

## *Adaptive Small Scale Database Design Using Objects Concept*

Dr. Taif S. Hassan

Computer Science Department/Al-Ma'moon  
University College

### **Abstract:**

Database plays a key role in the most modern applications. There are many software packages for programming and designing a database system. These packages have flexible and powerful tools for the designers and programmers. In spite of these tools, a problem arise in simple and special database design, where they covert a wide range of applications. Solving the problem needs simple and special design requirements. The available software such as Microsoft SQL Server, MySql, and Oracle could not provide suitable solution to these needs, because they have some complexity in designing the DataBase. This paper presents an efficient road map for designing these databases using the important concept of object oriented programming.

*Keywords: Object, Database, MySql, RSA, Oracle.*

## تصميم نظام متكيف لقواعد البيانات الصغيرة باستخدام مفهوم الكائنات

د. طيف سامي حسن

قسم علوم الحاسوب-كلية المأمون الجامعة

### **المستخلص:**

تؤدي قواعد البيانات دوراً رئيسياً في معظم التطبيقات الحديثة، وهناك العديد من الحزم البرمجية التي تستعمل لبرمجة أنظمة قواعد البيانات وتصميمها. وتمتلك هذه الحزم أدوات مرنة وكفاءة للمصمم وكذلك المبرمج. بالرغم من هذه الأدوات تظهر لنا مشكلة ولاسيما في التصميمات البسيطة والخاصة لقواعد البيانات، والتي تغطي مدى واسعاً من التطبيقات. تمتلك هذه التطبيقات متطلبات سهلة وخاصة. إن البرمجيات المتوافرة كالبرنامج Microsoft Sql Server والبرنامج MySql والبرنامج Oracle لا توفر حلول مناسبة لهذه المشكلة وذلك لتميزها بالتعقيد في ايجاد التصميم الجيد لقاعدة البيانات. إن البحث

يقدم خارطة طريق كفوءة لتصميم قواعد البيانات السالفة الذكر وذلك بتبني المفاهيم المتطورة للبرمجة الكائنية.

## 1. Introduction

In recent days, database system serves a lot of applications. The database systems and packages have witnessed many developments and adaptations in order to suit modern technology requirements, especially the internet. The old database is divided into many types: network database, hierarchical database, relational database, and the object database. The object database was invented in 1960, but it is not commonly used because of the limitation of the packages that support it. This paper presents efficient models that are used to design object oriented database, in order to overcome the limitation occur in the commonly used database[1].

## 2.Database

A database is an organized collection of data, typically, in a digital form. The data are typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies).

The term database is correctly applied to the data and their supporting data structures, and not to the database management system (DBMS). The database data collection with DBMS is called a database system.

## 3.Early Data Models

These models were popular in the 1960s, 1970s, but nowadays they can be found primarily in old legacy systems. They are primarily characterized by being navigational with strong connections between their logical and physical representations, and deficiencies in data independence.

### *Hierarchical Model*

The Hierarchical model has different record types (representing real - world entities) which are embedded in a predefined hierarchical (tree-like) structure. This hierarchy is used as the physical order of records in storage. Record access is done by

navigating through the data structure using pointers combined with sequential accessing[1,2].

This model has been supported primarily by the IBM IMS DBMS, one of the earliest DBMSs. Various limitations of the model have been compensated for in later IMS versions by additional logical hierarchies imposed on the base physical hierarchy.

### *Network Model*

In this model, a hierarchical relationship between two record types (representing real-world entities) is established by the set construction. A set consists of circular linked lists where one record type, the set owner or parent, appears once in each circle, and a second record type, the subordinate or child, may appear multiple times in each circle. In this way, a hierarchy may be established between any two record types, e.g., type A is the owner of B. At the same time, another set may be defined where B is the owner of A. Thus all the sets comprise a general directed graph (ownership defines a direction), or network construction. Access to records is either sequential (usually in each record type) or by navigation in the circular linked lists.

This model is more general and powerful than the hierarchical, and has been the most popular before being replaced by the Relational model. It has been standardized by CODASYL. Popular DBMS products that utilized it were Cincom Systems' Total and Cullinet's IDMS.

### *Inverted File Model*

An inverted file or inverted index of a first file, by a field in this file (the inversion field), is a second file in which this field is the key. A record in the second file includes a key and pointers to records in the first file where the inversion field has the value of the key. This is also the logical structure of contemporary database indexes. The related Inverted file data model utilizes inverted files of primary database files to efficiently directly access needed records in these files.

