eCourt with Asymmetric Key Security using Digital Signature

Surender I[1], Bharathi R[2]

[1]M.Tech (Computer Science & Eng), Prist University, Pondicherry, India

[2]Assistant Professor (Computer Science & Eng), Prist University, Pondicherry, India

Abstract: ***Creating a web-based project entitled 'eCourt with Asymmetric Key Security using Digital Signature', deals with automating the information retrieval regarding Cause Lists (i.e., the cases listed in a court on a particular day), Judgment of Cases, and Case Status information. This application software is maintained by a centralized database for providing Asynchronous Key security to data which are store in database can be administrated only by the super user and deals with automated information retrieval, E-mail / SMS intimation, and document storing. This web-application is developed using ASP.NET (.Net Framework 4.0) as the front end and SQL Server 2008 R2 as the back-end.***

Keywords: - Asymmetric Key, Digital Signature, Database, Design, RDBMS, xml.

1. Introduction

Presently in Techcellent Solutions, the employees have to give a large chunk of their time needlessly searching for records, document and the procedure of gaining access to information regarding the Cause Lists, the Judgment of a particular case or Case Status of a case is manual and involves repeated visits to legal section of Techcellent Solutions. The process is tiresome, involves cost, is time consuming and at times may lead to frustration. As a result, the new system was proposed.

So, we are creating a web-based project entitled '**eCourt with Asymmetric Key Security using Digital Signature'** for Techcellent Solutions, deals with automating the information retrieval regarding Cause Lists **(**i.e., the cases listed in a court on a particular day), Judgment of Cases, and Case Status information in various Region Courts of Puducherry. This application software is maintained by a centralized database for providing Asymmetric Key Security using Digital Signature to information which can be administrated only by the super user and deals with automated information retrieval, E-mail / SMS intimation, and document storing. This application flows through the following concern.

- Cause Lists i.e., the cases listed in a court on a particular day
- Judgment of Cases
- Case Status information
- Storing the Judgment Document(Image)
- Intimation of Case hearing dates and judgment details via E-mail / SMS.

This web-application is developed using ASP.NET as the front end and SQL Server 2008 R2 as the back-end.

2. Digital Signature Certificate

Digital Signature certificates are the digital equivalent (i.e. electronic format) of physical or paper certificates. Digital certificate could be used as follows:

- It allows you to access membership-based web sites automatically without entering a user name and password.
- It can allow others to verify your "signed" e-mail or other electronic documents.
  Finally, a digital certificate enables you to send private messages to others.

Asymmetric Key

Digital Signature Certificate itself contains an Asymmetric Key is a pair: *a Public Key and a Private Key.*

- Public Key: is made public and is distributed widely and freely.
- Private Key: is never distributed and must be kept secret.

3. Existing System

The objective of this project is to develop a user friendly package, to replace the existing manual system with a better computerized system. The primary goal of this project is to reduce the manual work of capturing the physical achievement and monitoring of Court Case Information's. Presently, the procedure of gaining access to information regarding the Cause Lists, the Judgment of a particular Case or Status of a Case is manual and involves repeated visits to legal section of Commercial Taxes Department. The process is tiresome, involves cost, is time consuming and at times may lead to frustration. As a result, the new system was proposed. The system integrates all the details regarding Court Case Information's from 4 regions (Puducherry, Karaikal, Mahe, and Yanam).

3.1 Disadvantage of Existing System

- **Involves Paper Work**: Searching Information in papers and getting them duplicated involves considerable paper work.
- **Time Consuming**: A person who needs any sort of information has to visit court office followed by a manual search of documents. This is a time consuming task.
- **Involves Cost**: Court Visits at times cost dear to people.

- **Involvement of Human Factor**: Dependence on Human factor may lead to propagating corruption at lower rungs.
- **Extra Work for Court employees:** Court employees have to give a large chunk of their time needlessly searching for records and document, which could be easily searched in an automated system.

4. Proposed System

Proposed system aims at automate Court Case Information's (Cause Lists, Judgments, Case Status) access and automating E-mail and SMS intimations through Windows Service Application (.Net Framework Template). This information will be accessible to **litigants, advocates and Court Employees** through the website.

