COMBINING PRIVATE AND PUBLIC KEY ENCRYPTION TECHNIQUES FOR PROVIDING EXTREME SECURE ENVIRONMENT FOR AN ACADEMIC INSTITUTION APPLICATION

[1]Syed S. Rizvi, [2]Aasia Riasat, [3]Khaled M. Elleithy

[1, 3]Computer Science & Engineering Department, University of Bridgeport, Bridgeport, CT

[1]srizvi,[3]elleithy@bridgeport.edu

[2]Computer Science Department, Institute of Business Management

[2]aasia.riasat@iobm.edu.pk

ABSTRACT

This paper presents the implementation of a secure application for an academic institution that offers numerous services to both students and the faculty. The primary focus of this paper is to provide a technical implementation of a new architecture for encrypting the database. The scope of this paper mainly includes but is not limited to symmetric and public-key cryptography, authentication, key management, and digital signatures. The final results of this paper demonstrate that what security features one should implement in order to achieve a highly secured application. This paper presents the implementation of a stand alone system that can be implemented on any legacy systems, and still operates effectively. In other words, it is self sufficient in terms of the data that it stores.

KEYWORDS

Data inscription standard, Rijndael Algorithm, secret Key Algorithm, & WEP

1. INTRODUCTION

Some of the major services that the intended application offers to both students and the faculty are as follows:

- The intended application is flexible in a sense that it gives ability to add/delete users, courses, students, and documents.
- Flexibility to change passwords. The secure application provides highly transparent environment to its users. There should be minimal input from the user due to security features.
- One of the key features that the proposed application offers is the “forgotten passwords”. In other words, the secure application makes sure that if a user forgets his/her password, they should not completely lose their documents.
- In addition, the proposed application ensures that an administrator should not be able to decrypt the documents.
- Finally we design and develop this secure application by assuming that the communication is not secure at all.

Some of the security measures that we consider during the design and development of the targeted secure applications are as follows: Log all accesses activities to the server and provide features in the secure application to search for unusual access patterns. If possible, put an upper limit on the number of document that a single user can access or we should have a warning mechanism in the application to ensure fairness. Our secure application should have a permission system to the document that determines if a user is permitted to access it. If the documents are read-only, add a software application called "Secure Viewer" that never stores the document to disk. A user should also have the capability to add a specialized crypto board on the server. This crypto card would be used to encrypt/decrypt files on the server.

One of the major objectives of the targeted secure application is to provide secure storage of the faculty documents as well as maintaining authorized access to the documents for the authorized users. In order to maintain this level of security, there is a need to design a strong and secured application that let the documents of the faculty being kept secret by implementing data Integrity and confidentiality as well as making the documents partially shared or available [5, 6]. Our design approach, therefore, implements a complete line of defensive authentication and authorization cryptographic standards to protect the data and to maintain its integrity while at the same time making it available for the authorized users. In particular, in order to design and implement such a secured application, the following are the minimum key security-elements that should be addressed by us in this paper: User authentication and Authorization, ACL Management & Access Availability, Data encryption and decryption, Data integrity, and Document Accountability. Fig. 1 shows the implementation of the above five security components for both faculty as well as the student-users.

2. Related Work

The best known types of symmetric encryption are the Data Encryption Standard (DES), Triple DES (3DES), the Advanced Encryption Standard (AES), and Rivest Cipher (RC4). The former three are block ciphers while RC4 is a true stream cipher [11]. The AES was developed as a replacement for DES and 3DES. It supports key lengths of 128, 192, and 256 bits and a variable block length. AES is based on the Rijndael encryption algorithm. Rijndael is a block cipher adopted as an encryption standard by the U.S. government, developed by Joan Daemen and Vincent Rijmen. It has been analyzed extensively and is now used widely worldwide as was the case with its predecessor, DES [17]. During the evaluation of candidates for the AES standard, Rijndael was analyzed by some of the world's best cryptanalysts. It has proven to be very effective against known attacks, very efficient, and simple to implement. [17]. Rijndael supports a larger range of block and key sizes; AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits, whereas Rijndael can be specified with key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits [12].

Our proposed research project uses the Rijndael cipher algorithm to perform data encryption and decryption. The key sharing will be secured by the implementation of the public key algorithm, RSA. The use of Rijndael cipher algorithm allows us to store the data in a compressed encrypted form which consequently results in a small-size database. Our implementation, therefore, addresses some of the common issues raised in [10]. Moreover, we combine the secure hash algorithm 1 (SHA1) [16] with the RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) public key algorithm to generate the digital signature for user authentication.

Previously, there were several attempts to combine the RSA algorithm with the other security mechanism to provide a fast and secure implementation. For instance, number of researchers combined RSA algorithm with the Chinese remainder theorem (CRT) [13] [14]. However, none of them described the implementation detail of these algorithms. The goal of our research work is to provide an extreme secure environment by appropriately combining the private key algorithms with the public key algorithms.

The attempt of combining these algorithms allows us to minimize the execution time (e.g., using private key algorithm such as DES rather than public key algorithm such as RSA) and maximize

