YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRIC AND ELECTRONIC ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

SENIOR PROJECT

DIGITAL NOTARY

Project Supervisor: Prof. Dr. A. Coşkun Sönmez

Project Group

00011042 Alper Yavuz

İstanbul, 2006

CONTENTS

PREFACE

While I was looking for a project subject, Prof Dr. A. Coşkun Sönmez guided me to work on a digital notary application. He said this would be a research project as well, because there have not been any usage of digital notary applications in Turkey. Also legal procedures in progress. Then I began to work on March 2006.

During 4 months, Prof. Dr. Sönmez encouraged me be free to design new structures. At the end, I may say, my Digital Notary application is only a suggestion. For developped applications and legislation this would be a primary sample application only.

This is a developing working area. There may be many future works. I am thankful to Prof. Dr. A. Coşkun Sönmez for supervising this project.

ÖZET:

Gerçek noter uygulamaları kökleri Roma dönemine kadar uzanan bir dizi yasal sürecin toplamından meydana gelmektedir. Noterlik işlemlerinin ayrıntıları, her devletin yasalarına göre değişiklikler gösterse de, özü ve kapsamı bakımından noterliğin genelgeçer bir tanımı yapılabilmektedir. En genel tanımıyla noter, yasal olarak yetkili, bu yetki çerçevesinde bir dokümanın geçerli olduğunu belgeleyen ve bu işlem karşılığında bir ücret alan kişidir.

Özellikle Internetin yaygınlaşması ile beraber yeni bir gereksinim ortaya çıktı: sayısal noter. Internet üzerinden gönderilen dokümanların, kimin tarafından gönderildiğinin bilinebilmesi, elimize ulaşan bir dokümanın gönderenin izni olmadan değiştirilememesi, değiştirildiği zaman bu durumun kanıtlanabilmesi, dokümanlarda en son üzerinde işlem yapılan tarihin değiştirilemez bir biçimde belgelenebilmesi ve bunun gibi gereksinimler 'sayısal noter'in kapsamını belirliyor.

Bu sayısal noter uygulamaları Türkiye için henüz başlangıç aşamasında. Kullanılmakta olan bir sayısal noter uygulaması, bizim bilebildiğimiz kadarıyla bulunmamaktadır. Bu alanda birçok yasal düzenlemenin eksikliği duyulmaktadır. O nedenle "Sayısal Noter" (Digital Notary) projesi üzerinde çalışarak, bu projenin yasal düzenlemeler ve diğer sayısal noter uygulamalar tartışılırken bir örnek uygulama olması düşünüldü.

Sayısal Noter uygulamasında bir sunucu bir de istemci taraflı çalışan iki tane program bulunmaktadır. Bunlardan noter görevini gören, sunucu tarafında çalışan bir web servis uygulamasıdır. Bu programa 'Noter Servisi' adını veriyoruz. Sayısal Noter uygulaması esas olarak iki bölümden oluşuyor.

1) Sertifika Alma
2) Sertifika Doğrulama

Birinci bölümde istemci taraflı bir program aracılığıyla kullanıcı, bir elektronik dokümanı (her türlü bilgisayar dosyası) seçerek Noter Servisi'ne ben bu doküman için sertifika almak istiyorum diyebilir. Bu sertifikada esas olarak şu 3 bilgi tutulur.

1) Sertifika talebinde bulunan kişinin kullanıcı adı

2) Sunucunun ürettiği tekil sertifika numarası
3) Sertifikanın verildiği zaman

Sunucu bu 3 bilginin tutulduğu sertifikayı sayısal imza ile imzalayarak kullanıcıya geri gönderir. Sertifikada yeralan tüm bilgilerin yanısıra, sertifikasının alınan dosyanın bir özeti de Noter Servisi'nin veritabanında tutulur.

İkinci bölümde ise ilk bölümde olduğu gibi bir elektronik doküman seçilir ve sertifikası ile birlikte Noter Servisi'ne gönderilir. Servis gelen verileri kendi veritabanı ile karşılaştırarak, şu 3 sonuçtan birisine varır.

1) Dosya ve sertifika geçerlidir.
2) Sertifika imzası geçersizdir.
3) Bu sertifika bu dosyaya ait değildir.

Bu sertifika işlemleri yapılırken bir yetki kontrolü de yapılmaktadır. Noter programını kullanarak Noter Servisi'ne erişim için bir kullanıcı adı ve şifreye gereksinim vardır. Bu kullanıcı adı ve şifreyi noter kurumu sağlamaktadır.

Bir doküman Internet üzerinden bir uçtan diğerine gönderilecekse, bu program iki uçta da bulunuyorsa ve iki uçtaki kullanıcıların da Noter Servisi'ne erişim hakları varsa dosyanın ve sertifikanın durumu sorgulanabilir. "Dosya ve sertifika geçerlidir" geçerlidir sonucuna ulaşan bir alıcı gelen dosyanın bütünlüğünden ve göndericisinden emin olabilmektedir. Gönderici ise gönderdiği belgenin değiştirilemezliğini kanıtlayabilmektedir.

1. INTRODUCTION

The Internet is becoming part of our lives. With electronic files rapidly taking the place of conventional paper documents, we are increasingly witnessing important documents being shared and exchanged in a digital format, whether you are in the office or at home.

In the meantime, we hear the news of hacking and other computer crimes on a daily basis, and many of us still feel uncomfortable in exchanging confidential and/or valuable documents over the Internet.

With electronic files, it is difficult to tell the difference between the original and its copies. In addition, in the network environment, transactions are made without seeing each other's face. This makes it hard in cyberspace to verify "what", "who", "when", and "whom", the attributes easily verifiable in real-life transactions.

In order to remedy this insecurity and risk, Digital Notarization(or Electronic Notarization) techniques are developing. The need for a trusted third party, with strict neutrality and objectivity is satisfied by some computer techniques. [1]

As in the paper world, digital notarization services provide evidence from an unbiased third-party that records were created as claimed. Digital notarization services go a step further — they provide direct evidence those electronic files, inclusive of their respective pages and other digital components, were not altered after they were notarized. It also proves that the notarized file is the only file to which the notarization applies. [2]

2. SYSTEM ANALYSIS

In this section we will observe the infrastructure of the system by denoting some technical concepts. Digital security and its fundamental applications including cryptography, digital signatures, digital certificates and timestamps are dealed with in this section.

2.1 Cryptography

Cryptography is the art and science of keeping messages secure.

When a message is transferred from one place to another, it contents are readily available to an eavesdropper. A simple network-monitoring tool can expose the entire message sent from one computer to another in a graphical way. For an N-Tier or distributed application to be secure, all messages sent on the network should be scrambled in a way that it is computationally impossible for any one to read it.

