A Technique for Handling Range and Fuzzy Match Queries on Encrypted Data

Shaukat Ali, Azhar Rauf, and Huma Javed
Department of Computer Science, University of Peshawar, Pakistan

Abstract: *Data is an important asset of today's dynamically operating organizations and their businesses. Data is usually stored in databases. An important issue for IT professionals is to secure such data from unauthorized access and intruders. For protecting business centric data, many levels of security are used. Among these levels, data encryption is the final layer of security. Although encryption makes it difficult to breach this level of security, but it has a potential disadvantage of performance degradation, particularly for those queries which require operations on encrypted data. This work proposes to allow users to query over the encrypted column directly without decrypting it. This improves the performance of the SELECT query. In this technique the query retrieves only those records fulfilling the user's search criteria and the data will be decrypted on the fly. The proposed algorithm handles range, fuzzy match, and exact match type queries. It has no "false positive hits". From experimental findings the algorithm has shown greater efficiency than the state of art technique.*

Keywords: *Database security, encryption, performance.*

Received June 27 2010; accepted March 1, 2011; published online March 1, 2012

1. Introduction

Data is one of the most valuable assets of organizations and it is important to keep this data secured for the efficient growth of an organization. Organizations' operative data is stored in databases. It is a continues challenge for IT professionals and companies to develop and implement strategies to keep this data protected from unauthorized access. There are many techniques used to secure database. Database security methods can be divided into four layers [6]: physical security [3], operating system security [9], DBMS security [12, 16], and data encryption [2, 13, 14, 15]. However, encryption is the final layer of security in the databases [11]. Even if an intruder bypasses all security layers and successfully penetrates into the database, still he will not be able to break the encryption scheme applied to sensitive data. It means that the hacker will get the data in the unreadable form and thus no information can be retrieved from it [11].

Data encryption is a strong layer of security specially in those organizations where the security risks are high. However, encryption degrades the SELECT query performance significantly because the Structural Query Language (SQL) queries cannot be executed directly on the encrypted data. As first the encrypted data needs to be decrypted and then SQL query can be run on it. This is expensive performance wise.

There are some techniques which address the problem of performance degradation. However, these techniques are somehow limited in their applicability. For example, hashing technique [20, 22], bucketization [7], order preserving technique [1] have got one or another problem. These techniques are limited to full text matches and give partial solution to range and fuzzy match type queries.

This paper proposes a novel searching technique which improves not only the performance of searching but also, handles the limitation of the existing techniques. The proposed technique works with all types of queries including range and fuzzy match type queries. It is also, compatible with all types of encryption algorithms.

The rest of the paper is structured as follow: Section 2 describes related work. Section 3 presents the actual research problem and hypothesis. Section 4 is focused on proposed security technique and section 5 describes how to search in the encrypted column. Section 6 is about the architecture of the proposed Security system and section 7 presents the algorithm. Section 8 gives the statistical results of the algorithm after testing and section 9 concludes the work done.

2. Related Works

Encryption can always be used to add additional layer of security to your database. But adding additional security layers to protect data against the threats require addition expertise and reduce the system performance [18].

Zhang *et al.* [22] propose a hashing technique which can execute faster over encrypted data. This hashing technique uses hash values as well as a number called "confuse number" which can distinguish two similar values in different records.

Cipher index method is used to improve the performance and keeps the data confidentiality in the un-trusted servers [20]. In cipher index method, the hash values are used for the searching of data on the un-trusted servers without decrypting it on the server side. This method extracts the matched records to the hash values and decrypts it on the client side.

Hacigümüs *et al.* [7] proposed a technique, which can execute directly over encrypted data using a mapping function for the translation of queries from client side to server side. It preserves the data privacy and confidentiality.

Tang *et al.* [21] discus the problems in the bucketization of sensitive data. Since bucket values are generated through a hash function which might be duplicated. According to Tang et al the duplication of buckets values can generate the problem of values guessing. To stop the problem of guessing in bucketization they introduced a new method Split and Merge (SAM). In SAM method they divide and then remerge the buckets with new sequence in order to reduce the guessing of data because of bucketization.

In homomorphic encryption technique some of mathematical operation can be performed directly on encrypted data without decrypting it, but it needs a specialized encryption technique and cannot be generalized to all encryption technique [4, 19].

Feng and Danfeng [5] used Cryptograph Index Technology to improve efficiency of query over encrypted data. But it has "false positive hits" which degrades the query performance. False positive hits are the unwanted records retrieved during a search query.

