INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

Effectuation of Blowfish Algorithm using Java Cryptography

Prachi Jain, Prof.Shubhangi Kharche

Abstract---The cognitive content of the paper delineates some key features like importing the data in Java using MySQL database, creating the text file and a pdf file for storing the output and importing the .csv (dot csv) file as raw data input. The paper is coaxing in a way that it clearly hashes out the procedure to efficiently execute the Blowfish Algorithm for securely transferring the data within the system. In the initial encryption phase, the blowfish algorithm generates the encrypted output which is written in text file addressed to any drive. The algorithm can be effectively implemented in Eclipse software as it helps to debug the code. In this paper algorithm is applied in voucher management .The application is coaxing in a way that it exemplifies the need to make Pre-paid Recharge system and other applied systems by providing security in intermediate path. Lineament of procedure is an extremely efficient method for securing the network that ends at card generator as the end terminal .The method finds its applications in a wide range of bailiwicks. The Blowfish algorithm is chosen for many benefits. If the world is to have a secure, unpatented, and freely- available encryption algorithm, we need to develop many implementations. These algorithms can then be subjected to years of public scrutiny and cryptanalysis. The paper serves its purpose and can be skeptically evaluated.

General Terms---MySQL, Java, Blowfish Algorithm, Encryptor, Decryptor.

Keywords---Import, Export, Class, Packages, Database, Text file, PDF file.

1. INTRODUCTION

1.1. Philosophy of algorithms

Methods of Inquiry: Algorithm is a science of computing. The most distinctive thing about Algorithms is its emphasis on computer science's methods of inquiry. Briefly, the methods of inquiry are…

1.1.1 **Design**. The planning and construction of algorithms, programs, computer hardware, or other devices that carry out computations.

1.1.2 **Theory**. The mathematical analysis of algorithms, programs, hardware, or other computing devices, often predicting the performance or correctness of those algorithms or devices.

1.1.3 **Empirical Analysis**. The verification of expectations about algorithms or other computing devices against real-world behavior, usually done via experiments.

The project's foremost goal is to demonstrate the methods as mutually supporting necessities for any task in computer application. Algorithms are subjects of deep study. Algorithms are simply the abstract outlines of programs, preliminaries to writing "real" code. Algorithms are much more than the main subjects of study in their own right, far more important than programs. This is because the thought processes surrounding algorithms are the important end result of studying any computer application. Any application understands the insights and invariants on which an algorithm is based, the process by which it was developed, and the reasons why it works and why it works efficiently (or not) can reproduce the algorithm in any programming language and can even invent new or better algorithms based on similar insights, processes, and analyses. On the other hand, anyone who understands only the code for the algorithm in a particular language can do little more than regurgitate that code in its original form. [1]

Therefore we emphasis to discuss a process by which the algorithm could be discovered, develop its design through multiple iterations, analyze its performance and correctness, and, almost as an after-thought, present it in Java. Two features characterize the book's presentation of algorithms: object oriented

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

abstraction and recursion. [1] Abstraction in some form is essential to make computing intellectually manageable, and object oriented design and programming seem to be particularly powerful abstraction mechanisms. Recursion is perhaps the ultimate control structure for algorithms (it can express everything iteration can, and more). [1] Recursion also provides a key connection between algorithms and mathematics, and is a definitional mechanism applicable far beyond algorithms. It is not that the ideas are unimportant, but simply because they are not prominent in the public perception of computing that they seem to be unimportant. No important idea appears until after some motivation for studying has been presented.

