Secure Multiple-Query Processing in Databases

Neha Chouhan
Govt. Holkar Science College, Indore (M.P.)
cneha7@gmail.com

Abstract: we know that traditional query processing and optimization concentrate on processing and optimization of the execution of each individual query. Multiple-Query Processing (MQP) optimizes a set of queries together by executing the common operations once in order to save query execution time and processing cost. To process multiple queries once in order to minimize the query execution time and its transmission, we identify common sub-expression and evaluate them rather than processing them individually. Thus, ***Multiple***-Query Processing typically offers significant improvement to the performance of a system. In this paper we propose an optimal security technique to transmit the result of the multiple queries once so that one user cannot access or view the result of other user. This cipher algorithm is one of the strongest, simplest and fastest encryption algorithms.

Keywords: MQP, CSE, Adhoc Query, Encryption, Decryption, Cipher Algorithm.

1. Introduction

As we know that query processing play an important role in any type of databases. The techniques of database queries are challenging today's database systems and promoting their evolvement. The research and development of processing the queries over a variety of databases are receiving more attention towards them. Query processing refers to the range of activities involved in extracting data from a database. All database systems must be able to produce result of the requests given by the user that is process the queries. Getting the desired information from a database system in a predictable and reliable manner is the scientific art of Query Processing. Obtaining these result back in a timely manner deals with the technique called Query Optimization [1, 2].

As we know that traditional query processing and optimization concentrate on processing and optimization of the execution of each individual query. More recently, it has been observed that by processing a sequence of multiple queries, some additional high-level optimizations can be performed. Multiple-Query Processing (MQP) optimizes a set of queries together by executing the common operations once in order to save query execution time and processing cost. Multiple-Query Processing typically offers significant improvement to the performance of a system [3]. Processing the group of queries simultaneously having some common expression rather processing them individually is known as Multiple Query Processing. To process the multiple queries, we have to choose a group of queries and process all these queries simultaneously. Therefore, by applying multiple-query optimization, we can minimize the total execution time by performing common tasks only once [4, 5]. In this paper , we focuses on an optimal method to process the multiple queries by executing their common sub-expressions and to produce the result of these multiple queries once in order to minimize the query execution time, its processing cost and also its transmission cost and thus improves the overall performance of a system. After studying the method to process multiple queries, we will propose a security technique to transmit the result of this multiple queries once, so that one user cannot access or view the result of some other user.

2. Related Work

Many researchers proposed an idea of processing a set of queries to reduce the total cost of executing a set of queries on the same database. Many queries may involve join operation and part of one query may subsume part of another query. Under these circumstances, grouping a set of queries together and processing them as a unit is clearly beneficial [6]. In multiple-query processing, a sub-expressions that appears in more than one query is called a common sub-expression (CSE). A common sub-expression needs to be evaluated once only to produce a temporary result that can then be used to evaluate all the queries containing the common sub-expression [7]. This technique is useful especially in subsumption cases (described in later section) and overlapping queries with common parts [8]. Multiple-query optimizers detect common operations and execute them only once, to optimize processing of a set of queries. Multiple-Query Optimization (MQO) is the on-line processing of Adhoc queries. An Ad-hoc queries are commonly utilized queries in traditional DBMS. An Ad-Hoc Query is a query that cannot be determined prior to the moment the query is issued. It is created in order to get information when need arises. So therefore, such a query is one that might suit a

Received 2012/11/25 Reviewed 2013/2/27

situation which is only there for the moment and later on will become irrelevant. This type of query explicitly mentions the required information in the query statement, and does not involve any context awareness information. Thus, the query result is only based on the actual query itself [9]. For example a query of this type could be:"University student wants to retrieve his/her academic record or personal details".

To illustrate the idea of multiple-query processing [10], we consider an example. The five queries Q1, Q2, Q3, Q4 and Q5 are sent simultaneously by different users U1, U2, U3, U4 and U5 respectively. The server processes the queries and sends the answers of each query individually to their respective user shown in Figure .1. The answers to Q1 and Q5 are identical, the answer to Q2 overlaps that of Q1, the answer to Q2 subsumes that of Q3 (i.e., answers for Q3 can be obtained by doing a simple selection operation on the result of Q2), and the answer to Q4 does not have any data in common with any other query answers.