In cryptography world the message that needs to be secured is called plaintext or cleartext. The scrambled form of the message is called ciphertext. The process of converting a plaintext to ciphertext is called encryption. The process of reconverting the ciphertext into plaintext is called decryption. Cryptography algorithms (ciphers) are mathematical functions used for encryption and decryptions.

For cryptography to be used in practical solutions algorithms used for encryption and decryption should be made public. This is possible by using a byte stream called Key. For the algorithm to encipher a plaintext to ciphertext and to decipher it back to plaintext it needs Key. [3]

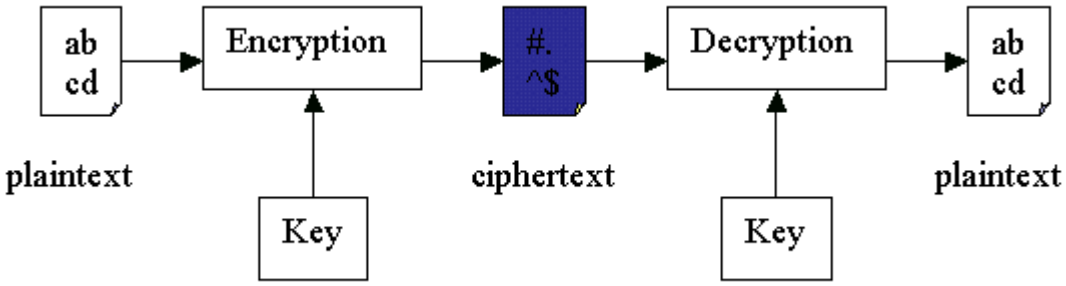

Figure 2.1: General Schema For Cryptography

2.1.1 Principles of Cryptography

In a distributed environment, for a server to trust the data it received from its peer-server there are three issues that must be addressed, namely

- Authenticity: There should be a way for the receiving server to ascertain that the message came from its peer-server only.

- Data Integrity: There should be a way for the receiving server to verify that the message was not altered in the way after it was sent by its peer-server.

- Non-repudiation: There should be a way for the receiving server to protect it from the sender falsely denying the knowledge of the message sender sent.

For a server to send any data confidently to its peer-server there is one more issue that must be addressed in addition to the above, namely

Privacy: There should be a way for the sending server to ensure that the message will be secure during transit, and except for the intended receiving server no one would be able to read it.

Cryptography, with its encryption and hashing algorithms, solves all these issues. [4]

2.1.2 Cryptographic Applications

Although asymmetric encryption, symmetric encryption and hash functions are powerful by themselves, any real world application of them in enterprise is possible only by their combination.

To epitomize their strengths,

- **Asymmetric Encryption**: Encrypting data with a key (public-key or private-key) of a key-pair makes it computationally infeasible to decrypt without using the complement key in the same key-pair.

- **Symmetric Encryption**: Encrypting data with a key (secret-key) makes it computationally infeasible to decrypt without using the same key. Also symmetric encryption is faster than asymmetric encryption.

- **Hashing**: Produces a unique message digest of known fixed size and it is computationally infeasible to get the original data from Message Digest.

Following applications of cryptography uses some or all of the above algorithms.

- Digital Envelope
- Digital Signature
- Digital Certificate

We are going to discuss two of these applications: Digital Signature and Digital Certificate. For discussing these applications, assume three users (or servers) named Alice, Bob and Eve. Alice is the sender of the message, Bob is the receiver of the message and Eve is the malicious eavesdropper. Alice has an asymmetric key-pair with a public-key called PBa and a private-key called PRa. Bob also has an asymmetric key pair with public-key called PBb and private-key called PRb. Since PBa and PBb are public-keys every one, Alice, Bob and Eve, knows them. [5]

2.2 Digital Signatures

A digital signature is like a paper signature, but it is electronic. A digital signature cannot be forged. A digital signature provides verification to the recipient that the file came from the person who sent it, and it has not been altered since it was signed.[6]

2.2.1 Digital Signature Concept

The authentication of computer-based business information interrelates both technology and the law, and calls for cooperation between people of different professional backgrounds and areas of expertise. Each field of expertise brings to the topic of authentication a different repertoire of concepts. Often the concepts from the information security field correspond only loosely to concepts from the legal field, even though both fields apply the same term to their differing concepts.

This interdisciplinary contrast exists even for basic, central concepts such as "authentication" or "digital signature". From a technical point of view, "digital signature" means the result of applying to specific information the technical processes described below. From a legal point of view, handwriting one's name on paper has been the principal means of signature for centuries. In addition, the legal concept of signature recognizes, in many cases, not only a handwritten name but any mark made with the intention of authenticating the marked document. In an electronic setting, today's broad legal concept of "signature" may well include markings such as digitized images of paper signatures, typed notations such as "s/John Smith", or even addressing notations such as letterheads, electronic mail origination headers, and the like. From an information security viewpoint, these simple electronic signatures are entirely different from the "digital signatures" described in this report and in technical documents, although "digital signature" is sometimes used colloquially or in some legal writing to mean another or any form of computer-based signature. To avoid confusion, this report uses "digital signature" only in the sense in which the term is used in information security terminology, as meaning the result of applying the technical processes described in this report.

The differences between digital signatures and other electronic signatures are significant, not only in terms of process and result, but also because those differences make digital signatures more serviceable for legal purposes. However, some electronic signatures, though perhaps

legally recognizable as signatures, may not be as secure as digital signatures, and may lead to uncertainty and disputes. [7]

2.2.2 Digital Signature Applications

Digital Signing is an application in which the sender signs the message in such a way that anyone can verify that the message is from the sender only and no one modified the message after sender signed it. It uses message-digest functions (hash algorithms) and message-digest encryption algorithm (asymmetric encryption).

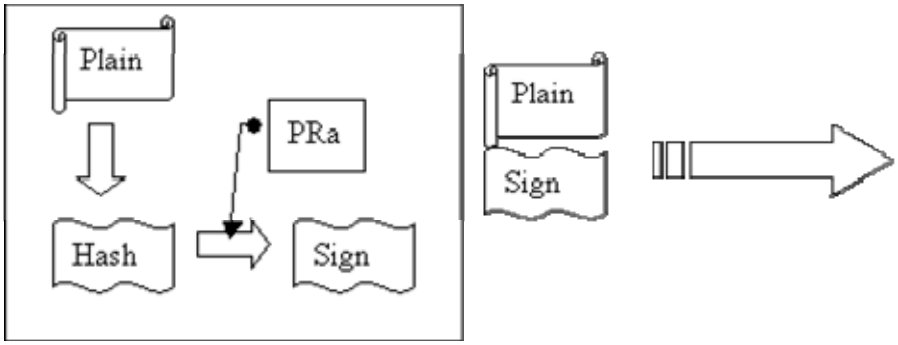

Figure 2.2: Signing Process

Alice creates the message-digest (hash) for the message (Plain) and encrypts it with her private-key (PRa) to create a digital signature (Sign) of the message. She sends the message along with the signature to the receiver, Bob.