1.2 Working of Blowfish Algorithm

Blowfish is a symmetric block cipher. It is effectively used for encryption and safeguarding the data. It takes a variable-length key, from 32 bits to 448 bits. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Blowfish is unpatented and license-free, and is available free for all uses. [2] Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard now receives more attention. [2]Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative to the aging DES and free of the problems and constraints associated with other algorithms. Blowfish is known to be susceptible to attacks on reflectively weak keys.[4] This means Blowfish users must carefully select keys as there is a class of keys known to be weak, or switch to more modern alternatives like the Advanced Encryption Standard, Salsa20, or Blowfish's more modern successors Twofish and Threefish. [3]Blowfish is a fast block cipher, except when changing keys. Each new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers. This prevents its use in certain applications, but is not a problem in others. In one application, it is actually a benefit:the password-hashing method used in OpenBSD uses an algorithm derived from Blowfish that makes use of the slow key schedule; the idea is that the extra computational effort required gives protection against dictionary attacks. Blowfish has a memory footprint of just over 4 kilobytes of RAM. This constraint is not a problem even for older desktop and laptop computers, though it does prevent use in the smallest embedded systems such as early smartcards .Blowfish was one of the first secure block ciphers not subject to any patents and therefore freely available for anyone to use. This benefit has contributed to its popularity in cryptographic software. [4] Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. In structure it resembles CAST-128, which uses fixed S-boxes. The diagram to the left shows the action of Blowfish. Each line represents 32 bits. The algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. [4] The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries. The diagram to the upper right shows Blowfish's F-function. The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 2^{32} and XORed to produce the final 32-bit output. Decryption is exactly the same as encryption, except that P1, P2... P18 are used in the reverse order. This is not so obvious because xor is commutative and associative. A common misconception is to use inverse order of encryption as decryption algorithm (i.e. first XORing P17 and P18 to the cipher text block, then using the P-entries in reverse order). Blowfish's key schedule starts by initializing the P-array and S-boxes with values derived from the hexadecimal digits of pi, which contain no obvious pattern (see nothing up my sleeve number). The secret key is then, byte by byte, cycling the key if necessary, XORed with all the

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

P-entries in order. A 64-bit all-zero blocks is then encrypted with the algorithm as it stands.

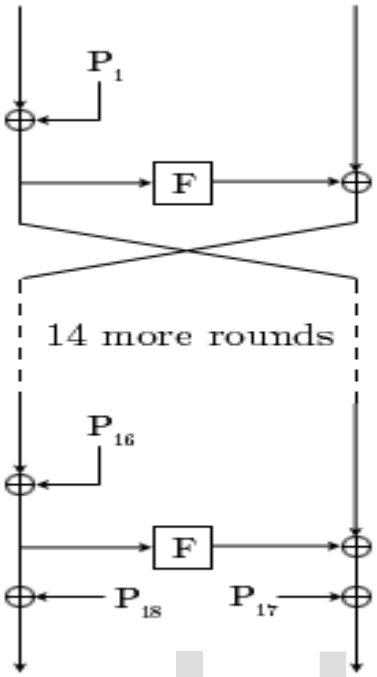

Fig.1 The Feistel Structure of Blowfish [5]

The resultant cipher text replaces P_1 and P_2. The same cipher text is then encrypted again with the new sub keys, and the new cipher text replaces P_3 and P_4. This continues, replacing the entire P-array and all the S-box entries. In all, the Blowfish encryption algorithm will run 521 times to generate all the sub keys - about 4KB of data is processed. Because the P-array is 576 bits long, and the key bytes are XORed through all these 576 bits during the initialization, many implementations support key sizes up to 576 bits. While this is certainly possible, the 448 bits limit is here to ensure that every bit of every subkey depends on every bit of the key, as the last four values of the P-array don't affect every bit of the cipher text. This point should be taken in consideration for implementations with a different number of rounds, as even though it increases security against an exhaustive attack, it weakens the security guaranteed by the algorithm.

Comparison of Algorithm

Symmetric encryption, also known as secret key encryption, uses a single key for encryption and decryption. The same key is used to transmit data to the encryption side and then used at the decrypting machine. Transferring key via the Internet exposes the key to malicious programs. Hence, symmetric encryption is more suitable for communication between those parties who are already aware of the key, and need not transfer it over the network. Asymmetric encryption, also known as public key-private key encryption, uses two different keys for encryption and decryption, public key and the private key respectively. For the purpose of storing and transferring, the key can be converted into an array of bytes by a process called wrapping. When the key is needed again, the byte array can be converted back into the key by a process called unwrapping. There are several algorithms for data encryption using symmetric and asymmetric encryption keys. Password-Based Encryption (PBE) derives an encryption key from a password. In order to make the task of getting from password to key very time consuming for an attacker, most PBE implementations will mix in a random number, known as a *salt*, to create the key. Given below is a list of symmetric key algorithms for data encryption: [6]

DES: DES stands for Data Encryption Standard. Developed by IBM and commonly used. Triple-DES (3DES) is a variant which uses DES thrice for ciphering data. DES operates with 56 bit key on 64 bit blocks of data. It is vulnerable to security attacks and can be easily retrieved.

