DBCrypto: A Database Encryption System using Query Level Approach

Asavari Deshpande
Assistant Professor
Computer Department
MIT COE Pune
Maharashtra India

Anup Patil
UG Student
Information Technology
MIT Pune
Maharashtra India

Saurabh Joshi
UG Student
Information Technology
MIT Pune
Maharashtra India

Suraj Bothara
UG Student
Information Technology
MIT Pune
Maharashtra India

ABSTRACT
Online applications are vulnerable to theft of sensitive information because adversaries can exploit software bugs to gain access to private data and because curious or malicious administrators may capture and leak data. *DBCrypto* provides practical and provable confidentiality to the database by using queries. The proposed system is a middleware between user application and DBMS. The encrypted data is stored in tables by preserving its format and decrypted data can accessible to the user through regular queries. The various encryption and decryption algorithms are implemented at Query Level to secure the data from malicious administrator or from information leak.

Categories and Subject Descriptors
Data Security

General Terms
Performance, Design, Experimentation, Security

Keywords
Database, Security, Data Security

1. INTRODUCTION
Data [1] is representation of qualitative or quantitative variables belonging to a set of items. Data is organized in terms of rows and columns where items are organized as rows and values are organized as columns. A database is a collection of data stored in digital form. The database system is defined as a combination of data and their supporting data structure with Data Base Management System (DBMS).
DBMS based on relational model is called as Relational Database Management System (RDBMS) [2]. Relational model is based on the theory of sets and relations of mathematics. It represents the data in the form of table. A table is a two dimensional array containing rows and columns. [1].The Structured Query Language (SQL) is a programming language to access database [1]. There are various types of queries like insert, delete, update, select etc. to perform operations on database stored in the database systems.
Security [2] is concern with protection against the danger, damage, loss and crime. In IT field the types of security are Application Security, Computing Security, Data Security, Information Security, and Network Security.
Database security [2] is an important and rapidly growing issue in now days. Huge amount of digital data is shared on internet this increases threat of data security. The Defense Information Systems Agency of the US Department of Defense [3], in its *Database Security Technical Implementation Guide*, states that database security should provide controlled protected access to the contents of a database as well as preserve the integrity, consistency and overall quality of the data.
In this paper Section 2 is about related work which includes many research papers related to data security where as section 3 is indicating basics of DBCrypto which refers to initial phase and control flow of our application. Implementation details along with pseudo code for various queries related to DBCrypto is discussed in section 4 where as in section 5 discussion about experimentation and result for various queries. Finally section 7 is focusing on conclusion and future work of the paper.

2. RELATED WORK
The concept of data security is discussed by various researchers to avoid attacks, information leakage etc. Early 1980-1990 the Us Army air force [3] proposed technique of database security using filter guards using Secret Key to avoid Trojan Horse attack. But after 90's many researchers [4] [5] worked to overcome attacks caused by database leakage.
The Popa et.al proposed [4] onion model for database encryption by using various approaches such as RND, DET, HOM, OPE and [5] is giving actual implementation of the concept proposed by [4] using authentication and proxy server. The main disadvantage discussed by author is implementation of onion layer is difficult and time consuming for execution as well. The encryption algorithm used in mainly considering levels of security. In our approach we are concentrating on the simplified data security model which is easy to implement and give secure data by considering time constraint.
The [6] **[**7] proposed data security on cloud by portioning database and applying randomization, k-anonymization and

