B. TECH. PROJECT REPORT

On

DIGITAL SIGNATURES WITHOUT HARDWARE TOKEN AND CENTRAL PUBLIC KEY STRUCTURE (PKI)

BY

CHALLAPALLI CHAKRADHAR REDDY
THOTA SRI RANGA VINEEL

DISCIPLINE OF COMPUTER SCIENCE ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

November 2018

DIGITAL SIGNATURES WITHOUT HARDWARE TOKEN AND CENTRAL PUBLIC KEY STRUCTURE (PKI)

A PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE ENGINEERING

Submitted by:
CHALLAPALLI CHAKRADHAR REDDY

THOTA SRI RANGA VINEEL

Guided by:
DR. BODHISATWA MAZUMDAR
Assistant Professor

INDIAN INSTITUTE OF TECHNOLOGY INDORE
November 2018

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"Digital Signatures without Hardware Token and Central Public Key Structure (PKI)"** submitted in partial fulfilment for the award of the degree of Bachelor of Technology in 'Computer Science Engineering' completed under the supervision of **Dr. Bodhisatwa Mazumdar, Assistant Professor, Computer Science Engineering,** IIT Indore is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

(C CHAKRADHAR REDDY)

DATE: __________________

(THOTA SRI RANGA VINEEL)

DATE: __________________

CERTIFICATE by BTP Guide

It is certified that the above statement made by the students is correct to the best of my knowledge.

Dr. Bodhisatwa Mazumdar

Assistant Professor

Computer Science Engineering

Preface

This report on "Digital Signatures without Hardware Token and Central Public Key Structure (PKI)" is prepared under the guidance of Dr. Bodhisatwa Mazumdar.

Through this report we have tried to give a method to implement digital signatures without using hardware token and public key structure.

We have tried to the best of our abilities and knowledge to explain the content of our method in lucid manner with a working prototype.

This report contains 5 chapters and outline is given in the table of contents page. Prototype details are also included.

CHALLAPALLI CHAKRADHAR REDDY
150001007
B.Tech. IV Year
Discipline of Computer Science Engineering
IIT Indore

THOTA SRI RANGA VINEEL
150001036
B.Tech. IV Year
Discipline of Computer Science Engineering
IIT Indore

Acknowledgements

We wish to thank Dr. Bodhisatwa Mazumdar for his kind support and valuable guidance.

CHALLAPALLI CHAKRADHAR REDDY
150001007
B.Tech. IV Year
Discipline of Computer Science Engineering
IIT Indore

THOTA SRI RANGA VINEEL
150001036
B.Tech. IV Year
Discipline of Computer Science Engineering
IIT Indore

Abstract

The intention of project is to design a simple method of digital signatures that works without using hardware tokens or public key structure. The project has been done by designing such method which overcomes the challenge. Also a prototype has been made using the design of the method to illustrate how the method succeeds to complete the challenge.

Table of Contents

Chapter 1 Overview

1.1 Introduction

A digital signature is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file, etc.) is **authentic**. Authentic means that you know who created the document and you know that it has not been altered in any way since that person created it. Its goals are to ensure authentication, integrity and non-repudiation.

Digital signatures rely on certain types of encryption to ensure authentication. Encryption is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Authentication is the process of verifying that information is coming from a trusted source. These two processes work hand in hand for digital signatures.

1.2 Motivation

The motivation for this project is derived from the fact that using a hardware token is expensive and reproducible by others, hence authenticity is a problem. Overcoming it is also complex. Similarly, public key infrastructure uses public and private keys in authenticity, where private key stays only with author and public key is shared to all. If private key is compromised, then the method of authenticity fails. Hence requirement of a method which doesn't use hardware token and public key structure has raised and has become the cause of motivation of this project.

Chapter 2 Project Overview

2.1 Statement of the Challenge

Digital Signature ensures the authentication of files/documents, it is required to have a technical solution for digitally signing a document without hardware token and implementation of digital signature without central Certifying Authority (CA).