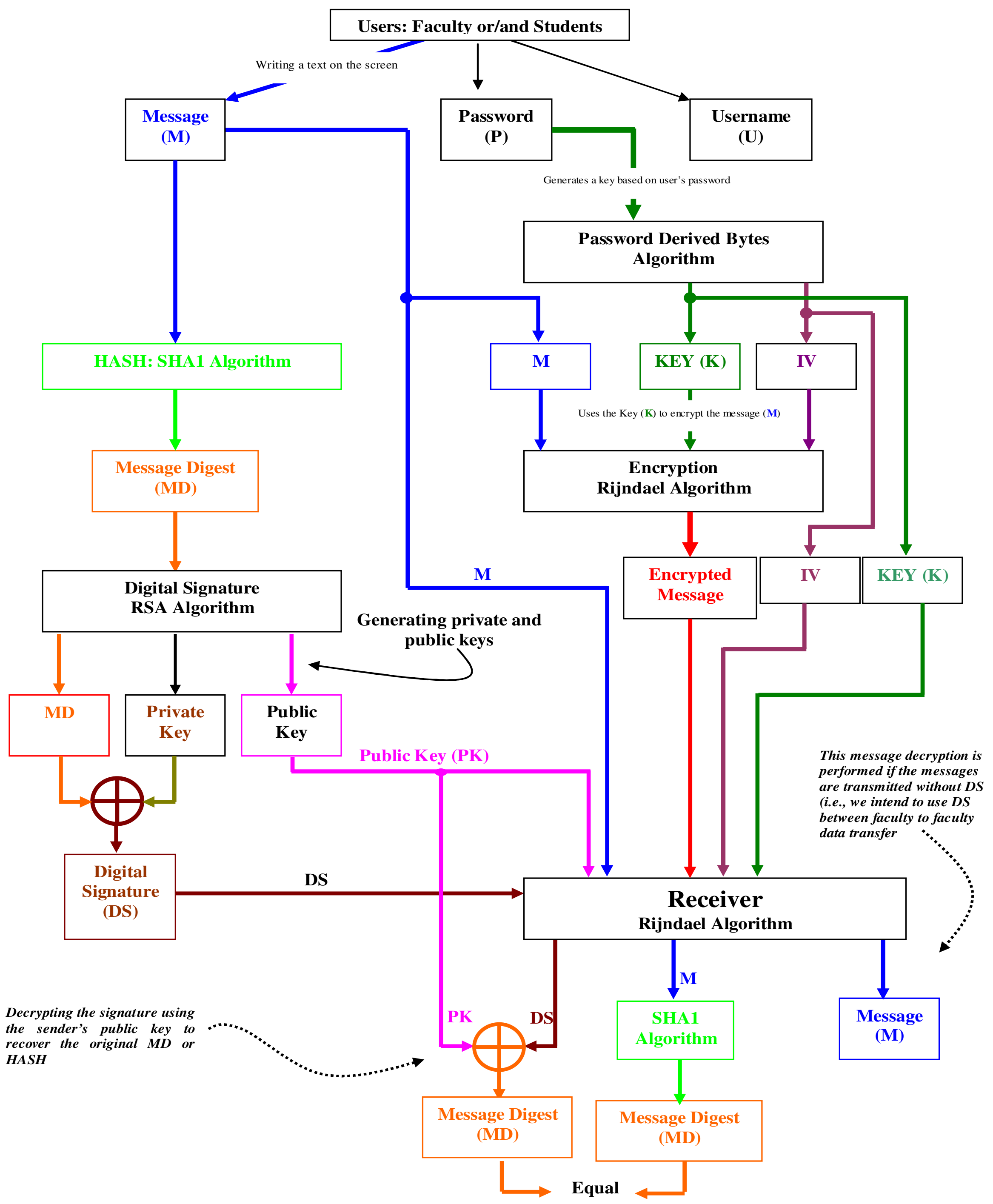

Users: Faculty or/and Students
Writing a text on the screen
Message (M)
Password (P)
Username (U)
Generates a key based on user's password
Password Derived Bytes Algorithm
HASH: SHA1 Algorithm
M
KEY (K)
IV
Uses the Key (K) to encrypt the message (M)
Encryption Rijndael Algorithm
Message Digest (MD)
M
Encrypted Message
IV
KEY (K)
Digital Signature RSA Algorithm
Generating private and public keys
MD
Private Key
Public Key
Public Key (PK)
This message decryption is performed if the messages are transmitted without DS (i.e., we intend to use DS between faculty to faculty data transfer
Digital Signature (DS)
DS
Receiver Rijndael Algorithm
M
Decrypting the signature using the sender's public key to recover the original MD or HASH
PK
DS
SHA1 Algorithm
Message (M)
Message Digest (MD)
Message Digest (MD)
Equal

Fig. 1. Proposed Architecture for combining various security features for the intended application

the security (e.g., using public key algorithm to avoid the use a secret key). For instance, RSA is about 1000 times slower than DES [10]. This is partly a result of the fact that secure key lengths for public key algorithms are about 100 times longer than comparable-strength symmetric keys [15]. It is also a result of the fact that the mathematical operations required to implement the popular flavours of public-key encryption are much more complicated than those required for popular symmetric-key algorithms [11].

3. Components Of The Proposed Architecture

3.1. User Authentication and Authorization

The secure application is certainly required to employ a strong mechanism to authenticate the users. The most frequently used strategy is asking for a user name and password to authenticate the user. Some key points that we should consider in the design of authentication mechanism are: transmitting the password in clear (i.e., we may use SSL to protect the user privacy and to safe the application by being played in the hand of some intruder after he capture the network traffic and thus get the password). Also, it is required that the secure application provides secure storage of the user names and passwords along with a method to manage them, including resetting or revoking the passwords or user accounts. Our another important concern during the preliminary design of secure application is whether to store the password in some hash format or storing it in the plain text format as the user entered.

3.2. ACL Management & Access Availability

One of the requirements of the secured application is making information always accessible to users who need it and who have sufficient permissions to access it. In order to achieve this task, the design of secure application should provide a robust mechanism to perform good management of document creation/accessing the documents in a secure manner, and access rights settings. For instance, Fig. 2 represents the implementation of secure HTTP.

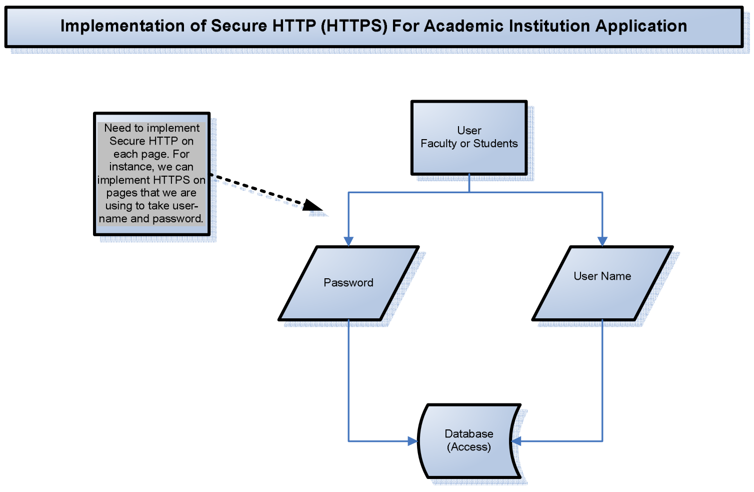

Implementation of Secure HTTP (HTTPS) For Academic Institution Application
Need to implement Secure HTTP on each page. For instance, we can implement HTTPS on pages that we are using to take user-name and password.
User Faculty or Students
Password
User Name
Database (Access)

Figure 2. Implementation of secure HTTP (HTTPS) for academic institution application

3.3. Date Encryption and Decryption

The design of a secure application is not possible without the use of some encryption and decryption techniques. The advantages of symmetric key cryptography make our design choice rather straightforward. However, since both parties need the same key for effective communication to occur, key distribution becomes an issue [1, 4]. For our secure application, the encryption will takes place at the server where as the keys can be generated by the owner of the files entering some text. In addition, if file gets corrupted, the owner should be able to produce the same set of the keys if needed. The keys can be stored in encrypted format on the secured server, while just the server side application can access the file that contains the set of all keys that are used to encrypt the documents. On the other hand, in order to decrypt the document and make accessible to all users, we prefer an approach where decryption takes place at the server before the actual transmission of the file to the user.

3.4. Data Integrity

Data integrity is one of the issues that we consider during the development phase of our secured application. The task is to make files secure by completely denying unauthorized access to the files while at the same time make sure that the files should be modified only by those (student or faculty) who are authorized to do so (If any) or can not be modified other than the owner of the document. We implement the concept of digital signatures that enable recipients to verify the integrity of an electronic document that is used.

3.5. Document Accountability

The secure application needs to keep track and to maintain the log of the activities that the user performed on the document. This is essential since some of the users may have privileges to modify some of the shared documents for which they have access rights. Document auditing [2, 3] is one of the important security measures that we desire to have in our designed application since it enables secure application to maintain accountability with regard to the use of protected files.