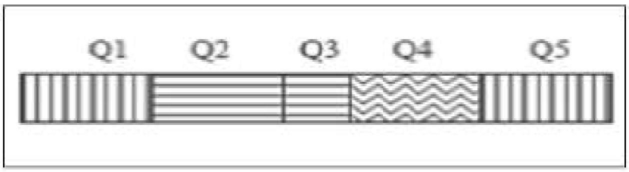


Figure 1 Result of Each Individual Query

Using MQP techniques, the resulting view is shown in Figure .2. Since Q1 and Q5 are identical, the results are transmitting just once. The overlapping portion of Q1 and Q2 is transmitting just once. The result of Q3 is obtained from the result of Q2 and Q4 is transmitted.

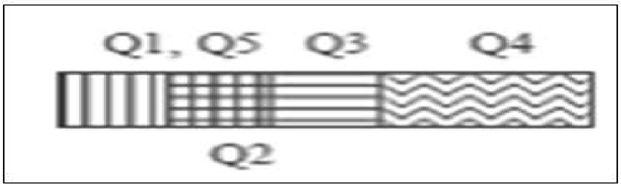


Figure 2 Result of Multiple Query Processing

2.1.Steps Followed in Multiple Query Processing

To perform Multiple Query Processing, we have to choose a group of queries and process them simultaneously. Each query requires a specific subset of tuples from each table in the group [18]. Here we are identifying the set of tuples required by each query in a group from each table (in the group's table list) in that group. Then we are reducing each table (in the group's table list) to one that contains tuples that are required by at least one query in that group. Then the tables so computed are joined to compute the superset. While joining these tables we eliminate spurious tuples (not a part of any query's result in the group) that will arise in the process. After the completion of optimization we are left with an efficiently computed superset Σ of results (ρ_i , i = 1..N) for each query (q_i, i = 1..N) such that ρ_i ε Σ for all i = 1..N and if a tuple t ε Σ then there exists a query q_i such that t ε ρ_i . Then the superset is transmitted.

The system performs its operation in the following steps:

1. Collection of queries.
2. Decomposition into groups.
3. Computation of supersets.
4. User specific encryption of the superset.
5. Transmission of encrypted superset.
6. Extraction of the individual results.

1) Collection of queries

The optimizer is a multithreaded program. In this step the main thread of the program waits for a specific time window and collects all the queries arriving at the server. Then a new thread is created to perform query optimization while the main thread goes on collecting queries for another time window. This process goes on. The optimal span of the time window depends upon several system parameters: the size and capacity of the system, the load on the system, query arrival pattern etc.

2) Decomposition into groups

Different queries have different structures. Some queries are best evaluated together while others are not. Based on structural similarity the queries collected in a specific time window are decomposed into groups such that the queries in a group have maximum possible commonality. The criteria used for group decomposition is the equality of the projection list and the table list so that the result of each query in a group has the same domain. For example, consider two databases Employee and Branch:

Employee:- EmpId, EmpName, Salary

Branch:- EmpId, Bname, Bcity, Work_Exp

Now, let us consider the following incoming queries in a specific time window.

Q6: select EmpId from Employee, Branch where Salary between 1000 and 16000 and Work_Exp>=10 and Employee.EmpId=Branch.EmpId;

Q7: select EmpId, Bname from Branch where Bcity =

‘Brooklyn’;

Q8: select EmpId from Employee, Branch where Salary between 2500 and 17000 and Work_Exp>=12 and Employee.EmpId=Branch.EmpId;

Q9: select EmpId, Bname from Branch where Bcity = ‘Brooklyn’;

The set of queries collected S = {Q6, Q7, Q8, Q9} is partitioned into disjoined subsets G1and G2. The projection list for Q6 = {EmpId} and the projection list for Q8 = {EmpId}. The table list for Q1 = {Employee, Branch}, table list for Q8 = {Employee, Branch}. From this we see that the queries Q6 and Q8 have the identical projection list and table list. Similarly the queries Q7 and Q9 have the same projection list and table list. So, there will be two groups G1 = {Q6, Q8} and G2 = {Q7, Q9}. The groups so formed are processed separately.