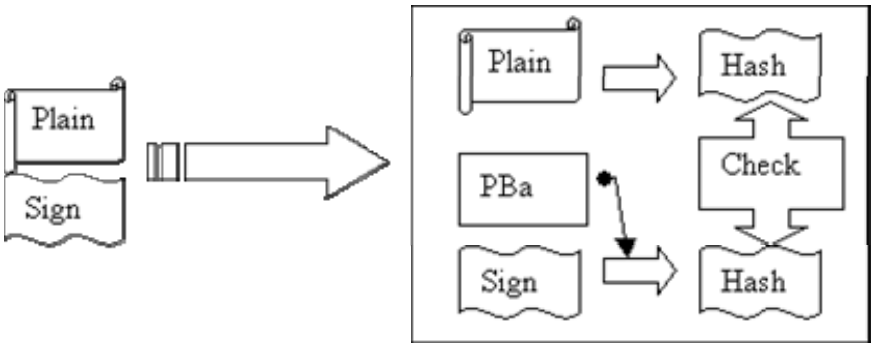

Figure 2.3: Validation Process

When Bob receives the message-signature pair from Alice, he hashes the message (Plain), with the same hash algorithm that Alice used, to generate a hash, and decrypts the Sign with Alices public-key (PBa) to generate another hash. If both the hashes are same then the message came from Alice only and no one did alter the message in the transit.

If Eve attempts to forge Alice, he would capture the message-signature pair send by Alice and send a different message-signature pair to Bob. Eve can either send a modified message with the same signature that Alice sent, or he can modify the message and create a new signature with his private-key. Bob can identify both, as the hash that he creates from the message wont match the hash that he gets by decrypting the signature using Alices public-key (PBa). [8]

2.3 Digital Certification

Digital certification is an application in which a certification authority signs a special message m, containing the name of user and the users public-key in such a way that any one can verify that the message was signed by no one other than the centralized certification authority. This message m along with its signature is called digital certificate or digital id. A typical digital certificate contains the subjects name, subjects public-key, subjects public-key algorithm and parameters, unique id of the certificate, validity period of the certificate, certificate issuer name and the issuers signature.

2.3.1 Digital Certification Applications

To understand the need for digital certification, revisit the scenario where our fictitious personalities Alice and Bob exchange digitally signed messages between them. In this Alice sends a digitally signed message to Bob; Bob ensures that the message was not altered in transit by verifying the mathematical validity of the signature using the public-key of Alice (PBa).

Alice signs the message with her private-key (PKa) of signature key-pair and sends the signature along with the message to Bob.

Bob verifies the validity of the signature using Alices public-key (PBa).

The big challenge in this solution is publicizing the Alices public-key.

1. How Bob gets Alices public-key?
2. How can Bob be sure that the key he received is Alices public-key and not someone elses?

An apparent quick solution for this problem is Alice handing over her public-key to Bob in a secured manner. But this presupposes that Alice and Bob have had some form of secured communication prior. Even If Alice publicizes her public-key in this way, it is not scalable. If she needs to have similar secured communication with say 100 more users then this process becomes a nightmare. Also for Bob getting public-key in this way and managing them is a nightmare.

If both Bob and Alice can trust some intermediary who in a secured way can bind a public-key to the owner of it, the problem will be solved. Alice can simply ask this intermediary to certify her public key. Bob needs to trust only this intermediary. He can verify that trusted intermediary certified the public-key. Since both Alice and Bob need to trust the public-key with one person, this scales for any number of users. For anyone with whom Alice needs to communicate she can send the same certificate. Also Bob can verify the public-key for all the users who are certified by the intermediary.

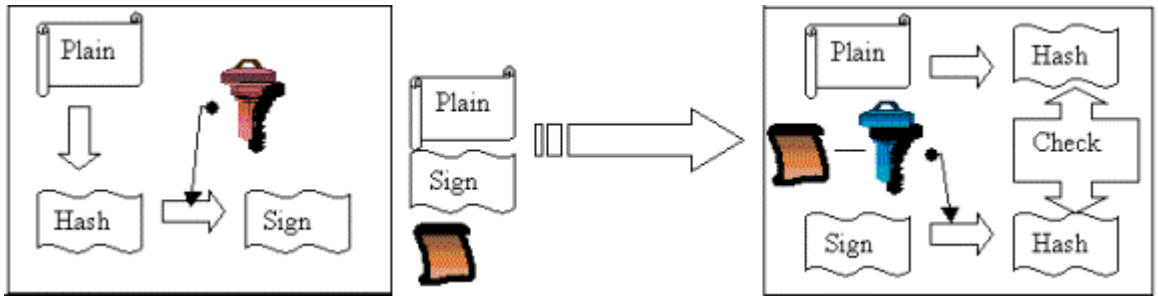

Figure 2.4: Certification Process Overview

Alice signs the message with her private-key and sends her certificate along with the message and signature to Bob. Bob verifies the validity of the signature using Alices public-key that he extracts from the Alice certificate issued by certifying authority. [9]

3. FEASIBILITY STUDY

Feasibility study is a proposal to determine whether a project is viable or feasible not only from an economic but also technical point of view. It is also establishes whether the project can be implemented successfully not only to yield meaningful returns on investment or bottom line profitability but also in aspects such as marketability of the type and nature of business envisaged and its technical viability as well.

3.1 Technical Feasibility

The purpose of the technical feasibility is to determine the technical feasibility of each implementation alternative. In technical feasibility, the possibility to build “Digital Notary” application is examined. Firstly, the technologies to be used on the project are investigated. Alternatives for each technology are identified.

3.1.1 Software Requirements

Software requirements consist of operating system, database server, application software and web server.

3.1.1.1 Operating System

Alternatives for Operating System:

- **Linux :** Linux is the most versatile Unix based platform that serves a number of functions. It is particularly suitable for meeting user’s Internet requirements, as the best OS for a web server, as well as mailing, streaming, and file serving. Linux is a very cost-effective choice; it uses hardware efficiently, and there are a huge number of free applications available on the Internet.

  Ideal uses: For all Internet functions; Web servers, mail server, streaming and file servers.

- **FreeBSD :** FreeBSD is a version of BSD Unix that was designed for the Intel X86 processor. FreeBSD is a very stable open source operating system, and a good alternative to Linux. It is an extremely well integrated and tested system. It is inexpensive and there are a large number of free applications available.
  This OS is for those who are experienced with administering a Unix OS from the command line.