2.2 Evolution of the Challenge

- Office automation is a central feature of all automation efforts. It involves Management and user mail system. The criticality of all electronic information sharing systems is the capability of a system to implement the following aspects as demanded by Information Technology (IT) Act:
  a. Authentication
  b. Non-repudiation
  c. Integrity

2.2.1 Authentication

Authenticity refers to genuinity in a file/message being generated from the specified author.

Usually digital signatures are assumed to be a mathematical scheme for presenting the authenticity of digital messages or documents based on Wikipedia.

There are several ways to authenticate a person or information on a computer:

Password: The use of a user name and password provide the most common form of authentication. You enter your name and password when prompted by the computer. It checks the pair against a secure file to confirm. If either the name or password do not match, then you are not allowed further access.

Checksum: Probably one of the oldest methods of ensuring that data is correct, checksums also provide a form of authentication since an invalid checksum suggests that the data has been compromised in some fashion. A checksum is determined in one of two ways. Let's say the checksum of a packet is 1 byte long, which means it can have a maximum value of 255. If the sum of the other bytes in the packet is 255 or less, then the checksum contains that exact value. However, if the sum of the other bytes is more than 255, then the checksum is the remainder of the total value after it has been divided by 256. Look at this example:

- Byte 1 = 212
- Byte 2 = 232
- Byte 3 = 54
- Byte 4 = 135
- Byte 5 = 244
- Byte 6 = 15
- Byte 7 = 179
- Byte 8 = 80
- **Total = 1151.** 1151 divided by 256 equals 4.496 (round to 4). Multiply 4 x 256 which equals 1024. 1151 minus 1024 equals **checksum of 127**.

Private Key Encryption: Private Key means that each computer has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to the other computer. Private Key requires that you know which computers will talk to each other and install the key on each one. Private Key encryption is essentially the same as a secret code that the two computers must each know in order to decode the information. The code would provide the key to decoding the message. Think of it like this. You create a coded message to send to a friend where each letter is substituted by the letter that is second from it. So "A" becomes "C" and "B" becomes

"D". You have already told a trusted friend that the code is "Shift by 2". Your friend gets the message and decodes it. Anyone else who sees the message will only see nonsense.

Public key encryption: Public key encryption uses a combination of a private key and a public key. The private key is known only to your computer while the public key is given by your computer to any computer that wants to communicate securely with it. To decode an encrypted message, a computer must use the public key provided by the originating computer and its own private key.

The key is based on a hash value. This is a value that is computed from a base input number using a hashing algorithm. The important thing about a hash value is that it is nearly impossible to derive the original input number without knowing the data used to create the hash value.

Digital certificates: To implement public key encryption on a large scale, such as a secure Web server might need, requires a different approach. This is where digital certificates come in. A digital certificate is essentially a bit of information that says the Web server is trusted by an independent source known as a Certificate Authority (CA). The Certificate Authority acts as the middleman that both computers trust. It confirms that each computer is in fact who they say they are and then provides the public keys of each computer to the other.

2.2.2 Non-repudiation

Non-repudiation refers to a situation where a statement's author cannot successfully dispute its authorship.

In digital security, non-repudiation means:

- A service that provides proof of the integrity and origin of data.
- An authentication that can be said to be genuine with high confidence.
- The most common method of verifying the digital origin of data is through digital certificates, a form of public key infrastructure that includes digital signatures.

2.2.3 Integrity

Integrity means ensuring that file/message has not been altered between source and receiver. This is done by using checksum, encryption, etc.

- The recommended means of implementing the above aspects through Digital Signatures is to implement PKI through a Central Certifying Authority and hardware based digital signature tokens. In absence of Central Certifying Authority this scheme fails and cannot be implemented under the present circumstances. Moreover, to obviate the security issues related to hardware tokens and to cater for digital signing of documents on the go, a technical solution for digitally signing a document without hardware token is required.

2.3 How is it Overcome Previously

Not yet overcome.