There will be queries that cannot be grouped with any other queries. For example, if only the queries Q6, Q7 and Q8 had arrived in this time window then Q7 could not be grouped with any other query. There can be multiple such queries. These queries are put in a separate group such that queries in that group are separately evaluated and results are separately dispatched. If in a group there is not much commonality then this procedure will lead to extra overhead and in that case we will get better results by processing and dispatching them separately.

3) Superset computation

For each group the optimizer computes the superset of the individual results of the queries in that group. The motivation behind superset computation is taking care of common data. If the results of multiple queries have common data then this common data is present in the superset only once. So, if we broadcast this superset instead of sending the results individually we will have to transmit less data and thus reduces the transmission cost.

Let us consider some example query Q10, Q11, Q12, Q13 having some common data. The results of each individual query are given below:

Table 1 Query Result

Query	Resulting Tuples from each Query				
Q10	T1	T4	T5	T6	T8
Q11	T2	T4	T6	T7	
Q12	T1	T2	T3	T5	T6
Q13	T1	T3	T6	T7	T8

The above table shows tuples comprising the result of queries Q10, Q11, Q12 and Q13. If the query results were dispatched separately, 19 tuples would have to be transmitted. But we note that there are only 8 distinct tuples. If we broadcast the superset Σ = {T1, T2, T3, T4, T5, T6, T7, T8} then only 8 tuples have to be transmitted. Therefore it saves transmission cost as well as processing cost of these multiple queries.

4) User specific encryption of the superset

We can see that in the above procedure the superset broadcast contains tuples that belong to multiple queries. Hence some tuples in the superset are not part of some queries. So there is a chance that a malicious user will view the tuples that are not part of his result. To solve this problem we have to encrypt the tuples belonging to a specific user by a randomly created key before transmitting the superset. We will describe the whole encryption technique in next section.

In this procedure, we keep track of which broadcast tuples belong to which query. The tuples in the superset are ordered in such a way that tuples belonging to the same query are as contiguous as possible (Table 2).

Table 2 Superset with Contiguous Tuples

Tuples
T8
T1
T5
T6
T2
T7
T3
T4

As a result, the superset to be transmitted is consists of 4 blocks of tuples BLK1, BLK2, BLK3 and BLK4. BLK1 is required by user U1, BLK2 by U1 and U2, BLK3 by U2 and U3, BLK4 by U3. Our scheme is illustrated below:

1). BLK1 = {T8} required by U1

2). BLK2 = {T1, T5, T6} required by U1 and U2

3). BLK3 = {T2, T7} required by U2 and U3

4). BLK4 = {T3, T4} required by U3

5) Transmission of encrypted superset

The superset so computed is encrypted using public key as well as using private key which provide strong encryption view of a result. After performing encryption of the superset, the server transmitted the superset result on the transmitting channel. This superset result for users is transmitted on same broadcast channel in order to save the transmission cost of result.