  Ideal uses: Free BSD is ideal for Web servers and virtual hosting servers. .

- **Windows NT:** Window's NT Server has a familiar interface for most IT teams to work with. It integrates well with other Microsoft applications and there is wealth of commercial applications available for this platform.

  Ideal uses: Windows NT/2000 is a SQL7 server, Web server and backoffice integration server.

- **Windows XP Professional:** The operating system using by client side application is Windows XP Professional. The advantages of Windows XP professional are defined below.

WIN XP Advantages:

Remote Desktop Remote: Desktop allows you to make your desktop computer a Terminal Server. This is very handy if you move from computer to computer and occasionally need to access your primary desktop from another machine. Remote Desktop is also nice when you are on the road, as you can connect to your machine over a VPN connection from a remote location.

Offline Files and Folders Offline files and folders allow you to store the contents of a network share on the local disk. This feature isn't much use for permanently connected desktops, but its great if you travel with a laptop are automatically synchronized so that the files in the network share are up to date.

Encrypting File System The Encrypting Files System (EFS) allows you to encrypt files so that unauthorized users cannot view their contents. This is helpful in a high security or a laptop environment. EFS depend on the NTFS file system. Both Windows XP Home and Professional support the NTFS file system, but EFS is disabled on XP Home. EFS are best

used on laptop computers. If you don't use XP on laptops, you might not consider this a must have feature.

Multi-language Support Multi-language support allows you to change the language used is various dialog boxes and applications "on the fly". This is a helpful feature is multinational corporations, but is probably not something you require if you work in one language exclusively.

Multiprocessor and Multiple Monitor Support Multiprocessor and multiple monitor hard support in Windows XP Professional allows you use up to two processors and 10 monitors. XP Home allows only a single processor and monitor.

Among these operating systems, Windows XP Professional is preferred, because software development infrastructure is .NET Framework and development environment is Visual Studio .NET. These software works with Windows operating systems. Another reason for choosing Windows XP Professional is the fact that users can take the advantage of the system's power and security, coupled with the familiarity and ease-of-use of the Windows graphical interface.

3.1.1.2 Application Software

Alternatives for Application Software:

- ✓ ASP.NET
- ✓ PHP
- ✓ JSP

Here it is shown the comparision of these in tables :

CLR – Common Language Runtime
COM TI - COM Transaction Integrator
EJB – Enterprise JabaBeans
J2EE – Java 2 Enterprise Edition
JAXP - Java API for XML Parsing
JCA – Java Connector Architecture
JMS - Java Message Service
JNI – Java Native Interface
JRE - Java Runtime Environment
JSP – Java Server Pages
MSMQ - Microsoft Message Queue
SOAP - Simple Object Access Protocol
UDDI - Universal Description, Discovery, and Integration
WSDL - Web Services Description Language

Feature	.NET	J2EE
Single Vendor Solution	Yes: Microsoft	Yes: Through legacy system vendors such as IBM and Oracle and BEA
Type of technology	Standard	Product
Middleware Vendors	Microsoft	30+
Interpreter	CLR	JRE
Dynamic Web Pages	ASP.NET	JSP
Middle-Tier Components	.NET Managed Components	EJB
Existing System Support and Integration	Yes: Host Integration Server 2000. COM TI, MSMQ, BizTalk	Yes: JCA, JMS, Web Services, CORBA, JNI
Language Support	Yes: All languages supported by Microsoft.	Java: Other languages can be bridged into J2EE through web services but they cannot be intermixed (ORBA, JNI and JCA)
Portability	Windows	All platforms: JRE
Web Services Support	Yes: Visual Studio .NET, SOAP, WSDL, UDDI	Yes: JAXP, SOAP, WSDL, UDDI,

Comparison information based on a report written by Chad Vawter and Ed Roman.

Table 3.1 The comparision of .NET and J2EE

Feature	PHP	ASP.NET
Coding Language	C, C++ style scripting language with older ASP style mark-up. Supports some OOP concepts.	Supports more than 25 languages, but the 2 that are most-commonly used are Visual Basic .NET and C#.
Compiled Application Logic	Compiled and can be run as an executable	Supported, in both dynamically-compiled and precompiled modes.
Full-Page Output Caching	No native support	Supported, caches different versions of the page based on one or more URL parameters, browser type, a custom function.
Partial-Page Output Caching	No native support	Built-in support through use of User Controls. Data and other objects can be cached with sophisticated expiration rules using the Cache API.
Database Access	Has drivers for most databases on the market as well as open-source databases	Supports OLE-DB and ODBC directly, and includes native drivers for Microsoft SQL Server™ and Oracle.
Database Output	Datasets are returned as PHP variables and can be outputted like any other variable	Templated data binding to server-side controls for ease of development, or manual looping.

Table 3.2 The comparision of .NET and PHP

According to these comparisions above we have chosen ASP.NET.

3.1.1.3 Database Server

Alternatives for Database Software:

- MS SQL Server
- MySQL
- DBase
- DB2
- Oracle

As the database server of "Digital Notary, MS SQL Server 2000, which stands for Microsoft - Structured Query Language Server database system, is chosen for the following reasons:

- Microsoft SQL Server 2000 is capable of supplying the database services needed by extremely large systems. Large servers may have thousands of users connected to an instance of SQL Server 2000 at the same time.
- SQL Server 2000 has full protection for these environments, with safeguards that prevent problems, such as having multiple users trying to update the same piece of data at the same time. SQL Server 2000 also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users.
- SQL Server 2000 can handle large files.

XML Support: SQL Server 2000 can use XML to insert, update, and delete values in the database, and the database engine can return data as Extensible Markup Language (XML) documents.

3.1.2 Hardware Requirements

Hardware requirements of DN application is decided by the means of considering minimum and recommended hardware requirements of the operating system of the project. The hardware configuration of DN project is designed according to recommended hardware for Windows XP Professional.