Chapter 3 Innovation

3.1 Method

The issue of digital signature without hardware token can be resolved by using software token. The details of which are given below:

3.1.1 Software Token:

- A software token is a type of two-factor authentication security device that may be used to authorize the use of computer services. Software tokens are stored on a general-purpose electronic device such as a desktop computer, laptop, PDA or mobile phone and can be duplicated. Contrast hardware tokens, where the credentials are stored on a dedicated hardware device and therefore cannot be duplicated (absent physical invasion of a device).
- Software tokens are something one does not physically possess, they are exposed to unique threats based on duplication of the underlying cryptographic material - for example, computer viruses and software attacks. Both hardware and software tokens are vulnerable to bot-based man-in-the-middle attacks, or to simple phishing attacks in which the one-time password provided by the token is solicited and then supplied to the genuine website in a timely manner Software tokens do have benefits there is no physical token to carry, they do not contain batteries that will run out, and they are cheaper than hardware tokens.

3.1.2 Security Architecture:

- There are two primary architectures for software tokens i.e. shared secret and public-key cryptography

- For a shared secret, an administrator will typically generate a configuration file for each end-user. The file will contain a user name, a personal identification number and the secret. This configuration file is given to the user.

- The shared secret architecture is potentially vulnerable in a number of areas. The configuration file can be compromised if it is stolen and the token is copied. With time-based software tokens, it is possible to borrow an individual's PDA or laptop, set the clock forward and generate codes that will be valid in the future. Any software token that uses shared secrets and stores the PIN alongside the shared secret in a software client can be stolen and subjected to offline attacks. Shared secret tokens can be difficult to distribute, since each token is essentially a different piece of software. Each user must receive a copy of the secret, which can create time constraints.

- Some newer software tokens rely on public-key cryptography or asymmetric cryptography. This architecture eliminates some of the traditional weaknesses or software tokens, but does not affect their primary weakness (ability to duplicate) A PIN can be stored on a remote authentication server instead of with the token client, making stolen software token no good unless the PIN is known as well. However, in the case of virus infection, the cryptographic material can be duplicated and then the PIN can be captured (via key logging or similar) the next time the user authenticated. If there are attempts made guess the PIN, it can be detected and logged on the authentication server, which can disable the token. Using asymmetric cryptography also simplifies implementation, since the token client can generate its own key pair and exchange public keys with the server.

3.2 Proposition of Solution

The solution that we have come up with is Online Digital Signatures. In this we register a user through online portal with his email id and mobile number to make sure it's owner is authorized user. Then the user can login and upload a file's hash that we store on our database. Any person with the file which may be of any type, can cross check the author details and if information in it is altered.

Our idea's base is built on the definition of digital signatures but simplified by making it online and storing fixed size of irreversible output generated by an algorithm whose inverse is not predictable or possible. We are naming this method as Online Digital Signature.

We felt there is no need for complicated mathematical schemes and hardware tokens. No need of key infrastructure. What we just need is a function/algorithm whose inverse doesn't exist and can't be estimated by any means or at least, not within a lifetime, a secured database to store, a secured web application to connect to the database and being online. The above is the solution to the problem.

3.3 Advantages of this method

- So one has to just register and share their email id and phone number for others to verify.
- It is simple and secured as no one can undo hash of the file to get the information in file.
- There is no need of large space as hash of each file registered occupies only at most 64 bytes even if we use 512-bit hashing.
- Now-a-days being online is far easy and there is only involvement of uploading but not downloading.
- Once a file is registered can't be undone, hence even the author can't deny that the file belongs to him.
- Hence it covers all the requirements of digital signatures, doesn't require hardware token or PKI and is easy to use.

3.4 Disadvantages of this method

- One has to have a valid email id and phone number and are willing to share it for verification.
- One has to be online to verify or sign.

Chapter 4 Prototype

4.1 Overview

We have developed a prototype which is based on our idea of online digital signatures.

