Secure Message Transferring System Using Arbitrated Digital Signature and RSA Algorithm

May Thu Myint; Renu
Computer University, Monywa
mthumyint@gmail.com

Abstract

Today, security is an important thing that needs to transport data from location to another safely. Therefore, many models and systems are looking for an ideal method to provide a secure environment for better of the secure communication. Cryptography accepts the challenge and plays the main role of the modern secure communication. The purpose of this system is to develop RSA algorithm with arbitrated digital signature for message security. In this system, Arbitrated Digital Signature handles several forms of dispute between two parties .And the message authentication protects between two parties who exchange message from Arbiter. So, both two parties cannot directly check whose signature. And then, this purposed algorithm was developed based on the asymmetric algorithm whereas the length of the key and digital signature is considered.

Keywords*: Security, Cryptography, Digital Signature, Arbiter, RSA "Rivest Shamir and Adleman".*

1. Introduction

Digital Signature are rapidly becoming ubiquitous in many aspects of electronic life. They are used to obtain security services such as authentication, data integrity and non-repudiation. Digital Signature is one of those developments which combine the legal and computer security discipline [1]. Cryptography provides the basics for authentication of messages as well as their security and integrity; carefully designed security protocols are required to exploit it. There are different types of encryption algorithms used to protect sensitive data including; symmetric, asymmetric encryption techniques. In this age of universal electronic connectivity of viruses and hackers, of electronic eaves-dropping and electronic fraud, there is indeed no time at which security does not matter.

Two trends have come together to make the topic of vital interest. First, the explosive growth in computer systems and their interconnections via network has increased the dependence of both organizations and individuals on the information stored and communicated using there systems. This, in turn, has led to a heightened awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks [2].

Second, the disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security [2].

In this paper, we proposed a message transferring, by using Arbitrated Digital Signature and RSA that is used for encryption and decryption methods. The remainder of this paper is organized as follows. In section 2, we describe Public key cryptography. In section 3, we present Authentication of Public key cryptography. In section 4 we implement the purposed of digital signature and RSA algorithm. Section 5 expresses Overview of the purposed system and finally, in section 6 concludes our purposed system.

2. Related work

Mihir Bellare and Phillip Rogaway proposed new scheme, both for signing and for signing with message recovery. They are as simple the efficient as the standardize ones. But, assuming the underlying hash function is ideal, their method were not provably secure in a strong sense: the security of their schemes can be tightly related to the security of the RSA function [1].

A.I.Khang and A.R.Ranli advised the problem of time execution and the authentication between sender and receiver are considered in their study. New algorithm was developed based on the combination of symmetric and asymmetric algorithm for Email encryption. It is a powerful tool in protecting the e-mail privacy [3].

S.Kadry and A.B.Bar proposed a new design to solve the authenticity problem in mobile communication. Their experimental result support shows that the proposed design was of a better

performance and security than public/private keys technique which used RSA algorithm. They were working on implementing their design while studying other security problems like privacy and confidentiality. Moreover, the extraction or composure of a key from the minutiae of Fingerprint is going to be studied as well. This technique works on the Fingerprint basis [4].

3. Public key cryptography

Public key cryptography is sometimes also referred to as asymmetric cryptography. Unlike secret key cryptography, keys are not shared. This cryptography has two keys: a private key that need not be revealed to anyone, and a public key that is preferably known to the entire world. The private key is called a private key and not a secret key. This convention is an attempt to make it clear in any context whether public key cryptography or secret key cryptography is being used. There are people in this world whose sole purpose in life is to try to confuse people. They will use the term secret key for the private key in public key cryptography, or use the term private key for the secret key in secret key technology. One of the most important contributions is to convince people to feel strongly about using the terminology correctly—the term secret key refers only to the single secret number used in secret key cryptography [5].