4. IMPLEMENTATION ISSUES AND DESIGN CHOICES

In this section, we present our overall design structure for the targeted application. In addition, this section provides a comprehensive discussion on implantation issues and our design choices for implementing each security component we discussed above. Furthermore, in this section we present the proposed architecture for the targeted application.

4.1. Project Design Phase

The design phase includes overall flow of the project, design of the database to suit the contents of the application, security features including authorization and authentication process, Access control list management, Session objects implementation, owner keys management, and secure http (https) implementation for transferring secure data between different entities of this application. The implementation of secure http is shown in Fig. 2.

4.2. Proposed Database Design

The database was designed in a way that it would suit the application flow and all the entities of the application (see Fig. 3). The database consists of the following main entities to record the application flow: *Faculty*, *Course*, *Student*, *Document*, and *System Administrator*. For the sake of simplicity, all the entities information are kept simple in database, although this information can be made comprehensive and complete in any real time implementation and as per the development requirements. Furthermore, the discussion of this section yields the class diagram as shown in Fig. 3. In addition, Fig. 4 shows a simple data flow diagram of the proposed architecture. The high level architecture of the proposed approach is presented in Fig. 5. The details of both flow diagram and the high level architecture will be discussed in the subsequent sections.

4.2.1. Faculty

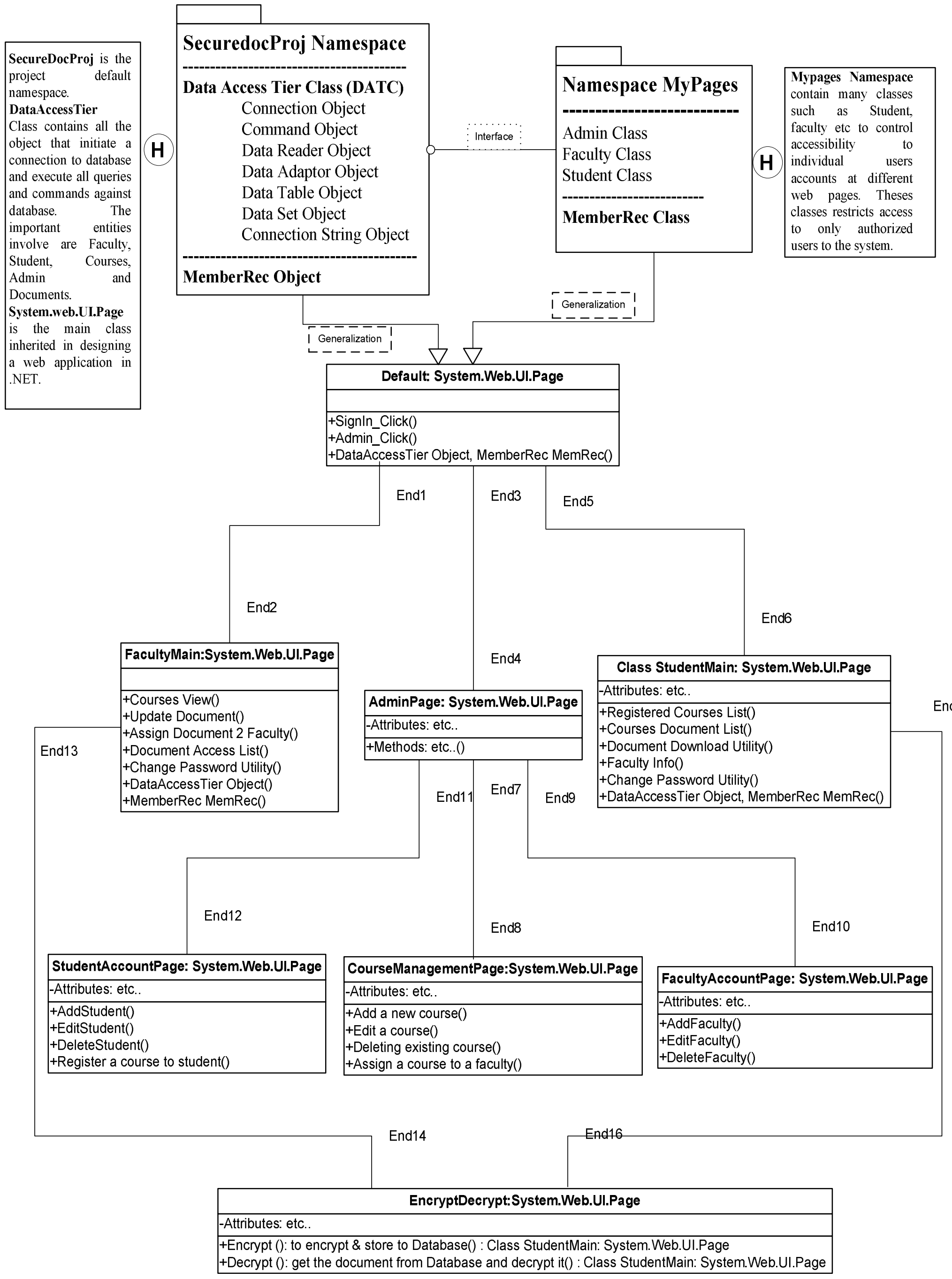

SecureDocProj is the project default namespace.
DataAccessTier Class contains all the object that initiate a connection to database and execute all queries and commands against database. The important entities involve are Faculty, Student, Courses, Admin and Documents.
System.web.UI.Page is the main class inherited in designing a web application in .NET.
H
SecuredocProj Namespace
Data Access Tier Class (DATC)
Connection Object
Command Object
Data Reader Object
Data Adaptor Object
Data Table Object
Data Set Object
Connection String Object
MemberRec Object
Interface
Namespace MyPages
Admin Class
Faculty Class
Student Class
MemberRec Class
H
Mypages Namespace contain many classes such as Student, faculty etc to control accessibility to individual users accounts at different web pages. Theses classes restricts access to only authorized users to the system.
Generalization
Generalization
Default: System.Web.UI.Page
+SignIn_Click()
+Admin_Click()
+DataAccessTier Object, MemberRec MemRec()
End1
End2
End3
End4
End5
End6
FacultyMain:System.Web.UI.Page
+Courses View()
+Update Document()
+Assign Document 2 Faculty()
+Document Access List()
+Change Password Utility()
+DataAccessTier Object()
+MemberRec MemRec()
AdminPage: System.Web.UI.Page
-Attributes: etc..
+Methods: etc..()
Class StudentMain: System.Web.UI.Page
-Attributes: etc..
+Registered Courses List()
+Courses Document List()
+Document Download Utility()
+Faculty Info()
+Change Password Utility()
+DataAccessTier Object, MemberRec MemRec()
End13
End15
End11
End7
End9
End12
End8
End10
StudentAccountPage: System.Web.UI.Page
-Attributes: etc..
+AddStudent()
+EditStudent()
+DeleteStudent()
+Register a course to student()
CourseManagementPage:System.Web.UI.Page
-Attributes: etc..
+Add a new course()
+Edit a course()
+Deleting existing course()
+Assign a course to a faculty()
FacultyAccountPage: System.Web.UI.Page
-Attributes: etc..
+AddFaculty()
+EditFaculty()
+DeleteFaculty()
End14
End16
EncryptDecrypt:System.Web.UI.Page
-Attributes: etc..
+Encrypt (): to encrypt & store to Database() : Class StudentMain: System.Web.UI.Page
+Decrypt (): get the document from Database and decrypt it() : Class StudentMain: System.Web.UI.Page

Figure 1. Class Diagram of a Secure Application for an Academic Institution

Figure 3. Class diagram for the implemented project

A Faculty entity contains the complete record about a faculty including the membership information. Membership information stores the faculty's login name, password (stored in the hashed format), secret question and answer, birth date and place. This information is included in the membership account to create the encryption/decryption keys and to reset the password in case of password forget or password reset request.

