:

J.PREMALATHA studying M.Tech (VLSI) from Chaitanya Institute of Technology and Science, Warangal, Telangana,India.

T. RATAN BABU working **as** Assistant Professor in Dept of E.C.E from Chaitanya Institute of Technology and Science, Warangal, Telangana, India.