Application of Digital Signature for Securing Communication Using RSA Scheme based on MD5

Stephen Fashoto
University of Swaziland Kwaluseni Swaziland

Application of Digital Signature for Securing Communication Using RSA Scheme based on MD5

Fashoto S.G, Gbadeyan J.A and Okeyinka E.A
Redeemer’s University, Redemption camp Mowe, Ogun State
gbengafash@yahoo.com

Abstract. Digital signature authentication scheme provides secure communication between two users. Digital signatures guarantee end-to-end message integrity and authentication information about the origin of a message. The focus of this paper is to discuss how to protect communications that occur in a transaction so as to guide against fraudsters and to make hash function collision free. MD4 is a predecessor of the MD5. At present MD4 is useless because collisions can be generated in a few seconds on a common Personal Computer (PC). The implementation based on python programming saves time and space compared to programming in C, C++ and others.

Keywords: Encryption, Decryption, Conventional signature, Digital signature, Hash functions and RSA

1 Introduction

Digital signature authentication schemes provide secure communication with minimum computational cost for real time applications such as electronic commerce, electronic voting etc. The sender generates the signature of a given message using his secret key; the receiver then verifies the signature by using sender’s public key [1].

Many organizations prefer going paperless by using electronic forms of sending and receiving data. In this context, it is essential that not only the sender needs to authenticate the receiver, the receiver should also authenticate the sender and ascertain himself from whom the message was received [2].

Everyday, people sign their names to letters, credit card receipts, and other documents, demonstrating they are in agreement with the contents. That is, they authenticate that they are in fact the sender or originator of the item. This allows others to verify that a particular message did indeed originate from the signer. However, this is not fool proof, since people can ‘lift’ signatures off one document and place them on another, thereby creating fraudulent documents. Written signatures are also vulnerable for forgery because it is possible to reproduce a signature on other documents as well as to alter documents after they have been signed.

A digital signature or digital signature scheme is a mathematical scheme for demonstrating the authenticity of digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions and in other cases where it is important to detect forgery and tampering.

Digital signatures and handwritten signatures both rely on the fact that it is very hard to find two people with the same signature. People use public-key cryptography to compute digital signatures by associating something unique with each person. When public – key cryptography is used to encrypt a message, the sender encrypts the message with a public key of the intended recipient. When public-key cryptography is used to calculate a digital signature, the sender encrypts the “digital finger print” of the document with his or her own private key. Anyone with access to the public key of the signer may verify the signature.

Digital signatures are often used to implement electronic signatures, a border term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signature. Encryption is the process of converting plaintext to cipher text. Its purpose is to ensure privacy by keeping information hidden form anyone for whom it is not intended, even those option is the have access to the encrypted data. Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form.

Encryption and decryption generally require the use of some secret information, Referred to as a key. For some encryption mechanisms such symmetric encryption, the same key is used for both encryption and decryption; for other mechanisms such as asymmetric encryption, the keys for encryption and decryption are different [3].

Authentication is used almost on daily basis. For instance when the need arises to sign ones name on documents and even as in situations where decisions and agreements are communicated electronically.

The field of cryptography is growing increasingly diverse cryptography is fundamentally based on problems that are difficult to solve. A problem may be difficult because its solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital document.

Authentication is any process through which one proves and verifies certain information. Sometimes one may want to verify the origin of a document, the identity of the sender, the time and date a document was sent and /or signed, the identity of a computer or user, and so on.

A digital signature is a cryptographic means through which many of these may be verified. The digital signature of a document is a piece of information based on both the document and the signer's private key. It is typically created through the use of a hash function and a private signing function (encryption with the signer's private key).

1.1 Comparison of Conventional Signature and Digital Signature

According to [4] conventional signature and digital signature can be compared based on the following:

a) Inclusion

A conventional signature is included in the document; it is part of the document. When we write a cheque, the signature is on the cheque; it is not a separate document. But when we sign a document digitally, we send the signature as a separate document. The sender sends two documents: the message and the signature. The recipient receives both documents and verifies that the signature belongs to the supposed sender. If this is proven, the message is kept; otherwise rejected.

b) Verification method

The second difference between the two types of signatures is the method of verifying the signature. For a conventional signature, when the recipient receives a document, she compares the signature on the document with the signature on the file. If they are the same, the document is authentic. The recipient needs to have a copy of this signature on file for comparison. For a digital signature, the recipient receives the message and the signature.

A copy of the signature is not stored anywhere. The recipient needs to apply a verification technique to the combination of the message and the signature to verify the authenticity.

c) Relationship

For a conventional signature, there is normally a one-to-many relationship between a signature and documents. A person uses the same signature to sign many documents. For a digital signature, there is a one-to-one relationship between a signature and a message. Each message has its own signature. The signature of one message cannot be used in another message. If Bob receives two messages, one after another, from Alice, he cannot use the signature of the first message to verify the second. Each message needs a new signature.

d) Duplicity

Another difference between the two types of signatures is a quality called *duplicity.* In conventional signature, a copy of the signed document can be distinguished from the original one on file. In digital signature, there is no such distinction unless there is a factor of time (such as a timestamp) on the document. For example, suppose Alice sends a document instructing Bob to pay eve intercepts the document and the signature, she can replay it later to get money again from Bob.