4.1 Advantage of Proposed System

- **Time Saving**: The process will save people of paying repeated visits to court offices to gain information.
- **Effort Saving for Court Employees:** The new system will save time of court employees who would otherwise devote a large amount of time furnishing information to people.
- **Efficient**: The search for documents and information will be faster.
- **Cost effective:** The total cost of the process will go down for litigants.
- **Will prevent any possible corruption:** Since, no human intervention will be required to gain information, any possible corrupt practices at lower level will be prevented.

4.2 Physical Architecture

Physical Architecture represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that comprise the system, and the interrelationships that occur among all architectural components of a system.

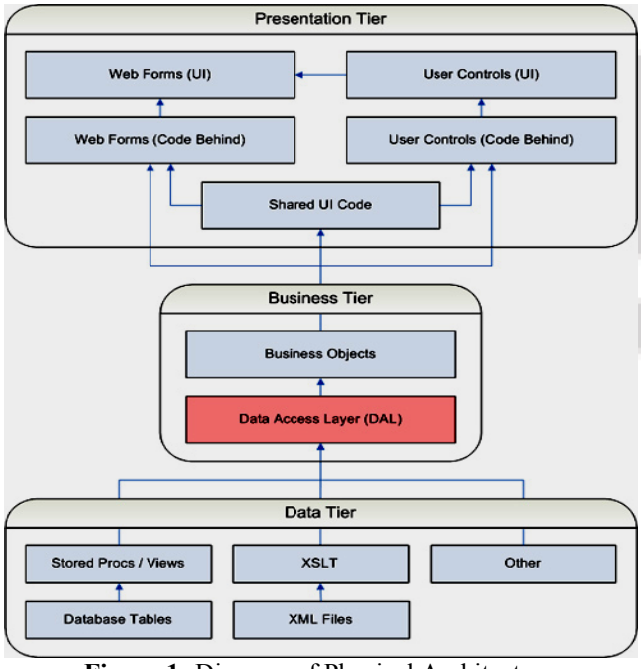

Figure 1: Diagram of Physical Architecture

Figure shows the architecture diagram that is common for any web application. Any web application is divided into three main parts, which are as follows:

- **Presentation Tier:**

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing and shopping cart contents. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. (In simple terms it is a layer which users can access directly such as a web page, or an operating systems GUI).

- **Business Tier:**

The logical tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing. Business logic could be anywhere in a program. For example, given a certain format for an address, a database table could be created which has columns that correspond exactly to the fields specified in the business logic, and type checks added to make sure that no invalid data is added.

- **Database Tire:**

This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

4.3 Block Diagram

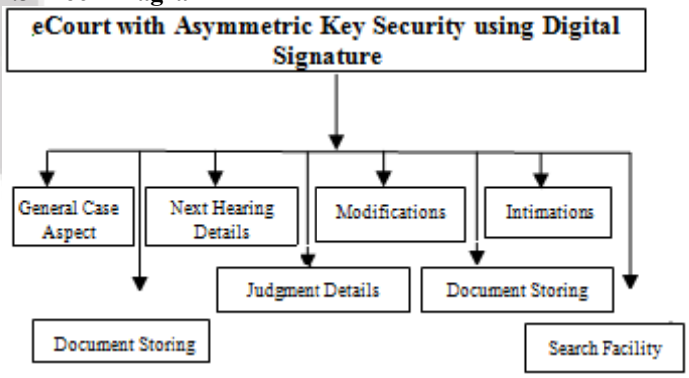

Figure 2: Block Diagram for eCourt with Asymmetric Key Security using Digital Signature

5. Modules Description

5.1 General Case Aspects

This module contains the following sub modules;

I. Case General Details
II. Petitioner Details
III. Respondent Details

I. Case General Details

It concerned with the entries Case General Details. The details will be saved into the database if it contains all the mandatory fields are entered. Also this page is validating the client input. And it controls, some validation process are as follows:

- Verifies the duplication of case numbers.
- Sanction date must not greater than the filing date of case

- Next hearing dates are must greater than the sanction and Filing date.

II. Petitioner Details
It concerned with the entries of case petitioner details. After completing all the mandatory entries of data, the page gets validated.

III. Respondent Details
It concerned with the entries of case petitioner details. After completing all the mandatory entries of data, the page gets validated.