Public key cryptography can do anything secret key cryptography can do, but the known public key cryptographic algorithms are orders of magnitude slower than the best known secret key cryptographic algorithms and so are usually only used for things secret key cryptography can't do. Public key cryptography is very useful because network security based on public key technology tends to be more secure and more easily configurable. It is mixed with secret key technology [5]. For example, public key cryptography might be used in the beginning of communication for authentication and to establish a temporary shared secret key, then the secret key is used to encrypt the remainder of the conversation using secret key technology.

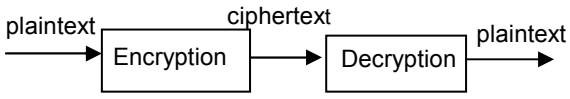

Figure 1. Process of encryption and decryption

4. Authentication of public key cryptography

Authentication is an area in which public key technology potentially gives a real benefit. With secret key cryptography, if Alice and Bob want to communicate, they have to share a secret. If Bob wants to be able to prove his identity to lots of entities, then with secret key technology he will need to remember lots of secret keys, one for each entity to which he would like to prove his identity. Possibly he could use the same shared secret with Alice as with Carol. Public key technology is much more convenient. Bob only needs to remember a single secret, his own private key. Alice chooses a random number r, encrypts it using Bob's public key eB, and sends the result to Bob. Bob proves he knows dB by decrypting the message and sending r back to Alice.

Another advantage of public key authentication is that Alice does not need to keep any secret information. With secret key based authentication, if Carol stole a backup tape and read the key that Alice shares with Bob, she could then trick Bob into thinking she was Alice. In contrast, with public key based authentication, the only information on Alice's backup tapes is public key information, and that cannot be used to impersonate Bob.

In large-scale systems, like computer networks with thousands of users and services, authentication is usually done with trusted intermediaries. Public key based authentication using intermediaries has several important advantages over secret key based authentication.

5. The proposed digital signature and RSA algorithm

In this section, the purposed digital signature and RSA algorithm is presented.

5.1. Digital signature

It is often useful to prove that a message was generated by a particular individual, especially if the individual is not necessarily around to be asked about authorship of the message. This is easy with public key technology. The signature for a message m can only be generated by someone with knowledge of private key. And the signature depends on the contents of m. If m is modified in any way, the signature no longer matches. So, digital signatures provide two important functions. They prove who generated the information, and they prove that the information has not been modified in any way by anyone since the message and matching signature were generated. An important example of a use of a signature is in electronic mail to verify that a mail message really did come from the claimed source. The digital signature ensures that the signatory is indeed the originator of the message. Any changes made to the document after it was signed invalidate

the signature, thereby protecting against forgery. There are three common reasons for applying a digital signature to communications:

- Authentication
- Integrity
- Non-repudiation

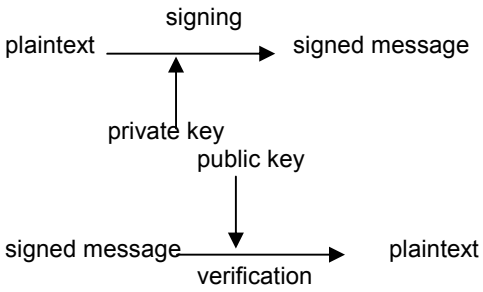

Figure 2. Digital signature using public key cryptography

5.2. Arbitrated digital signature techniques

The arbiter could from an alliance with the sender to deny a signed message, or with the receiver to forge the sender's signature.

All the problems can be resolved by public-key scheme. According to the following equation, X double encrypts a message M first with X's private key, KR_x , and then with Y's public key, KU_y . This is a signed, secret version of the message, this signed message, together with X's, identifier, is encrypted again with KR_x , and together with ID_x, is sent to A.

The inner, double-encrypted message is secure from the arbiter (and everyone else except Y). A can decrypt the outer encryption to assure that the message must have come from X (because only X has KR_x.) A checks to make sure that X's private and public key pair is still valid and verities the message. Then A transmit a message to Y, encrypted with KR_x, message includes ID_x, the double-encrypted message, and a timestamp.