Notable for using this data model is the ADABAS DBMS of Software AG, introduced in 1970. ADABAS has gained considerable customer base and exists and is supported until today. In the 1980s it has

adopted the Relational model and SQL in addition to its original tools and languages.

### *Relational Model*

The **relational model** is a simple model that provides flexibility. It organizes data based on two-dimensional arrays known as **relations**, or tables as related to databases. These relations consist of a heading and a set of zero or more **tuples** in arbitrary order. The heading is an unordered set of zero or more attributes, or columns of the table. The tuples are a set of unique attributes mapped to values, or the rows of data in the table. Data can be associated across multiple tables with a key. A key is a single, or set of multiple, attribute(s) that is common to both tables. The most common language associated with the relational model is the Structured Query Language (**SQL**), though it differs in some places[2].

### *Entity-Relationship Model*

#### *Object Model*

In recent years, the **object-oriented** paradigm has been applied in areas such as engineering and spatial databases, telecommunications and in various scientific domains. The conglomeration of object oriented programming and database technology led to this new kind of database. These databases attempt to bring the database world and the application-programming world closer together, in particular by ensuring that the database uses the same **type system** as the application program. The aim is to avoid the overhead (sometimes referred to as the **impedance mismatch**) of converting information between its representation in the database (for example as rows in tables) and its representation in the application program (typically as objects). At the same time, object databases attempt to introduce key ideas of object programming, such as **encapsulation** and **polymorphism**, into the world of databases.

A variety of these ways have been tried for storing objects in a database. Some products have approached the problem from the application-programming side, by making the objects manipulated by the program **persistent**. This typically requires the addition of some kind of query language, since conventional programming languages do not provide language-level functionality for finding

objects based on their information content. Others have attacked the problem from the database end, by defining an object-oriented data model for the database, and defining a database programming language that allows full programming capabilities as well as traditional query facilities.

### *Object Relational Model*

### *XML as a Database Data Model*

## 4.Object Oriented Database

In the late 1980s and early of 1990s, venders have faced complex data requirements, such requirements are so difficult to be handled by using standard relation as database system. Some of these complex data forms are: graphical structure, audio, video, and some other have a complex structure.

Object oriented is a set of design and development principles based on conceptual autonomous computer structure known as object. Each object represents real world entity with the ability to interact with it and with other objects[3].

There are some important related remarks to the discipline of object-oriented database:

1. **OID (Object Identify):** It is a unique identifier that identifies object and can't be re-used, or changed (like, the primary key in the relational database).
2. **Attribute (Instance Variable):** Tuble (same as field in relational database), such as Age, Name.
3. **Object State:** A set of values that the object attribute has at a given time (record at given time).
4. **Method:** like a procedure that has a name and body (code). In order to implement a method, a user sends message to declare the execution for a specific method. This will enforce the encapsulation feature; the object can't be accessed directly; it is a way for ensuring object integrity.
5. **Class:** Objects that share common characteristics are grouped into a class (similar attribute and behavior methods) and respond to the same message
6. **Protocol:** Class collection of messages constitutes object or class protocol. This is the public object aspect.

7. Relation could be done by the containing class into another a class (the first is sub class, while the second is super class). It could also be done by using the inheritance feature for the object-oriented approach.

## 5.The Object Database Design

Object-oriented approach is based on modern concept that gives the programmer a great deal of flexibility, simplicity, security, and state of art in modeling and representing complex structure. Object oriented database offers the programmer the ability to create and define the suitable data structure, this structure can be defined as object, which contains members of various data type (could be the normal data type or other objects), and the members methods which can be used to implement a wide range of activities. The object oriented database is also characterized by the object-oriented feature (encapsulation, polymorphism, and information hiding).

In the object-oriented database, the “attribute” term is similar to “field” term in the relational database, the term “object” is similar to the “record” term, while “class” term is similar to the “table” term. The relationship in the object oriented database is implemented by designing the super class (container class) and sub class (contained class) and by utilizing the “inheritance” feature among classes (tables).

In the market, till now there is no standard software for object oriented database, but there are some database application softwares that support some features of the object oriented approach. The main available software is Oracle database server.

