NETWORK SECURITY ON DIGITAL SIGNATURES

Purushoth Kumar.T[1] Dhanasekaran.D[2]
[1]Student [2]Professor
[1,2]Saveetha School Of Engineering, India.

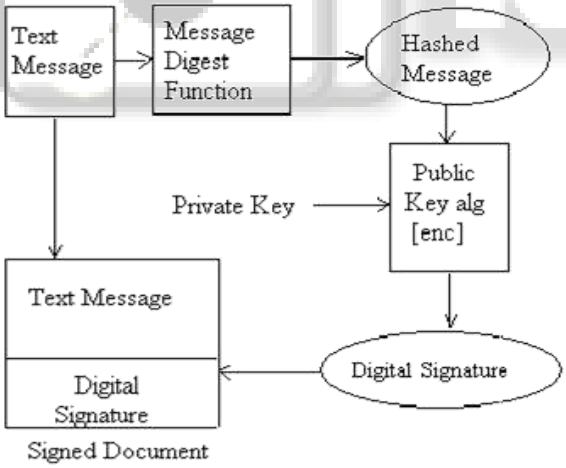
Abstract— Our objective is to provide the tools for certifying the origin of the file where it come from. When transferring important documents it is often necessary to certify in a reliable way who is actually the sender of a given document. One approach for certifying the origin of documents and files is by using the digital signature. The digital signing is a mechanism for certifying the origin and the integrity of transmitted information. In the process of digitally signing, additional information called digital signature is added to the given document, calculated using the contents of the document and some private key. At a later stage, this information can be used to check the origin of the signed document

***Key words*:** Digital signature.Private Key,Cryptography.

I. INTRODUCTION

Digital Signatures are messages that identify and authenticate a particular person as the source of the message and indicate such person's approval of the information contained in the message. They help users to achieve basic security building blocks such as authentication and integrity

The idea behind digital signature is the same as the handwritten signature. A digital signature doesn't involve signing something with a pen and paper then sending to the receiver. But like a paper signature, it attaches the identity of the signer to a transaction.



The digital signing is a mechanism for certifying the origin and the integrity of electronically transmitted information. In the process of digitally signing, additional information—called a digital signature—is added to the given document, calculated using the contents of the document and some private key. At a later stage, this information can be used to check the origin of the signed document.

The digital signature is a number (sequence of bits), calculated mathematically when signing a given document (message). This number depends on the contents of the message, the algorithm used for signing, and the private key used to perform the signing. The digital signature allows the recipient to check the actual origin of the information and its integrity. Digital signatures can be generated by using technique public key cryptography combined with a one way hash function. This requires public-private key pair.

The public key is a number (sequence of bits), which is usually bound to a person. A public key can be used to check digital signatures, created with the corresponding private key, as well as for encrypting documents that can then be decrypted only by the owner of the corresponding private key. The public keys are not secret to anybody and are usually publicly available.

The private key is a number (sequence of bits), known only to its owner. With his or her private key, a person can sign documents and decrypt documents that are encrypted with the corresponding public key. To a certain extent, the private keys resemble the well-known access passwords, which are a widespread authentication method over the Internet. The similarity is that with the private key, as well with the password, a person can prove his or her identity, i.e. to authenticate himself or herself.

The public and private keys are a mathematically bound cryptographic key pair (public/private key pair). To each public key corresponds exactly one private key and vice versa; to each private key corresponds exactly one public key. To use public key cryptography, one must have a public key and its corresponding private key.

II. NEED FOR DIGITAL SIGNATURES

Unlike paper documents, digital documents such as Word documents and e-mail messages have little to vouch for their authenticity and integrity. For example, e-mails do not have the identifying components that are found in a letter, such as letterhead, address of the recipient, date and traditional hand-written signature. It is also easy to forge message headers so that they appear to come from someone else.

In this environment there are circumstances when it is important for the recipient to be confident that the message has come from the person it seems to have come from and also that it has not been altered in transit. With the growing use of attachments to distribute information, it has become more important to be able to identify the sender of an e-mail message. The risk of viruses from attachments means that people will only want to open attachments from a trusted source.

III. DIGITAL SIGNATURE GENERATION

The following procedure is involved in the digital signature generation, If a sender wants to send receiver a text message, with digital signature, first the sender creates the text message to be signed and generates a hashed message using the message digest function. Message digest function is a mathematical function that generates a 160-bit hash code.

The hash code has the property that we can generate the hash code from original message but we can not generate original message from the hash code. Once sender has the hashed message he uses the public key cryptographic algorithm and his private key to sign the hash to generate the digital signature .The signed document is sent to the receiver.

IV. DIGITAL SIGNATURE VERIFICATION

Once the receiver receives the digital signature and the corresponding text message he need to calculate two values. First the hashed message of the received text is calculated using the same hashing algorithm. Then once he has the hash value he uses the decryption algorithm with sender's public key and the digital signature to retrieve the signed hash.

If he can decrypt digital signature which implies that sender's private key was used to encrypt the hashed message. The final step is to compare the hash he calculated with the hash he retrieved from the encryption process.

The key generation and distribution is done by the trusted central authority called Certification Authority (CA).CA accepts the certificate applications from entities, authenticates application issues certificates to users and devices and maintains and provides status information about the certificates.

