Chapter 4

Design and Implementation of Threshold Multi-Attribute Secure Decision Making Algorithm (MASDMA)
DESIGN AND IMPLEMENTATION OF THRESHOLD MULTI-ATTRIBUTE SECURE DECISION MAKING ALGORITHM (MASDMA)

This chapter presents the design and implementation of the threshold Multi-Attribute Secure Decision Making Algorithm. Section 4.1 and Section 4.2 give information about decision making and its importance. In section 4.3 various decision making algorithms are discussed. Section 4.4 and Section 4.5 provide details of the proposed threshold Multi-Attribute Secure Decision Making Algorithm (MASDMA) and its implementation respectively. Section 4.6 presents solved example and outputs which are generated after execution of the proposed algorithm.

4.1 Introduction

Health-Insurance organizations today are capable of generating and collecting large amounts of data for processing. This increase in volume of data requires automatic way of extracting, transforming data when needed to get useful information. With the use of data mining techniques it is possible to extract interesting and useful knowledge, helpful for Decision making system to make an effective decision enriched by knowledge acquired by careful processing of data [11]. Not only insurance companies but also clients find difficulty in selecting the best policy for their requirements, so it is necessary to have a forceful decision making system which is the main aim of this research work.

Information technologies are being increasingly implemented in healthcare or in insurance organizations in order to respond to the needs of doctors or customers of insurance companies to make decisions perceptively. Data mining tools can be very useful to control limitations of people such as subjectivity or error due to fatigue, and to provide indications for the decision-making processes [10]. The essence of data mining lies in the identification of relations, frameworks and models with algorithmic support which provide support for predictions and decision making process for selecting the best Hospital and the best Insurance Company, in customer town. These models can be called predictive, and they are being integrated into information systems of hospitals and insurance companies as models for decision making.
reducing the subjectivity and time complexity in accomplishment of decisions. In addition, to its implementation, information technology in hospital and insurance company’s information enables comprehensive management of health insurance knowledge and its secure exchange between recipients and providers of healthcare services. Extensive use of information technology enables the elimination of complex task of manual data extraction from charts or from specialized questionnaires. Mining of data directly from electronic records of insurance companies and hospitals will help user to make decisions accurately in lesser time, which minimizes time of searching or finding the information about the health insurances or hospitals suitable. Retrieval/processing of information with the help of computers can help the quality of decision making and avoiding human errors. When there is a large volume of data that needs to be classified, decision making accurately by people is an issue. [4].

To a large extent decision making depends on Data mining which represents the process of analyzing raw data with the help of computer and extraction of their meaning. It is often defined as discerning previously unknown and potentially useful information from large volume of (unstructured) data [5]. Various techniques of data mining and classification helps in making decisions, regarding customer tendencies from his view depends on data set patterns from a database of historical information about selections [24][6].

4.2 Literature Study related To Existing Decision Making Algorithms

The important aspects of decision-making processes in operation of insurers, in terms of improving the collaboration between insurance companies and insurance providers is the key factor in making financial success of both the insured and insurers and this is because the transfer of massive data between the insurance companies and insurance providers can only be effectively managed by means of sophisticated computer algorithms [7, 8]. A decision making process is one of the most important and complicated human activity. The influence of decision making will effective client business in various ways like enabling client in selecting best hospital, best insurance company, and optimum policy suitable for his requirements.
Unfortunately, decision making algorithms can be defined only for some kinds of (simple) management problems. In more complex cases, an "optimal" or "universal" algorithm for decision making does not exist, even though classy artificial intelligence tools have increasingly been applied. Due to above stated reasons the possibility of construction of an automated decision making process is limited [9, 10, 11]. Therefore, decision making processes, particularly in an insurance sector, requires a sophisticated decision making system with suitable decision making algorithm based on real world facts. Whilst complex computer science mechanisms, including data-base models and artificial intelligence machinery used by the actuaries constitute the "brain" in the prediction world environment at the top management level of a large insurance company [2, 4], the "heart" of a massive data circulation system between the brokers and the insurance company at its medium management level is the Microsoft Excel environment overwhelmingly operated and commonly used by the brokers to make what if analysis in a limited manner. In fact, in their everyday influx of customers, brokers could not admit any other, more complex programming environment, so it is necessary to develop a system which will provide effective decisions which uses an efficient algorithm.

Handling massive amounts of policy-related data at the medium management level has posed significant challenges for the insurance industry [13, 15]. Similarly in the case of customers opting for insurance policy faces complexity in selecting an optimal policy among huge number of health insurance companies which in turn provides large number of policies at different premiums. So by considering clients requirements which consists of variables such as income, premium, age, number of dependents etc., acts as inputs which are subjected to different processing and analytics, resulting in information outputs that in turn served for decision making resolutions to get an optimum decisions is the task in hand which we have given solution.