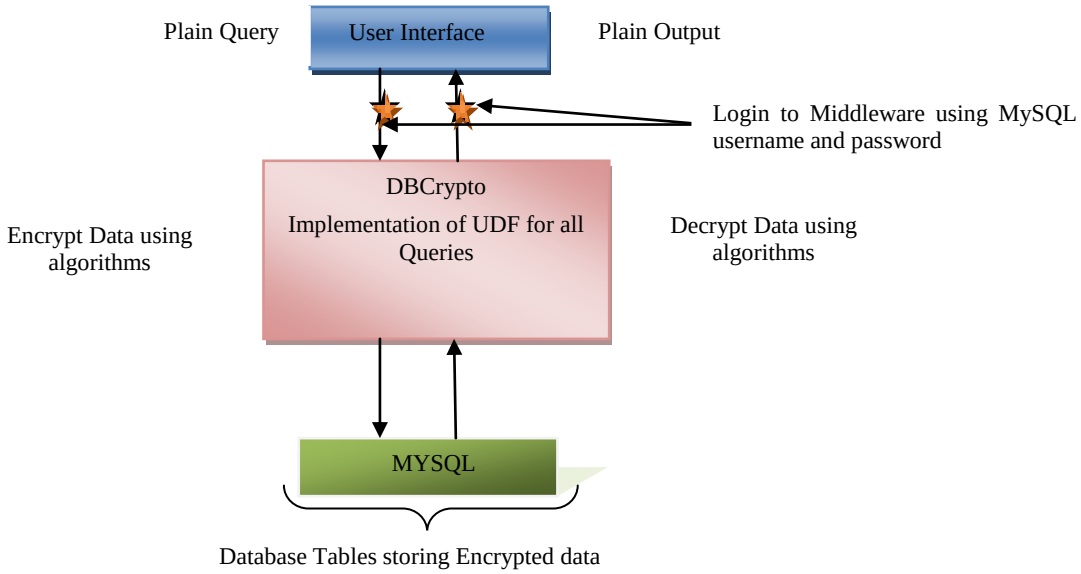


Figure 1. DBCrypto Flow Diagram

1 diversity and distributed data preservation whereas [8] [9] [10] are discussing various possible attacks and solutions on those attacks. Here we are considering attacks on our proposed system as future work.

We have considered various security algorithms such as AES, RND, Symmetric Key, and Asymmetric Key [2] to perform encryption and decryption operation. Within all of above algorithms Advanced Encryption Standards (AES) is most secure algorithm. It stores all encrypted data is terms of array of bytes. As per our requirement the encrypted data must preserve its original format l.e after encryption integer must be stored as an integer. As AES storing all encrypted values in byte format it is not applicable to DBCrypto implantation requirement. The [11][12][13][14][15] are discussing about format preserving encryption (FPE) but they considering methods all in terms of mathematical conventions which are more complex to understand and implement. We have proposed our own substitution tables for numeric values and string characters including symbols by using the substitution method called "Vigenere Ciphers" [2] which are discussed in section 4.

3. BASICS OF DBCRYPTO

The DBCrypto is concentrating on the data security of database tables. The data stored in the tables is in encrypted format by preserving related data types. DBCrypto presents a practical relational DBMS that provides provable privacy guarantees without having to trust the DBMS server or the DBAs who maintain and tune the DBMS. In DBCrypto, unmodified DBMS servers store all data in an encrypted format, and execute SQL queries over encrypted data without having access to the decryption keys.

DBCrypto is a system that provides practical and provable confidentiality in the face of these attacks for applications backed by SQL databases.

The DBCrypto addresses two threats. The first threat is a curious database administrator (DBA) who tries to learn private data (e.g., health records, financial statements, personal information) by snooping on the DBMS server; here, DBCrypto prevents the DBA from learning private data. The second threat is an adversary that gains complete control of application and DBMS servers. DBCrypto works by rewriting SQL queries, storing encrypted data in regular tables, and using an SQL user-defined function (UDF) to perform cryptographic operations shown by Figure 1.

4. IMPLEMENTATION DETAILS

From implementation point of view of the problem statement we have come across various implementation strategies and after experimentation on those strategies we come across various advantages and disadvantages of those approaches. All strategies along with problem statement are discussed in next subsection.

