**1. INTRODUCTION**

Data Mining has wide applications in many areas such as banking, medicine, scientific research and among government agencies. Classification is one of the commonly used tasks in data mining applications. For the past decade, due to the rise of various privacy issues, many theoretical and practical solutions to the classification problem have been proposed under different security models. However, with the recent popularity of cloud computing, users now have the opportunity to outsource their data, in encrypted form, as well as the data mining tasks to the cloud. Since the data on the cloud is in encrypted form, existing privacy preserving classification techniques are not applicable. In this paper, we focus on solving the classification problem over encrypted data. In particular, we propose a secure k-NN classifier over encrypted data in the cloud. The proposed k-NN protocol protects the confidentiality of the data, user’s input query, and data access patterns. To the best of our knowledge, our work is the first to develop a secure k-NN classifier over encrypted data under the standard semi-honest model. Also, we empirically analyze the efficiency of our solution through various experiments.

**1.2 SYSTEM SPECIFICATION**

**1.2.1 HARDWARE CONFIGURATION**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 4 GB.

**1.2.2 SOFTWARE SPECIFICATION**

* Operating System : Windows 7
* Front End : ASP.NET
* Back End : MY-SQL
* Document : MS-Office

**1.2.3 SOFTWARE DESCRIPTION**

**ASP.Net**

**Server Application Development**

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and MS Access can perform standard operations while your application logic executes through the managed code.

**Server-side managed code**

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development. Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

**Active Server Pages.NET**

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

* Enhanced Performance. ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
* World-Class Tool Support. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.
* Power and Flexibility. Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.
* Simplicity. ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.
* Manageability. ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.
* Scalability and Availability. ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.
* Customizability and Extensibility.ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.
* Security. With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.
* Language Support.The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

**What is ASP.NET Web Forms?**

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

* The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.
* The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
* The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

An ASP.NET page can be created simply by taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required). For example, the following sample demonstrates a simple HTML page that collects a user's name and category preference and then performs a form post back to the originating page when a button is clicked:

ASP.NET provides syntax compatibility with existing ASP pages. This includes support for <% %> code render blocks that can be intermixed with HTML content within an .aspx file. These code blocks execute in a top-down manner at page render time.

Introduction to ASP.NET Server Controls

In addition to (or instead of) using <% %> code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a runat="server" attributes value. Intrinsic HTML tags are handled by one of the controls in the System.Web.UI.HtmlControls namespace

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an <input type="hidden"> form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the <asp: adrotator> control can be used to dynamically display rotating ads on a page.

* ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
* ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
* ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
* ASP.NET server controls provide an easy way to encapsulate common functionality.
* ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
* ASP.NET server controls can automatically project both up level and down-level HTML.
* ASP.NET templates provide an easy way to customize the look and feel of list server controls.
* ASP.NET validation controls provide an easy way to do declarative client or server data validation.

**Features of SQL-SERVER**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services. SQL-SERVER database consist of six type of objects, They are,

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types,

1. Design View

2. Datasheet View

**DESIGN VIEW**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

**DATASHEET VIEW**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:** A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

**AJAX:** ASP.NET Ajax marks Microsoft's foray into the ever-growing Ajax framework market. Simply put, this new environment for building Web applications puts Ajax at the front and center of the .NET Framework.

**2. SYSTEM STUDY**

A detailed study to determine whether, to what extent, and how automatic data processing equipment should be used it usually includes an analysis of the existing system and the design of the new system, including the development of system specifications which provide a basis for the selection of equipment.

**2.1. EXISTING STUDY**

To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy.

**2.1.1 DRAWBACKS**

* One of the major problems of privacy preserving data mining is the abundant availability of personal data.
* Many technologies exist for supporting proper data handling, but much work remains, and some barriers must be overcome in order for them to be deployed.
* The above discussed privacy preserving data mining techniques are remarkably good, but there is always extent for more enhancements.

**2.2 PROPOSED SYSTEM**

In this paper, we utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in mind. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-users.

**2.2.1. FEATURES**

* PPDM is very advantageous in development of various data mining techniques.
* It allows sharing of large amount of privacy sensitive data for analysis purposes.
* It has a ability to track and collect large amounts of data with the use of current hardware technology..

**2.3 FEASIBILITY STUDY**

All projects are feasible given unlimited users and infinite time. Unfortunately the development is more likely to be plagued by a scarcity resources and difficult delivery date. It is both necessary and prudent to evaluate the feasibility of an object, no such limitation imposed that is not feasible as all the resources are easily available and time given was sufficient. Feasibility study was made according to four key considerations

**TECHNICAL FEASIBILITY**

Technical feasibility centres on the exiting computer and to what extent it can support the proposed addition. This involves financial considerations to accommodate technical enhancement. If the budget is a serious constraint, the project is judged as not feasible.

**BEHAVIORAL FEASIBILITY**

An estimate should be made of how strong a reaction the user staff is likely to have toward the development of computerized system. It is common knowledge the computer installation have something to do understandable that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of considering business.

**OPERATIONAL FEASIBILITY**

Operational feasibility is the method used to evaluate whether the proposed system is user friendly. They must be easy to handle, maintain and modify. The main objectives is that they should be enough flexible.

**ECONOMICAL FEASIBILITY**

Economic analysis is the most frequently used method for evaluating the effectiveness of a system more commonly known as cost/benefit analysis, the procedures is made to determine the benefits and savings that are expected from system and compare them with costs. The result of comparison for found out and changed if needed. This is an ongoing effort that improves in accuracy at each phase of the system life cycle. Feasibility is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of providing quality is reduced.

**3. SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

There are numerous types of files used for storing data needed for processing, reference or back up. The main common types of processing files include

* Master,
* Transaction,
* Reference,
* Backup, Report and
* Sort.