An input to a decision making system is a policy-related, massive data coming both from the insurance company management and, in particular, the collaborating brokers or client's status. In the former case, data includes monthly, raw (invalidated) policies' details such as their numbers, the assigned premiums (new and renewed ones), losses, expiry dates, and possible policy cancellation notifications, without
divisions to particular brokers and their campaigns (business branches). In the latter case, each campaign-assigned broker provides (only once) on-line data concerning newly-written policies, in terms of their numbers, booked premiums, and expiry dates on time. Since there are hundreds of brokers handling variety of policies, the raw data can be biased with numerous liabilities, sometimes leads to financially detrimental errors. Due to these reasons it is need an automated and secure Decision making algorithms in health insurance industry to provide an ideal solution in minimum time [23].

The following algorithms are in practice for making decisions about insurance by insurer/client in the case of decisions [9, 24]. These algorithms exhibit several inherent advantageous features.

4.2.1 Decision support system as feedback-feed forward control system

Management and control are quite similar notions, even though they have been grown on different application fields. Making decisions for the purpose of (broadly understood) management is a sort of computation of control decisions by a controller. Both management decisions and control decisions are typically executed in a feedback, or closed-loop system, hopefully supplemented with feed forward paths arranged on measurable/countable disturbances acting upon the management/control plant. Fig. 1 presents a feedback-feed forward decision making system at the medium management level of the plant, which is an insurer (insurance company + broker). The system is arranged in a control-like fashion, with specifying the system variables as follows:

- a control/management decision \( n_u \)-vector \( u(k) \) feeding the plant, where \( k \) is the discrete time, with the sampling interval equal to the company's reporting period (at the medium management level), that is one month,
- controlled output \( n_y \)-vector \( y(k) \) from the plant,
- output reference (or set point) \( y_r(k) \),
- error \( e(k) = y_r(k) - y(k) \) between the reference and controlled output,
- measurable/countable disturbance \( n_d \)-vector \( d_m(k) \),
- Un-measurable/uncountable disturbance \( d_u \).
Elements of a control/management decision vector $u(k)$ constitute the corrective decisions onto underperforming brokers:

- shifting the broker(s) to some more suitable campaigns,
- recommending additional training sessions to the broker(s),
- suspending the existing business arrangements with the broker(s),
- disconnecting the existing business arrangements with the broker(s).

Now, the size of the vector $u(k)$ is $n_u=4$. It is important however that an actual control decision vector is always of a lower size, e.g. 3 or even 1. This is because, typically, not all the "rectifying" control means for underperforming broker(s) are simultaneously applied. Moreover, there are often situations that there are no underperforming brokers during monthly reporting periods (corresponding to $n_u=0$), so the control decision is "no change indecision".

The company's decision making process at the medium management level is subject to a number of un-measurable/uncountable disturbances $dun(k)$ related with altering financial situation of the company and its environment on the one hand, and a market-sensitive tissue of customers on the other. However, the main source of the
disturbances is just the massive policy-related data itself. With hundreds of brokers operating on thousands of policies, the data transferred between the brokers and management and vice versa is inevitably prone to human-originated errors. The sources of those errors are specified and commented in the sequel. Elimination of those errors by an automated data processing system is a challenging goal of this work.

The control/management task of the decision maker of Fig.1 is to produce decisions \( u(k) \) in order to keep the error \( \epsilon(k) \) as close to zero as possible. We have shown above that even precise specification of the control input \( u(k) \), controlled output \( y(k) \) and measurable/countable disturbance \( d_m(k) \), in terms of determining both sizes of the vectors and their specific elements, is hardly possible here because of the very uncertain and unclear matter of the medium level management environment. Unfortunately, the problem is empowered by the inability to precisely define the set point values for the output reference \( y_d(k) \). In fact, it is not possible to precisely demand for a reference value of e.g. WLR; it should be as low as possible, but the practice is that in some critical cases it would be satisfied when, temporarily, this value is equal to even 100% and it does not increase. Quite similar is with the reference values for total premiums, which clearly do not exist. From business perspective the values should be as high as possible, but in some exceptional situations it might be acceptable when the premiums fall down provided that losses get lower.

In general, a control/management law could be as follows

\[
u(k) = u_f(b)(k) + u_f(f)(k)
= \varphi_1[\epsilon(k)] + \varphi_2[\epsilon(k)]
\]

Where \( u_f(b)(k) \) and \( u_f(f)(k) \) are the feedback and feed forward components of the control vector \( u(k) \) and \( \varphi_1[\epsilon(k)] \) and \( \varphi_2[\epsilon_m(k)] \) are certain (vector) functions which, in the classical control framework, could be obtained on a basis e.g. of some control performance criterion. However, with the above issues of time-varying sizes of the vectors \( y(k) \) and \( d_m(k) \) (and their elements sometimes overlapping), in addition to a blurry determination of the reference components \( y_d(k) \), a formulation and solution of any optimization problem is, most likely, unfeasible, except perhaps of
some trivial, in fact unreal situations. All the above is intended to demonstrate that it is most likely impossible to construct an automated expert decision maker that could effectively do the control/management job as theoretically required in Fig. 1. The extremely complex matter of a decision making process at the medium management level of a large insurance company can only be, most likely, implemented in the advisory, decision support mode, the solution being overwhelmingly accepted in the management practice. In fact, there is no example world-wide that such an automated decision-making system has been implemented. An extremely simple example of a fuzzy-like realization of the decision-making process by a human expert is: if "written loss ratio (WLR) in %, for new and renewed policies, per broker, for the current year to date" is "low" or "low enough" and "total premiums in $, for new and renewed policies, per broker, for the current year to date" are "high" or "high enough" then "no change indecision". Even if the fuzzy-like reasoning [17] could be attractive in possible development of an automated decision making process at the medium management level in a large insurance company, it has still never been implemented in such a company world-wide. One reason is that, with a poor performance of a manual data processing system supporting the decision making process, a human expert could not be quite sure if those "low enough" or "high enough" quantifiers were not falsified. And without them, the above expert, fuzzy reasoning example would be too trivial in the insurance practice.