Jinbiao [10] uses hybrid encryption technique for securing a database. According to his work, traditional encryption does not provide adequate security.

A query optimization technique is used in [8], in order to improve the performance of query. This technique is used in un-trusted server environment.

All these techniques works well in the case of full text search but fails or gives a partial solution to fuzzy match and range type queries. The novelty of proposed technique is that it works for all types of range and fuzzy match type queries and has no "false positive hits".

3. Research Problem and Hypothesis

The search process in an encrypted column of a table is performed by decrypting the entire column before the data retrieval which takes a significant amount of time and reduces the performance of SELECT query in relational databases. Although, there are some methods which could resolve the problem of performance, but these methods are not suitable for the range and fuzzy match queries.

Following is our hypothesis of research: "Storing the encrypted column in decrypted form along with encrypted key column in a different table increases the performance of the data retrieval process. By using a separate table for searching on encrypted data resolves the problem of range and fuzzy match queries".

4. Proposed Security Technique

The proposed technique suggests an additional table with actual_table to introduce security. The actual_table show in Table 1 contains the actual data and other named search_table show in Table 2 containing only that data on which the search query runs. The search_table is used for searching purpose only. The sensitive column in the actual_table is encrypted using strong encryption algorithm. A copy of the sensitive data column along with the primary key column is taken in the "search_table". In the search_table, the data column copied from the actual_table is kept in unencrypted form and the key column in encrypted form. Records in the search_table are shuffled and given another order than the order of actual_table. The encryption of the sensitive data column in the actual_table and the encryption of the primary key column in the search_table hide the relationship between the actual_table and search_table. The search_table is stored in the secure_schema. The Secure_Schema is that schema which can be accessed only by those users who are authorized to access the encrypted data.

In the proposed technique, the actual security is introduced by hiding the relationship between the actual_table and search_table and search_table is stored in secured schema. In addition to deceive automated schema generation tools, different column headings are used for the columns of search_table than that of the actual_ table.

Table 1. Actual_table.

Key	Emp_Name	Salary	Job Title
1	Bob	Encrypted	Manager
2	John	Encrypted	Assist manger
3	Alias	Encrypted	N/A admin
…	…	…..	…

Table 2. Search_table.

ABC (Key column of the Actual_Table)	XYZ (Salary column of Actual_Table)
Encrypted	12000
Encrypted	10000
Encrypted	9000
……	……

5. Proposed Search Methodology

Whenever an authorized user wants to search some records from the Actual_Table and the search condition is on the encrypted column, the search process will be performed over the Search_Table. The

search query returns keys to actual_table based on the search condition, and using those keys, records are returned to a user. This technique returns only those records which satisfy the user query and no extra records are returned. This improves performance and data confidentiality.

Whenever a query is issued on the encrypted data column, the proposed algorithm performs decryption at two points: first decryption is in the search_table to decrypt Keys and second decryption is in actual_table to decrypt actual column values. The experimental findings show that the proposed technique has a considerable performance gain over the existing techniques of querying over encrypted data. It is due to the fact that the proposed approach does not need to decrypt all values of entire encrypted column; rather it decrypts only those values which satisfy the user query. The proposed technique is very efficient whenever the amount of data retrieval is less than 45% of the total data. In a typical environment less amount of data is retrieved in the search query which suits well for the proposed technique. The searching mechanism of the algorithm can be described by the following example:

Reference to Table 1 suppose a user poses the following query over the actual_table.

```
Select emp_name, salary
From actual_table
Where Salary = 12000
```

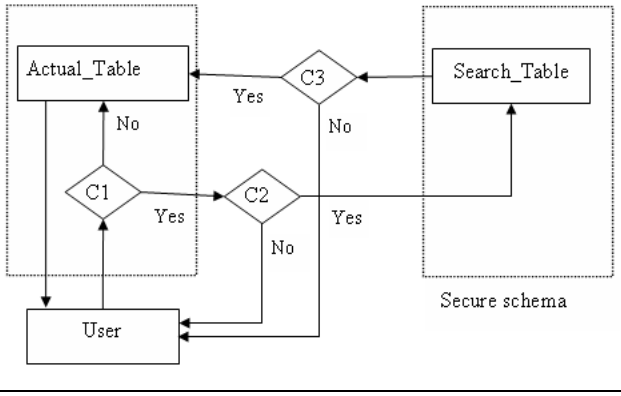
The algorithm interprets this query and transforms it as following:

```
Select emp_name, decryptfunction (salary)
From actual_table
WHERE key in (select decryptfunction (ABC)
From search_table where Xyz =12000)
```

Here in this query, the user wants to retrieve data of those records whose salary is 1200$. The proposed algorithm performs searching on the search_table, as the encrypted column is selected in WHERE clause of the query. The inner query performs the searching in the search_table for keys of those particular records which satisfies the user's search criteria. After that, keys are returned in the WHERE clause of the outer query. Now, the outer query uses those keys to retrieve exactly the records of the user interest. Here, the decryption function is called twice, once for the decryptions of keys in the inner query from search_table and the other for decryptions of the actual values in the outer query from the actual_table.