AES: AES stands for Advanced Encryption Standard. It replaced DES as the official standard of US National Institute for Standards and Technology (NIST). AES operates on variable key sizes and variable block sizes of 128, 192 or 256 bits. Due to the number of permutations and combinations in AES contributing to its higher complexity, AES has a higher security as compared to DES.

IDEA: It stands for International Data Encryption Algorithm. It uses 128 bit key and 64 bit blocks of data.

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

RC4 to RC6: RC is a stream cipher that stands for Ron's Code or Rivest Cypher, attributed to the name of its developer. It can adopt variable key lengths and block sizes. It is based on random permutation.

Blowfish: It is a fast running substitute for DES and IDEA. It adopts key lengths in the range of 32 to 448 bits and block size of 64 bits.

Twofish: Flexible algorithm which can run on PC, smart Card microprocessors or dedicated encryption hardware. It uses 128, 192 or 256 bit keys and operates on 128 bit-blocks. Asymmetric encryption or Public Key Cryptography (PKC) was developed by Prof. Martin Hellman and Whitefield Diffie in 1976. It is based on the concept of using two different keys for encryption and decryption. Asymmetric Encryption Algorithms are listed below as:
RSA: This is a block cipher algorithm is named after its developers – Ronald *R*ivest, Adi *S*hamir and Leonard *A*leman. It uses variable size keys. The keys can be integers from 0 to n-1, where n is typically 1024. It can adopt stronger keys as the computational power increases.

Diffie-Hellman: Algorithm proposed by the duo who published public key cryptography. The purpose of this algorithm is to allow the secure the exchange of a key between two users that can be used for encryption.

DSA: Digital Signature Algorithm is used for authentication of messages, by allowing the sender to attach a code at the end of the message. Digital signature ensures that the data was not tampered during its transfer from the original machine to the user machine.

PKCS: Stands for Public Key Cryptography Standards proposed by RSA Systems Inc.

KEA: Key Exchange Algorithm. It is a variant of Diffie-Hellman algorithm for exchange of key.

LUC: Designed on the basis of Lucas sequence (a form of Fibonacci sequence), in which the first two numbers are 2 and 1. The rest of the sequence follows the rule that it is the sum of the previous two numbers. [7]

Advantages of Blowfish algorithm over other algorithm: Blowfish is a fast running substitute for DES and IDEA. Also Blowfish provides a good encryption rate in software. Blowfish is unpatented and available in public domain. This has led to its popularity and people can freely access them the latest versions are Twofish and Threefish algorithms.

Technical specifications of Blowfish algorithm:

Packet Size(Kb)	Time (millisecond)
1024.00	508
1500.02	621
2100.50	854
2512.67	978
3124.21	1021
5100.50	1520

Table.1 Time consumption (Hexadecimal Encoding) [8]

In March 2012, Anand Kumar et al [9] discussed that the time consumption for AES Algorithm is more than that of the Blowfish Algorithm. The Time taken by AES for 1024Kb file is more than 700 milliseconds and that of Blowfish is between 600 and 700 milliseconds. For 3124Kb size file the time taken by AES is approximately 1600 milliseconds and that of Blowfish algorithm is 1500 milliseconds. For the file size of 2101 Kb, the time taken for encoding Hexadecimal input is approximately same and it remains constant for Blowfish though the file size increases from 2101Kb to 2513Kb. [9]

2. ENCRYPTION AND DECRYPTION PROCEDURE

The following are the steps that will draw consummate flow of the steps to be carried out in implementing the Blowfish algorithm using

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

importing and exporting functions in MySQL and Java.

The first step is that if the file plied is a Text file it need to be is be converted to .csv file. The file saved as .csv file is imported to MySQL database. MySQL database is saved in the Library document register and it is imported by the java code written in Eclipse - (Java connector J 5.1.14 is used for importing the file from MySQL Database). Then the Blowfish Algorithm is called by the java code and the Database files are encrypted Line-by-line using random Key Generator. Finally, at the encryptor side, the encrypted output and the Random key generated are stored in text file. The text files are send to other end. The Decryptor code decrypts the Cipher text Line-by-line using the respective random keys that are stored in the text file. The decrypted output is stored as PDF file in any desired drive.