4.2.2 Location of Automated Data Processing System (ADPS)

The invention constituting the subject of this dissertation is an Automated Data Processing System (ADPS) whose location in the feedback-feed forward control/management system of Fig.4.1 is shown in Fig.4.2. The system collects massive data from the insurer, that is the insurance company and brokers, and performs a series of analytical, Excel-based data validation procedures followed by error correction supporting schemes and presentation of reports on summary of the results (incorporated in the output y(k)). Consequently, the quality of the output y(k), supplied from ADPS to the Decision Maker, is very high as compared to that for manual processing, both in terms of the accuracy and speed of processing. As for ADPS itself, it is composed of a number of Excel implemented formulae representing both feedback (closed-loop) and feed forward (open loop) operations on data.
ADPS – System Interworking's

In order to monitor the effectiveness of targeted marketing, it is desirable to analyze the performance or profitability of the policies generated using the marketing and sales application. Policy data is stored in the policy administration system that is separate from the marketing and sales application executed on the web server. The policy data includes, but is not limited to, policy renewal data, policy premium data, claim data and policy termination data. By combining the policy data with the marketing and sales data, the performance of the policies generated as a result of such campaigns can be measured. The policy performance can be summarized by broker, by campaign, by year, etc.

Fig. 4.3 shows a Microsoft Excel workbook 800 into which the policy data and marketing and sales data are imported for analysis. The workbook 800 has a plurality of worksheets, among which are a data import sheet 804, a set of annual summary sheets 808, a policy summary sheet 812, a campaign summary sheet 816, a broker summary sheet 820, a new policies summary sheet 824 and a broker summary sheet 828 broken down by their campaigns. The data import sheet 804 receives policy data that is inputted for processing. Each of the annual summary sheets 808 looks up data for the appropriate years in the data import sheet 804, validates it and summarizes the projected and actual policy activity for each broker/campaign combination for a particular calendar year. Each of the policy summary sheet 812, the
campaign summary sheet 816, the broker summary sheet 820, the new policies summary sheet 824 and the broker summary sheet 828 broken down by their campaigns references the data processed in the annual summary sheets 808. In turn, the policy summary sheet 812 provides an overall analysis of the performance of the policies generated as a result of all of the campaigns.

The campaign summary sheet 816 summarizes the performance of the policies generated as a result of each campaign by calendar year of issuance. The broker summary sheet 820 summarizes the performance of the policies generated as a result of each broker by calendar year of issuance. The new policies summary sheet 824 summarizes the performance of the policies generated as a result of each campaign and by each broker that are newly issued in the current calendar year. The broker summary sheet 828 broken by their campaigns summarizes the performance of the policies generated as a result of each broker by calendar year of issuance.

Fig 4.3 Schematic diagram of a Microsoft Excel workbook for analyzing the performance of Policies generated as a result of the targeted marketing method
4.2.2.1 Data validation formula

As can be seen, the uniquely created formula 900 uses a set of nested IF (condition, then, else) functions to serially perform a number of checks on the data [10]. In a first condition 902, it is determined whether the number of records in the data import sheet 804 that match the policy with which the formula 900 is associated is not equal to one. The first condition 902 checks all of the policy numbers in the data import sheet 804 using an array calculation. If no records in the data import sheet 804 match the policy number with which the formula 900 is associated, data for the specific policy number is deemed to be missing. Alternatively, if two or more records in the data import sheet 804 match the policy number with which the formula 900 is associated, the data in the data import sheet 804 for the specific policy is deemed to be suspect and unreliable. In both cases, any data that could be returned by a formula may not be accurate and it is desirable to correct the data instead of relying on potentially inaccurate results reported by the workbook 800. As a result, if the first condition 902 is true, a first error message 904 is visually displayed. Data validation formula shows What if Analysis of data about an insurer with manual input of attribute values to make decision regarding monthly premium etc., if used.

\[
=I(F(SUM(IF(R1A1:R2A2=0001002,TRUE=0))=1","Detected policy number error in the System")
\]

904

IF(SUM(IF(R1A1:R2A2=0001002
TRUE=0))=1","Detected policy number error in the System")

908

IF(R1A1:R2A2=0001002 TRUE=0)="Detected policy number error in the System"

910

"Cancelled policy"

922

\[
=I(F(SUM(IF(R8A1:R9A9=0001002,TRUE=0))=1","Incorrect Ex-Date in the System")
\]

914

IF(SUM(IF(R8A1:R9A9=0001002
TRUE=0))=1","Incorrect Ex-Date in the System")

916

SUM(IF(R8A1:R9A9=0001002
TRUE=0))","Incorrect Ex-Date in the System")

Fig 4.4 Showing the What-If-Analysts of data using formulas in the workbook
ADPS model gives an example of application of analytical methodology for automated data processing through Excel, which is used to analyse data to make decisions based on What If conditions. Though ADPS model is simple but cannot handle multiple data sets with different classifications which is used for obtaining optimum decision.