4.1 Problem Statement

Online applications are vulnerable to theft of sensitive information because adversaries can exploit software bugs to gain access to private data, and because curious or malicious administrators may capture and leak data. The system should provide practical and provable congeniality in the face of these attacks for applications backed by SQL databases.

4.2 Proposed Solution

DBCrypto only empowers the server to execute queries that the users requested, and achieves maximum privacy given the mix of queries issued by the users. The database server fully evaluates queries on encrypted data and sends the result back to the application for final decryption; client machines do not perform any query processing and client-side applications run without changing.

4.3 Implementation Strategies

By considering above problem statement we have considered our own strategies such as, first strategy is to design the system either by writing PLSQL Block [16] or embedding Java code of the encryption algorithms in to the MySQL as MySQL is an

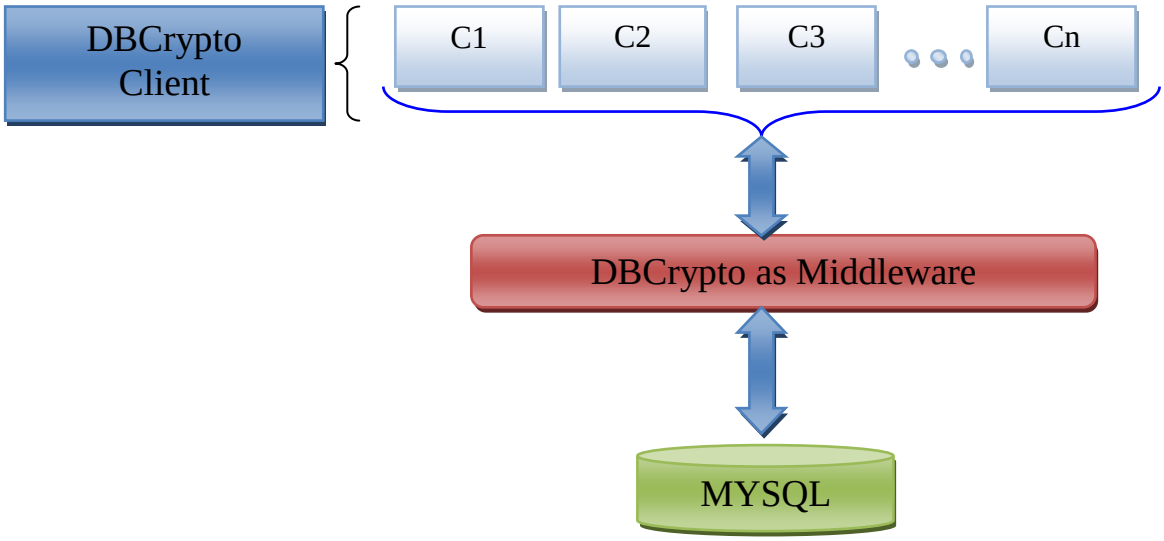


Figure 2. Architecture of DB Crypto

Open source database platform. As source code of MySQL can't be altered, the approach was not fulfilling our goal.
Second strategy is to implement Plug-in for MYSQL [16].To activate MYSQL plug-ins, we have started with a modified query which eventually led to deadlock. Deadlock in the manner, when we want to activate the plug-in as a service in MySQL itself, we have to fire a query with its parameters. The structure of query is predefined which we cannot change according to our convenience.
Third strategy is finalized for our implementation where DBCrypto Module is acting as middleware between Client and MySQL Server. The architecture of DBCrypto is shown by Figure 2.
As shown in the diagram, one or more clients are connected to the Application Server. When a query is fired by client, it is transferred to DBCrypto application server to parse query, apply required security algorithms and generate modified query along with secure data. Then the modified query is transferred to the MYSQL for execution. The working of the middleware is different for different queries as shown by Table 1.