3. CREATION OF DATABASE

Setup is quick and easy if you follow the step-by-step installation documentation.

3.1. Step One: Creating your new database

When you log in to your PhpMyAdmin welcome page, the first step is to enter a name for your new database in a text box provided. You can name your database anything that you wish, however if you are creating the database to use with a script or software package that you purchased somewhere, the script provider will often suggest a "preferred" database name. You should always create your database using the following format: [10]

username_ database name Example: allows the server to know which user is in control of the new database, and it will also provide permission to access the database to only specific users. This also allows different users on the same server to use the same name for their own database, as you did, without interfering with your data – that is helpful if more than one user on your server bought similar software for their own site. They can then also use the software providers "preferred" database name. [10]

3.2. Step Two: Creating a table for your new database

After you have created a database, the next step is to create a table, or even multiple tables, for you to store data. A table is the part of your new database that actually stores data.

Create a table by selecting the database that you created from the drop box list of databases. Once a database is selected a new form appears and asks for you to create a new table. One must decide what you want to name your table and enter that name into the name box. Try to choose a name that reflects the type of data that will be stored in the table, such as orders, users, or inventory. Then decide how many "fields" or columns of data that you want to store for each record. If you need for the table to store five (5) different items, such as username, users email address, users telephone number, users account number, and the users age, than you would need five (5) fields. Simply enter the number 5 in the appropriate box. Once you hit create, the system will create a table and will add those fields into the table for you. Don't worry about the number of fields you might need right now, as you can always add or delete fields later.

3.3. Defining Fields

Once you have created your table you will be prompted to tell the database what features that you want each field to have. Select the best choice for storing your data. If you make a mistake you can go back and edit the field. [10]

If the field is to be used to store numbers, here are some choices:

TINYINT – A very small integer. The signed range is -128 to 127. SMALLINT – A small integer. The signed range is -32768 to 32767. MEDIUMINT – A medium-size integer. The signed range is -8388608 to 8388607. INT – A normal-size integer. The signed range is -

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

2147483648 to 2147483647. BIGINT – A very large integer.

Some other less common number options include:

FLOAT- A floating-point number. DOUBLE – A double-precision floating-point number. DECIMAL – A packed exact fixed-point number.

If the field is to be used to store text or both text and numbers combined, here are some choices:

VARCHAR is for varying characters and can be up to 255 characters in length. TEXT is a column with a maximum length of 65,535 characters – easy to search. BLOB is a column with a maximum length of 65,535 characters – case- sensitive.

Once you have selected the data type for your fields you will need to let the system know how many characters that you will need to store in the field.

The last step to creating your data fields is to select any special attributes that you may find helpful. Some examples are:

Auto Increment: Auto-Increment fields are useful for assigning unique identification numbers for users, products, and customers, etc. By default, fields are incremented using number characters (like "1", "2").

Primary Key: The primary key is a data column that uniquely identifies a specific instance of that data. At least one of your fields must be a Primary Key. Username is an example of a good primary key. You do not want to have more than one individual having the same username.

Index Key: Allows you to speed up searches by designating a field as a preferred data source, especially when combining data from multiple tables. [10]

4. ALGORITHM

4.1 Encryption Code

Initially we define the package as blowfish. The following packages are imported: java.security.*, javax.crypto.*, javax.crypto.spec.*, java.io.*, java.sql.Connection, java.sql.DriverManager, java.sql.ResultSet, java.sql.Statement, java.nio.*, java.util.*.

Next step in the algorithm is creation of class which uses public excess specifier. In the class the generation of key is carried out by import the class SecretKeySpec, for which object is created. Another static class Cipher is created which is defined in the program.

cipher = Cipher.getInstance("Blowfish")

Here cipher is an object of class Cipher.

The main class returns no values and uses excess specifier as public. The main class connects MySQL Database with java code. In the encryption program we import the database file.

The syntax is:

Class.*forName* ("com.mysql.jdbc.Driver");

Connection connection = DriverManager.*getConnection* ("jdbc:mysql://localhost:3306/database_name", "root", "password");

Statement statement = connection.createStatement ();

ResultSet resultSet = statement.executeQuery ("SELECT idvoucherraw FROM voucher_raw");

Here idvoucherraw and voucher_raw are Database specifications.