4.2.3 ID3 and C4.5 Techniques

The basic Decision making algorithm ID3 was used for tasks such as learning good game playing strategies for chess end games. Since then ID3 has been applied to a wide variety of problems in both academia and industry. ID3 picks predictors and their splitting values based on the gain in information that the split or splits provide. Gain represents the difference between the amount of information that is needed to correctly make a prediction before a split is made and after the split has been made. If the amount of information required is much lower after the split is made then that split has decreased the disorder of the original single segment. Gain is defined as the difference between the entropy of the original segment and the accumulated entropies of the resulting split segments. ID3 is suitable for simple data sets with less attributes which may not suitable for current big data sets available [44].

ID3 was later enhanced in the version called C4.5. C4.5 improves on ID3 in several important areas such as predictors with missing values can still be used, predictors with continuous values can be used, pruning is introduced, rule derivation. Even C4.5 cannot address problem of multiple attributes analysis in terms of decision making and preserving decision securely.

4.2.4 CART Decision making technique

CART stands for Classification and Regression Trees and is a data exploration and prediction algorithm developed by Leo Breiman et.al. CART technique performs validation of the model and the discovery of the optimally general model. CART accomplishes this by building a very complex tree and then pruning it back to the optimally general tree based on the results of cross validation or test set validation. The tree is pruned back based on the performance of the various pruned version of the tree on the test set data. The most complex tree rarely fares the best on the held aside data as it has been over fitted to the training data.
The CART algorithm is relatively robust with respect to missing data. If the value is missing for a particular predictor in a particular record that record will not be used in making the determination of the optimal split when the tree is being built. In effect CART will utilizes as much information as it has on hand in order to make the decision for picking the best possible split. When CART is being used to predict on new data, missing values can be handled via surrogates. Surrogates are split values and predictors that mimic the actual split in the tree and can be used when the data for the preferred predictor is missing. This technique concentrates on making decision on missing values in data sets on a limited data sets but at the expense of performance as it is statistical based method, and does not addressed any decision making on multiple attribute data sets and how they can be retrieved without error[44]. The above generalized algorithms do not meet the requirements of the client for minimum data set with small no of attributes for effective decision making, under these situations, we have provided an effective decision making algorithm, which will fulfill the optimum needs of the client. To improve the performance of larger data sets for decision making with large no of policies and insurance companies. So, to overcome above limitations we have proposed Multi- Attribute Secure Decision Making Algorithm.

Moreover, security is major concern to maintain integrity for the benefit of insurer which is appropriate for my present study [In the present chapter we have applied and implemented "Selective Secure Field RSA Public key Cryptosystem ",which was proposed in CHAPTER-3]

4.3 Proposed Secure Decision Making Algorithm

Threshold Multi-attribute Secure Decision making algorithm (MASDMA) which provides secured decision making solutions in designing and development of protocols to provide a fair and non-repudiation services to safeguard the transactions in E-Health Insurance System.

Many health care organizations have implemented various types of automated medical record systems which support quick access of Clients record, patient history etc., but does not provide any decision making system which provides suggestions in
selecting Insurance schemes, hospital based on E-Knowledge regarding health care, also security regarding decisions, protocols used during transactions.

4.3.1 Threshold Multi Attribute Secure Decision Making Algorithm (MASDMA)

The MASDMA algorithm works in three phases such as

1. AttributeKeyGeneration: In this phase the public key and private keys are generated which is used to secure the decision.
2. EHDM A (E-Health decision making algorithm): It returns the decision in encrypted format
3. Decrypt Decision: Decrypts the decision generated in phase-2 by algorithm.

MASDMA(P: data, T: table of Attributes for objects)

1. Begin
2. Call AttribKeyGeneration(P);
3. EIHD = Call EHDM A(P, T, D) ED: encrypted Decision, D-default Attribute ie.. root)
4. Call Decrypt Decision(EIHD)
5. End

In phase-1 the following algorithm generates keys for encryption and decryption of data or decision from which the client chooses insurance plan or hospital based on attributes like income, place, number of dependents etc., in data set of client. As there is a possibility of modification of decision by client by a competent authority we ensure security by generating private and public keys in phase-1 which can used by third party or client to view his decision with precision.

Attribute Key Generation(P) (Generates keys in Phase-1)

1. Generate two random large prime numbers p, q with equal size (n/2) bits.
2. Compute \( n = p \times q \) and \( Eu = (p-1) \times (q-1) \).
3. Compute e value such that gcd (e, Eu) = 1 and \( 1 < e < Eu \).
4. Compute d value such that \( d = e^{-1} \mod (Eu) \).
5. Select a Random integer ‘f select field attribute with range h.