4.2.2. Course

Course entity as the name implies contains all the information related to an offered course. This information includes Course-No, CRN, Name, Course Level, Credit hours, semester and year in which the course if offered. Along with this course entity a unique ID is stored too that identifies this course uniquely among all the other courses records.

4.2.3. Student

A student entity contains the complete record of a student including the membership information. Membership information stores the student‘s login name, password (stored in the hashed format), secret question and answer, birth date and place. This information is included in the membership account to reset the password in case of password forget or password reset request. This information serves to maintain the student personal information and to process authentication and authorization request. Along with this student entity a unique ID is stored too that identifies student uniquely among all the other student records.

4.2.4. Document

This entity contains all the information that is related to a document that belongs to a faculty and optionally can belong to a course. The information in this entity includes a unique ID allotted to each record in this table to uniquely identifying the document, the faculty ID who owns this document, document name, location and description. The original document is stored in encrypted format in the database. For encrypting the documents the keys are generated at the runtime by using some personal information of the faculty to whom the document belongs. Since this is entirely dynamic, it ensures the document security from the time when the document resides in the database.

4.2.5. System Administrator

System administrator is the second important entity of this application since all the above described entities including faculty, courses, students, registration, accounts setup and course assignments management is being carried out by the system administrator. The following are some of the typical responsibilities of the system admin: System admin is responsible to perform additions, modifications and deletions of all the faculty, courses and students records. A system admin can add new course information to the database and to the list of the courses that are offered at any given semester. He or she can set up the membership records and the login accounts for both students and the faculty. Only a system administrator can perform the registration process.

Some of the design and implementation issues that can arise here include the level of authority and accessibility for the system admin. The project draft indicated that introducing a new course, assigning it to a faculty member, and registering a student for a course can be or should be done by any of the faculty member. However, a design issue that is involved here is that these tasks require a central authority that neutrally could do all these tasks and that entity should be some thing else other than the faculty entity. In order to resolve this issue, we developed the application as our faculty can not assign a course to him/her self, while an administrator could do that.

4.3. Proposed Security Design

The proposed security design includes various security measures that are incorporated in the intended application.

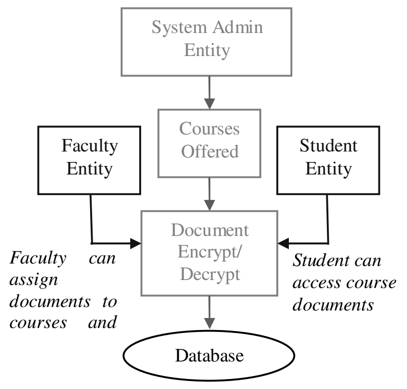

System Admin Entity
Courses Offered
Faculty Entity
Student Entity
Document Encrypt/ Decrypt
Faculty can assign documents to courses and
Student can access course documents
Database

Figure 4. A simple data flow diagram

4.3.1. Custom Base Class

In our research project we have used ASP.NET Custom Base Class feature to secure access to all the project web pages, data and services available on them [7]. For this purpose, we created custom base class called "My Pages" which is derived for System.Web.UI.Pages and consists of those classes that contain the code that put the security checks and take care of the process of authorization. All the web form's codes behind classes are derived from the Custom Base Class that provides the basic infrastructure for the web page's information access security. To implement this hierarchy, we implemented the .Net's most prominent feature: session management to maintain the user's identity at each step of the application [8, 9]. By using the custom based class implementation, we have avoided the URL spoofing in which a person who is not authorize to view the page contents or to access the resources offered by it can be able to access the page's contents

4.3.2. Dynamic Key Generation and Management

In order to prevent the unauthorized access to the keys that are used to secure the documents upon storage, the keys for encryption and decryption are chosen entirely at run time. With this approach, we avoided to store them at any place which consequently avoided any security threats. The system will be a bit slow in the response but will save us the cost of being insecure. The keys are generated based on the session objects information of the person which is being signed at the time of the document upload and encryption request

4.4. Basic Concepts Design and Flow Diagram

The basic concept includes the users, custom, validation and calendar controls. Validating the user inputs throughout the pages include telephone number and date information. Updating the database based on the calendar when the user specify the date. The retrieved information from the database is displayed using data adapters, data sets, and data grids and data list. The main tools used as a basic concept in .Net framework are: User Controls, Image Controls, Html File Control, Data List, Data Grid, Calendar Controls, Validation Controls, Regular Expressions, Data Readers, Data Adapters, and Data Sets. It can be seen in Fig. 4 that the data flow from top to bottom where system administrator initiates and introduces students, courses, and faculty. A detailed flow diagram or a high level architecture of the proposed secure application is shown in Fig. 5.

5. SECURE DOCUMENT APPLICATION IMPLEMENTATION

In this section, we present a discussion on the technicalities we encountered during the development phase of this project. This includes implementation detail and interface choice. In