Next step is to create a print writer for writing to a file stored in G drive for this code

PrintWriter <object> = **new** PrintWriter(**new** FileWriter("G:\\file_name.txt"));

PrintWriter <object> = **new** PrintWriter(**new** FileWriter("G:\\key_file.txt"));

Next step includes creation of loop that reads every text line-by-line and store it in text file.(Instead of text file one can also use pdf file format where changes cannot be easily made). Then the String plaintext is defined. Initially the bits are defined as 32. The

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

function generateKey() is used to generate random key for random cipher. Cipher text is generated by using the function getEncoded(). The raw array is set to zero value at index zero.
Finally after importing all the functions, the variables are printed. The Blowfish Algorithm defined in the Javax package is passed generated random key values.
The syntax is defined as:
Blowfish <object> = **new** Blowfish (key)
After decryption each read plain text is written in text file. The following syntax is used:

<object>.println(*asHex*(key))

The three classes- Blowfish, encrypt, decrypt that are used in the main are defined in the later program.

4.2 Raw .csv File (looprawdata.xlsx)

The raw file has many fields according to the service provider's specifications. It includes code length, serial number, batch size, and initial batch size, face value, expiration, and expiration months (not in the same format). For example,

0222.000903201800008069489497385612467110000 2

4.3 MySQL Connector/J 5.1.14

JDBC is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. From a technical point of view, the API is as a set of classes in the java.sql package. To use JDBC with a particular database, we need a JDBC driver for that database.[11]

MySQL is a leading open source database management system. It is a multi user, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular **LAMP** platform. Linux, Apache, MySQL, PHP. Currently MySQL is owned by Oracle. MySQL database is available on most important OS platforms. It runs under BSD Unix, Linux, Windows or Mac. Wikipedia and YouTube use MySQL. These sites manage millions of queries each day. MySQL comes in two versions. MySQL server system and MySQL embedded system.

Before we star - install mysql-server and mysql-client packages. The first package has the MySQL server and the second one contains, among others, the mysql monitor tool. We need to install the JDK, Java Development Kit, for compiling and running Java programs. Finally, we need the MySQL Connector/J driver. Inside the Projects tab, right click on the Libraries node and select Add Library option. From the list of options, select MySQL JDBC Driver. [11]

Few changes made are in the latest version of MySQL connector:
Bugs Fixed - The Statement.cancel() was made less susceptible to deadlocking, to allow application servers to immediately cancel operations that fail due to transaction timeouts or network issues. The abortInternal() method that was previously part of the MySQLConnection interface is now available through the Connection interface, so it can be tested for and called if available. When a statement was accessing a table in streaming mode, terminating the statement by issuing KILL QUERY from another session could cause a SQLException "Query execution was interrupted" and a stall when the original session issued a ResultSet.close() or Statement.close() method call. [12]

4.4 Encryptor Output File (output.txt)

The Encryptor output is 48 byte value.

For example: 8a 90 04 04 72 1a a7 f4 54 75 c4 6c 23 12 9a 19 61 e6 65 ae 85 e0 79 20 db 1f 2a 0e 37 1d 83 bd fc cd 5f ad ed 25 77 b1 ed ea 25 9a 51 4c 31 ef

4.5 Random Key Generated (keyinput.txt)

The random keys are generated by class key generator which imports the function from other class. The key generated is of 32 bits/ 4 bytes. Example,00 fb d3 82 00 77 0c f8

This output is stored as a text file in G drive according to the program.

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

4.6 Decryption Code

The following packages are imported: java.security.*, javax.crypto.*, javax.crypto.spec.*, java.io.*, java.sql.Connection, java.sql.DriverManager, java.sql.ResultSet, java.sql.Statement, java.nio.*, java.util.*.