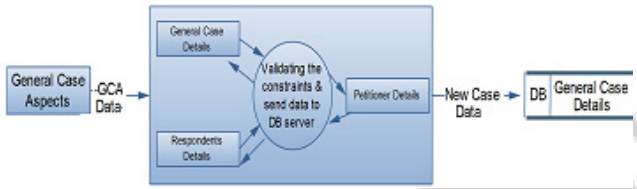

Figure 5.1: Diagram for General Case Details

5.2 Next Hearing Details

It fretful with the entries of next hearing details. After completing all the successive entries of data, the page gets validated and is allowed to store into the database. And it controls, some validation process are as follows:

- Hearing dates of a particular case is validated using their case number and thus the user is allowed to enter next hearing date.
- Next hearing date should be greater than current date.

Figure 5.2: Diagram for Next Hearing Details

5.3 Intimation (SMS / E-Mail)

In this process, the intimations are made through two ways:

I. The E-Mail Intimation
The hearing dates and judgment declaration details are sent automatically through email and which can be carried out by windows services. This email alert is made one day earlier to the hearing date. Similarly the judgment details are intimated after one hour of the judgment declaration.

II. The SMS Intimation
The hearing dates and judgment declaration details are sent automatically through SMS and which can be carried out by windows services. This SMS alert is made one day earlier to the hearing date. Similarly the judgment details are intimated after one hour of the judgment declaration.

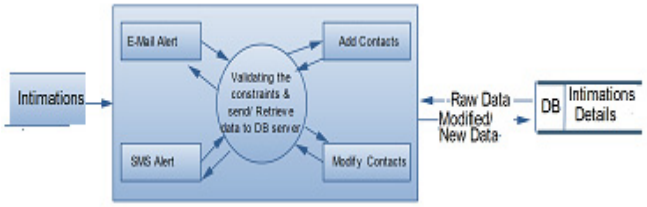

Figure 5.3: Diagram for Intimation

5.4 Judgment Details

In this process the user is allowed to enter the case Judgment details. Once the Case Number is entered, details of Petitioners and Respondents are generated and displayed. After completing all the successive entries of data, the page gets validated and is allowed to store into the database. And it controls the validation process are as follows:

- It validates, whether the particular case have any other hearing date.

Figure 5.4: Diagram for Judgment Details

5.5 Modification Process

This module contains the following sub modules;

I. Case General Details
II. Petitioner Details
III. Respondent Details

I. Case General Details
General Case Details Modification provides the facility to modify any of the information entered in respect of Case Filed. The Case Number is entered (by selecting Region Name, Division Name, and Case Type) and the existing information is displayed on the screen, which can be modified by the user. If the case that are not declared judgment or dismissed can only be modified. Also this page is validating the client input. And it controls, some validation process are as follows:

- Verifies the duplication of case numbers.
- Sanction date must not be greater than the filing date of a particular case
- Next hearing dates must be greater than the sanction and Filing date.

II. Petitioner Details
Modification of Petitioner details provides the facility to modify any of the information entered in the Petitioner Details. The Case Number is selected and the existing information is displayed on the screen, which can be modified by the user.

III. Respondent Details
Modification of Respondent Details provides the facility to modify any of the information entered in the Respondent Details. The Case Number is selected and the existing information is displayed on the screen, which can be modified by the user.

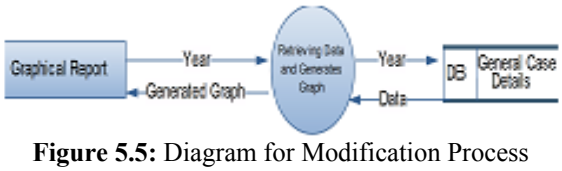

Figure 5.5: Diagram for Modification Process

5.6 Document Storing

In this process, the scanned judgment document images are stored into the database and we can carry out this process only after the judgment declaration of a particular case.

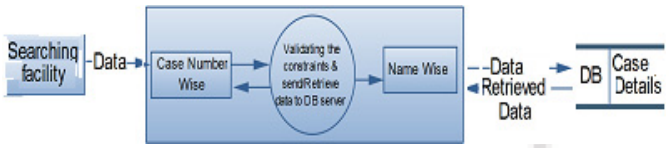

Figure 5.6: Diagram for Document Storing

5.7 Graphical Report

In Graphical Report, a Graph is generated according to the user's input and shows the graphical analysis results of No. of case filed, No of case declared and No. of case pending of a particular year.