**MASTER FILE**

A master file is the main that contains relatively permanent records about particular items or entries. For example a customer file will contain details of a customer such as customer ID, name and contact address.

**TRANSACTION FILE**

A transaction file is used to hold data during transaction processing. The file is later used to update the master file and audit daily, weekly or monthly transactions. For example in a busy supermarket, daily sales are recorded on a transaction file and later used to update the stock file. The file is also used by the management to check on the daily or periodic transactions.

**REFERENCE FILE**

A reference file is mainly used for reference or look-up purposes. Look-up information is that information that is stored in a separate file but is required during processing. For example, in a point of sale terminal, the item code entered either manually or using a barcode reader looks up the item description and price from a reference file stored on a storage device.

**BACKUP FILE**

A backup files is used to hold copies (backups) of data or information from the computers fixed storage (hard disk). Since a file held on the hard disk may be corrupted, lost or changed accidentally, it is necessary to keep copies of the recently updated files. Incase of the hard disk failure, a backup file can be used to reconstruct the original file.

**REPORT FILE**

Used to store relatively permanent records extracted from the master file or generated after processing. For example you may obtain a stock levels report generated from an inventory system while a copy of the report will be stored in the report file.

**SORT FILE**

It stores data which is arranged in a particular order. Used mainly where data is to be processed sequentially. In sequential processing, data or records are first sorted and held on a magnetic tape before updating the master file.

**3.2 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.3 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.4 DATABASE DESIGN**

The development of database systems is not a major breakthrough in computer technology; rather it is a logical development in the methods used by computers to access and manipulate data stored in various parts of computer systems. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database management systems allow the data to the protected and organized separately from other resource. Defining the term ‘database’ is an integrated collection of items. Computer base systems which support a centralized data management function can support a database. If the data management software can provide centralized access to the data from the programs, possible to treat data as a separate resource.

Usually the centralized data management software is the software is a package called a database management system. Within it will be found components which exist as separate software packages, such as a complex file handler, an information retrieval system, or a comprehensive report generator. The organization of data in a database aims to achieve three major objectives such as data integration, data integrity and data independence.

**3.5 SYSTEM DEVELOPMENT**

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software. Written standards and procedures must guide all information systems processing functions. The organization’s management must define and implement standards and adopt an appropriate system development life cycle methodology governing the process of developing, acquiring, implementing, and maintaining computerized information systems and related technology.

**3.5.1 DESCRIPTION OF MODULES**

**PRIVACY-PRESERVING PUBLIC AUDITING MODULE**:

Homomorphic authenticators are unforgeable verification metadata generated from individual data blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. Overview to achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server’s response is masked with randomness generated by a pseudo random function (PRF). The proposed scheme is as follows:

* Setup Phase
* Audit Phase

**BATCH AUDITING MODULE**:

With the establishment of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different users’ requests. The individual auditing of these tasks for TPA can be tedious and very inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side.

**DATA DYNAMICS** **MODULE:**

Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. We can adopt this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics.

**4. SYSTEM TESTING AND IMPLEMENTATION**

### 4.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**UNIT TESTING:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**TEST STRATEGY AND APPROACH**

Field testing will be performed manually and functional tests will be written in detail.

**TEST OBJECTIVES**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**FEATURES TO BE TESTED**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**TEST RESULTS:**

All the test cases mentioned above passed successfully. No defects encountered.

**ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**TEST RESULTS:**

All the test cases mentioned above passed successfully. No defects encountered.

### 4.2 IMPLEMENTATION:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

**5. CONCLUSION**

In this paper, we propose a privacy-preserving public auditing system for data storage security in privacy preserving data mining, where TPA can perform the storage auditing without demanding the local copy of data. We utilize the homomorphic authenticator and random mask technique to guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users’ fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Extensive security and performance analysis shows that the proposed schemes are provably secure and highly efficient. We believe all these advantages of the proposed schemes will shed light on economies of scale for Data Mining.

**5.1 SCOPE FUTURE ENHANCEMENT**

Preserving privacy on the web has an important impact on many web activities and web applications. In particular, privacy issues have attracted a lot of attention due to the growth of e-commerce and e-business. These issues are further complicated by the global and self-regulatory nature of the web. Privacy issues on the Web are based on the fact that most users want to maintain strict anonymity on web applications and activities. The ease access to information on the web coupled with the ready availability of personal data, also made it easier and more tempting for interested parties (e.g., businesses and governments) to willingly or inadvertently intrude on individuals' privacy in unprecedented ways.

**BIBLIOGRAPHY**

**Book Reference:**

* User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
* Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
* Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.
* Data Communications and Networking, by Behrouz A Forouzan.
* Computer Networking: A Top-Down Approach, by James F. Kurose.
* Operating System Concepts, by Abraham Silberschatz.

**Website References**

* <http://www.sourcefordgde.com>
* <http://www.networkcomputing.com/>
* <http://www.ieee.org>
* <http://www.almaden.ibm.com/software/quest/Resources/>
* <http://www.computer.org/publications/dlib>
* <http://www.ceur-ws.org/Vol-90/>
* <http://www.microsoft.com/isapi/redir.dll?prd=ie&pver=6&ar=msnhome>

**APPENDIX**

**A. DATA FLOW DIAGRAM**

A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or combination of both.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

It is usually beginning with a context diagram as the level 0 of DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of details for decomposing particular function really depending on the complexity that function.

**DFD SYMBOLS:**

In the DFD, there are four symbols

Process that transforms data flow.

Source or Destination of data

Data flow

Data Store

**DATA FLOW DIAGRAM**

****

**B. TABLE STRUCTURE**

**C. SAMPLE CODING**

**D. SAMPLE INPUT AND OUTPUT**