The main class reads the text file that contains the encrypted output and the generated key. Syntax used for importing is:

File <object> = **new** File("G://file_name.txt")
File <object> = **new** File("G://key_name.txt")

Data from this file is read using a file reader. The syntax is:
FileReader <object> = **new** FileReader(encrypted output object)
FileReader <object> = **new** FileReader(key output object)

The contents read via File Reader is then stored in a new object defined by BufferReader and the FileReader variable is passed through it. The syntax for reading is:
BufferedReader <object> = **new** BufferedReader(object)
BufferedReader <object> = **new** **B**ufferedReader(object)

The String variables are defined for importing the text line-by-line. But before the start of the loop the file in which the decrypted output is transferred is defined. Had the file been defined in the later part only the output of the last line is written in the file.PrintWriter out = **new** PrintWriter(**new** FileWriter("G:\\decrypted_output.pdf")

For this program the input file are text files and the output file containing the decrypted text is pdf file.

While loop takes line-by-line every text statement in string variable. The string output is then converted into byte format to pass it to the function. In this case decryption function accepts key as well as encrypted output read from the file by readLine function which is system defined. At the end of the loop we pass the decrypted text to the destined file through out.println(variable).
The output generated is then given to the card generator as the input. The data is processed by the card generator to create the voucher output.

4.7 Decrypted File (decrypt.pdf) - Result

The output decrypted file contains the result in byte format by default as all the functions defined in the algorithm accepts the argument in byte format. The Byte format can be converted into string by using the syntax:
<Byte variable>.toString()

The Output in the byte format will be as: e2eec8

The result is further send to the card generator where the actual clandestine codes are generated and printed on the vouchers.