1.2 Statement of the Problem

Cryptography is about communication in the presence of an adversary. However, the most ancient and basic problem of cryptography is secure communication over an insecure channel.

Suppose Alice wants to send a signed document or message to Bob. The first step is generally to apply a hash function to the message, creating what is called a message digest. The message digest is usually considerably shorter this saves than the original message. In fact, the job of the hash function is to take a message of arbitrary length and shrink it down to a fixed length. To create a digital signature, one usually signs (encrypts) the message digest as opposed to the message itself. This saves a considerable amount of time, though it does create a slight insecurity (the example is illustrated below).

Alice sends Bob the encrypted message digest and the message, which she may or may not encrypt. In order for Bob to authenticate the signature he must apply the same hash function as Alice to the message she sent him, decrypt the encrypted message digest using Alice’s public key and compare the two. If the two are the same he has successfully authenticated the signature. If the two do not match there are few possible explanations. Either someone is trying to impersonate Alice, the message itself has been altered since Alice signed it or an error occurred during transmission.

There is a potential problem with this type of digital signature. Alice not only signed the message she intended to but also signed all other messages that happen to hash to the same message digest. When two messages hash to the same message digest it is called a collision; the collision-free properties of hash functions are a necessary security requirement for most digital signature schemes. A hash function is secure if it is very time consuming, if at all possible, to figure out the original message given its digest. However there is an attack called the *birthday attack* that relies on the fact that it is easier to find two messages that hash to the same value than to find a message that hashes to a particular value. Its name arises from the fact that for a group of 23 or more people the probability that two or more people share the same birthday is better than 50%.

2 Literature Review

RSA is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. It is named for the three MIT mathematicians who developed it — Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, *n*, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an *n* with roughly twice as many digits as the prime factors. The public key information includes *n* and a derivative of one of the factors of *n*; an attacker cannot determine the prime factors of *n* (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure. Regardless, one presumed protection of RSA is that users can easily increase the key size to always stay ahead of the computer processing curve. As an aside, the patent for RSA expired in September 2000 which does not appear to have affected RSA's popularity one way or the other [5].

RSA is very widely used today for secure Internet communication (browsers, S/MIME, SSL, S/WAN, PGP, and Microsoft Outlook), operating systems (Sun, Microsoft, Apple, Novell) and hardware (cell phones, ATM machines, wireless Ethernet cards, Mondex smart cards, Palm Pilots).

a) Diffie-Hellman

After the RSA algorithm was published, Diffie and Hellman came up with their own algorithm. D-H is used for secret-key key exchange only, and not for authentication or digital signatures.

b) Digital Signature Algorithm (DSA)

The algorithm specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for the authentication of messages

c) ElGamal

El Gamal is a public key algorithm that can be used for digital signatures and key exchange. It is not based on the difficulty of factoring large numbers, but is based on calculating discrete logarithms in a finite field [6].

d) Elliptic Curve Cryptography (ECC)

A Public Key Cryptography(PKC) algorithm is based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited computation power and/or memory, such as smartcards and PDAs.

e) Cramer-Shoup

A public-key cryptosystem proposed by R. Cramer and V. Shoup of IBM in 1998.

f) Key Exchange Algorithm (KEA)

A variation on Diffie-Hellman; proposed as the key exchange method for Capstone

2.1. Hash Functions

A ***one-way hash*** is a function (usually mathematical) that takes a variable-length string, a message, and compresses and transforms it into a fixed-length value referred to as a hash value. A hash value is also called a ***message digest***. Just as fingerprints can be used to identify individuals, hash values can be used to identify a specific message. If Kevin wants to send a message to Maureen and he wants to ensure that the message does not get altered in an unauthorized fashion while it is being transmitted, he would calculate a hash value for the message and append it to the message itself. When Maureen receives the message, she performs the same hashing function Kevin used and compares her result with the hash value that was sent with the message.

If the two values are the same, Maureen can be sure that the message was not altered during transmission. If the two values are different, Maureen knows that the message was altered, either intentionally or unintentionally, and she discards the message.

The hashing function, usually an algorithm, is not a secret—it is publicly known. The secrecy of the one-way hashing function is its "one-wayness." The function is only run in one direction, not the other direction. This is different than the one-way function used in public key cryptography. In public key cryptography, the security is provided because it is very hard, without knowing the key, to perform the one-way function backwards on a message and come up with readable plaintext. However, one-way hash functions are never used in reverse; they create a hash value and call it a day. The receiver does not attempt to reverse the process at the other end, but instead runs the same hashing function one way and compares the two results.