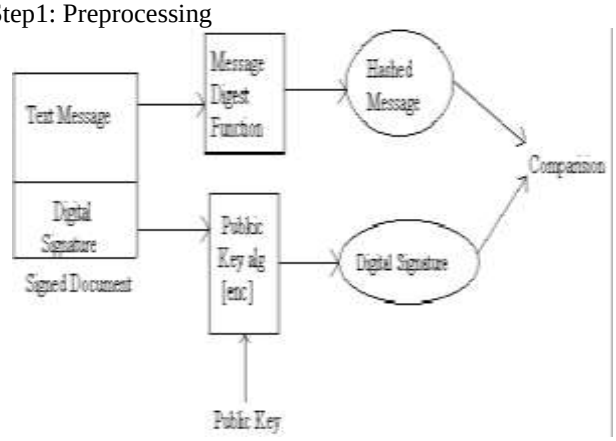
V. ALGORITHMS USED

A. SHA-1

The Secure Hash Algorithm was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standards(FIPS 180) in 1993 a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1.

The maximum length of a message *(in bits)* that can be hashed using the SHA-1 algorithm is one less than 2 to the 64th power. Implementation of the SHA-1 algorithm consists of six major steps and a variety of minor steps.

Step1: Preprocessing



The first major step is to prepare the message for hashing. There are three minor steps included in this preprocessing step. The first minor preprocessing step is to pad the length of the message so as to guarantee that the final length is a multiple of 512 bits or 64 bytes. The second minor preprocessing step is to set the initial hash value to a standard 160-bit value. The third minor preprocessing step is to parse the padded message into blocks of 512 bits or 64 bytes each.

1) Each block is processed separately

Each block is processed separately and the hash value is updated during the processing of each block. The initialized hash value from above is the input hash value for processing the first block. The updated hash value produced by processing each block serves as the initial hash value for the processing of the next block.

2) Padding the message

All messages are padded regardless of the original length of the message. The number of bits in the pad must be such as to cause the final length to be a multiple of 512 bits. The first bit in the pad must have a value of 1. The last 64 bits in the pad must contain a binary representation of the length of the original message. All other bits in the pad must have a value of 0.

3) Processing the message blocks:

Each message block, consisting of 64 bytes, is processed in sequence. The hash value output from processing one message block forms the initial hash value for processing the next message block. The hash value output from processing the final message block is the message digest for the message.

4) The five remaining steps

The five remaining major steps take place and are repeated during the processing of each message block.

Step 2: Initialize the message schedule

Initialize the message schedule **W** by using the incoming 64 bytes that constitute a message block to populate the first sixteen 32-bit elements of the 80-element message schedule.

Step 3: Populate the remainder of the message Schedule

Populate the remaining 64 elements of the message schedule **W** by propagating the values in the first sixteen elements upward into the remaining 64 elements.

Step 4: Initialize the working variables

Set the initial values of the variables **A** through **E** to the five 32-bit segments of the incoming 160-bit hash value.

Step 5: Process the message schedule

Individually process each of the elements in the 80-element message schedule. A different process is applied to the elements in each group of 20 elements. The processing of each element results in an updated set of values for the working variables **A** through **E**.

Step 6: Update the hash value

When all 80 elements in the message schedule have been processed, use the values in the working variables **A** through **E** to update the five 32-bit segments of the hash value. This updated hash value is used as the input hash value for processing the next message block. If all message blocks have been processed, the updated hash value is the message digest for the message that has been processed.

B. RSA

The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.

C. Key Generation Algorithm

(1) Generate two large random primes, p and q, of approximately equal size such that their product $n = pq$ is of the required bit length, e.g. 1024 bits. [See note 1].

(2) Compute n = pq and (φ) phi = (p-1) (q-1).
(3) Choose an integer e, 1 < e < phi, such that gcd (e, phi) = 1. [See note 2].
(4) Compute the secret exponent d, 1 < d < phi, such that
ed ≡ 1 (mod phi). [See note 3].
(5) The public key is (n, e) and the private key is (n, d). The values of p, q, and phi should also be kept secret.

- n is known as the **modulus.** `
- e is known as the **public exponent *or* encryption exponent**.
- d is known as the **secret exponent *or* decryption exponent.**

D. Encryption

Sender A does the following:-

(1) Obtains the recipient B's public key (n, e).
(2) Represents the plaintext message as a positive integer m [see note 4].
(3) Computes the cipher text $c = m^{\wedge e} \bmod n$.
(4) Sends the cipher text c to B.

E. Decryption

Recipient B does the following:-

(1) Uses his private key (n, d) to compute $m = c^{\wedge d} \bmod n$.
(2) Extracts the plaintext from the integer representative m.

VI. Conclusion

With the advancement in technology, it is expected to get even more reliable media for communication. Our system will provide one extra layer of authentication for communication.

One application is used for certifying the origin of documents and files is by using the digital signature. It is important for the developer to give conclusions or state the results of his work. This application will transfer the message with authentication provided with in it. It will fulfill all the needs for a reliable transfer by using efficient digital signature algorithms.

References

[1] www.crypto.com
[2] www.cryptography.com
[3] www.infosyssec.net
[4] www.uow.edu.au
[5] www.amazon.com
[6] www.phptr.com
[7] www.csrc.nist.gov