6. select the particular attributes values like (Income, Tax, Benefit, Age, Dependents, Occupation) from P to

Compute \( pf = \sum_{i=1}^{k} p_i \); where \( i = 1, 2, \ldots, k \) fields as selected by Client
concatenate \( pf \) add into the in middle of the \( d \) as

i) Find the length of \( d \) and divide the \( d \) into two half's as \( fd \) and \( sd \).

ii) \( pf \) is concatenate to the first half (\( fd + pf \)) and the result concatenate to second half \( sd \) as \( fnld=(fd + pf + sd) \).

iii) \( d = fnld \)

7. **Public key is (h, e, n) and private key is (x, d);**

MASDMA performs two tasks in Phase-2, in first task focuses on providing an effective decision as current competitive world there are number of companies which provides insurance with various policies but the selection of policy is the major problem for the client and the second task is to hide the decision of client from other competitors from modification by encrypting it with \( S^2FRSA \) algorithm which uses key generated in phase-1 based on selective fields to secure decision.

**Phase-2 (Task-1) to generate an effective decision for client.**

**EHDMA: (Decision Making Algorithm)**

1. create a node \( N; \) (\( N = \) Insurance data record/Policy data record)
2. if \( P[i] \) (sample training data) are all of the same class(Insurance Scheme/Hospital) then

2.1 \( EIHD=Encrypt(N) \); where the \( N \) is the final decision

2.2 Return \( EIHD(\)decision\);

3. if \( P[i] \) is Null then

3.1 \( N = \) Best Insurance/Hospital with default decision(majority selection)

3.2 \( EIHD=Encrypt(N) \)

3.3 Return \( EIHD(\)decision\);

70
4. Select test-attribute, the attribute among attribute-list with the highest Information gain; by using Entropy
5. Label node N with test-attribute (i.e., Income, Tax_Exemption etc..)
6. for each known value P[i] of test-attribute;
7. Construct a branch from node N for the condition test-attribute = P[i];
8. Let S_i be the set of samples (subset of P[i]) in samples for which test-attribute = P[i];
9. If S_i is empty then
   9.1.1 N= Best Insurance/Hospital with default decision (majority decision chosen by Clients already)
   9.1.2 EIHD=Encrypt (N)
   9.1.3 Return EIHD (decision);
   else
   9.2.2 attach the node returned b as null decision

**Phase-2(Task-2) to encrypt the decision of client for decision integrity**

**Encryption Process**

The Sender or in this case Third party which provides the service does the following:

1. Obtain the recipient public key (h, e, n)
2. Select a Random integer f,h (where f is starting field and h is the range 1…f_n)
3. calculate \( x = (f^h)^e \mod (n) \);
4. Message will be represented as positive integer \( m < n \).
5. Encrypt the message \( D_m \) with the public key \( (e, n) \) like
   \( EHIC_d = D_m^e \mod (n) + f^h \).
6. Sends Encrypted decision \( C_d \) regarding Insurance Scheme/Hospital Selection
   \( EHIC_d \) to recipient.

In Phase-3 to view his decision client should use his private key to view his decision which is in encrypted format by decrypting it which is given below.
Decryption Process (Views clients decision in Phase-3)

To view the decision Client does the Following:

1. Uses the private key (x, d) to decrypt the Decision EHIC\textsubscript{d} as follows

2. Before going to decryption process, the x will be decrypted as \( y = x^d \mod (n) \)

3. Now finally decrypt the Cipher text as \( \text{PCD} = (EHIC_d)^d \mod (n) - y \) to the clients decision PCD.

4.4 Implementation of Proposed Multi-Attribute Secure Decision Making Algorithm (MASDMA)

```java
import java.io.*;
import java.sql.*;
import java.util.*;

public class MASDMA
{
    public MASDMA()
    {
    }
    public static void main(String args[])
    {
        Connection con;
        ResultSet rs, rs1, rs2, rs3, rs4, rs5, rs6;
        PreparedStatement ps, ps1, ps3, ps4, ps5, ps_I;
        String cname, ctype, str, city;
        int noofemp, noofclients, noloc, noofyears, profit, successrate[], rcnt, hrcnt, i, j;
    }
```
String compname[];
String itaxbnft,ioccupation,ifullcashbnft,imedicinebnft;
String isurgicalbnft,insurancecvgtoalld,ipremiumperyear,ictimy;
intiage, idependents, idiagnosistimes, ipremiumamount;

String taxbnft, occupation, fullcashbnft, medicinebnft;
String surgicalbnft, insurancecvgtoalld, premiumperyear;
String decision, insname, tempcname[], hnames[];
intiage, dependents, diagnosistimes, premiumamount;
int temp=0, count;
ResultSetMetaData rdmt;
con=null;
rs=null;
rs1=null;
rs2=null;
rs3=null;
ps3=null;
rs4=null;
rs6=null;
ps4=null;
ps=null;
ps1=null;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
try {
Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/healthportal?user=hospitalinsurance&password=hinsurance2013");

System.out.println("Enter the taxbnft(yes/no):");
itaxbnft=br.readLine();