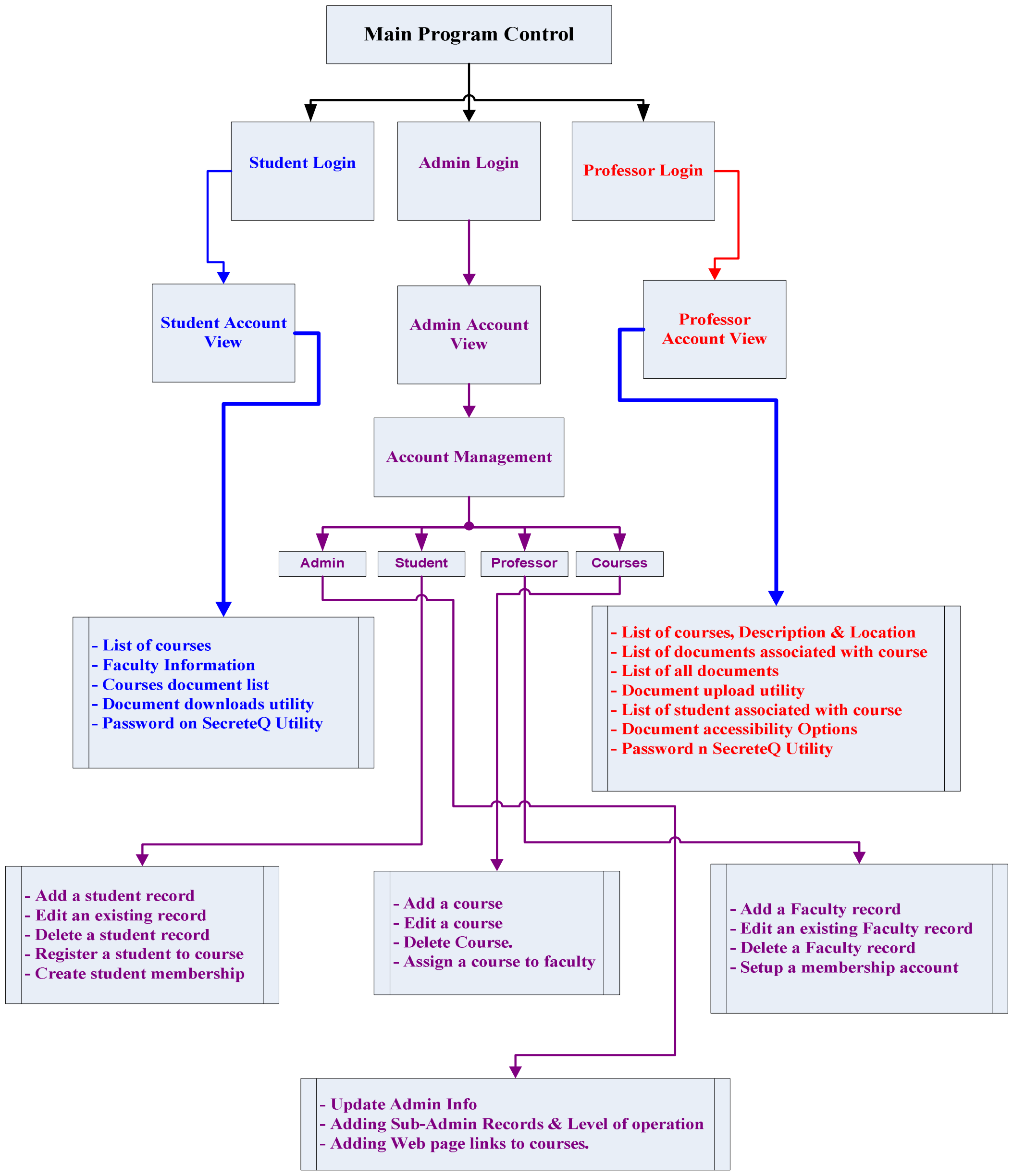
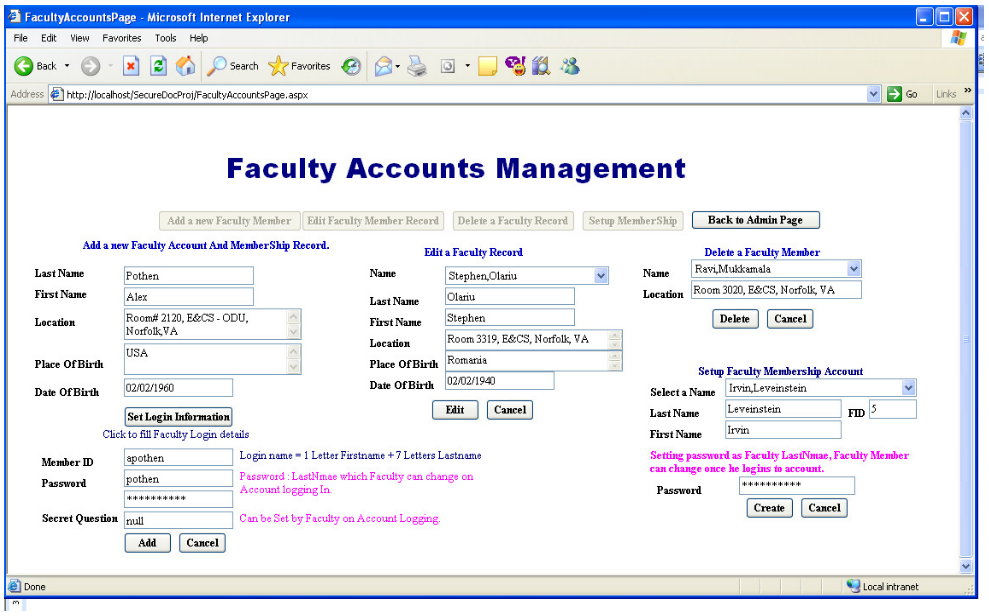

Main Program Control
Student Login
Admin Login
Professor Login
Student Account View
Admin Account View
Professor Account View
Account Management
Admin
Student
Professor
Courses
- List of courses
- Faculty Information
- Courses document list
- Document downloads utility
- Password on SecreteQ Utility
- List of courses, Description & Location
- List of documents associated with course
- List of all documents
- Document upload utility
- List of student associated with course
- Document accessibility Options
- Password n SecreteQ Utility
- Add a student record
- Edit an existing record
- Delete a student record
- Register a student to course
- Create student membership
- Add a course
- Edit a course
- Delete Course.
- Assign a course to faculty
- Add a Faculty record
- Edit an existing Faculty record
- Delete a Faculty record
- Setup a membership account
- Update Admin Info
- Adding Sub-Admin Records & Level of operation
- Adding Web page links to courses.

Figure 5. Detailed Flow Diagram of Secure Application for an Academic Institution

this application, the flow of the application starts from the main (default) page where a person sign in and then based on its role or membership, he/she will be then directed to specific web pages and resources he can access. The main entities in this implementation include System Admin interface and Faculty and Student Interfaces as outlined below.

5.1. System Admin Interface

The system admin interface contains the links to the pages where a system admin can perform course management, faculty & students accounts managements In addition, a system admin can assign courses to faculty and can register students to specific courses. The links at the system admin interface include faculty accounts, admin accounts, student accounts, courses management interfaces. System administrator manages student's accounts by adding, modifying, deleting student record. He/she can setup their login accounts and can register them to the desired courses offered by a certain semester. Similarly for performing courses management, a system administrator can introduce new courses, can modify and delete existing courses and can assign different courses to faculty members as agreed by some formal meetings. Figures 6, 7, and 8 shows different parts of the system administrations.


FacultyAccountsPage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/SecureDocProj/FacultyAccountsPage.aspx
Go Links
Done
Local intranet

Faculty Accounts Management

Add a new Faculty Member | Edit Faculty Member Record | Delete a Faculty Record | Setup MemberShip | Back to Admin Page

Add a new Faculty Account And MemberShip Record.

Last Name	Pothen
First Name	Alex
Location	Room# 2120, E&CS - ODU, Norfolk,VA
Place Of Birth	USA
Date Of Birth	02/02/1960

Set Login Information
Click to fill Faculty Login details

Member ID	apothen	Login name = 1 Letter Firstname + 7 Letters Lastname
Password	pothen	Password : LastNmae which Faculty can change on Account logging In.
	**********	
Secret Question	null	Can be Set by Faculty on Account Logging.