Table 1. Working as DBCrypto as Middleware

Sr. No.	Query/Clause / Keys/Others	Targeted Data	Working as DBCrypto as Middleware
1.	Create Table	--	Generates Key File for table
2.	Insert	Column Values in Tuples	Encrypt table tuple values by using combination of Key file assigned to the table and substitution Table.
3.	Select	Column Values in Tuples	Decrypt table tuple values by using combination of Key file assigned to the table and substitution Table.
4	Update	Column Values in Tuples	Identifies specified column name, Decrypt data of identified columns, fire the query as it is. Encrypt updated tuple values.
5	Delete	Tuple/ Tuples	Removes the row/rows specified in query.
6	Drop	--	Deletes whole table schema from database & also delete key entry for specified table from key file.
7	Where	Column	Decrypt columns specified in where clause temporary, perform query execution as it is on server and again encrypt specified column.
8	Null	Work is in progress	
9	Foreign Key		
10	Aggregations		
11	Join		
12	Nested Queries		

4.4 Security Algorithms

The data security algorithms are implemented in the middleware to perform encryption and decryption operation. As mentioned in section 2 we implemented Vigenere Cipher text algorithm [2] for encryption and decryption. For Example the Table 2 for integer values encryption and decryption is as follows,

Now let's see how Cipher Text generates. Let the plain text be 5436 and we have assumed the key as 3412.So the result is as follows.

5436 → Plain Text

3412 →	Key
8848 →	Cipher Text

Table 2. Table for Integer Values

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	0	1	2	3	4	5	6	7	8	9
2	0	2	3	4	5	6	7	8	9	1
3	0	3	4	5	6	7	8	9	1	2
4	0	4	5	6	7	8	9	1	2	3
5	0	5	6	7	8	9	1	2	3	4
6	0	6	7	8	9	1	2	3	4	5
7	0	7	8	9	1	2	3	4	5	6
8	0	8	9	1	2	3	4	5	6	7
9	0	9	1	2	3	4	5	6	7	8

The encryption and decryption given above is with respect to Integer data type. Now let's look regarding String data type. (Table of ASCII values and keys of string data type is in Excel Document mailed along with this document).
Consider,
Key = 89534
Plain Text = 'suraj'
Plain Text with ASCII = 46 35 33 32 45
Encryption:
Here we consider that Plain Text as with ASCII as column value and key as row value. Map these values in table. It will generate following Cipher Text
Cipher Text with ASCII=53 45 39 36 50
Cipher Text = (a12d
Decryption:
Taking Cipher Text as table value and key as row value, retrieve appropriate column value. It will give plain text.
Plain Text with ASCII= 46 35 33 32 45

5. PSEUDO CODE FOR ALGORITHMS

The pseudo code of various algorithms implemented for queries such as create, insert, update etc. are discussed in the subsections.

5.1 Pseudo Code for Create

1. Retrieve the table name from query.
2. Generate unique key by using KeyGenerator class.
3. Store key value pair as table name, key into 'DBCryptoKeyFile.file'.
4. Send create query as it is to mysqlserver for its execution.
5. Display message to user.

5.2 Pseudo Code for Insert

1. Retrieve tablename from query.
2. If tablename present in DBCryptoKeyFile.file then Retrieve key
   Else
   Give error message 'Table Not Present'
   Exit
   End if
3. Retrieve the column names specified in the query also retrieve their datatype.
4. Retrieve the values from the query.
5. Check datatype of values.
6. If datatype =string then
   Encrypt value using encryptString() of StringPolyAlpha.
   If datatype=integer then
   Encrypt value using encryptDigit() of IntegerPolyAlpha.
   End if
7. Reconstruct query using encrypted values.
8. Send reconstructed query as it is to mysqlserver for its execution.
9. Display message to user.

5.3 Pseudo Code for Update

1. Retrieve table name from query.
2. If tablename present in DBCryptoKeyFile.file then
   Retrieve key
   Else
   Give error message 'Table Not Present'
   Exit
   End if
3. Retrieve the column names specified in the query also retrieve their data types.
4. Decrypt those columns by the key.
5. Fire query as it is on server database.
6. Encrypt the columns which are decrypted in step 4.