Computer:	IBM or 100% compatible
Processor:	Intel Pentium 133 MHz or equivalent
Memory:	32 MB
Drives:	650 MB Disk space CD-ROM / DVD drive
Video:	VGA or higher
Controls:	Microsoft mouse / keyboard or compatible
Direct X:	Direct X 7.0
Other:	NIC required for network installation

Table 3.3 Minimum hardware requirements for Windows XP Professional

Computer:	IBM or 100% compatible
Processor:	PIII/AMD K6 450 megahertz or higher
Memory:	256 megabytes of RAM or higher
Drives:	6 gigabytes hard drive or larger CD-ROM / DVD drive
Sound:	Sound Card recommended
Video:	Super VGA (800 × 600) or higher-resolution video adapter and monitor
Controls:	Microsoft mouse / keyboard or compatible
Direct X:	Direct X 7.0
Other:	NIC required for network installation

Table 3.4 Recommended hardware for Windows XP Professional

Processor:	Pentium 450 megahertz
Memory:	256 megabytes of RAM
Drives:	6 gigabytes hard drive
Video:	Super VGA (800 × 600) resolution video adapter
Main Board:	478-Pin Pentium 4
Screen:	Screen Card (AGP4x, 32MB)
Monitor:	15" Digital, 1024x768
CD-ROM:	52X CD-ROM driver
Tower:	Midi P4 ATX Tower
Floppy:	1.44 Floppy
Keyboard:	Q Keyboard
Mouse:	PS/2 Mouse
UPS:	1000VA UPS (Uninterruptible Power Supply) -Provides up to 75 minutes backup time. Battery load capacity: 1000VA (550 Watts)
Modem:	ADSL Modem

Table 3.5 Hardware requirements of “Digital Notary” project

3.1.3 Requirements for Internet Service

For DN project, there is need for web hosting, because one part of DN is designed to run on a web server. Web Server alternatives are dealed below. Also users should have a requirement for Internet access. There are mainly three alternatives for Internet access. These are listed below;

- **56K Internal Modem**
- **DSL Modem**
- **ADSL Modem**

Among these alternatives, 56K Internal Modem is eliminated, because it provides low speed access to Internet and it is on a standard analog phone line that causes extra payment for duration of Internet access. DSL Modem is also not preferred because DSL Modem is convenient when the speed of receiving data is almost equal to the speed of sending data.

ADSL Modem is the best choice for this project, because in ADSL Modem, " the speed of receiving data (downstream rate) is different than the speed of sending data (upstream rate) . In DN project, the speed of sending data (client side sends low amount of data, such as search words, search options etc) is much lower than the speed of receiving data (large amount of data is received, such as search results, words box, page titles etc). Because of this structure of DN project ADSL Internet Service with ADSL modem is convenient to this system.

In addition to this, Asymmetric Digital Subscriber Line (ADSL) provides 24 Hour Internet access at speeds up to 140 times faster than a traditional 56k Kbps modem on a standard analog phone line. Furthermore, with ADSL there are no dial-up delays or busy signals to contend with.

Specifications of ADSL modem which will be used in DN;

ADSL Modem:

- USB interface
- Data Transfer Rate - Upstream: 1 Mbps
  - Downstream: 8 Mbps
- Bridge/Router Properties

4. DIGITAL NOTARY (DN) SYSTEM DESIGN

4.1 Secure Hash Functions

The foundation of the Digital Notary system is the use of cryptographically secure collision-free hash functions. As illustrated in Figure 4.1, these functions take an arbitrary length bit stream as input (e.g., a digital document) and produce a fixedsized condensed representation of the bit stream as output. The condensed representation of the bit stream is referred to as a hash value, also known as a message digest or digital fingerprint.

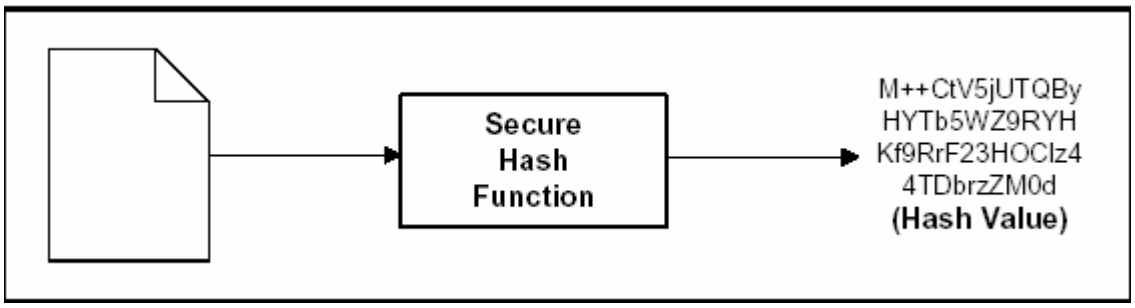

Figure 4.1: Secure Hash Computation

Hash functions have several important properties that make them particularly useful:

- **They are one-way**. Given a hash value produced by a hash function, there is no way to determine anything about the bit stream that produced that hash value.
- **The hash value is a function of all bits in the input**. Changing even a single bit in the input bit stream will cause the hash function to produce a completely different hash value.
- **They are collision free**. Given an input stream and corresponding hash value, it is computationally infeasible to find another input stream that will produce the same hash value.

In this project MD5 algorithm is used in digesting functions

4.2 Notarization

Notarization is the process of sealing the contents of data in time by providing a fail-safe means of detecting tampering, and attaching a witnessed time-stamp to the data. The notarization process, illustrated in Figure 4.2, consists of the following steps:

1. The application computes the secure hash of the document to be notarized.
2. The hash is sent to the Digital Notary Web Service via a notarization request.
3. The Digital Notary Web Service associates a time-stamp with the hash and records the hash and time-stamp in a database called the Timestamp Database.
4. The Digital Notary Web Service constructs a XML Certificate containing the document hash and assigned time-stamp and returns it to the application in a notarization response. The Web Service also signs the XML Certificate against changing and store the sign in XML Certificate. The certificate validity provided by this signing mechanism.
5. The application stores the XML Certificate, along with the associated document, so that it can be used to verify the integrity of the document at any point in the future.