Add Cancel

Edit a Faculty Record

Name	Stephen,Olariu
Last Name	Olariu
First Name	Stephen
Location	Room 3319, E&CS, Norfolk, VA
Place Of Birth	Romania
Date Of Birth	02/02/1940

Edit Cancel

Delete a Faculty Member

Name	Ravi,Mukkamala
Location	Room 3020, E&CS, Norfolk, VA

Delete Cancel

Setup Faculty Membership Account

Select a Name	Irvin,Leveinstein
Last Name	Leveinstein
FID	5
First Name	Irvin

Setting password as Faculty LastNmae, Faculty Member can change once he logins to account.

Password	**********

Create Cancel

Figure 6. System Admin Control Panel: Faculty Accounts Management

5.2. Faculty Interface

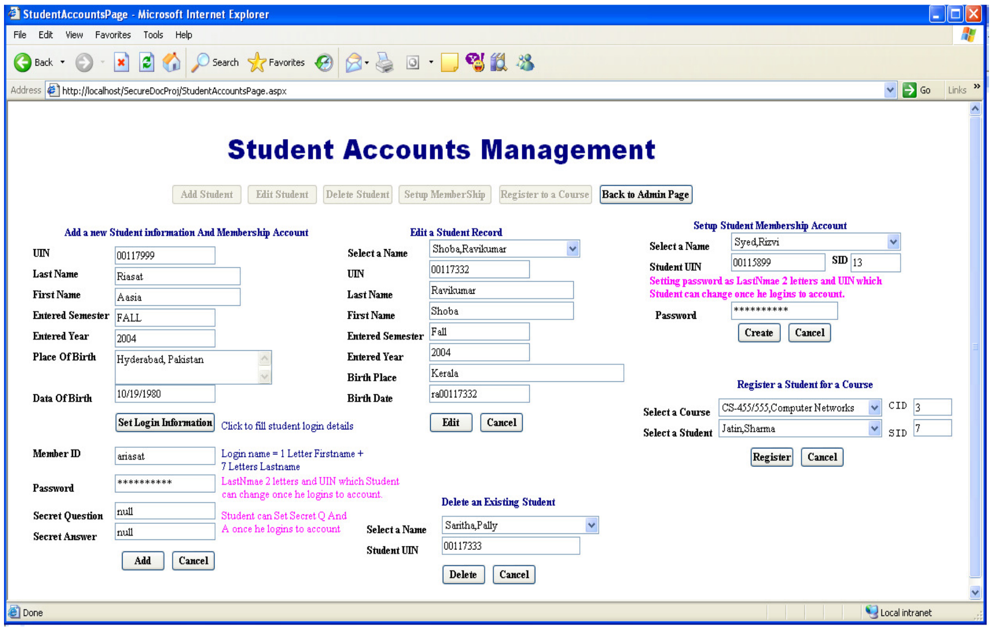
When a faculty member logs in to the application, he/she is directed to a web page that provides the information and services that are only related to that faculty member. As one can see in Fig. 9, the faculty member has provided the information regarding the courses that are assigned to him and the documents (encrypted) that he has in his folder at the server. In addition when a course is selected, the page shows the documents that are related to that specific course. The list of students who have given the access to his (faculty member) documents are also shown here. The faculty member has given the option to change the accessibility permissions of the student by deleting the student record form the list for whom he doesn't want to allow the accessibility of the document. The documents are uploaded to the server in encrypted format and then stored into the data base as a BLOB. During the uploading and encryption, the secure Http Protocol is being used, so that the transfer of the documents takes place securely.

5.3. Student Interface

When a student logs into the secure document application, he has shown the list of his registered courses and their complete description including faculty information. He can choose any of the documents that he want to access and can click the download button. The download button extract the document that are stored in the database in the BLOB form and then decrypt it on to the-sever; finally the document is made available in the browser for the student. During the document transfer we again implemented secure Http protocol to securely transfer the document.

6. Features And Security Considerations

Following are some of the features that we have successfully implemented. As this project demands, in designing this web application, a lot of emphasis we have given to the document security, authentication and authorization process. The security features we implemented in this project are outlined below.



StudentAccountsPage - Microsoft Internet Explorer

http://localhost/SecureDocProj/StudentAccountsPage.aspx

Student Accounts Management

Add Student | Edit Student | Delete Student | Setup MemberShip | Register to a Course | Back to Admin Page

Add a new Student information And Membership Account

UIN	00117999
Last Name	Riasat
First Name	Aasia
Entered Semester	FALL
Entered Year	2004
Place Of Birth	Hyderabad, Pakistan
Data Of Birth	10/19/1980
Member ID	ariasat
Password	**********
Secret Question	null
Secret Answer	null

Set Login Information — Click to fill student login details

Login name = 1 Letter Firstname + 7 Letters Lastname

LastNmae 2 letters and UIN which Student can change once he logins to account.

Student can Set Secret Q And A once he logins to account

Add | Cancel

Edit a Student Record

Select a Name	Shoba,Ravikumar
UIN	00117332
Last Name	Ravikumar
First Name	Shoba
Entered Semester	Fall
Entered Year	2004
Birth Place	Kerala
Birth Date	ra00117332

Edit | Cancel

Setup Student Membership Account

Select a Name	Syed,Rizvi
Student UIN	00115899
SID	13
Password	**********

Setting password as LastNmae 2 letters and UIN which Student can change once he logins to account.

Create | Cancel

Register a Student for a Course

Select a Course	CS-455/555,Computer Networks
CID	3
Select a Student	Jatin,Sharma
SID	7