(1) $X \rightarrow A: ID_X \| E_{KRx}[ID_X \| E_{KUy} (E_{KRx}[M])]$

(2) $A \rightarrow Y: E_{KRa} [ID_X \| E_{KUy} [E_{KRx}[M]] \| T]$

Equation 1. Pubic key encryption, arbiter does not see message

Notation: X=Sender M=Message
IDx=Identifier Y=Recipient
T=Timestamp E=Encryption
A=Arbiter

The presence of A solves the problem faced by directly signature schemes that X might disown the message. The arbiter plays a sensitive a crucial role in this sort of scheme. The preceding scenario also implies that A is able to read message from X to Y and, indeed, that any eavesdropper is able to do so. Public-key scheme can be used to avoid all these problems.

5.3. RSA algorithm

The scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n. The plaintext is encrypted in blocks, with each block having a binary value less than n. Encryption and Decryption are of the following form:

For Encryption, $C = M^e \bmod n$

For Decryption, $M = C^d \bmod n$

where C is the encrypted ciphertext, and M is the original plaintext. Both the sender and the receiver know the value n, with the value 'e' is known to the sender, but only the receiver knowing the value of 'd'. Thus the public key of this algorithm is {e, n} and the private key is {d, n}. The following paragraphs describe the selection of the public key and private key members, and the encryption and decryption process mathematically.

5.4. RSA key generation

Select p,q privately: where p and q are any two prime numbers:

Calculate $n = p*q$

Calculate $\Phi(n) = (p-1)(q-1)$

where $\Phi(n)$ is an Euler totient function which is the number of positive integers less than n, but prime to n [3].

Select an integer e such that the (greatest common divisor),

$\gcd(\Phi(n), e) = 1$ and $1 < e < \Phi(n)$

Calculate d privately, such that

$d = e^{-1} \bmod \Phi(n)$

Select the Public Key, KU = {e, n} and Private Key KR = {d, n}

The prime numbers p and q are responsible for the complexity of the algorithm. In practice, the key sizes are large enough to prevent brute-force attacks, but small enough for practical encryption and decryption [3].

5.5. RSA encryption

Once the key-pair of KU and KR has been determined encryption and decryption are simple mathematical computations. The plaintext M is chosen to be smaller than 'n', and the ciphertext C is computed using the public key {e,n}. The pseudo code below describes the encryption process [3].

Plaintext M<n
Ciphertext C = M^e(mod n)

5.5. RSA decryption

The private key KR of the key-pair is retained only with the owner. When a ciphertext C is received, the decryption process uses the private key {d,n} as shown in the pseudo code below to obtain the plaintext M [3].

Plaintext M = C^d (mod n)
Ciphertext C = M^e(mod n)

The following steps are the example for RSA key generation

Select prime: p = 17 and q =11
Compute n = pq = 17*11=187
Compute $\phi(n) = (p-1)(q-1) = 16*10 = 160$
Select e : gcd(e,160) =1; choose e = 7
Determine d: de = 1 mod 160 and d<160 Value is d = 23.
Public key KU ={7, 187 }
Private key KR = {23, 187}

And here is the example for RSA encryption and decryption steps;

Given message M = 88 (note: 88<187)
Encryption: C = 88^7 mod 187 =11
Decryption: M = 11^{23} mod 187 = 88

6. Proposed system architecture

The following Figure 3 shows the overview of the system, the sender encrypts a message M with its private key, and then receiver's public key. This is a signed, secret version of the message. This signed message, together with sender's identifier, is encrypted message that is secure from the arbiter.

The arbiter receives the message from the sender. The arbiter can decrypt the outer encryption to assure that the message have come from sender (because only sender has private key).Then the arbiter checks to make sure that the sender's identifier. If the sender's identifier is invalid, reject and go to listen state. If the sender's identifier is valid, identifier and timestamp is saved. And the arbiter appends with timestamp and encrypts with the message with the arbiter's private key. After this step, the arbiter with a message is send to receiver.