6) Extraction of individual results

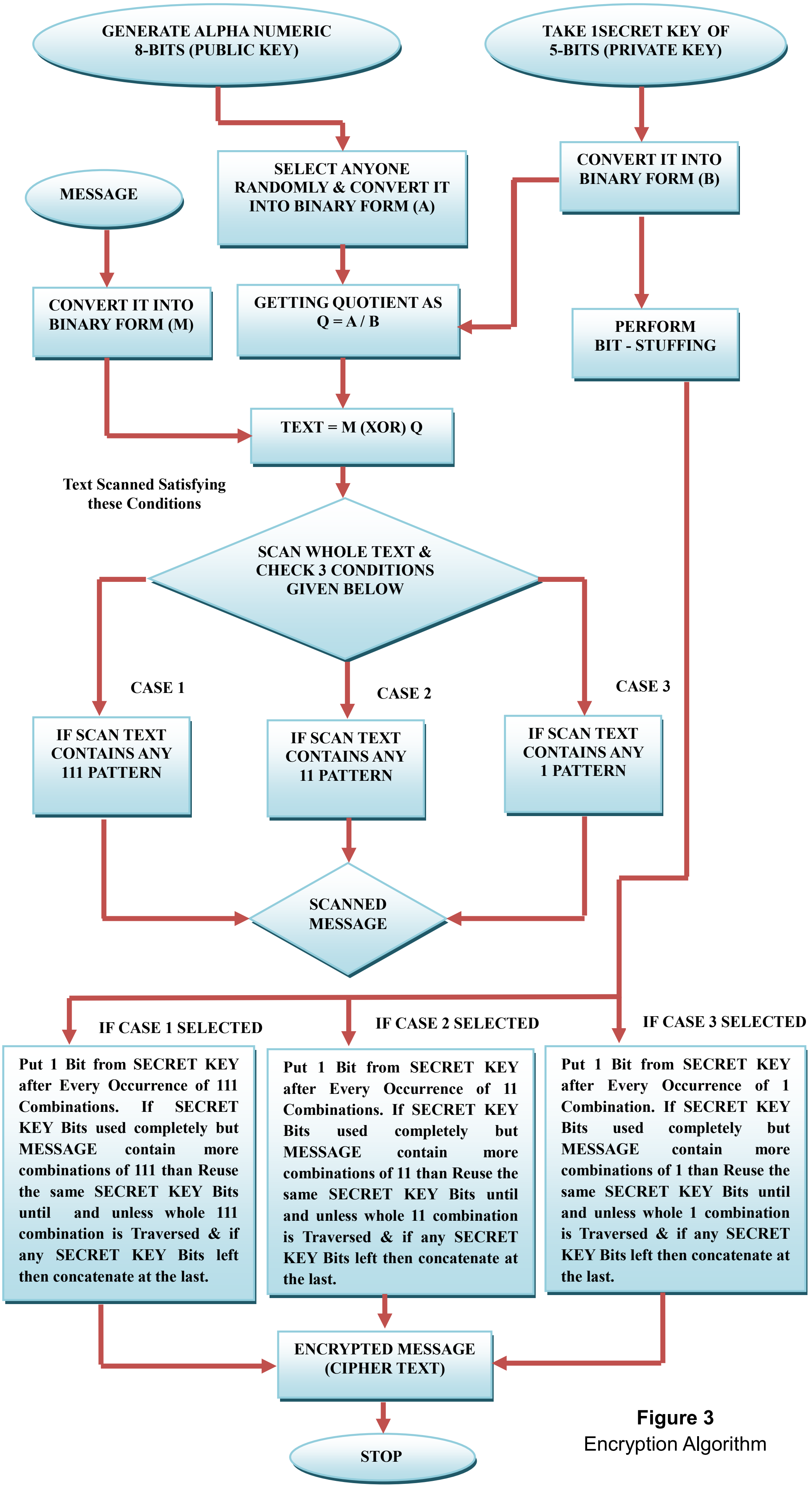
The superset transmitted by server is available to each individual user and then each individual user decrypts the information using their corresponding keys that is using their own secret key to view their result from the superset and thus receive their own tuples. Thus all users can only access or see their own result and cannot access or view the result of other user.

3. Proposed cipher algorithm for secure transmission

In this section, we are going to propose a cipher algorithm which can be applied in multiple query processing for secure transmission of the result so that one user cannot view the result of some other user. Cryptography is the technique that is used to ensure the safety of communication over the network in most of the computer communication systems. The strength of a cryptographic algorithm is based on the difficulty of cryptanalysis imposed over system. Several algorithms are available for the cryptography like DES, RSA, IDEA etc. All these algorithms are used for the purpose of data security [11, 12, 13, 14]. In this work, we use asymmetric key technique in which two keys are used, one key that is public key and other key that is secret key of user. In our work, we apply the concept of RAC i.e. Randomized Alphanumeric Characters and bit stuffing to produce cipher text in which we use some existing algorithm that uses the strength of one algorithm to compensate the weakness of other [15, 16].

In this proposed algorithm we are using the concept of XOR operation and bit stuffing in a specific way. We used asymmetric private key and the public key which increases the efficiency level of encryption algorithm. Here one 8-bit public key is randomly selected among several generated Alphanumeric Characters and one 5-bit private key.

The steps involve in our proposed Cipher technique is described in flowchart which is given in Figure 3 (Encryption Technique) and in Figure 4 (Decryption Technique) which is shown below:-



Put 1 Bit from SECRET KEY after Every Occurrence of 111 Combinations. If SECRET KEY Bits used completely but MESSAGE contain more combinations of 111 than Reuse the same SECRET KEY Bits until and unless whole 111 combination is Traversed & if any SECRET KEY Bits left then concatenate at the last.