5.4 Pseudo Code for Update

1. Retrieve table name from query.
2. If where clause is present then
   i. If tablename present in DBCryptoKeyFile.file then
      a) Retrieve key
      b) Decrypt the column specified in where clause.
      c) Fire the query on database as it is.
      d) Encrypt the column retrieved in step a)
   ii. Else
      a) Give error message 'Table Not Present'
      b) Exit
   iii. End if
   Else
   Fire the query as it is
   End if

5.5 Pseudo Code for Update

1. Retrieve the table name from query.
2. If tablename present in DBCryptoKeyFile.file then
   a. Fire the query as it is on Database
   b. Remove the key from key file.
   Else
   Give error message 'Table Not Present'
   Exit
   End if

6. DBCRYPTO INTERACTIVE TOOL

The implementation of DBCrypto as user interactive tool is providing DBCrypto consol as shown in Figure 3.The client can write queries on the console and can see results by using Result Window as shown in the Figure 4. At the backend actual data stored in MYSQL tables in the encrypted format as shown by Figure 5.

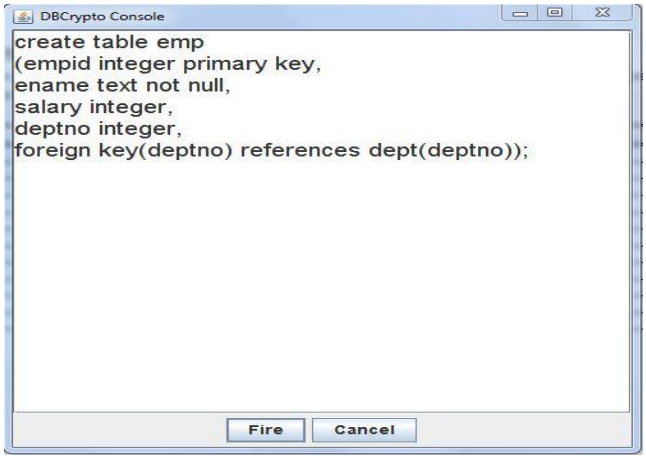


Figure 3. *DBCrypto* Client Interface to Execute Query

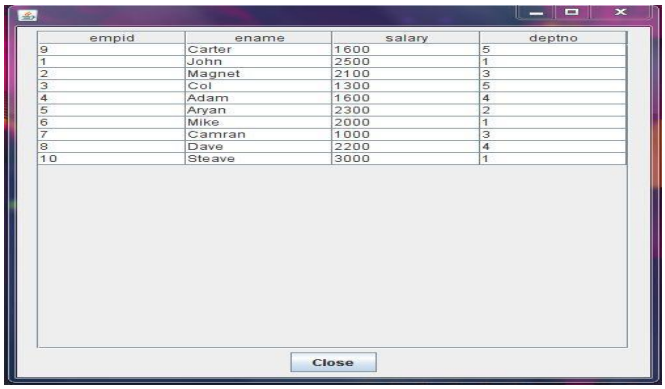