## 6.Object Database Advantages

- Database Independance: Database objects are built from the ground up to be database independent . They do not rely on the features of a specific underlined database to provide their functionality. This allows an application to be built using database objects to be ported from one database platform to another in literally moments, providing the potential for great scalability.
- Automatic Connection Handling:
- Declarative Referential Integrity: By embedding access logic directly in the database object, you can achieve the same advantages as using stored procedures for accessing a database

without the platform-dependence. Referential integrity becomes database independent, and complex relationships between database objects become portable. For example, business rules can be integrated into the database object, so that all applications accessing the object are assured of following the established rules.

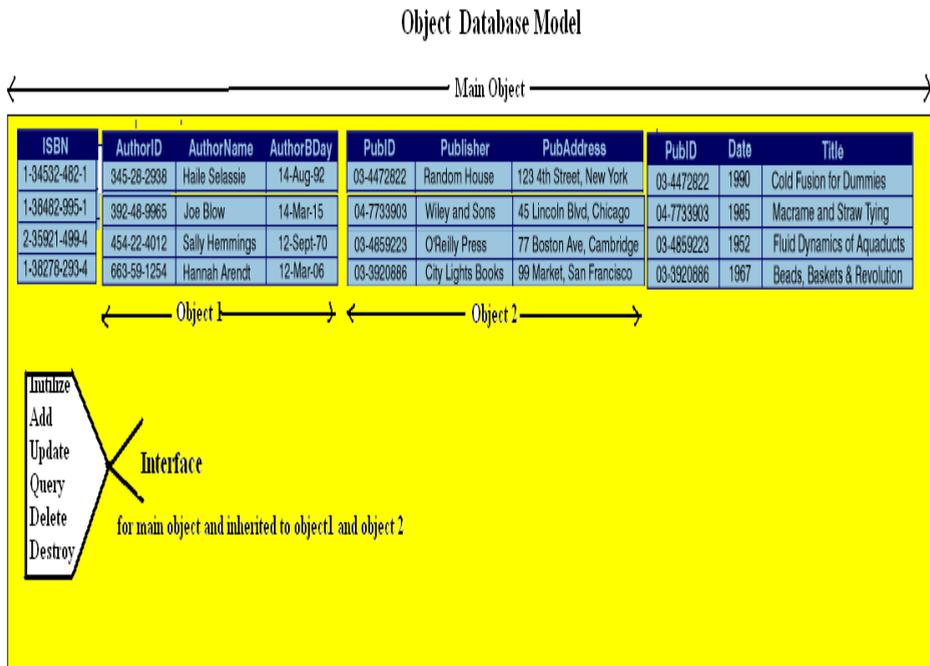
- **Multi-Level Validations:**
- **Avoiding embedded SQL:** Using database objects allows an application to avoid embedded SQL and its maintenance difficulties and system dependencies. Applications simply interact with other Java objects, allowing the design of a system to be entirely object-oriented.
- **Security, and much more:** Every interaction with a database object can be secured, and the security data is easily maintainable via Expresso's built-in capabilities.

## 7.The Adopted Design

Our design consists of the following:

1. **The Main Object:** it consists of
  - a. **Data section:** this section contains the main data type. There are many types such as
    1. Short integer data type.
    2. Integer data type.
    3. Float data type.
    4. Double data type.
    5. Character data type.
    6. String data type.
    7. Array of any data type.
    8. Structure contains a collection of data types
    9. Array of structures.These configurations could suit any database problems.
  - b. **Methods:** the methods is used for implementing the main database operation. They are:
    1. **Initialize method:** the initialize method includes constructor for deserializing the object and a function `GetObjectData` which describes how to serialize the object. The deserializing process load the data section members of the class with the information stored in file at media.
    2. **Add method:** is used to add new record to class.

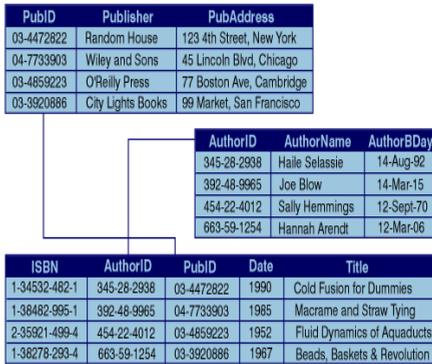
3. **Update method:** is used to update record.
  4. **Query method:** is used to query for specific records.
  5. **Delete method:** is used to delete specific records.
  6. **Destroy method:** is a destruction method that is used to serialize the class data in which the data members are stored in disk .
2. **The sub object:** resemble to the table in the relational database. The sub object inherits the method of the main object (interface).
- Also the sub object will contain also sub a object(with its data and methods)
  - The hierarchy of the final model will suit the existent data.
- Figure (1) shows object database model



Fig(1) Object Database Models

Also Figure(2) shows relational and object database models.