Register | Cancel

Delete an Existing Student

Select a Name	Saritha,Pally
Student UIN	00117333

Delete | Cancel

Figure 7. System Admin Control Panel: Student Accounts Management

6.1. Custom Base Class and Code-Behind Classes

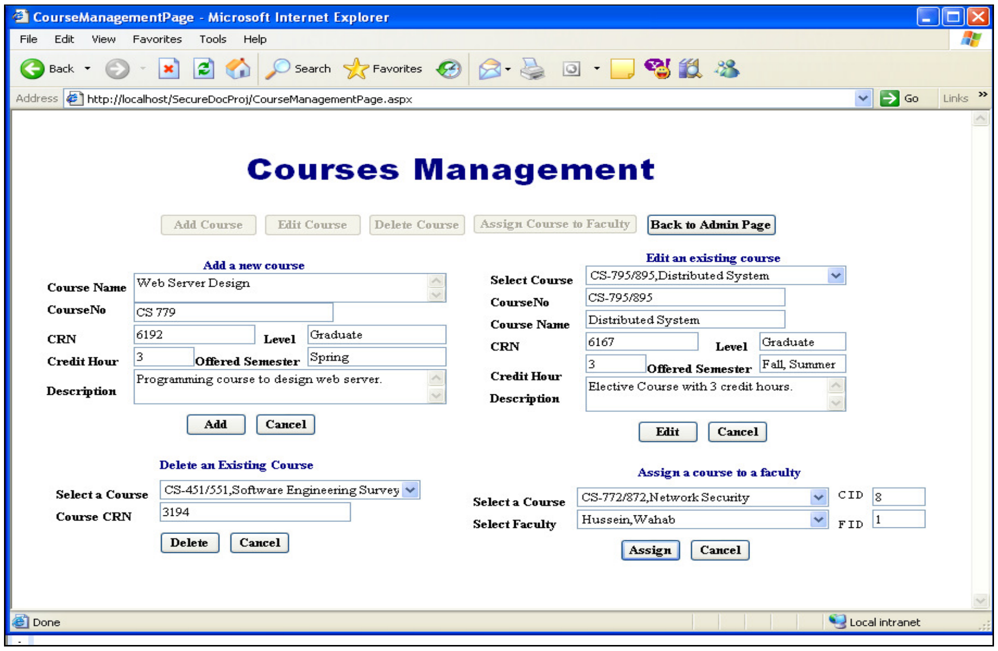
The base functionality for all ASP.NET pages is spelled out by the Page class in the system. Web. UI name space [8, 9]. This class defines the essential properties, methods, and events for an ASP.NET page. While all ASP.NET pages must be derived from the Page class, they need not be directly derived. Some time it becomes very convenient to create a base class for a particular ASP.NET Web application that extends the Page class and have all code-behind classes derive from this class, rather than directly from the Page class. This universal base class can contain its own properties, methods, or events that are common to all pages in the particular ASP.NET application, or it can extend the functionality of existing methods or properties.

6.2. Role Base Security for Securing Page & Resource Access

.NET Framework provides support for the implementation of role based security which consists of authentication and authorization. Authentication is verifying identity while authorization is determining whether user has the permission for the request he placed. Authorization takes the identity of the user and information based on which the application grant or deny permissions. In this application, we have our main entities as system administrator, faculty, and students. Depending on these roles .NET authenticates a user and provides access to the services or the documents he/she is authorized for. With this Project idea, we basically assumed three different levels or angles of data access and authorization. We designed the web pages dividing the data accessibility and its access rights for faculty, student and administrator at the time of signing in the application. The two interface categories that are made available in this case are: *System administrator Interface* and *Classified members interface.*

6.3. System Admin Interface & Admin Responsibilities

System admin interface is the back bone of the project as it directly affects the application's data processing and retrieval. It actually implements the role based security by creating roles to faculty, student & other administrator, and in this way it drives the data access security and data retrieval of the entire application.



Add a new course

Course Name: Web Server Design
CourseNo: CS 779
CRN: 6192 Level: Graduate
Credit Hour: 3 Offered Semester: Spring
Description: Programming course to design web server.

Edit an existing course

Select Course: CS-795/895,Distributed System
CourseNo: CS-795/895
Course Name: Distributed System
CRN: 6167 Level: Graduate
Credit Hour: 3 Offered Semester: Fall, Summer
Description: Elective Course with 3 credit hours.

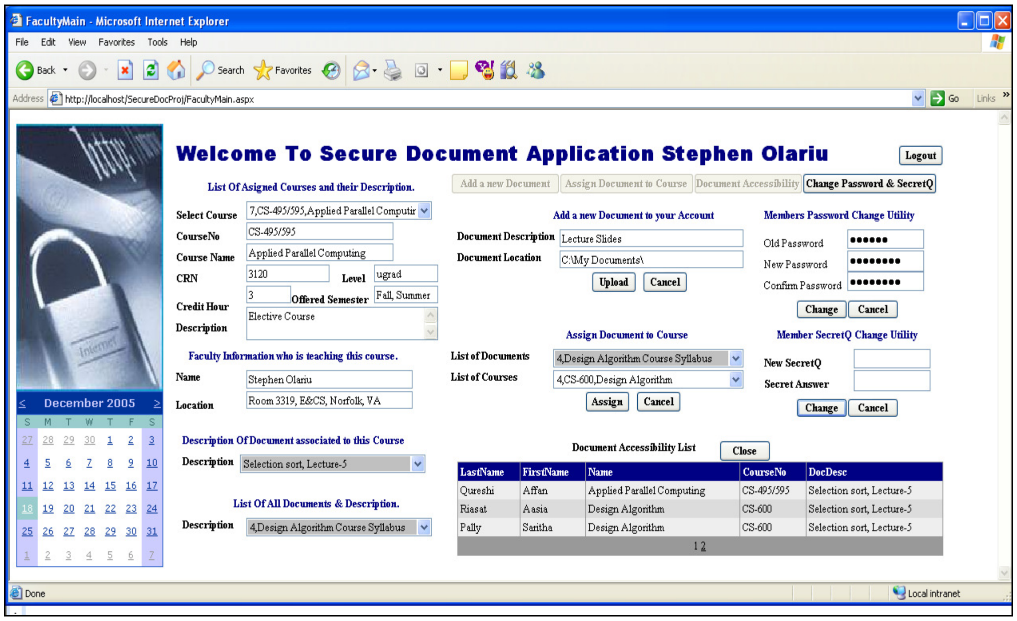
Delete an Existing Course

Select a Course: CS-451/551,Software Engineering Survey
Course CRN: 3194

Assign a course to a faculty

Select a Course: CS-772/872,Network Security CID: 8
Select Faculty: Hussein,Wahab FID: 1

Figure 8. System Admin Control Panel: Courses Management


FacultyMain - Microsoft Internet Explorer
http://localhost/SecureDocProj/FacultyMain.aspx
Welcome To Secure Document Application Stephen Olariu
Logout
Add a new Document
Assign Document to Course
Document Accessibility
Change Password & SecretQ
December 2005
Upload
Cancel
Change
Cancel
Assign
Cancel
Close
Done
Local intranet

List Of Asigned Courses and their Description.

Select Course: 7,CS-495/595,Applied Parallel Computir
CourseNo: CS-495/595
Course Name: Applied Parallel Computing
CRN: 3120 Level: ugrad
Credit Hour: 3 Offered Semester: Fall, Summer
Description: Elective Course

Faculty Information who is teaching this course.

Name: Stephen Olariu
Location: Room 3319, E&CS, Norfolk, VA

Description Of Document associated to this Course

Description: Selection sort, Lecture-5

List Of All Documents & Description.

Description: 4,Design Algorithm Course Syllabus

Add a new Document to your Account

Document Description: Lecture Slides
Document Location: C:\My Documents\

Members Password Change Utility

Old Password: ••••••
New Password: ••••••••
Confirm Password: ••••••••

Assign Document to Course

List of Documents: 4,Design Algorithm Course Syllabus
List of Courses: 4,CS-600,Design Algorithm

Member SecretQ Change Utility

New SecretQ:
Secret Answer:

Document Accessibility List

LastName	FirstName	Name	CourseNo	DocDesc
Qureshi	Affan	Applied Parallel Computing	CS-495/595	Selection sort, Lecture-5
Riasat	Aasia	Design Algorithm	CS-600	Selection sort, Lecture-5
Pally	Saritha	Design Algorithm	CS-600	Selection sort, Lecture-5

1 2

Figure 9. Faculty Member Interface

6.4. Session Management with ASP.NET Session Object

Whenever a user requests a web page from an application, the server object checks to see if the user has a session ID, The session ID is included in the requesting HTTP header. If the user has a valid session ID, the user is treated as "Active" user and is allowed to continue with the application. The session object is used to store information about the user. This information is retained for the duration of user session.