The hashing one-way function takes place without the use of any keys. This means that anyone who receives the message can run the hash value and verify the message's integrity. However, if a sender only wants a specific person to be able to view the hash value sent with the message, the value would be encrypted with the key. This is referred to as the *message authentication code* [6].

Hash algorithms that are in common use today include:

a) Message Digest (MD) algorithms

MD is a series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.

- **MD2 (RFC 1319)**

This is designed for systems with limited memory, such as smart cards.

- **MD4 (RFC 1320)**

Developed by Rivest, similar to MD2 but designed specifically for fast processing in software.

- **MD5 (RFC 1321)**

Also developed by Rivest after potential weaknesses were reported in MD4; this scheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996 [6].

2.2 Algorithm of MD5

Step 1 : start
Step 2: Get a long string of any length as input
Step 3: The processes produce a fixed length string as output
Step 4: stop

2.3 Design of the Modified RSA Scheme

This segment is used to describe the methods employed in this paper based on hash function algorithm. The message digest value could be generated using any of the Message Digest algorithm on hashing. The message digest algorithm that will be considered is the one that can overcome collision problem and that is MD5. MD4 is a predecessor of the MD5. At present MD4 is useless because collisions can be generated in a few seconds on a common Personal Computer (PC).

The digital signature scheme for a message is generated using three steps

a) A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
b) A signing algorithm which, given a message and a private key, produces a signature.
c) A signature verifying algorithm which given a message, public key and a signature, either accepts or rejects the messages claim to authenticity.

2.4 Implementation

In the implementation we have used MD5 algorithm to generate the message digest value using python software version 2.6. The message digest value needs to be expressed as 32bit hexadecimal number.

HASH FUNCTION CODE USING PYTHON SOFTWARE

```
Import  md5

Message1 = ‘ this is an important message’

Obj1 = md5.new(message1)

Digest =obj1.digest()

Message2 =’this is a tampered message’

Obj2 = md5.new(message2)

Digest2=obj2.digest()

If  digest2 == digest1:

        Print “message intact”

Else:

        Print “message not intact”
```

3 Conclusion

In this paper we have been able to prove and verify how we can ensure that hash function algorithm is collision free based on digital signature using RSA scheme. With the implementation based on python programming, time and space is saved.

RSA uses public and private keys and so files encrypted with the Public key can only be decrypted by the holder of the Private Key. Also, the difficulty in factoring out these keys makes RSA secured.

Apart from the security advantage, this system has been proved to maintain the integrity of the information secured since the length of the message is preserved during any of the operations of encryption and decryption. The system has therefore eliminated the fear of losing vital information to an eavesdropper or an enemy at any point in time.

Cryptography is important for more than just privacy, however. Cryptography protects the world’s banking systems as well. Many banks and other financial institutions conduct their business over open networks, such as the Internet. Without the ability to protect bank transactions and communications, criminals could interfere with the transactions and steal money without a trace [7].

References

1. Ramasamy, R. R. and Prabakar, M.A. (2009): Digital signature scheme with message recovery using knapsack-based ECC, International Journal of Network Security, Vol. 12 No. 1, pp. 15-20.
2. Roja, P. P. and Avadhani, P.S. (2007): Digital signature development using truncated polynomials, International Journal of Computer Science and Network Security vol. 7 No. 7.
3. http://www.rsa.com
4. Forouzan, B. A. Cryptography and network security Published by McGraw-Hill (2008)
5. Kessler, G.C. (1998): An Overview of Cryptography, http://www.garykessler.net/library/crypto.html
6. Harrisx retrieved from www.cccure.org/documents/cryptography/cisspallinone.pdf (2001)
7. Encarta, M. (2007): Cryptography 1993-2006 Microsoft corporation.

Participants in the SEIS 2010 Doctoral Symposium

1. **Reusing Software Component in a Revolving World -** AJAYI, Olusola Olajide
Department of Computer Science, Adekunle Ajasin University
Akungba-Akoko, Ondo State, Nigeria

2. **DEVS-Based Simulation of Vibration Signature Detection System for Structural Health Monitoring of Remote Engineering Facilities -** Imouokhome, F. Aien-Akho. Uangbaoje.
Department of Computer Science, University of Benin, Benin City. Nigeria.

3. **Capturing Organisational Policies in Granting Access to Database Resources Using Fuzzy Logic -** Nicholas Oluwole Ogini
Department Of Computer Science, Faculty of Science, Delta State University, Abraka, Nigeria.

4. **Framework For A Service-Oriented Mobile E-Health Infrastructure For Rural/Suburban Healthcare** - OLADOSU, John Babalola
Department of Computer Science and Engineering
Ladoke Akintola University of Technology, Ogbomoso, Nigeria

5. **E-Government Evaluation: A Citizen Centric Approach -** Chete F. O.
Department of Computer Science, University of Benin, Benin-City

6. **Computer Immunity Using Intrusion Detection System -** Konyeha Susan
Department of Computer Science, University of Benin, Benin City