Put 1 Bit from SECRET KEY after Every Occurrence of 11 Combinations. If SECRET KEY Bits used completely but MESSAGE contain more combinations of 11 than Reuse the same SECRET KEY Bits until and unless whole 11 combination is Traversed & if any SECRET KEY Bits left then concatenate at the last.

Put 1 Bit from SECRET KEY after Every Occurrence of 1 Combination. If SECRET KEY Bits used completely but MESSAGE contain more combinations of 1 than Reuse the same SECRET KEY Bits until and unless whole 1 combination is Traversed & if any SECRET KEY Bits left then concatenate at the last.

Figure 3
Encryption Algorithm

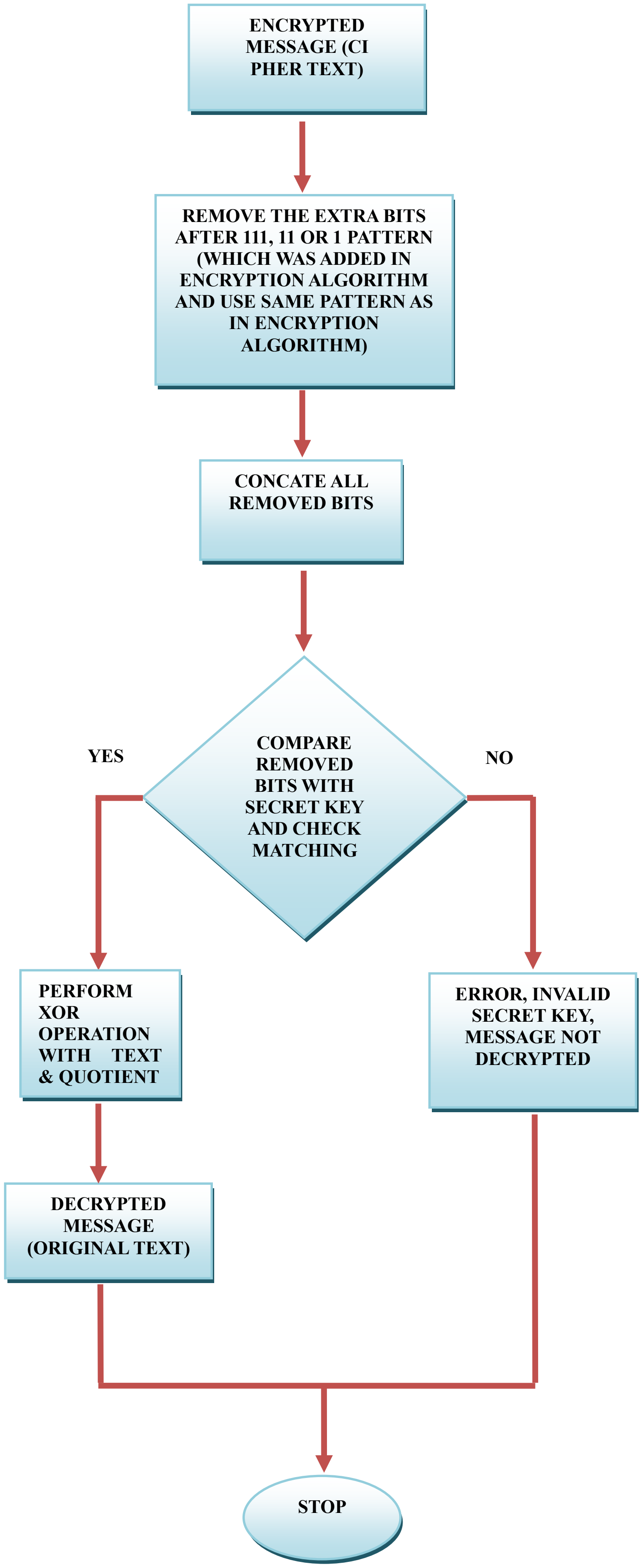

ENCRYPTED MESSAGE (CI PHER TEXT)
REMOVE THE EXTRA BITS AFTER 111, 11 OR 1 PATTERN (WHICH WAS ADDED IN ENCRYPTION ALGORITHM AND USE SAME PATTERN AS IN ENCRYPTION ALGORITHM)
CONCATE ALL REMOVED BITS
COMPARE REMOVED BITS WITH SECRET KEY AND CHECK MATCHING
YES
NO
PERFORM XOR OPERATION WITH TEXT & QUOTIENT
ERROR, INVALID SECRET KEY, MESSAGE NOT DECRYPTED
DECRYPTED MESSAGE (ORIGINAL TEXT)
STOP