Figure 5.7: Diagram for Graphical Report

5.8 Search Facility

According to the user's request, the searching facility generates the report and displayed on the screen. The search facility is basically in ways:

1. Name wise search,
2. Case Number wise search

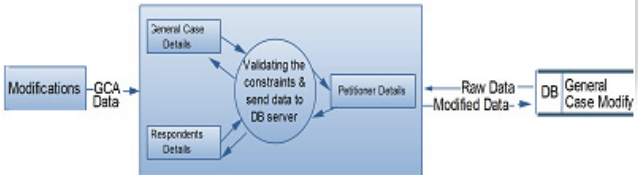

Figure 5.8: Diagram for Search Facility

6. RSA Algorithm

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman introduced a cryptographic algorithm, which was essentially to replace the less secure National Bureau of Standards (NBS) algorithm. Most importantly, RSA implements a public-key cryptosystem, as well as digital signatures. RSA is motivated by the published works of Difie and Hellman from several years before, who described the idea of such an algorithm, but never truly developed it. Introduced at the time when the era of electronic email was expected to soon arise, RSA implemented two important ideas:

I. Public-key encryption. This idea omits the need for a \courier" to deliver keys to recipients over another secure channel before transmitting the originally-intended message. In RSA, encryption keys are public, while the decryption keys are not, so only the person with the correct decryption key can decipher an encrypted message. Everyone has their own encryption and decryption keys. The keys must be made in such a way that the decryption key may not be easily deduced from the public encryption key.

II. Digital signatures. The receiver may need to verify that a transmitted message actually originated from the sender (signature), and didn't just come from there (authentication). This is done using the sender's decryption key, and the signature can later be verified by anyone, using the corresponding public encryption key. Signatures therefore cannot be forged. Also, no signer can later deny having signed the message.

This is not only useful for electronic mail, but for other electronic transactions and transmissions, such as fund transfers. The security of the RSA algorithm has so far been validated, since no known attempts to break it have yet been successful, mostly due to the difficulty of factoring large numbers n = pq, where p and q are large prime numbers.

Public-Key Cryptosystems

Each user has their own encryption and decryption procedures, E and D, with the former in the public file and the latter kept secret. These procedures are related to the keys, which, in RSA specifically, are sets of two special numbers. We of course start out with the message itself, symbolized by M, which is to be "encrypted". There are four procedures that are specific and essential to a public-key cryptosystem:

a) Deciphering an enciphered message gives you the original message, specifically

$$D(E(M)) = M : (1)$$

b) Reversing the procedures still returns M:

$$E(D(M)) = M : (2)$$

c) E and D are easy to compute.
d) The publicity of E does not compromise the secrecy of D, meaning you cannot easily figure out D from E.

With a given E, we are still not given an efficient way of computing D. If C = E(M) is the ciphertext, then trying to figure out D by trying to satisfy an M in E(M) = C is unreasonably difficult: the number of messages to test would be impractically large.

An E that satisfies (a), (c), and (d) is called a \trap-door one-way function" and is also a "trap-door one-way permutation". It is a trap door because since it's inverse D is easy to compute if certain "trap-door" information is available, but otherwise hard. It is one-way because it is easy to compute in one direction, but hard in the other. It is a permutation because it satisfies (b), meaning every ciphertext is a potential message, and every message is a ciphertext of some other message. Statement (b) is in fact just needed to provide "signatures".

Now we turn to specific keys, and imagine users A and B (Alice and Bob) on a two-user public-key cryptosystem, with their keys: EA, EB, DA, DB.

Signatures

For complete assurance that the message originated form a sender, and was not just sent through him by a third party who may have used the same encryption key (that of the receiver), we need a digital signature to come with the message. This has obvious implications of importance in real-life applications.

Bob wants to send a private message to Alice. To sign the document, we pull a clever little trick, all assuming that the RSA algorithm is quick and reliable, mostly due to property (c). We decrypt a message with Bob's key, allowed by properties (a) and (b), which assert that every message is the ciphertext of another message, and that every ciphertext can be interpreted as a message. Formally,

DB(M) = S : (3)

Then we encrypt S with Alice's encryption key.