Following are details of our prototype:

1. Languages used to write the web app: PHP, HTML, and CSS
2. Database language: MySQLi
3. Database integration: Apache, phpMyAdmin
4. Tested locally on chrome browser using xampp
5. Irreversible algorithm: sha1 and sha1_file functions of PHP

4.2 User Interface

Our prototype's user interface has 4 components:

1. New user component
2. File upload component
3. Login component
4. File details component

4.2.1 New User Component

- It registers a user with the details provided by him/her.
- Currently we are not using any cross checking techniques.

4.2.2 File Upload Component

- There are two types in this.
- One type is used for file verification and another type is used for new file uploading after log in.
- In this part our project's main component of digital signatures is implemented.
- Using the irreversible hashing algorithm (file_sha1 function in our prototype), we get the file hash which will be around 40 bytes and is stored on our database.
- The hashing function is running on the user's browser itself, so only the hash is transferred across the network.
- Even if total uploaded files reach 1billion, the total size will be around 40 GB which is small in terms of server’s cloud storage.
- Hence in this method file details are secure and signing is storage efficient, authentic and verifiable.
- Also in this method original file is unaltered and user’s need not worry about keys and hardware tokens.

4.2.3 Login Component

- It can be used by registered users to log in and upload new files.
- We are using salted SHA1 for secured logging.

4.2.4 File Details Component

- This component displays the details of the file that is being verified.
- If the uploaded file doesn't exist it returns with no details, else with author details.

4.3 Usage Details

- First a user can register his details on new user page.
- Once registered, a user can log in using his credentials.
- Once logged in, a user can upload a file for signing it.
- Once digital signing is done, any user can upload the same file to check author details and hence know if the file has been modified.

4.4 Details of default functions used

4.4.1 sha1() function definition and usage

- It is a default function provided in PHP.
- The sha1() function calculates the SHA-1 hash of a string.
- The sha1() function uses the US Secure Hash Algorithm 1.

From RFC 3174 - The US Secure Hash Algorithm 1: "SHA-1 produces a 160-bit output called a message digest. The message digest can then, for example, be input to a signature algorithm which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature."

4.4.2 sha1_file() function definition and usage

- It is a default function provided in PHP.
- The sha1_file() function calculates the SHA-1 hash of a file.
- The sha1_file() function uses the US Secure Hash Algorithm 1.

From RFC 3174 - The US Secure Hash Algorithm 1: "SHA-1 produces a 160-bit output called a message digest. The message digest can then, for example, be input to a signature algorithm which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature."

This function returns the calculated SHA-1 hash on success, or FALSE on failure.

Chapter 5 Conclusion and Future Scope

5.1 Conclusion

By implementing the method as prototype we have reached to the conclusion that the method has successfully fulfilled the requirements of the challenge. The base of the method of using online environment and storing the irreversible function's output of a file/message for comparing without taking the entire file/message. The simplicity of method and its implementation are the highlights of this project. Hence, the method of Online Digital Signature is the solution to the project challenge.

5.2 Potential and Scope for the Future Work

- Implementation of proper mobile and email authentication for registration and signing.
- Better hashing algorithm in the place of irreversible function that creates unique, never coinciding file hash.
- Better security protocols for the online signing and verification.
- More developed scheme to find percentage of similarities with previously signed files and hold restriction and so on.

Prototype Link

Below Attached is the Google Drive Link to our prototype

https://drive.google.com/open?id=17f_Pni6ky-aKtQlx55olP7Z8SrcGyozU

We used XAMPP Control Panel to run our Prototype

NOTE:

All the programs are executed in machine configuration

- 5th Gen Intel® CoreTM i7-5500U Processor 4M Cache, up to 3.00 GHz
- 8 GB DDR3 Memory

References

[1]. https://en.wikipedia.org/wiki/Digital_signature

[2]. https://www.w3schools.com/

[3]. https://www.cryptomathic.com/news-events/blog/major-standards-and-compliance-of-digital-signatures-a-world-wide-consideration