Figure 4. Inserted Data threw *DBCrypto* in decrypted form

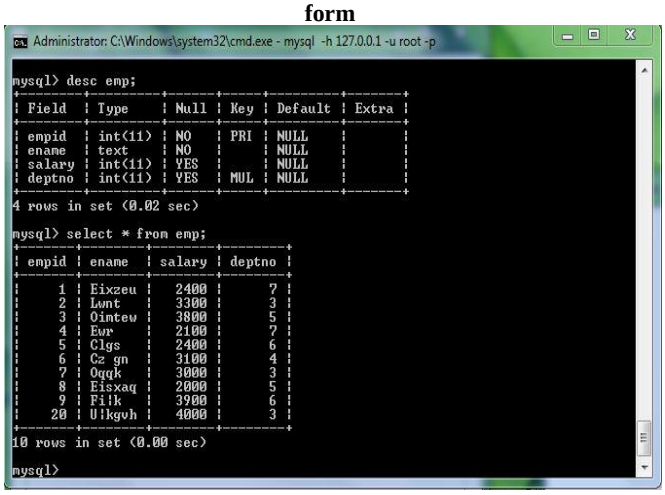


Figure 5.Actual Data in MySQL in Encrypted Form

7. EXPERIMENTATION AND RESULTS

In this section we are considering the query evaluation for 'emp' table and 'dept' table .We are using MYSQL 5.5 and JDK 1.6 for implementation of the tool. The different queries are evaluated for execution performance in terms of Time in seconds for the given table. The Table 2 and 3 are representing tuples.

Table 2. Data in table 'emp'

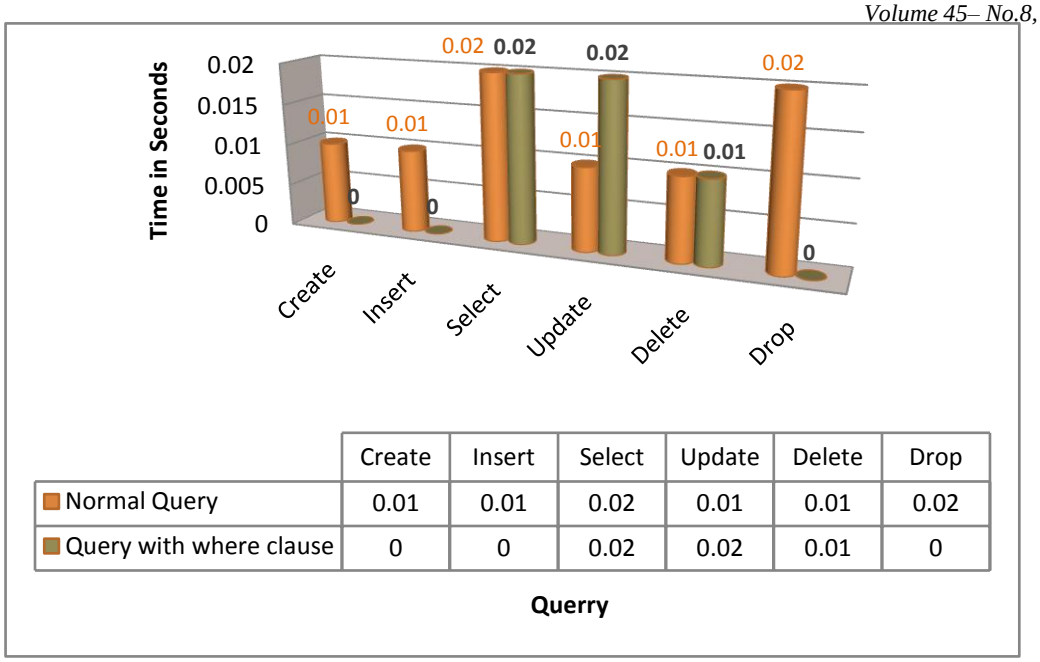
Table ***emp***			
empid	ename	salary	deptno
1	John	2500	1
2	Magnet	2100	3
3	Col	1300	5
4	Adam	1600	4
5	Aryan	2300	2
6	Mike	2000	1
7	Camran	1000	3
8	Dave	2200	4
9	Carter	1600	5
10	Steave	3000	1

Table 3. Data in table 'dept'

Table dept	
Deptno	dname
1	Physics
2	Math
3	Chemistry
4	Bio
5	DBMS

The query evaluation is based on two criteria's such as,
1. Query with where clause
2. Query without where clause

The performance for various queries is shown by Figure 6.As per results shown in figure time taken to perform various is operation is in seconds which is negligible. So the DBCrypto tool Query performance is not time consuming and it can be easily used by the client by securing data.