System.out.println("Enter Your age:");
age=Integer.parseInt(br.readLine());

System.out.println("Enter your occupation(govt/any):");
ioccupation=br.readLine();

System.out.println("Enter number of dependents(1/2/3/4):");
idependents=Integer.parseInt(br.readLine());

System.out.println("Enter the full cashbnft(yes/no):");
ifullcashbnft=br.readLine();

System.out.println("Enter the medicinebnft(yes/no):");
imedicinebnft=br.readLine();

System.out.println("Enter the full surgicallbnft(yes/no):");
isurgicallbnft=br.readLine();

System.out.println("Enter the how many diognositimes you want(50/100/150):";)
idiagnositimes=Integer.parseInt(br.readLine());

System.out.println("Enter the insurance coverage to all dependents(yes/no):");
iinsurancecvgtoalld=br.readLine();

System.out.println("Enter premium amount per year(2000/3000/4000/5000/6000/7000):");
ipremiumamount=Integer.parseInt(br.readLine());

System.out.println("nEnter premium per year(once/twice):");

ipremiumperyear=br.readLine();

System.out.println("nEnter your city
(delhi/banglore/kolkata/tirupati/hyderabad/nellore"): ");

city=br.readLine();

ps1=con.prepareStatement(" select inscompanyname from insavailablecities where availablecity=?");

ps1.setString(1,city);

rs=ps1.executeQuery();

rs.last();

rcnt=rs.getRow();

rs.close();

compname=new String[rcnt];

successrate=new int[rcnt];

tempcname=new String[rcnt];

rs1=ps1.executeQuery();

i=0;

while(rs1.next())
{

compname[i]=rs1.getString(1);

i++;
}

rs1.close();

75
j=0;
for(i=0;i<cnt;i++)
{
    ps=con.prepareStatement("select
        inscompanyname,companytype,nooflocations,noofclients,noofemployees,profit,noofyears,successrate from insurancecompanies where inscompanyname=?");
    ps.setString(1,compname[i]);
    rs2=ps.executeQuery();
    while(rs2.next())
    {
        noloc=rs.getInt(3);
        noofclients=rs.getInt(4);
        noofemp=rs.getInt(5);
        profit=rs.getInt(6);
        noofyears=rs2.getInt(7); /*
        successrate[j]=rs2.getInt(8);
    }
    j++;
}
System.out.println("nTarget Attribute: InsCompanyname");
System.out.println("nOther Attributes...");
System.out.println("ninscompanyname,companytype,nooflocation,noofclients,noofemployees,profit,noofyears,successrate");
System.out.println("nSuccessRate---->");
j=0;
for(i=0;i<rcnt;i++)
{
 if(successrate[i]<=40&&successrate[i]>20)
 {
 System.out.println("it"+successrate[i]+"% ="+compname[i]+"... Preferable");
 tempcname[j]=compname[i];
 j++;
 }
 else
 System.out.println("it"+successrate[i]+"% ="+compname[i]+"... Not Preferable");
 }
 rs2.close();

DriverManager.getConnection("jdbc:mysql://localhost:3306/healthportal?user=hospitalinsurance&password=hinsurance2013");

ps_1=con.prepareStatement("select *from insurances");

//ps_1.setString(1,disid);

rs=ps_1.executeQuery();

while(rs.next())
{
  insname=rs.getString(1);
  taxbnft=rs.getString(2);
  age=rs.getInt(3);
  occupation=rs.getString(4);
  dependents=rs.getInt(5);
fullcashbnft = rs.getString(6);
medicinebnft = rs.getString(7);
surgicalbnft = rs.getString(8);
diagnosistimes = rs.getInt(9);
insurancecvgtoalld = rs.getString(10);
premiumamount = rs.getInt(11);
premiumperyear = rs.getString(12);

if(age >= iage)
{
    count = 0;
    if(taxbnft.equals(itaxbnft))
    {
        count++;
    }
    if(dependents >= idependents)
    {
        count++;
    }
    if(fullcashbnft.equals(ifullcashbnft))
    {
        count++;
    }
    if(medicinebnft.equals(imedicinebnft))
{  
count++;
}

if(surgicalbnft.equals(isurgicalbnft))
{
  count++;
}

if(diagnosistimes>=idiagnosistimes)
{
  count++;
}

if(insurancecvrgtoalld.equals(iinsurancecvrgtoalld))
{
  count++;
}

if(premiumamount>=ipremiumamount)
{
  count++;
}

if(premiumperyear.equals(ipremiumperyear))
{
  count++;
}

if(count==9)
{ System.out.println("\n (100%) Suitable Insurance Scheme is: "+insname);
}
else if(count!=9&& temp==0)
{
System.out.println("\n NO (100%) Suitable Insurance Schemes are available based on your options");
System.out.println("\nThe following Insurance Schemes are Nearest to your options");
temp++;
}
if(count==8)
{
System.out.println("\n(90%) Suitable Insurance Scheme is: "+insname);
}
if(count==7)
{
System.out.println("\n (80%) Suitable Insurance Scheme is: "+insname);
}
if(count==6)
{
System.out.println("\n (70%) Suitable Insurance Scheme is: "+insname);
}
if(count==5)
{ 
    System.out.println("\n (60%) Suitable Insurance Scheme is: " + insname);
});