Figure 4 Decryption Algorithm

From the above flow chart, the cipher algorithm is described as below:

Encryption Algorithm

1. Generate N number of 8-bit Alphanumeric Characters randomly.

2. Randomly select any one among them, denoted as ‘A’ (PUBLIC KEY), and 5-bit SECRET KEY assigned as ‘B’.

3. Convert ‘A ’ and ‘B’ in binary format.

4. Compute the quotient of these two Q = A / B.

5. Perform XOR operation with Message ‘M’ and quotient ‘Q’,

TEXT = M XOR Q.

6. Scanned TEXT and perform the Bit-Stuffing as per flowchart given below.

7. In this way we get the encrypted message as CIPHERTEXT.

To decrypt the Ciphertext that is encrypted message, we use Decryption Algorithm which is as follow:-

Decryption Algorithm

1. Remove the extra bit from CIPHERTEXT.

2. Perform the XOR operation with ‘TEXT’ and quotient ‘Q’.

3. In this way we get the original message ‘M ’.

A. Comprehensive Example

The steps used in above algorithms will be well understood by suitable example. The above Encryption Algorithm and Decryption Algorithm working as shown below:-

Encryption:

1. Input: Public Key as ‘A’ = SCSE04, Secret Key as ‘B’ = VIT and Message as ‘M’ = 11MSC.
2. Conversion to binary format:

Message as M =11MSC
[1 = 0000001, 1= 0000001, M = 1001101, S = 1010011, C = 1000011]

M = 00000010000001100110110100111000011 (11MSC)

Public key A = SCSE04

[S = 1010011, C = 1000011, S = 1010011, E = 1000101, 0 = 0000000, 4= 0000100]

A= 1010011100001110100111000101000000000000100
(SCSE04)
Secret key B = VIT

[V = 1010110, I = 1001001, T = 1010100]

B = 101011010010011010100 (VIT)

3. Quotient ‘Q’;
Q = A / B
Q = 11110110111111011 0001

4. TEXT = M XOR Q.
TEXT = 00000010000000101101101101100111 0010

5. CIPHERTEXT = TEXT with Bit-stuffing using SECRET KEY, (Using Case 2)
CIPHERTEXT=100000010111011001110110001111001010100 10011010100

[Encrypted Message (CIPHERTEXT) =
10000001011101100111011000111100101010010011010100]
Now the above encrypted message is decrypted by user as:

Decryption:

It is just reverse of encryption.

1. Input as CIPHERTEXT =
10000001011101100111011000111100101010010011010100
Using Case 2: Remove the bit after 11 patterns (because 11 pattern is used in encryption algorithm for bit-stuffing) and concatenate.

Removed bits=101011010010011010100

Comparison of removed bit with SECRET KEY bits, here it matched, so go to next phase. (if removed bits doesn’t match with secret key then we do not take any step further and display error message to show that secret key is invalid).

2. Performing XOR operation with TEXT and Quotient. In this manner we get the original message.
Message M = 00000010000000110011011010011100011
(11MSC)

So in this way we can encrypt and decrypt the message and provide secure view of result to the user. Hence our cipher algorithm is a combination of RSA and IDEA algorithm by removing the weaknesses of these two algorithms as well as it uses both the keys i.e. public key and private key to encrypt the message and thus provides a confidence to the receiver that the data has been encrypted by one who has the possession of that private key and therefore improving the overall performance of a database system.

5. Conclusion and Future Work

Query processing in database is one of the latest research areas. It has taken up the challenge to develop techniques that can help people to retrieve their information efficiently within a minimum time period. In order to minimize processing cost of query as well as to improve the response time, we process the group of queries simultaneously rather than processing each query individually. Multiple-Query processing in database has contributed to the existing system for the query operations significantly. It is essential for large information retrieval system where time factor is important. As the database is growing gradually the retrieval time is also increased in exponential order. The presented approach in big databases that has high information volume and their aim is retrieving information in less time as well as with security is suitable and efficient. This approach makes processing cost less as well as improves the response time and also makes secure transmission of request to the user. In other words, Multiple-query processing in database management systems provides competitive as well as better performance than single query processing.