	Create	Insert	Select	Update	Delete	Drop
Normal Query	0.01	0.01	0.02	0.01	0.01	0.02
Query with where clause	0	0	0.02	0.02	0.01	0

Querry

Figure 6. Query performance

8. CONCLUSION AND FUTURE WORK

The DBCrypto is an efficient solution to the data security by using query level encryption and decryption approach. The data secured by this system is much authentic than we sought of. The authorized client can able to retrieve data however unauthorized client can get un-useful encrypted data so the theft of information leakage can be eluded.

The researchers [8] [9] [10] are discussing various possible attacks and solutions on those attacks. Here we are considering attacks such as SQL Injection, Man in the Middle attack as future work to make the data more secure.

9. REFERENCES

[1] Korth Henry F., Silberschatz Avi, Sudarshan S., Database System Concepts 5th Edition

[2] Stallings William Cryptography and Network Security

[3] Cryptographic Checksum for Multilevel Database Security.US Army of Airforce 1987.

[4] Popa Raluca, Catherine M. S., Zeldovich Nickolai, Balakrishnan Hari 2011. CryptDB: A practical encrypted relational DBMS.

[5] Popa Raluca, Catherine M. S., Zeldovich Nickolai, Balakrishnan Hari 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing .In Proceeding of 23rd ACM Symposium on Operating Systems Principles(SOSP 2011),cascais,Portugal,October 2011

[6] Curino Caralo, Jones Evan P. C., Popa Ada Raluca Malviya Nimesh, Wu Eugene, Madden Sam, Balkrishnan Hari & Zeldovich Nickolai. January 2011 Relational Cloud: A Database-as-a-service for the cloud. In proceedings of 5th biennial conference on innovative data systems data research (CIDR 2011).

[7] V. Narmada, B. Narasimha Swamy, D. Lokesh Sai Kumar, 2011 An enhanced security algorithm for distributed databases in privacy preserving databases. International Journal of Advanced eng. Sciences and Technologies Vol No. 8 Issue No2. 2 page no 219-225

[8] TransSQL: A Translation and Validation-based Solution for SQL-Injection Attacks, 2011 Kai-Xiang Zhang,Chia-Jun Lin, Shih-Jen Chen, Yanling Hwang, Hao-Lun Huang, Fu-Hau HsuFirst International Conference on Robot, Vision and Signal Processing

[9] Kim Geom-Go, May 23 2011, Injection Attack Detection using the Removal of SQL Query Attribute Values. IEEE International Conference on Information Science And applications(ICISA 2011) Page 1-7

[10] Desai Anand 2011 New Paradigms for Constructing Symmetric Encryption Schemes Secure Against Chosen-Cipher text Attack, Crypto 00' Proceeding of 20th annual cryptology Conference on Advanced Cryptography Springer Varlang London 2000

[11] Marten van Dijk, Gentry Craig, Halevi Shai, VaikuntanathanVinod December 11, 2009 Fully Homomorphic Encryption over the Integers

[12] Zheli Liu, ChunfuJia, Jingwei Li, Xiaochun Cheng 2010 Format-Preserving Encryption For Date Time IEEE International Conference on Intelligent computing and intelligent Systems(ICIS) Pages 201-205

[13] XuRuzhi, Guojian, Deng Liwu A Database Security Gateway to the Detection of SQL Attacks, 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)

[14] Bellare Mihir, Ristenpart Thomas, Rogaway Phillip, Stegers Till .Format-Preserving Encryption Dept. of Computer Science & Engineering, UC San Diego, La Jolla, CA 92093, USA Dept. of Computer Science, UC Davis, Davis, CA 95616, USA 2009

[15] Mattsson Ulf T. Format-Controlling Encryption using Data type-Preserving Encryption. 2009 IACR Cryptography e-Print Archive

[16] MySQL Reference Manual 5.5