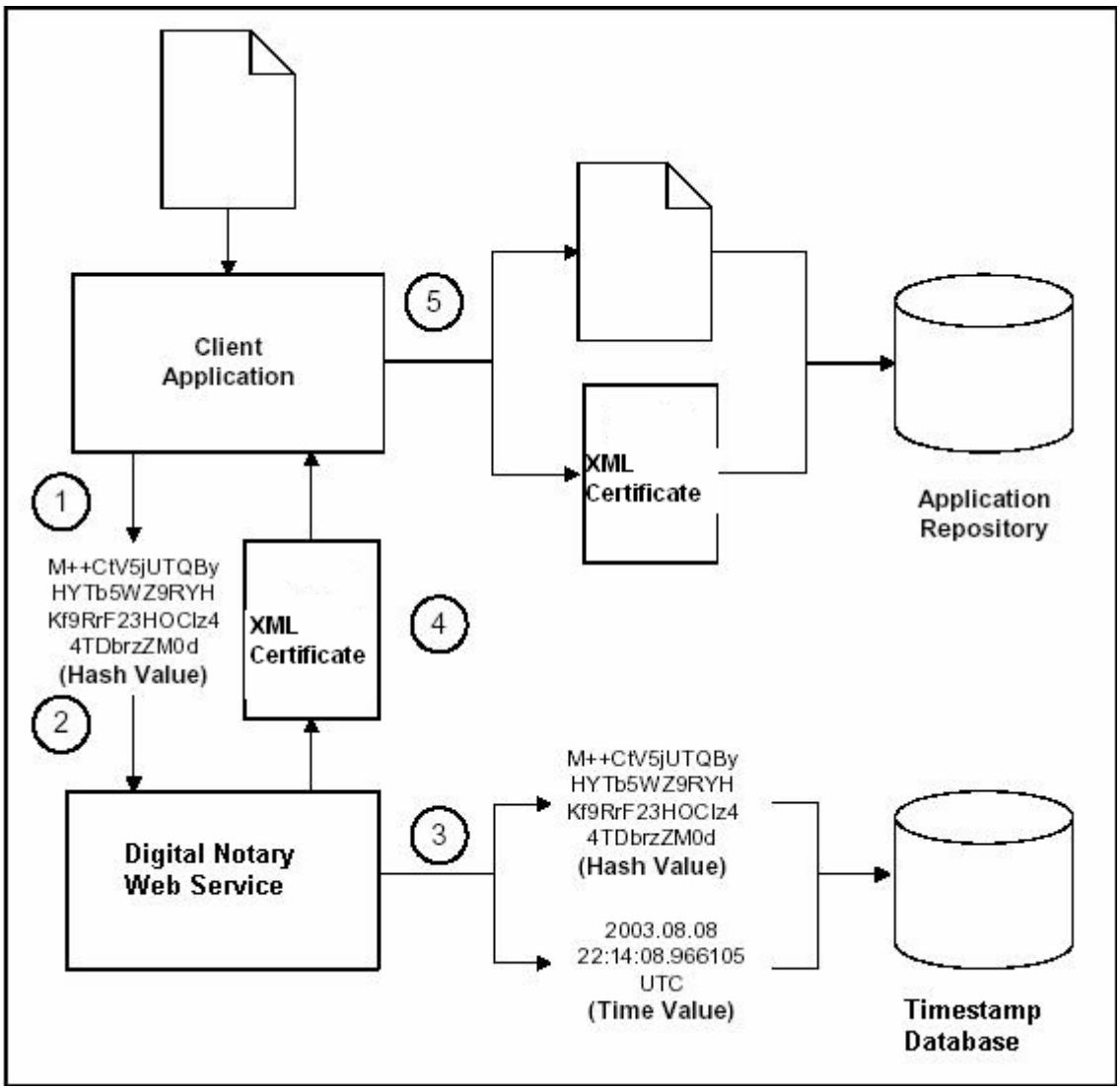

Figure 4.2: Notarization Process

4.3 Validation

Validation is the process of proving to any interested individual that a piece of notarized digital data is a "true copy" and that the time-stamp is valid. Validation proves conclusively that a piece of notarized digital data existed in an unaltered form at exactly the stated point in time. Any tampering with the contents of the data, or with the time-stamp, results in failure when validating a notarized piece of data.

The validation process, illustrated in Figure 4.3, consists of the following steps:

1. The application retrieves the document and associated XML Certificate for validation.
2. The application computes the secure hash of the document.
3. The application sends a validation request containing the XML Certificate and the document hash to the Digital Notary Web Service.
4. The Digital Notary Web Service validates that the hash value and time-stamp stored in the XML Certificate match the values that are recorded in the Timestamp Database.
5. The Digital Notary Web Service sends a validation response to the application that indicates whether the XML Certificate was valid or invalid. [10]

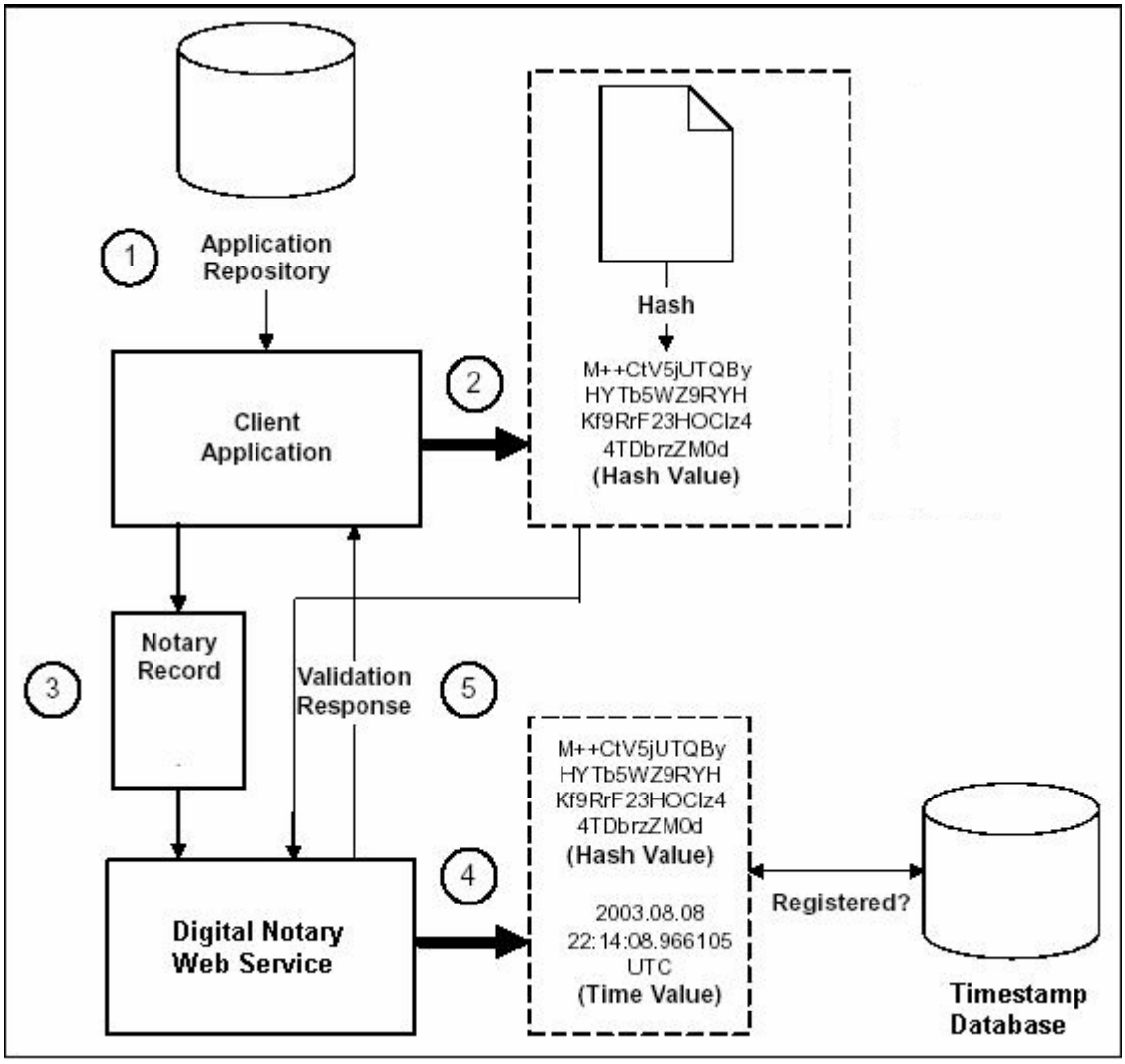

Figure 4.3: Validation Process

5. PROGRAM INTERFACE

In this section, we will show Digital Notary software's screen outputs and their explanations. The project consists of two main parts. One is client side code, the other is server side code.