EA(S) = EA(DB(M)) (4)

This way, we can assure only she can decrypt the document. When she does, she gets the signature by DA(EA(DB(M)) = S. She now knows the message came from Bob, since only his decryption key could compute the signature. The message need not be sent separately, since Alice can deduce it from the signature itself by using Bob's publicly available encryption key, formally EB(S) = E B(DB(M)) = M. Since S depends on M, and the encrypted transmission Bob sent depends on S, we have a transmission that depends on both the message and the signature, so both can be deduced from the transmitted document.

This brilliantly assures the message could not be modified (if needed to be presented to, say, a \judge"), since a modified M in the form of M0 would have to generate a signature S0 = DB(M0) as well, which is impossible, since she does not known DB by property (d).

So not only does Alice possess proof that Bob signed the message and indeed sent it, but she also cannot modify M nor forge a signature for any other message.

Now, say an “intruder" attempted to lie and tell you he was from the public file? This is not a problem in RSA, since “signatures' are used. Signature just needs to assure it came from the public file (PF) itself. Every time a user joins a network, everybody gets a securely sent copy of the most recently updated PF, which is stored on their system, and they never have to look it up. Anyone trying to send a message pretending to be in the public file would not have the appropriate signature, and would be singled out as an “intruder". He would also never receive the PF, since he never joined it.

Rsacryp to service provider
It Performs asymmetric SignData() and VerifyData() using the implementation of the RSA algorithm provided by the cryptographic service provider (CSP).

SIGNDATA():
Computes the hash value of the specified byte array and signs the resulting hash value.

Example:

```
byte[] data = Encoding.UTF8.GetBytes("Message to sign");
byte[] publicKey;
byte[] signature;
object hasher = SHA1.Create(); // Our chosen hashing algorithm.
// Generate a new key pair, then sign the data with it:
using (var publicPrivate = new RSACryptoServiceProvider())
{
signature = publicPrivate.SignData(data, hasher);
publicKey = publicPrivate.ExportCspBlob(false); // get public key
}
```

VERIFYDATA():
Verifies that a digital signature is valid by determining the hash value in the signature using the provided public key.

Example:

```
using (var publicOnly = new RSACryptoServiceProvider())
{
publicOnly.ImportCspBlob (publicKey);
Console.Write (publicOnly.VerifyData (data, hasher, signature)); // True
}
```

7. Future Enhancement

Every system is vulnerable to changes in requirements or some new requirements may crop up in the enterprise after sometime. Though, the system has been designed in a manner so as to keep the future needs of the company in mind, changes in requirements can still be accommodated into the system by either attaching new modules to it or by altering the existing ones depending on the requirements.

8. Conclusion

Working on ‘**eCourt with Asymmetric Key Security using Digital Signature**’ has been an enriching experience for me in multiple ways. Not only was it wonderful to work on a project of such magnitude in my training period, it was absolute pleasure to work among people who knew so much. The project provided for me practical knowledge of not just ASP.NET but also JavaScript, SQL Server 2008 R2 and exposed to so many new software’s. This shall always help me in my future projects. Following are benefits and limitations of the system developed.

Benefits of the System:
- The software provides an easy to use interface for user to deal with and thus, can be put in the category of **user friendly** software.
- Software provides proper validation and assistance to the user in situations when it is needed.
- Proper considerations have been made for accuracy.
- Litigants, Public and Advocates can get case related information all at one place.
- It will be a step towards paperless office.

Limitations of the System:
- The project caters to information gathering needs of the user.
- Only data that has been added in the database can be accessed.
- There is no on-screen help option for the user.

References

[1] C# 4.0 in a Nutshell (Cryptography Overview), Fourth Edition by Joseph Albahari and Ben Albahari.
[2] John Alexander, Billy Hollis, "Developing Web Applications with C#.NET and ASP.NET", Wiley Computer Publishing John Wiley & Sons, Inc., 2002.
[3] http://www.msdn.microsoft.com/architecture – Reference site for ASP.NET Architecture
[4] http://www.asp.net/Tutorials/quickstart.aspx - Reference site for ASP.NET coding
[5] http://www.support .microsoft.com/ - Reference site for ASP.NET coding
[6] http://www.databases.about.com/od/sqlserver/a/sqlserver 2k.htm - Reference site for SQL Server Introduction
[7] http://www.javascriptkit.com/ - Reference site for JavaScript.