The receiver decrypts the outer encryption first with the arbiter's public key and checks the sender's identifier and timestamp. It decrypts the message first with its private key and then the sender's public key. Finally the receiver displays the message.

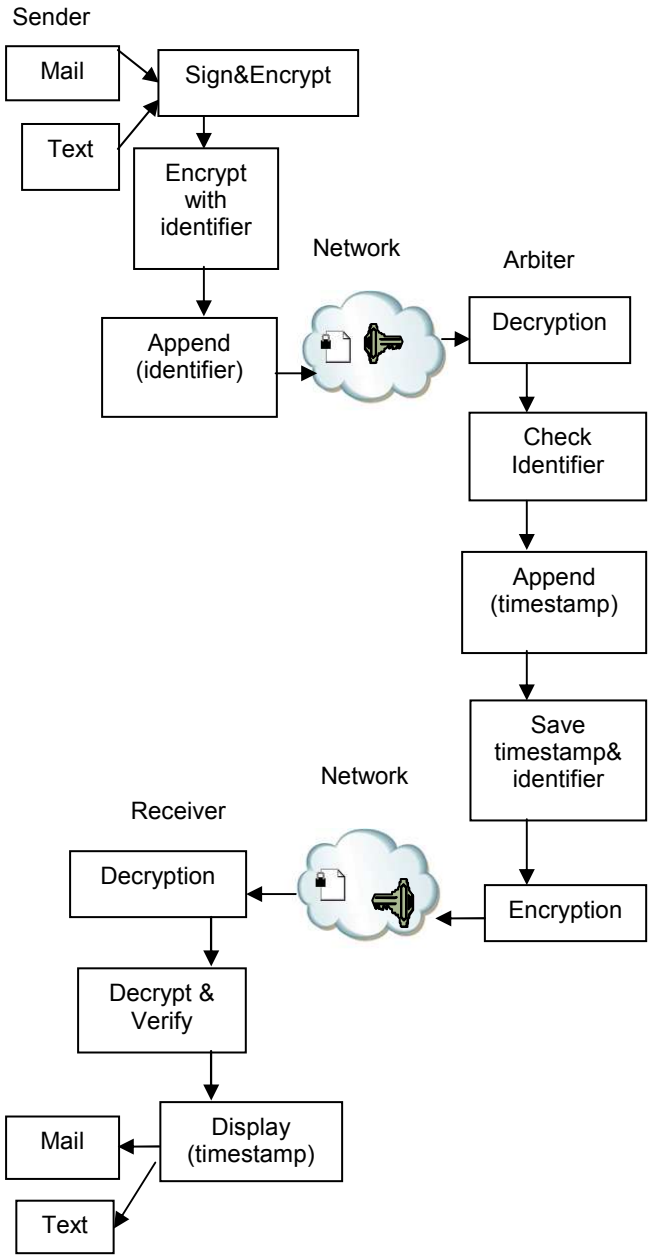

Figure 3. Overview of the proposed system

6.1. Process flow of the sender

In Figure 4, it describes the system design of the sender. In the first , Sender X listen and load the message M. Sender X double encrypts a message M first with X's Private Key KRx. It encrypts with Y's Public Key KUy. This signed message C, together with IDx of Sender X and then is encrypted again with KRx. This message C append with IDx of X and this message is a message C^1. The message C^1 is sent to Arbiter.