System.out.println("\n Preferable Insurance Company wise Insurances.....\n");
ps4 = con.prepareStatement("select insurancename from companyinsurances where inscompanyname=?");
for(i=0; i<tempcename.length; i++)
{
    if(tempcename[i]!=null)
    {
        System.out.println(tempcename[i] + "-----");
        ps4.setString(1, tempcename[i]);
        rs4 = ps4.executeQuery();
        while(rs4.next())
        {
            System.out.println("\t\t" + rs4.getString(1));
        }
    }
}
ps4.close();
rs4.close();
String srate;
ps3 = con.prepareStatement("select hospitalname from hospitalcities where city=?");
ps3.setString(1, city);
rs3=ps3.executeQuery();
rs3.last();
hrcnt=rs3.getRow();
rs3.close();
hnames=new String[hrcnt];
rs5=ps3.executeQuery();
System.out.println("nPossible Hospitals based on your city and your insurance Schemes");
System.out.println("n Target Attribute is... Hospitalname");
System.out.println("n Success rate is 'high or medium' decision=yes");
i=0;
while(rs5.next())
{
    hnames[i]=rs5.getString(1);
i++;
}
ps5=con.prepareStatement("select successrate from hospitals where hospitalname=?");
for(i=0;i<hnames.length;i++)
{
    ps5.setString(1,hnames[i]);
    rs6=ps5.executeQuery();
    while(rs6.next())
    {
}
srate=rs6.getString(1);

if(srate.equals("high")||srate.equals("medium"))
System.out.println(hnames[i]+"... preferable");
else
System.out.println(hnames[i]+"... NOT preferable");
}
}
rs6.close();
rs5.close();
ps3.close();
con.close();
}catch(Exception ex)
{
ex.printStackTrace();
}}

import java.math.BigInteger;
.import java.util.*;
.import java.io.*;

public class SFRSA1 {

    staticBigInteger cipher[];
    staticinti,j=0, k,count=0,c=0;

    staticBigInteger g, e = new BigInteger("1"), o = new BigInteger("1"), n, d = new BigInteger("1"), Eu;

    83
static BigInteger p, q,u,eval,bi;
static BigInteger Me, M1, C,tempe,x,y,ua,ewords[];
static long startTime;
static long stopTime,ptime;
static Random rng;

static int a, tlen, wno[], wnumber=0, cp=0;
static String str="", wstr="";
static String words[], swords[];
static String wc="", fans="";

public static BigInteger[] genKeys(int len) {

rng=new Random();
tlen=8;
do
{
  p=BigInteger.probablePrime(len, rng);
  q=BigInteger.probablePrime(len, rng);
  n  = p.multiply(q);
  Eu = (p.subtract(o)).multiply(q.subtract(o));

}
for(i=0;i<20;i++)
{
    tempe=BigInteger.primeProblem(tlen,rng);

    if(((Eu.gcd(tempe)).intValue()==1))
    {
        e=tempe;
        //System.out.println("tempe:"+tempe);
        break;
    }
}

e=new BigInteger("137");
d = e.modInverse(Eu);

}while((p.equals(q)));

a=8;
u=new BigInteger("6");

BigInteger[] keys = new BigInteger[5];
keys[0] = n;
keys[1] = Eu;
keys[2] = e;
keys[3] = d;
keys[4]=u;

return keys;

}  

public static BigInteger enCrypt(BigInteger e, BigInteger n, BigInteger M) {
    returnM.modPow(e, n);

}  

public static int deCrypt(BigInteger C, BigInteger n, BigInteger d) {
    returnC.modPow(d, n).intValue();

}  

public static void main(String[] args)  
{

try

    {
        int size;
           
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        Scanner cin = new Scanner(System.in);
        String Msg, Plaintxt=""", S, Ci;

        int M[], Ptxt[], k;
        String ans="";
System.out.print(" Enter the p, q size : ");

size=Integer.parseInt(br.readLine());

System.out.print(" Enter the Plain Text : ");

Msg=br.readLine();
Ptxt = new int[Msg.length()];

System.out.println("\nKey Generation Phase...\n");

startTime=System.currentTimeMillis();

BigInteger[] key = genKeys(size);

stopTime=System.currentTimeMillis();

ptime=stopTime-startTime;

System.out.println("\nKeys Generation Time:"+ptime+" Milli Seconds");

n = key[0];

Eu = key[1];

e = key[2];

d = key[3];

ua=u.pow(a);

x=ua.modPow(e,n);

//System.out.println("ua: \t"+ua);

System.out.println("n: \t"+n);

System.out.println("n (bits):\t"+n.bitLength());

System.out.println("Eu: \t"+Eu);

System.out.println("e: \t"+e);