Hypothetical Relational Database Model



Object Database Model

ISBN	AuthorID	AuthorName	AuthorBDay	PubID	Publisher	PubAddress	PubID	Date	Title
1-34532-482-1	345-28-2938	Haile Selassie	14-Aug-92	03-4472822	Random House	123 4th Street, New York	03-4472822	1990	Cold Fusion for Dummies
1-36482-995-1	392-48-9965	Joe Blow	14-Mar-15	04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	Sally Hemmings	12-Sept-70	03-4859223	O'Reilly Press	77 Boston Ave, Cambridge	03-4859223	1952	Fluid Dynamics of Aqueducts
1-36278-293-4	663-59-1254	Hannah Arendt	12-Mar-06	03-3920886	City Lights Books	99 Market, San Francisco	03-3920886	1967	Beads, Baskets & Revolution

Fig(2) The Relational and Object Database Models

## 8.Communication with Remote Object

One common way to move groups of multiple data values between systems on network is to create a class that contains all the data, along with a specific method for converting the data into a byte array. The basic class contains variables for the data elements used in the communication.

```
class Employee
{
    public int EmployeeID;
    public string LastName;
    public string FirstName;
    public int YearsService;
    public double Salary;
```

```

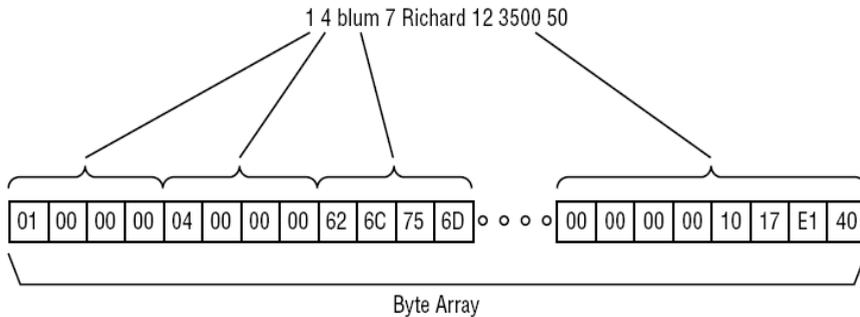
public int LastNameSize;
public int FirstNameSize;
public int size;
}
    
```

Here, the class Employee can be considered similar to a record, with the variable representing the fields in the record. Each instance of the class represents a record in the database.

Because the two string elements can have variable lengths, you should include additional elements to define the size of those elements. This is comparable to the variable text field method.

Eventually, a data element is created to hold the size of the total byte representation of the class instance- again a necessity because the class instance itself will be a variable length as illustrated in Figure(3), converting all field elements into byte stream, suitable for sending out on the network. The NET classes that are used in creating network programs. Whereas the Socket class allows to manually create network programs and with the classes

1. TcpClient
2. TcpListener.
3. UdpClient for UDP protocol.



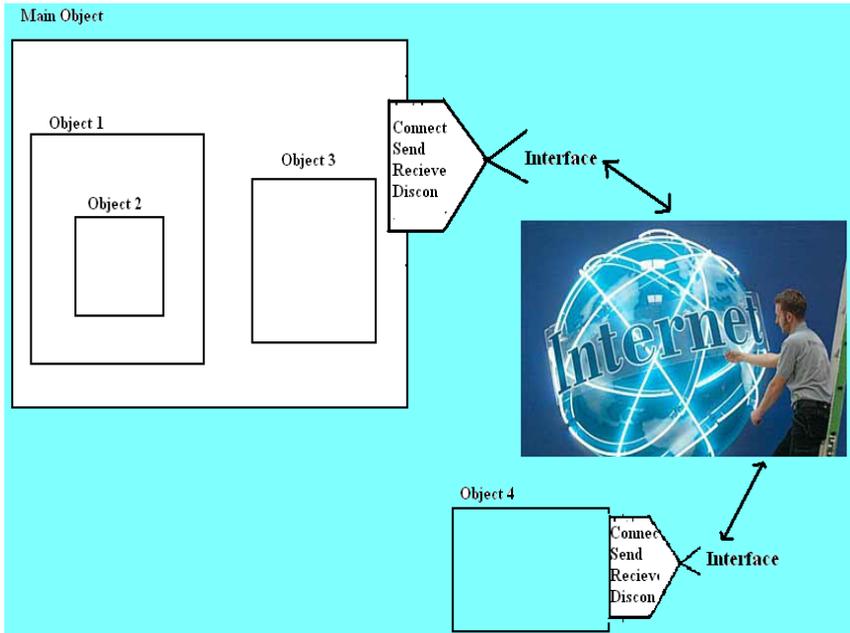
Fig(3) The variable data type Byte Stream