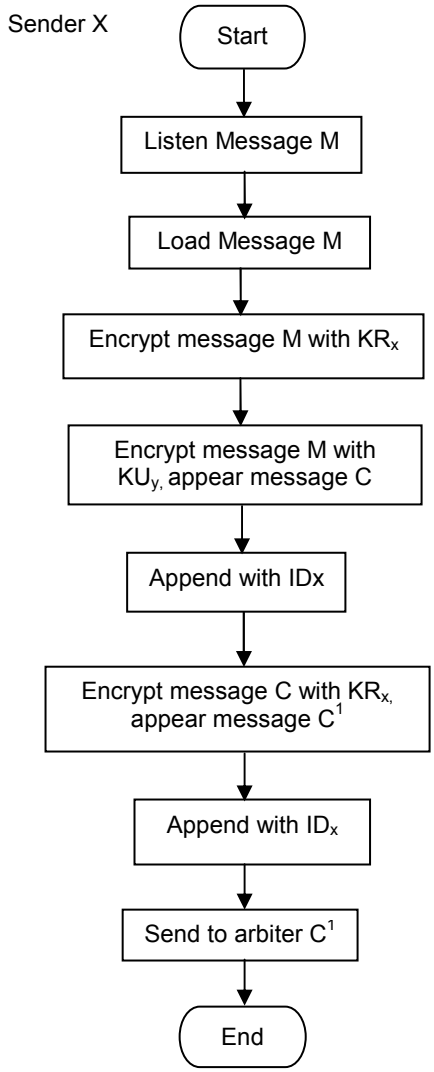

Figure 4. System design for the sender

6.2. Process flow of the arbiter

In the Figure 5 scheme, it expresses the system design for the Arbiter. The arbiter A listen the message C^1. The arbiter A can decrypt the received C^1 with the public key KUy. So, the arbiter A checks the message C^1 by X's identifier. If IDx is equal to X's identifier IDx, the message is saved with the timestamp T. Then, it appends with timestamp T and encrypts C^1 by Arbiter's Private Key KRa. This message is a message C^2. Finally, C^2 is send to the receiver. If IDx is not equally by X's IDx, this message is reject.

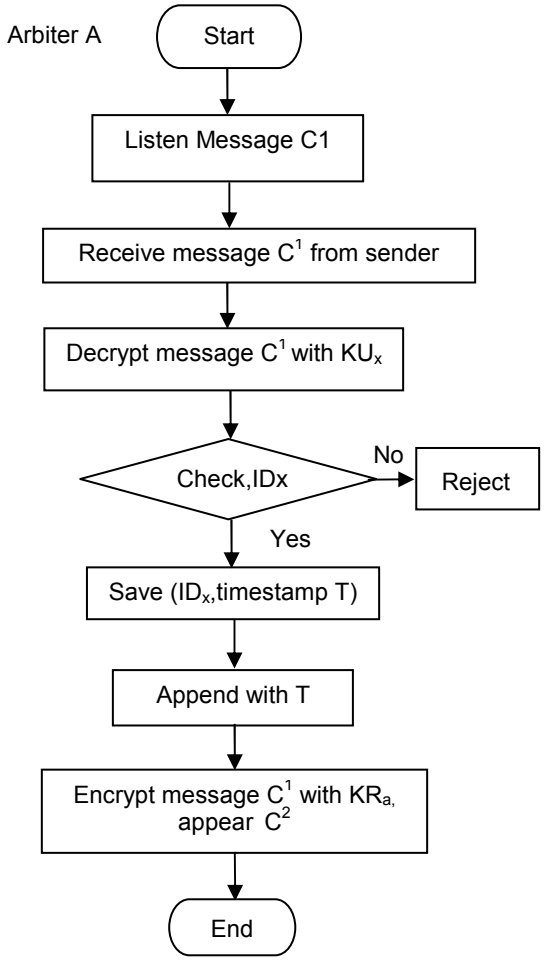

Figure 5. System design for the arbiter

6.3. Process flow of the receiver

Figure 6 show that the system design of the receiver. Initially, receiver Y is listened and received the message C2 from arbiter. This message C2 is decrypted by arbiter's Public Key KUa. This message is received by message C1.The message C1 is decrypted with Y's Private Key KRy, this message is transformed to message C. The message C is decrypted again with arbiter A's Public Key KUa and then this message C is transformed to message M. The transforming message M is displayed with timestampT.