In this work, we also proposed a cipher algorithm that can be applied to transmit the result of multiple queries once in order to provide secure view of result to the user so that one user cannot view or access the result of some other user. The major achievement of this work is to propose an efficient algorithm for security in the context of multiple-query processing in database. For future work, we will try to implement the above Cipher Algorithm so that we can extend this work in future also. Our proposed cipher algorithm is one of the strongest, simplest and fastest encryption algorithms. It uses both the keys i.e. public key and private key to encrypt the message and thus provides a confidence to the receiver that the data has been encrypted by one who has the possession of that private key and therefore improving the overall performance of a database system.

Acknowledgment

We would like to pay my sincere thanks to Dr. P. K. Sharma, Head of Computer Science Department and Dr. V. Raich, Professor in Govt. Holkar Science College, Indore for providing environment, motivation and guidance towards this research work.

References

[1] Li Yan and Zongmin Ma, "Advance Database Query Systems: Techniques, Applications and Technologies", Copyright © 2011 by IGI Global.

[2] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", Sixth Edition.

[3] Rajeswari Malladi and Karen C. Davis, "Applying Multiple Query Optimization in Mobile Database", proceedings of the 36th Hawaii International Conference on System Sciences – 2003.

[4] J. R. Alsabbagh and V. V. Raghavan, "A Framework for Multiple-Query Optimization", © 1992 IEEE.

[5] Murat Ali Bayir, Ismail H. Toroslu, and Ahmet Cosar, "Genetic Algorithm for the Multiple-Query Optimization Problem", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 37, No. 1, January 2007.

[6] U. S. Chakravarthy and J. Minker, "Multiple Query Processing in Deductive Databases using Query Graphs", Proceedings of the Twelfth International Conference on Very Large Data Bases, August 1986.

[7] Jamal R. Alsabbagh and Vijay V. Raghavan, "Analysis of Common Subexpression Exploitation Models in Multiple-Query Processing", © 1994 IEEE.

[8] Ali-Asghar Safaeei, Mehran Kamali, Mostafa S. Haghjoo and Kamyar Izadi, "Caching Intermediate Results for Multiple-Query Optimization", © 2007 IEEE.

[9] Samidha Dwivedi Sharma and Dr. R. S. Kasana, "Mobile Database System: Role of Mobility on the Query Processing", International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010.

[10] D. Saha and N. Chowdhury, "A Method for Secure Query Processing in Mobile Databases", Engineering Letters, 14:1, Advance online publication: 12 February 2007.

[11] "What is Encryption", what is encryption- Feature-www.techworld.com/encryption.

[12] Encryption in SAS ®, Copyright © 2009, SAS Institute Inc., Cary, NC, USA, ISBN 978-1-59994-865-2.

[13] Shaligram Prajapat,"A Novel Approach For Information Security With Automatic Variable Key Using Fibonacci Q-Matrix",International Journal of Computer & Communication Technology (IJCCT), Vol-3, Iss-3, 2012.

[14] Shaligram Prajapat et. al.," Implementation of AVQ-Using Fibonnacci-Q Matix", ICICIS-2012.

[15] Sudhakar Kumar Singh, "Design and Implementation of Cipher Algorithm using Randomized Alphanumeric Characters", International Journal of Scientific and Research Publications, Vol. 2, Issue 6, ISSN 2250-3153, June 2012.

[16]http://www.cccure.org/Documents/Cryptography/cisspallinone.pdf.

Biography:

Neha Chouhan received his B.Sc. degree in Electronics from P. M. B Gujarati Science College, Devi Ahilya University of Indore (M. P.) in 2006. She completed her Master's degree in Computer Science from School of Computer Science, Devi Ahilya University of Indore (M. P.) in 2008. After completion of her master's degree, she joined Govt. Holkar Science College as a faculty and having a teaching experience of three years. She received her M. Phil degree in Computer Science from Govt. Holkar Science College, Devi Ahilya University of Indore (M. P.) in 2012. Her research interests are in Database, Query processing in database, Security and Encryption/Decryption technique.