6. Architecture of The Proposed System

The architecture for the proposed security model, shown in Figure 1, consists of three main parts which are user interface, actual_table and search_table. search_table is stored in the "Secure Schema".



C1: Checks the condition that whether the query is on encrypted column?
C2: Checks the user validity to the Secure_schema.
C3: Are there any records?

Figure 1. Architecture of the proposed model.

A user issues a query against the actual_table. The WHERE clause of query is checked for encrypted nature. If WHERE clause of the query contains an unencrypted data column then data is retrieved from actual_table otherwise user is authenticated to the "Secure Schema" and retrieval of data is performed indirectly from actual_table viva search_table.

7. Algorithmic Outlines and Flowchart

The formal outlines of the proposed technique are given in the form of an algorithm shown in Figure 2:

```
1. [User query]
   User poses query
2. [Check the searching column]
   If (searching column is not encrypted)
            Goto step 3
   Else If (authorized user)
            Goto step 4
        Else
            Goto step 5
3. [Retrieval of data]
   Retrieve data from actual_table
            Goto step 5
4. [Passing control to the secure schema]
   [Check for query match]
   If (no query match) Then
        4.1. Display ("search is unsuccessful")
        4.2. Goto step 5
   Else
      [Retrieval Of Encrypted Keys]
        a. Retrieve the corresponding
           encrypted key(S)
        b. Decrypt the Key(S)
        c. Retrieve the data from actual_table
           based on key(S)
5. Exit
```

Figure 2. Outlines of the algorithm.

Figure 3 shows the flowchart for the algorithm shown in Figure 2, which explains each step of the algorithm.

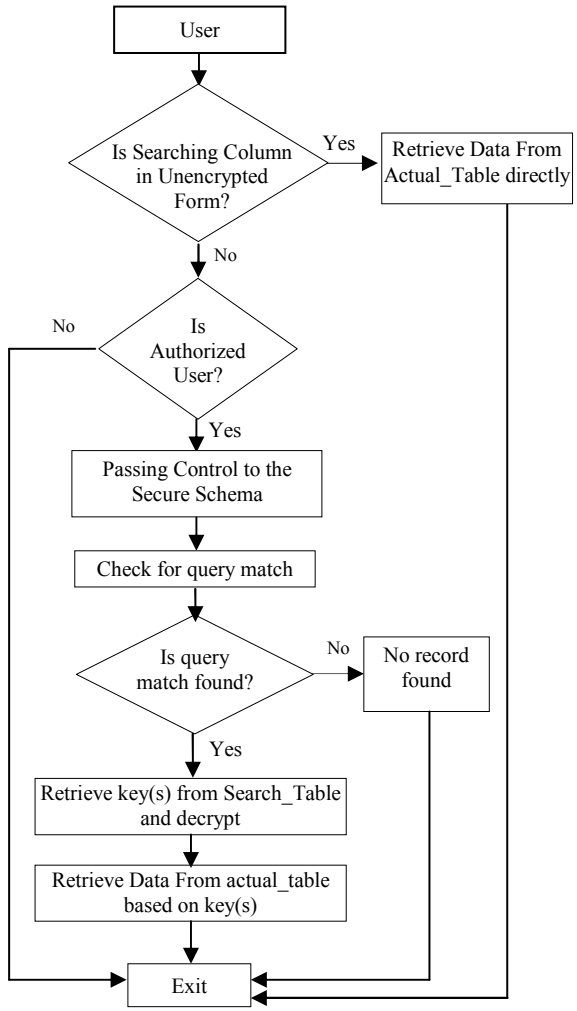


Figure 3. Flowchart of the proposed algorithm.

8. Testing and Results

All experiments are conducted on the TPC-E schema of Transaction Processing Performance Council (TPC) [17]. Two tables are used for testing purposes first one is the CUSTOMER table which is used for Fuzzy match type query and the second table is the CASH_TRANSACTION table which is used for Range type query. Data for testing purposes is generated by a software named "EGen" version 1.9.0, which is provided by TPC.