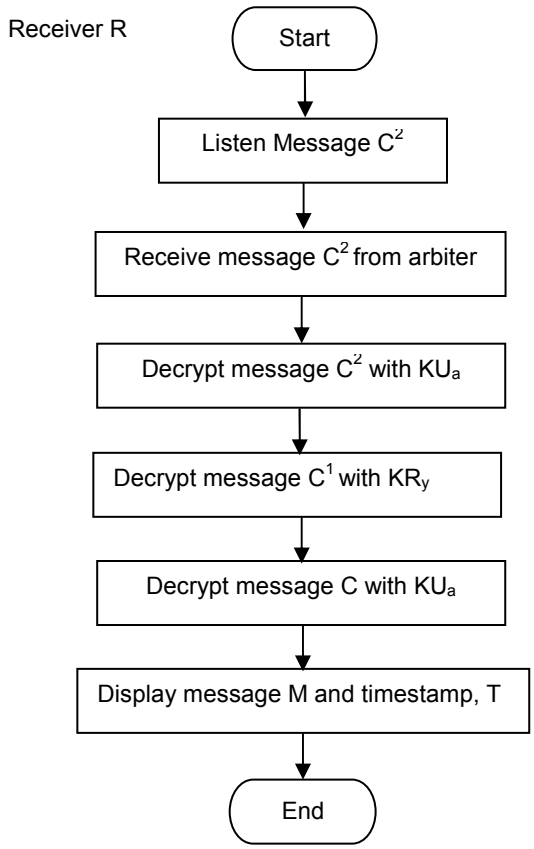

Figure 6. System design for the receiver

7. Conclusion

The dispute cases between the sender and receiver can be eliminated by using arbitrated digital signature. According to the RSA algorithms, key management of asymmetric algorithm is more efficient than symmetric algorithm. This system involves encryption of the message twice with public key algorithm (RSA). In summary, asymmetric (RSA) algorithm is used to develop the arbitrated digital signature cryptosystem. The proposed system prevents disputes between two parties. So, by using this system it can achieve the effective cryptosystem for communication of sensitive data. The content of the message from the sender to the receiver is secret from the arbiter and anyone else.

The system applies public key cryptography, therefore the user can achieve the authentication services and non-repudiation service.

10. References

[1] M.Bellare and P.Rogaway, "The Exact Security of Digital Signature_ How to Sign with RSA and Rabin", March 1996.

[2] S.Goldwasser, S.micali and R.Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosem_Message Attacks", Siam Journal of Computing, Apiul 1998.

[3] A.I.Khang and A.R.Ranli, "Implementation and Evaluation of New Cryptography Algorithm for E mail Applications", Department of Computer and Communication Systems Engineering Faculty of Engineering-University, Putra Malaysia, 2002.

[4] S.Kadry and A.B.Bar, " Design of Secure Mobile Communication Using Figureprint", American University of Science and Technology Department of Computer Science, European Journals of Scientific Research, Vol.30, 2009.

[5] C.K.Koc,C.D, "High-speed RSA Implementation", Technical Report TR201,RSA laboratories, November 1994.

[6] V.K.O.Paathangi and P.K.Tipparti, "High Performance Implementation of the RSA Algorithm", Department of Computer Science, St.Cloud State University, 2003.

[7] R.L.Rivest, A.shamir, and L.Adleman, "A Method for Obtaing Digital Signatures and Public_Key Cryptography", Communication of the ACM, VOL;21, 1978.

[8] B.Schneidler, Applied Cryptography, NewYork : Wiley, 1996.

[9] http://www.rsasecurity.com/rsalabs/node.asp

[10] http://www/acm.org/awards/turing_multimedia/t1_mm-2002.htm.

[11] http://www.stcloudstate.edu/jhealth/csci969/RSAAcLearn.htm

[12] http://en.wikipedia.org/wiki/Cryptography#Public-key cryptography.