The adopted model has the ability to import and export its data to other objects through the internet and the http protocol with simple steps. The object has the following methods for communications [3]:

- Connect: connecting to the remote object.

- **Send:** send data to the remote object.
- **Receive:** receive data from remote object.
- **Disconnect:** disconnect from remote object.

It could send any object or sub object data through the four methods that could be inheriting from the main object as is shown in Figure(4).



Fig(4) Interfacing with Remote Object

## 9.Storing Object

The Object Serialization enables us to write any complex object directly to a file stream without converting values of individual properties into a text. It could make the data written, to the disk, at least not human readable. To enable users to read the data files, they have to use our program.

### 9.1The.NET and Object Serialization

Now what does .NET framework provide us with to do Serialization of Custom-build class objects? With the advent of any

programming language, developers would first look whether it is an OOP. If so, how to create our classes with our properties and functions? And then comes inheritance and all others.

## 9.2 Serialization Explained

Serialization is the process of converting complex objects into a stream of bytes for storage. Deserialization is its reverse process, that is unpacking stream of bytes to their original form. The namespace which is used to read and write files is System.IO. For Serialization we are going to look at the System.Runtime.Serialization namespace. The ISerializable interface allows you to make any class Serializable.

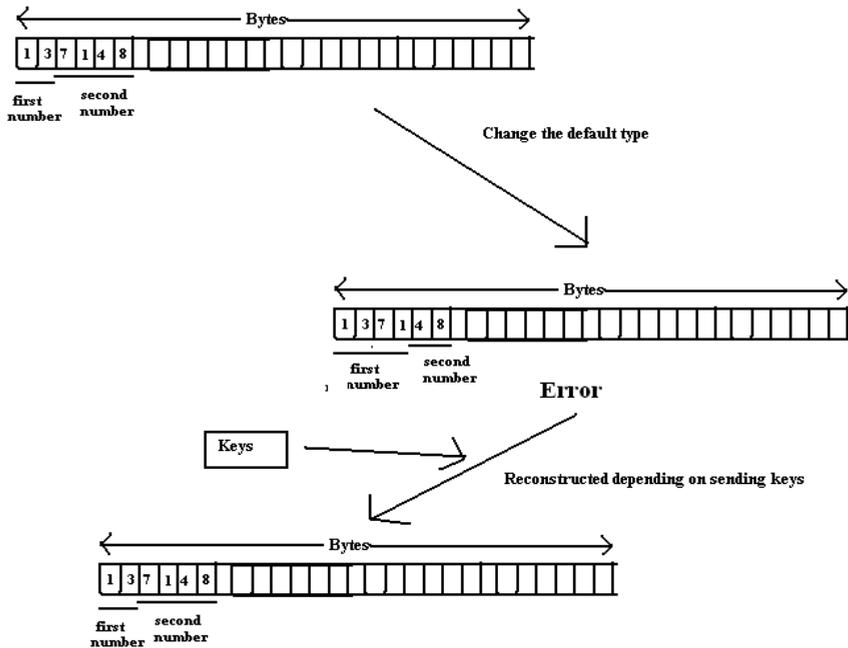
Here are the following steps that we are going to do to create a serializable class and test it.

- Create a custom class named such as Employee and assign properties.
- Define the serialization functions.
- Create a main class and instantiate our Employee class.
- Serialize the object to a sample file.
- Deserialize the values by reading it from the file.