The experiment of fuzzy match query is performed on the CUSTOMER table of TPC-E schema. The table consists of 5000 records. The query condition is changed frequently to retrieve different amount of data.

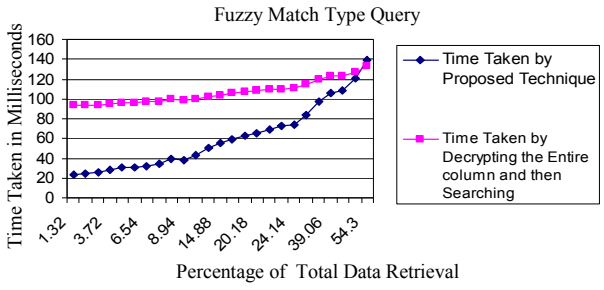


Figure 4. Analysis graph of the Fuzzy match query.

Consider Figure 4, the performance result of the proposed technique is good whenever the data retrieval is fewer. The figure depicts that the proposed technique works well whenever the ratio of data retrieval is less than 45% of the total data. For 5% of data retrieval an average performance is 72% improved. For 10% performance is 67% greater than the exiting. For 20% of data retrieval it 60% and for 30% data retrieval an average improvement in performance is 54%. For 40% and 45% of data retrieval the average performance improvement is 49% and 47%. The results are even better in case of lower percentage of data retrieval. In the typical environment less amount of data is retrieved, therefore the proposed algorithm is efficient for a typical environment.

The experiment on range type query was conducted on the TPC-E schema and data. The CASH_TRANSACTION table of the TPC-E schema was taken for testing purposes. The table contains 197485 records.

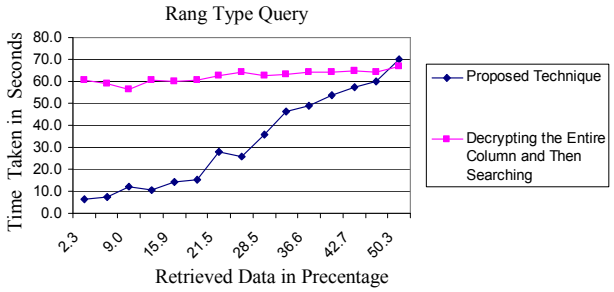


Figure 5. Analysis graph of range type queries.

It is obvious from Figure 5 that the proposed technique is also, efficient for range type queries whenever the retrieval of data is less than 45% of total data. For 5% of record retrieval, the proposed technique gives an improvement of 87% over the state of the art technique. For 10% it is 84%. For 20% of data retrieval performance improvement is 76% and for 30% of data retrieval the performance is 66% improved. For 40% the performance improvement is 58% and for 45% is 52%. Therefore, proposedtec hnique works well in a typical environment where less number of rows are retrieved in range queries.

The comparisons Figures 4 and 5 of proposed technique are done with "full column decryption technique". The proposed technique is compared with full column decryption technique because both of them support the exact match, fuzzy match and range type queries at the same time. There are some other techniques, for example the Order Preserving Technique [1] and Hashing technique [22]. The Order Preserving Technique supports range type queries, but does not support the fuzzy match queries. It also, works on a specialized encryption technique and cannot be generalized to all encryption techniques. Similarly, Hashing technique [22] supports fuzzy

match type queries but does not support the range type queries.

9. Conclusions

This work proposes an efficient algorithm for searching over encrypted data. Novelty of the proposed algorithm is that it efficiently eliminates the limitations of the existing techniques for fuzzy match and range queries and has no false positive hits. The algorithm is efficient for searching of data whenever the ratio of data retrieval is less than 45% of the total data.

References

[1] Agrawal R., Kiernan J., Srikant R., and Xu Y., "Order Preserving Encryption for Numeric Data," *in Proceedings of ACMSIGMOD*, USA, pp. 563-574, 2004.

[2] Arockia J., Rengansivagurunathan R., Ramasamy B., and Perumal E., "Hi-Tech Authentication for Palette Images Using Digital Signature and Data Hiding," *The International Arab Journal of Information Technology*, vol. 8, no. 2, pp. 117-123, 2011.

[3] Coper A*., Computer & Communication Security, Strategies for the 1990s*, McGrawHill, New York, 1989.