6.5. Multi-Tier Database Application Logic

The most prominent feature of our project is the custom class "DataAccessTier.cs" that separates out the interfaces or the web pages and their code behind files and the custom class's methods that get connection with database and retrieved application specific information. This we did by means of a custom class called "Data Access Tier" class that acts as a middle tier in between database and the application persons.

7. CONCLUSION

In this paper, we presented a new design for providing comprehensive security for a secure application by combining many different security techniques using the .NET framework. The most prominent feature of the .NET is its full fleshed Cryptography-API that provides techniques of encryption and decryption while hiding all the technical details. This is one of the main reasons that we achieved the goal of completing this secure application. Secure HTTP communication provided by ASP.NET's API is also another most important and handy feature worth to mention here. Some of the tools used in the application include data access controls that avoid repetitive database programming, built in authentication features and security controls that enable automated management of user accounts and roles and simplified web deployment. The proposed project consists of different tools and techniques for building secure web applications with strong database accessibility and crypto graphic techniques. During the design phase, we learned and practiced many new techniques that we found very useful and interesting in the context of building a secure and powerful web application along with strong and real time database functionality.

REFERENCES

[1] M. Hansen. Asynchronous group key distribution on top of the cc2420 security mechanisms for sensor networks. *Proceedings of the second ACM conference on Wireless network security*, pp. 13 – 20, March 2009.

[2] J. Albath and S. Madria. Practical algorithm for data security (PADS) in wireless sensor networks. *Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access*, pp. 9 – 16, Jan 2007.

[3] H. Bulbul, I. Batmaz, M. Ozel. Wireless network security: comparison of WEP (Wired Equivalent Privacy) mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) security protocols. *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, Article 9, Jan 2008.

[4] J. Brustoloni. Laboratory experiments for network security instruction. *Journal on Educational Resources in Computing (JERIC)***,** Vol. 6, Issue 4, December 2006.

[5] L. Catuogno and A. Santis. An internet role-game for the laboratory of network security course. *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pp. 240 – 244, June 2008.

[6] Z. Zhang, F. Abdesselam, X. Lin, P. Ho. A model-based semi-quantitative approach for evaluating security of enterprise networks. Proceedings of the 2008 ACM symposium on Applied computing, pp. 1069-1074, March 2008.

[7] L. Moningi, "Authentication and Authorization in ASP.NET," September 09, 2003. Available at: http://www.c-sharpcorner.com/mrsharp.asp

[8] D. Watkins, "An Overview of Security in the .NET Framework," Project 42, Sebastian Lange, Microsoft Corporation. January 2002.

[9] J. Meier, A. Mackman, B. Wastell, P. Bansode, A. Wigley, K. Gopalan, "Security Practices: ASP.NET 2.0 Security Practices at a Glance," Microsoft Corporation, August 2005.

[10] S. Diesburg, C. Meyers, D. Lary, and A. Wang. When cryptography meets storage. *Proceedings of the 4th ACM international workshop on Storage security and survivability*, pp. 11 – 20, October 2008.

[11] A. Risley, J. Roberts, P. LaDow, "Electronic security of real-time protection and SCADA communications", Schweitzer Engineering Laboratories, SEL 2003 Inc. Pullman WA USA.

[12] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, "Report on the Development of the Advanced Encryption Standard (AES)", October 2, 2000.

[13] J. Blömer, M. Otto, J. Seifert. A new CRT-RSA algorithm secure against bellcore attacks. Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 311 – 320, Washington D.C., USA, October 2003.

[14] D. Wagner. Cryptanalysis of a provably secure CRT-RSA algorithm. Proceedings of the 11th ACM conference on Computer and communications security, pp. 92 – 97, Washington D.C., USA, 2004.

[15] K. Yumbul and E. Savaş. Efficient, secure, and isolated execution of cryptographic algorithms on a cryptographic unit. Proceedings of the 2nd international conference on Security of information and networks, pp. 143 – 151, Famagusta, North Cyprus, 2009.

[16] S. Sanadhya and P. Sarkar. A new hash family obtained by modifying the SHA-2 family. Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 353 – 363, Sydney, Australia, 2009.

[17] Jalpa Bani and Syed S. Rizvi. A New Dynamic Cache Flushing (DCF) Algorithm for Preventing Cache Timing Attack. *International Journal of Computer Science and Information Security (IJCSIS)*. Vol. 4, No.1, pp. 110 - 116, 2009.

Author

SYED S. RIZVI is a Ph.D. student of Computer Engineering at the University of Bridgeport. He received a B.S. in Computer Engineering from Sir Syed University of Engineering and Technology and an M.S. in Computer Engineering from Old Dominion University in 2001 and 2005 respectively. In the past, he has done research on bioinformatics projects where he investigated the use of Linux based cluster search engines for finding the desired proteins in input and outputs sequences from multiple databases. For last one year, his research focused primarily on the modelling and simulation of wide range parallel/distributed systems and the web based training applications. Syed Rizvi is the author of 76 scholarly publications in various areas. His current research focuses on the design, implementation and comparisons of algorithms in the areas of multiuser communications, multipath signals detection, multi-access interference estimation, computational complexity and combinatorial optimization of multiuser receivers, peer-to-peer networking, and reconfigurable coprocessor and FPGA based architectures.

Aasia Riasat is an Associate Professor of Computer Science at Collage of Business Management (CBM) since May 2006. She received an M.S.C. in Computer Science from the University of Sindh, and an M.S in Computer Science from Old Dominion University in 2005. For last one year, she is working as one of the active members of the wireless and mobile communications (WMC) lab research group of University of Bridgeport, Bridgeport CT. In WMC research group, she is mainly responsible for simulation design for all the research work. Aasia Riasat is the author or co-author of more than 41 scholarly publications in various areas. Her research interests include modelling and simulation, web-based visualization, virtual reality, data compression, and algorithms optimization.

KHALED ELLEITHY received the B.Sc. degree in computer science and automatic control from Alexandria University in 1983, the MS degree in computer networks from the same university in 1986, and the MS and Ph.D. degrees in computer science from the Center for Advanced Computer Studies at the University of Louisiana at Lafayette in 1988 and 1990, respectively. From 1983 to 1986, he was with the Computer Science Department, Alexandria University, Egypt, as a lecturer. From September 1990 to May 1995 he worked as an assistant professor at the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. From May 1995 to December 2000, he has worked as an Associate Professor in the same department. In January 2000, Dr. Elleithy has joined the Department of Computer Science and Engineering in University of Bridgeport as an associate professor. In May 2003 Dr. Elleithy was promoted to full professor. In March 2006, Professor Elleithy was appointed Associate Dean for Graduate Programs in the School of Engineering at the University of Bridgeport.
Dr. Elleithy published more than seventy research papers in international journals and conferences. He has research interests are in the areas of computer networks, network security, mobile communications, and formal approaches for design and verification.