## 10.Object Security

Before storing data, one of the encryption methods must be applied to the data stream. There are many methods that could be used for the encryption step, but a simple and flexible suitable algorithm could be applied[5,6]. The algorithm is shown in Figure (5):

1. Changing the type of data when storing them(ex integer with ASCII)
2. Exchange between objects body, but there is a map that determines the correct position which must be followed when reconstructing the original data.



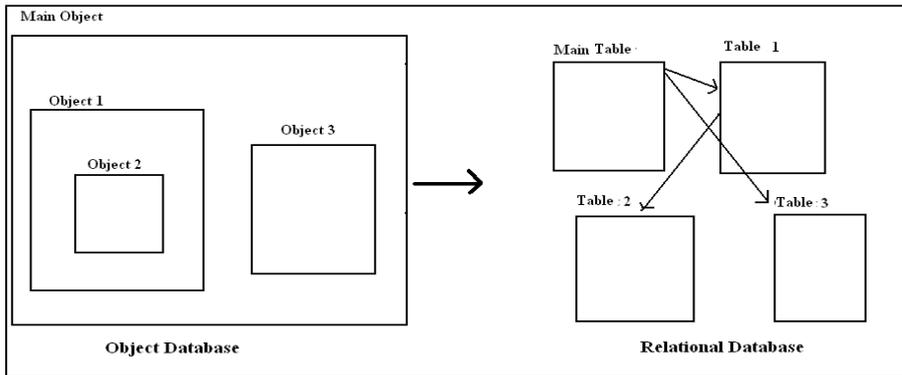
Fig(5) The encryption Diagram

### 11. Interfacing with Microsoft Access

Microsoft Access is a simple database software. It covers a wide range of application due to its simplicity, flexibility, and availability. There is a great need for our database object to communicate with Microsoft Access (export data to or import data from). There are specific routines that handle the interfacing with Microsoft Access. They are[7]:

- Export to: convert the object data to related relational table in Access.
- Import from: convert relational Access relational tables to its related object tables.

The strategy is shown below in the following figure:



Fig(5) Converting from Object to Relational Database

## 12.Result and Implementation

The adopted model is suited for many users and a wide range of application problems. Simple software system was built in order to implement the model using Visual C# and for the College problem.

The system is simple and efficient for responding to the problems that are difficult in representing the relational database using any software such as Microsoft Access, Microsoft SQL Server, etc.

The model objects and their properties are shown in the following tables:

Table(1) The Adopted Database Problem

Class	Member Data	Type
College	College_Name	String
	Department	Class
Department	Department_Name	String
	Stage	Class
Stage	Stage_Name	String
	Subject_Name	String
	Student	Class
	Date	Date
	Existent	Boolean
Student	Student Name	String
	Address	String
	Age	Byte
	School	String

### 13. Conclusion and Suggestions for Future Work

The paper presents an efficient database handling system. The system basically depends on the object oriented principle. This concept is the basic motivation behind many modern applications and softwares including the operating systems, and many others. Unfortunately, till now there is no standard software that support the object database. Only small range of facilities were supported by some database package such as Oracle.

Our adaptation model could be used with many problems that could not be solved using the traditional database software, and to solve the problem that could be solved using the traditional database software. The system is also secure and gives the user a great deal of freedom to choose the suitable tool. An efficient encryption algorithm is applied. The model has the ability to communicate with a remote site.

Our suggestion for future work is to adopt the model as a basic unit and trying to build an attractive environment system capable of handling the object database in a manner that resembles the environment of Microsoft Access. This system could be built using the run time code generation and execution that are offered with the .NET technology.

### 14. References

1. Date, C. J.; "An Introduction to Database Systems"; 7th edit; Addison Wesley.
2. Rob, P.; Coronad, C.; "Database Systems Design Implementation and Management"; 3rd edit; 1997.
3. Loney, K.; Theriault, M.; "Oracle 9i DBA Handbook"; McGraw Hill/ Osbome; 2002.
4. Beharous A. Forouzan, "Data Communication and Network", McGRAW. HILL international edition, 4 edit, 2007'
5. Jianying Zhou and Moti Yung, "Applied Cryptography and Network Security", 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010,
6. Proceedings (Lecture Notes in Computer Science / Security and Cryptology). (Aug 11, 2010)
7. Harris, A.; "Microsoft C# Programming for the Absolute Beginner"; Premier Press; 2002.