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

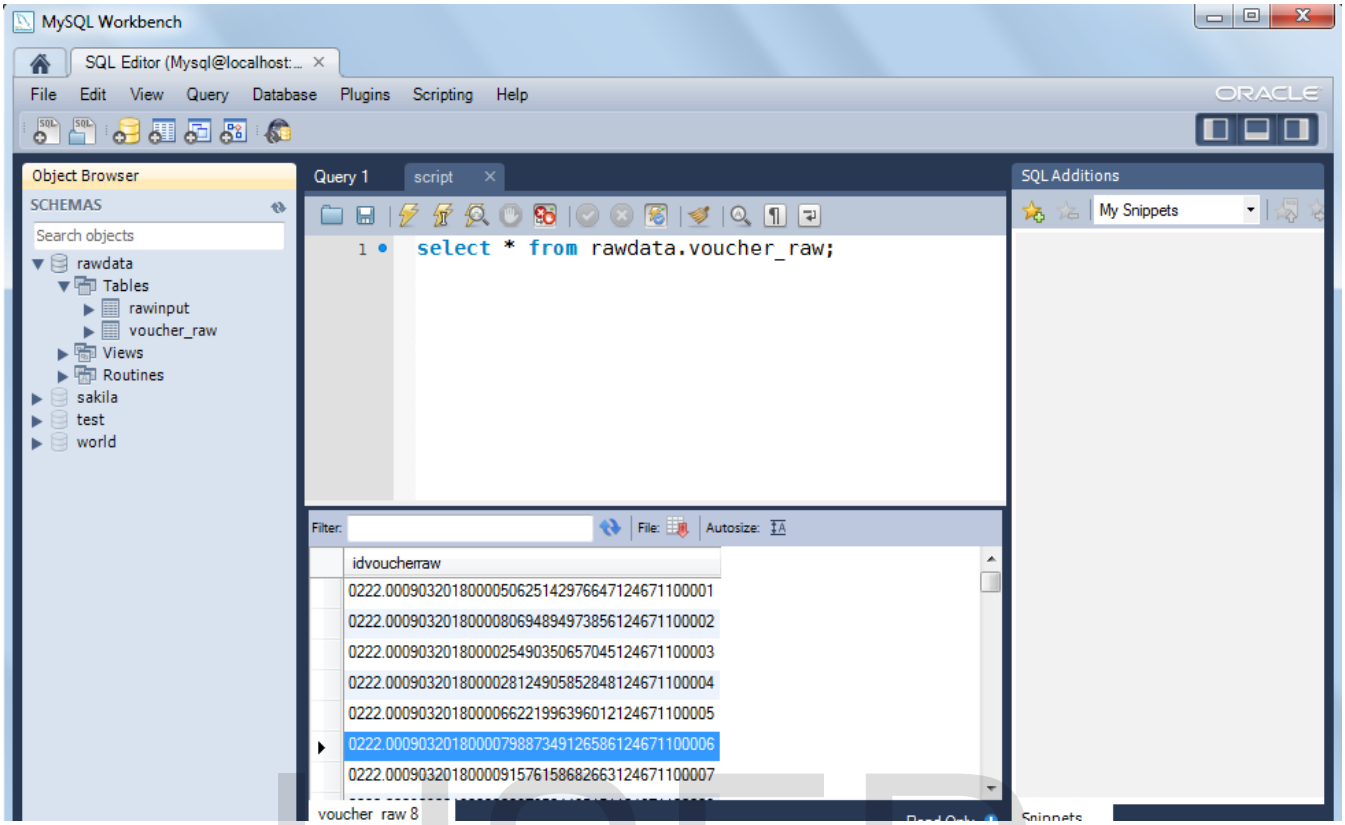

Fig.2 Created MySQL Database

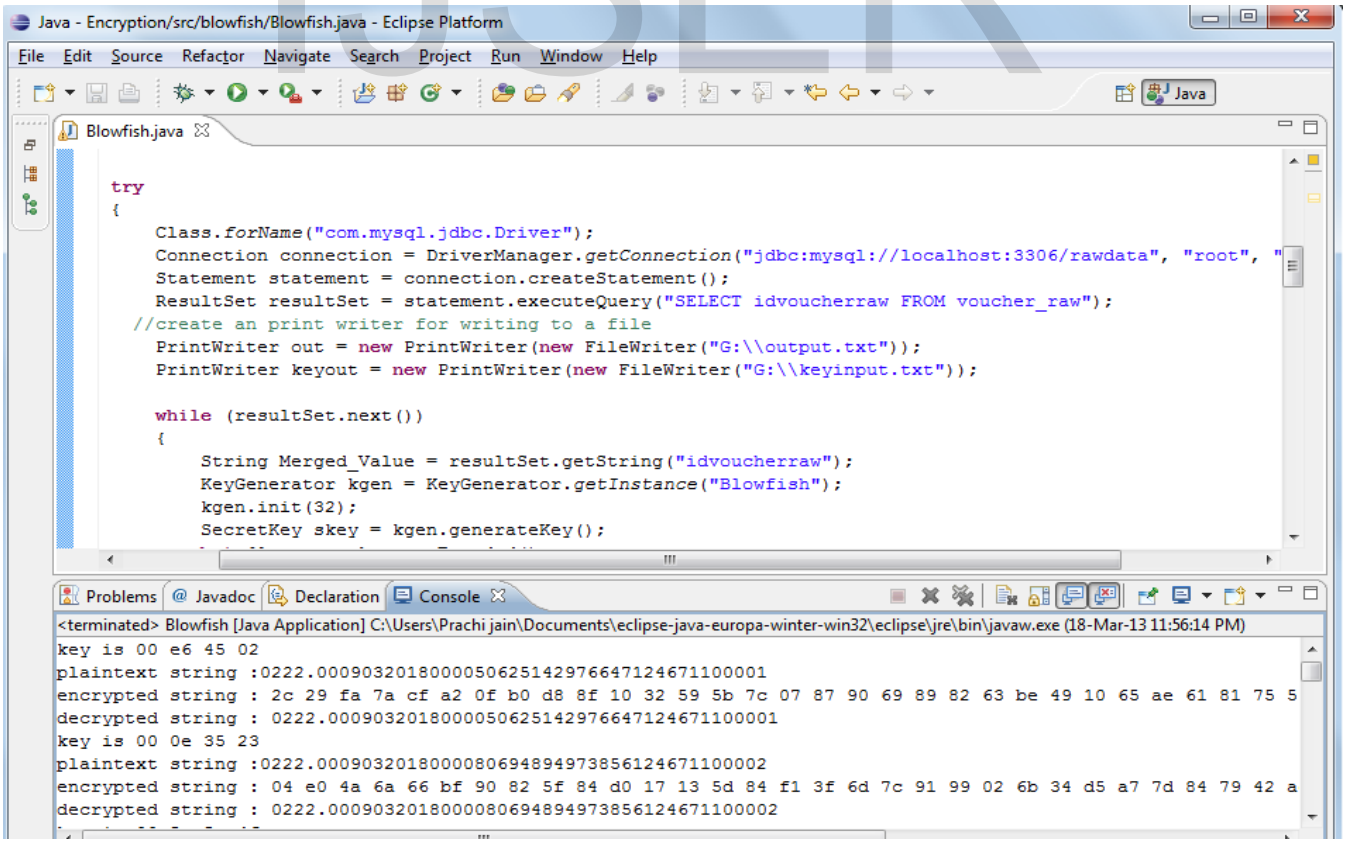

Fig.3 Java Encryption code in Eclipse

INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH VOLUME 4, ISSUE 5, MAY-2013
ISSN 2229-5518