5.1 Digital Notary Service

The server side part is a web service. It has also a web interface but it is not used via this interface, so only general view will be shown.

Figure 5.1 Web Service Interface

5.2 Digital Notary Client

On client side, there is a windows application in which user can notarize and verify his documents. These processes are placed in a tab control which help us show them seperately.

5.2.1 Getting Digital Certificate

This process is done in three steps:

1) Via “Choose A File” button and a dialog, user may choose an electronic document (such as office document, an image etc.) If chosen file is a readable one, its content can be read in a text box under the “Choose A File” section.
2) In this software the whole document are not sent with Internet, because of security causes, instead, only file digest is sent. User may generate file digest by “Generate File Digest” button. The file digest is shown in a text box
3) “Get Timestamp and Certificate” button is used in third order to send file digest and other file infos to Digital Notary Server and to get server’s certificate. Certificate is shown in a text box and saved in a folder. Certificate folder is shown on the left side in a list box.

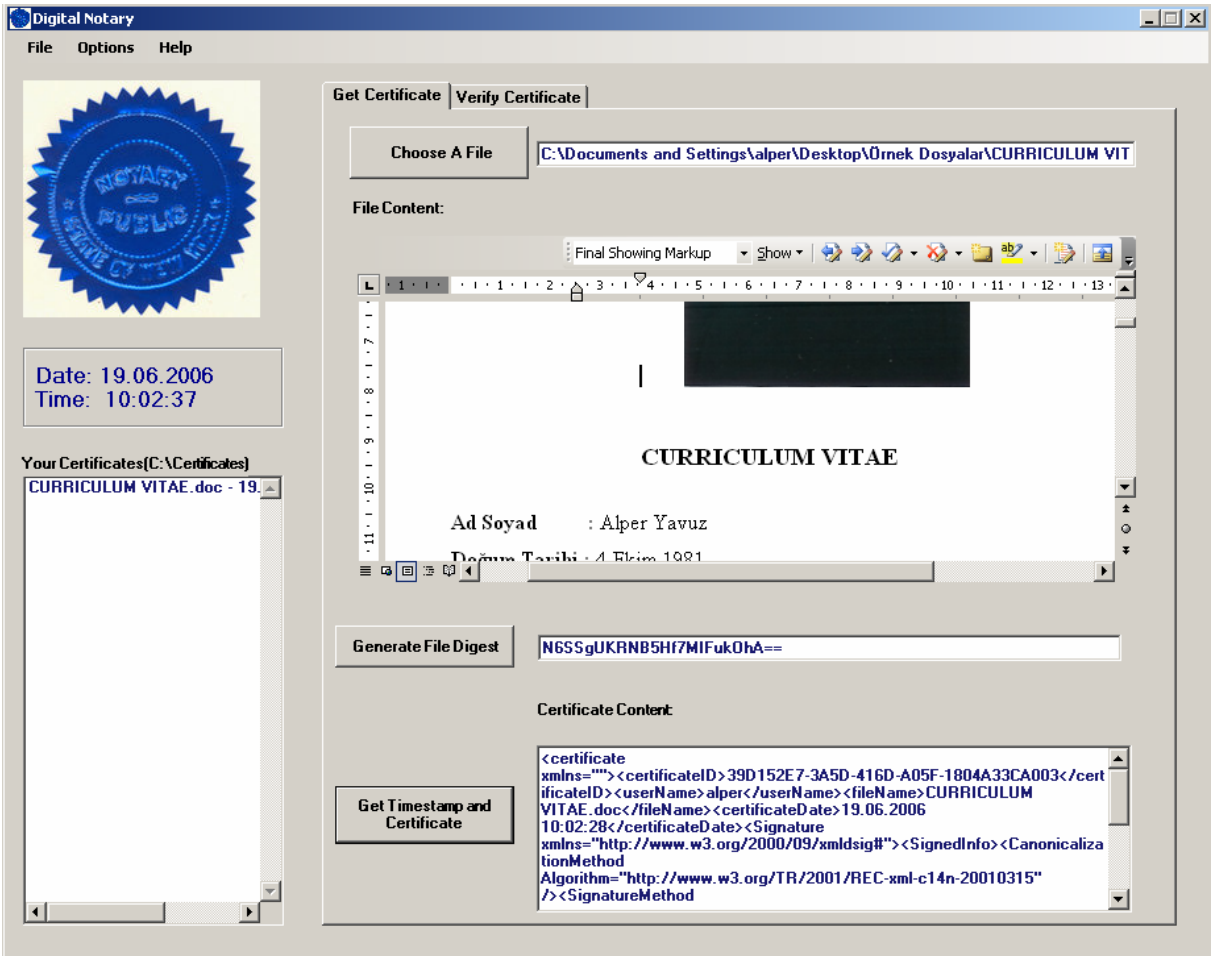

Figure 5.2 Get Certificate tab

Before accessing Digital Notary Service users have to register to system. They have username and password. In a dialog box they write them and get the result if they can pass or not.