[4] Craig G., "Fully Homomorphic Encryption using Ideal Lattices," *in Proceedings of STOC ACM*, USA, pp. 169-178, 2009.

[5] Feng G. and Danfeng Z., "A Cryptograph Index Technology Based on Wrong Hit Expectation," *in Proceedings of IEEE International Conference on Electronic Computer Technology*, Shanghai, pp. 301-305, 2009.

[6] Fernandez B., *Database Security and Integrity*, Massachusetts Addison-Wesley, 1980.

[7] Hacigümüs H., Iyer R., Li C., and Mehrotra S., "Executing SQL over Encrypted Data in the Database-Service-Provider Model," *in Proceedings of ACM SIGMOD*, USA, pp. 216-227, 2002.

[8] Hacigumus H., Iyer R., and Mehrotra S., "Query Optimization in Encrypted Database System," *in Proceedings of Database Systems for Advanced Applications*, Berlin, pp. 43-55, 2005.

[9] Hwang S. and Yang P., "A New Dynamic Access Control Scheme Based on Subject-Object-List", *Data and Knowledge Engineering*, vol. 14, no. 1, pp. 45- 56, 1994.

[10] Jinbiao H., "Research on Database Security of E-Commerce Based on Hybrid Encryption," *in Proceedings of the International Symposium on Web Information Systems and Applications*, China, pp. 363-366, 2009.

[11] Kiely D., "Protect Sensitive Data Using Encryption in SQL Server 2005," *SQL Server Technical Article*, 2006.

[12] Lunt F., Denning E., Schell R., Heckman M., and Shockley R., "The Sea View Security Model," *IEEE Transaction on Software Engineering*, vol. 16 no. 6, pp. 593-607, 1990.

[13] National Bureau of Standards., "Data Encryption Standard," *Federal Information Processing Standards Publication*, Washington, pp. 45-52, 1977.

[14] Rivest R., Shamir A., and Adleman L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[15] Smid E. Branstad K., "The Data Encryption Standard: Past and Future," *IEEE Transaction*, vol. 76, no. 5, pp. 550-559, 1988.

[16] Stachour D. and Thuraisingham B., "Design of LDV: A Multilevel Secure Relational Database Management System," *IEEE Transaction on Knowledge and Data Engineering*, vol. 2, no. 2, pp. 190-209, 1990.

[17] TPC Benchmark Specification, available at http://www.tpc.org/, last visited 2009.

[18] Ulf M. and Protegrity C., "A Practical Implementation of Transparent Encryption and Separation of Duties in Enterprise Databases," *in Proceedings of the 7th IEEE International Conference on E-Commerce Technology*, USA, pp. 559 - 565, 2005.

[19] Van M., Gentry C., Halevi S., and Vaikuntanathan V., "Fully Homomorphic Encryption over the Integers," *in Proceedings of Advances in Cryptology*, Berlin, pp. 24-43, 2010.

[20] Wang F., Wang W., and Shi B., "Storage and Query over Encrypted Character and Numerical Data in Database," *in Proceedings of the 5th International Conference on Computer and Information Technology*, Washington, pp. 77-81, 2005.

[21] Zhang L. Tang Y., "Adaptive Bucket Formation in Encrypted Databases," *in Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service on e-Technology, e-Commerce and e-Service*, USA, pp. 116-119, 2005.

[22] Zhang Y., Li W., and Niu X., "A Secure Cipher Index Over Encrypted Character Data in Database," *in Proceedings of the 7th International Conference on Machine Learning and Cybernetics*, Kunming, pp. 1111-1116, 2008.

Shaukat Ali is a PhD scholar in the Department of Computer Science, University of Peshawar, Pakistan. He got his Msc degree in computer science from the Same University of Peshawar. Apart from this, He is also, working as a lecturer in the Department of Computer Science, Islamia College Peshawar, Pakistan. His area of interest is database security.

Huma Javed is a senior lecturer in the Computer Science Department, University of Peshawar, Pakistan. She holds PhD in software engineering from School of Computing and Mathematical Sciences, Liverpool John Moores University, UK. Her areas of interest are middleware, sensor networks, embedded systems, pervasive computing and wireless communications. She has seventeen year experience at undergraduate and graduate levels in Pakistan and also, taught in Liverpool John Moores University at undergraduate level.

Azhar Rauf is working as an assistant professor at the Department of Computer Science, University of Peshawar, Pakistan. He received a PhD degree in computer sciences from Colorado Technical University, USA. His areas of interests include database security, data warehousing, data mining, and database watermarking. He worked as a data modeler.