5. APPLICATION- PRE-PAID VOUCHER CREATION

Voucher management is very important application of cryptography, which handles all aspects of the prepaid cards, like the bonus card or recharge top up cards. Voucher management provides encryption, voucher tracking and also takes care of the address fraud. A voucher has a unique serial number and may have a PIN code by which it is identified. The application performs automation of the voucher lifecycle, and should support a wide variety of voucher types. For a successful prepaid business, efficient generation of the PIN, voucher printing, distribution, warehousing and PIN redemption, have to be efficiently done. Customer satisfaction is most an important parameter that contributes to the success of the voucher companies. [13] The solution is prepaid platform independent, highly secure, supports both physical and virtual vouchers, is highly scalable, and integrates with the operators existing prepaid IN platforms. Subscribers can either apply the full recharge value to any one service, or spread the value across multiple services, with the System seamlessly managing the recharge. Benefits include easier operator logistics, lower retailer working capital requirement & simpler marketing communication to subscribers. [13]

Voucher management is of extravagant commercial importance. Vouchers are out sourced by the service provider company from other companies voucher software service provider companies. The cost of these voucher dealers may be as high as half a million US Dollar. This charge is too high for small scale the company to start as service provider who wants to create its business. Creating raw data processing, encrypting raw data and decrypting it is very significant process in handling vouchers and enabling prepaid facility for customer's benefit. Hence, need for Voucher is beneficial for the customer using prepaid billing system. And scope is security against the theft who steals this code and use the monetary value of vouchers to sell illicitly and earn profits.

6. CONCLUSION

This method of implementation prove to be very efficient and user friendly. It is very easy to implement at the service provider's end as well as at the printer end where the decrypted output data is further send to the card generator and then printed. The algorithm is very fast and effective. It is easier to implement compared to AES 128, AES 192, and AES 256.

REFERENCES

[1] Algorithms and Data Structures: The science of computing and the Study of Computer Science, supplement material for Bladwin and Scragg, Charles River Media, 2004

[2] Descryption of New Variable-Length Key, 64-Bit Block Cipher (Blowfish), B. Schneier. Site: http://www.schneier.com/paper-blowfish-fse.html

[3] Tom Gonzalez, "A Reflection Attack on Blowfish", Journal of Latex Files, Vol.6, No.1, January 2007.

[4] Orhun Kara and Cevat Manap, "A New Class of Weak Keys for Blowfish", Site: http://www.iacr.org/archive/fse2007/45930168/4593168.pdf

[5] Vaudenay, serge, "A Classical Introduction to Cryptography", Springer. Pg. 32, 53.

[6] Cryptography & Network security by William Stalling. Pg.30-56,134-165.

[7] Haiyong Xie, Li Zhou, and Laxmi Bhuyan, "Architectural Analysis of Cryptographic Applications for Network Processors".Site:http://citeseerx.ist.psu.edu/showciting;jsessionid=88B64B6CB12275FD576BD55E5DC2C73?doi=10.1.1.15.1229.

[8] Abdel-Karim Al Tamimi, "Performance Analysis of Data Encryption Algorithm". Site: http://www.cse.wustl.edu/~jain/cse567 06/ftp/encryption_perf/index.html

[9] IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.3, March 2011 - Blowfish_efficiency.pdf

[10] Don Beavers, "My SQL for Beginners How to create a MySQL Database", webpronews, 2005.

[11] Create the database, 2013 http://drupal.com/documentation/install/create-database_2013.

[12] "MySQL Connector/J Release Notes", site: http://dev.mysql.com/doc/relnotes/connector-j/en/news-5-1-14.html

[13] Converse, "Card Generator User Manual-RTB_4.6 UR15 Card generator Manual.pdf", release 13, Issue 1.