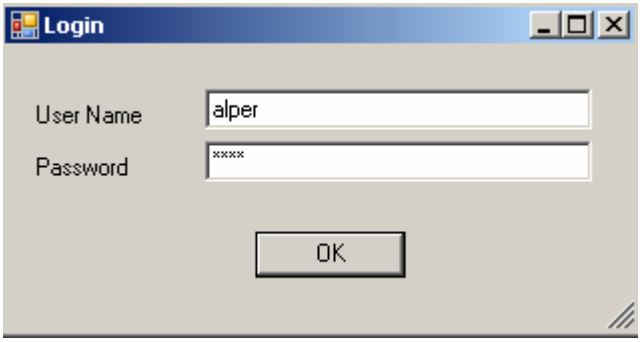

Figure 5.3 Login dialog

A sample digital certificate which generated by Digital Notary Server is shown below.

```
- <certificate xmlns="">
    <certificateID>63D83048-AF0B-43B6-BFDF-2B279C255408</certificateID>
    <userName>bonanza</userName>
    <fileName>CURRICULUM VITAE.doc</fileName>
    <certificateDate>31.05.2006 14:29:07</certificateDate>
  - <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    - <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      - <Reference URI="">
        - <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <DigestValue>INLMpE+1kERF5gO7bJ3a/M2q718=</DigestValue>
        </Reference>
      </SignedInfo>

        <SignatureValue>y3rSjljjLsgi0ktvmEKl1vZzpyMbxFLPN07H5u6dHD2yGHC/h68YxUXDSndV6mz1YzwQhcUJTJ1ErCY677+CD1OL9rq8cBOLOKzB9DlZ0xpfGu3XW
    </Signature>
  </certificate>
```

Figure 5.4 An XML certificate

5.2.2 Verifying Digital Certificate

This process is done in four steps:

1) Via “Choose A File” button and a dialog user may choose an electronic document (such as office document, an image etc.) If chosen one is a readable file, its content can be read in a text box under the “Choose A File” section.

2) User may generate file digest by “Generate File Digest” button. The file digest is shown in a text box
3) “Choose a Certificate” button is used in third order to chose certificate which will be used to verify file if it is changed. Certificate content is shown in a text box. Certificate folder is shown on the left side in a list box.
4) By clicking “Verify Certificate” button user will see the result of this whole process. There may be four kind of response message. These are: “File and certificate is valid”, “The XML signature is not valid”, "Cerificate does not belong to this file" and "Certificate timestamp is not valid."

   By clicking “Verify Certificate” button, user will also be prompted to login system if he does not login at that session yet.

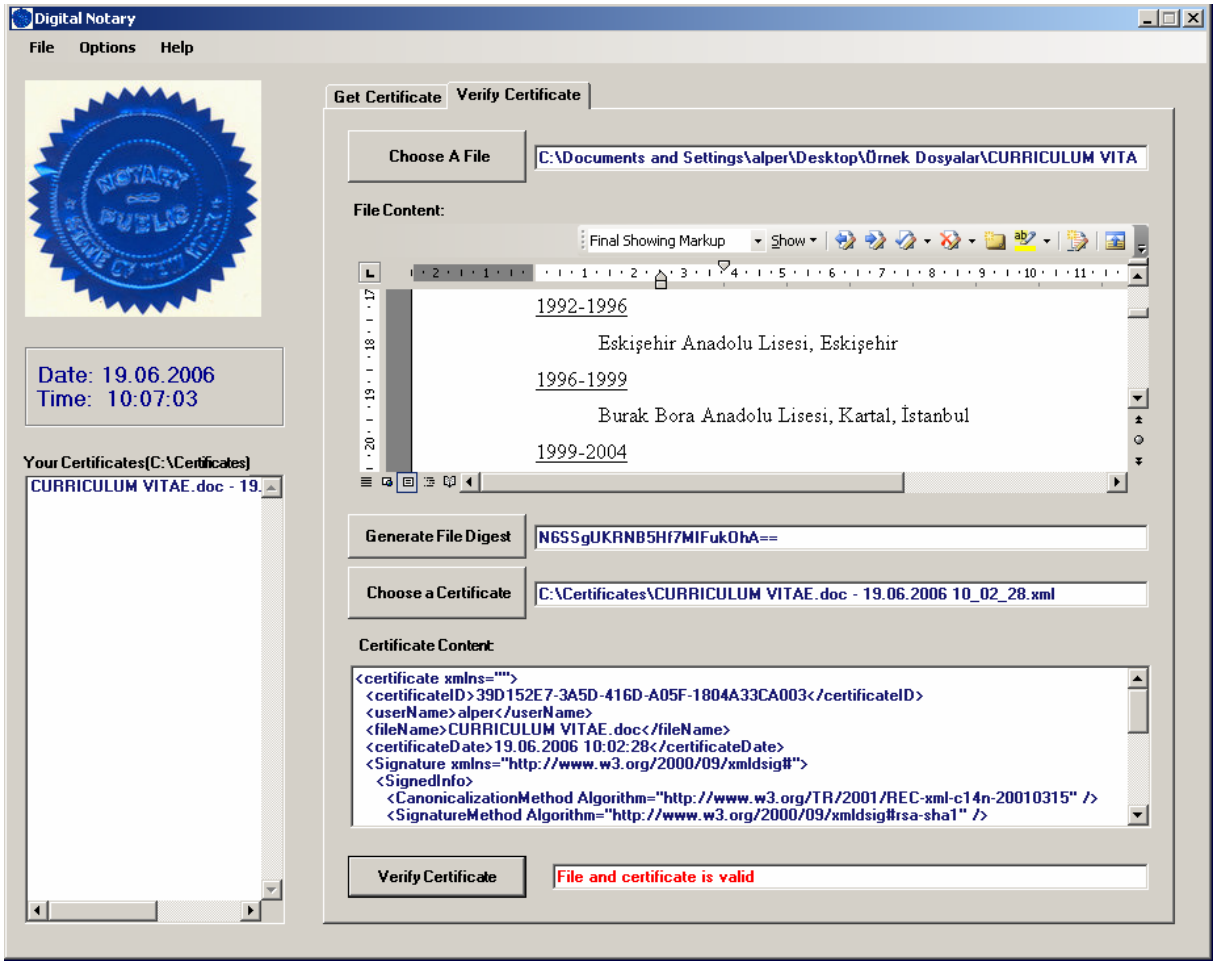

Figure 5.5 Verify Certificate tab

6. REFERENCES

1) http://www.jnotary.com/eng/epage.html
2) http://www.exchangeproof.com/applications.asp
3) http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/CryptEncryption11232005065300AM/CryptEncryption.aspx?ArticleID=2f4b0ae5-28e8-424b-825f-85a7043b5247
4) http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/Cryptography211242005003308AM/Cryptography2.aspx?ArticleID=cf900b08-35ed-4524-aeff-6806265a4196
5) http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/Cryptography211242005003308AM/Cryptography2.aspx?ArticleID=cf900b08-35ed-4524-aeff-6806265a4196
6) http://sig.nfc.usda.gov/pki/glossary/glossary.html
7) http://www.commerce.utah.gov/digsig/tutorl.htm
8) http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/Cryptography211242005003308AM/Cryptography2.aspx?ArticleID=cf900b08-35ed-4524-aeff-6806265a4196
9) http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/Cryptography311242005015640AM/Cryptography3.aspx?ArticleID=69844e8f-7364-4cf0-8f68-928952d1182d
10) http://www.surety.com/AbsoluteProofWhitepaper_final1.pdf