System.out.println("d: \t"+d);
```java
StringTokenizer token = new StringTokenizer(Msg);

while (token.hasMoreTokens())
{
    count++;
    str = token.nextToken();
}

words = new String[count];
swords = new String[count];
wno = new int[count];
i = 0;
token = new StringTokenizer(Msg);
while (token.hasMoreTokens())
{
    str = token.nextToken();
    words[i] = str;
    //System.out.println((i+1)+"\t"+words[i]);
    i++;
}
System.out.println("\n\nEncryption phase........................\n");
startTime = System.currentTimeMillis();
try{
```
for(i=0;i<count;i++)
{
    Msg=words[i];
    M = new int[Msg.length()];
    cipher=new BigInteger[Msg.length()];
    for (k = 0; k <M.length; k += 1)
    {
        M[k] = (int) Msg.charAt(k);
        S = "" + M[k];
        Me = new BigInteger(S); //Ascii value of each Character
        C = (enCrypt(e, n, Me)); //Encrypted text for each Character based on Ascii value;
        C=C.add(ua);
        fans=fans+C.toString(16)+"h";
        ans=ans+C.toString(16)+"h";
        cipher[k]=C;
    }
    swords[j]=ans;
    ans="";
    j++;
}
y=x.modPow(d,n);
//System.out.println("y:\t"+y);
}catch(Exception ex){System.out.println(ex.toString());}

stopTime=System.currentTimeMillis();

ptime=stopTime-startTime;

System.out.println("nEncryption Time:"+ptime+" Milli Seconds");

for(k=0;k<Msg.length();k++)
System.out.print(cipher[k].toString(16));

System.out.println("n");

for(i=0;i<count;i++)
{
    System.out.print("nEnter your Field numbers to decrypt :");
    wnumber=Integer.parseInt(br.readLine());

    if(wnumber>count)
    {
        break;
    }
    else{
        cp++;
        wno[i]=wnumber;
        wstr=wstr+""+wno[i];
    }
}

BigIntegersf=new BigInteger(wstr);

String sd=d.toString();
intldlen=sd.length();

inhlen=dlen/2;

String fsbstr=sd.substring(0,hlen);

String ssbstr=sd.substring(hlen,sd.length());

String fstr=fsbstr.concat(wstr);

String ffsr=fstr.concat(ssbstr);

d=new BigInteger(ffsr);

System.out.println("nPrivate key :."+d);

sd=d.toString();

intrlen=cp+hlen;

fsbstr=sd.substring(0,hlen);

ssbstr=sd.substring(rlen,sd.length());

String fffsr=fsbstr.concat(ssbstr);

d=new BigInteger(fffstr);

int temp;

System.out.println("n Decryption Phase........\n");

startTime=System.currentTimeMillis();

if(ua.equals(y))
{
    for(i=0;i<count;i++)
    {
        temp=wno[i];
        if(temp!=0)
        {
            temp=temp-1;
        }
    }
}
StringTokenizer st = new StringTokenizer(swords[temp], "h");

k = 0;

while (st.hasMoreTokens())
{

eval = new BigInteger(st.nextToken(), 16);

eval = (eval.subtract(y));

Ptxt[k] = deCrypt(eval, n, d);

k++;
}

PlainTxt = new String(Ptxt, 0, words[temp].length());

System.out.println("\nWord "+(temp+1)+" : "+PlainTxt+"\n");
}
}

} else
{

System.out.println("\n'rc' and 'c' are not equal");

return ;
}

stopTime = System.currentTimeMillis();

ptime = stopTime - startTime;

System.out.println("\nDecryption Time:" + ptime + " Milli Seconds");

} catch (Exception ex) {

System.out.println(ex.toString());
}

}
4.5 Test Results for Proposed Multi-Attribute Secure Decision Making Algorithm

The MASDMA algorithm takes the input from the user as taxbenft, age, occupation, dependents, surgical-benft, medical-benft etc and considers the hospital attributes and calculates the entropy value on sample records.

**Output 1:**

![Output Image]

*Fig 4.5 Showing possible decisions for client due to MASDMA algorithm and Keys generation as given in Phase-1*
Fig 4.6 Showing results of MASDMA algorithm for sample data of Client for selection of Insurance in Phase-2
Fig 4.7 Showing results of MASDMA algorithm with selected company and hospital for client showing in plain text in phase-3
Fig 4.8 Showing possible decisions for client due to MASDMA algorithm and keys generation as given in Phase-1
Fig 4.9 Showing results of MASDMA algorithm for sample data of Client for selection of Insurance for Client-2 in Phase-2
Fig 4.10 Showing results of MASDMA algorithm which shows clients choice from encrypted format in plain text in Phase-3
The above results show the proposed MASDMA can be used in Health Insurance sector which will secure privacy of decisions regarding Clients of Insurance companies and hospitals when Third party is involved in providing health insurance services.

4.6 Conclusion

In this chapter, decision making and its importance is discussed with examples which show how it can be used in Health insurance sector. In current health care market there are number of health insurance companies which provide various policies, and its selection is difficult for client to choose which hospital and insurance scheme/company is suitable for requirements of the client and also how decision should be secured from viewing by other competitors. So to address above problems in this chapter, it is proposed that the Threshold Multi Attribute Secure Decision Making Algorithm (MASDMA) and implemented in Java. The proposed MASDMA algorithm is verified with sample data sets and results are generated.