ARBITRATED DIGITAL SIGNATURE FOR E-AUTHENTICATION TECHNIQUE OF A DIGITAL MESSAGE

G.Ranjith[1], B.Prathusha[2] and P.Sagarika[2]
[1]HOD & Assistant Professor, [2,3]Assistant Professor,
Department of Computer Science & Engineering,
Warangal Institute of Technology and Science, Warangal, Telangana, India.

ABSTRACT:
Internet became a part of human life and these days online communication gained more popularity than traditional communication. Digital Signature technology more and more shows its important position in information security. Eventually paper based documents are replaced by electronic documents. Digital Signature is a security mechanism used to check the authenticity and integrity of an electronic document. A Digital Signature is an authentication mechanism that enables the creator of a message to attach a code that acts as signature. However, their security is a critical issue. In this paper, I provide security using Authentication Protocols - Mutual authentication and one-way authentication. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key. Confidentiality can be provided by further encrypting the entire message plus signature with either the receiver's public key or shared private key. It is an electronic signature used to authenticate the identity of the sender and it assures that original content of the message or document received is unchanged or same. The problems associated with Direct Digital Signature can be addressed by using an Arbiter. At the end, conclusion and further extension of this work is discussed.

KEYWORDS: *Digital Signature, Mutual authentication, one-way authentication, Arbiter.*

I. INTRODUCTION

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. Digital signature plays an important role in online communication. In these days most of the electronic documents are identified by the digital signature only. Digital signature is a branch of cryptography. Cryptography is defined as secret writing. Cryptography mainly consists of symmetric key algorithms, asymmetric key algorithms and message digest algorithms. The asymmetric key and symmetric key differs from each other by number of keys. Single key is used in the symmetric key algorithm and two keys (public and private) are used in the asymmetric key algorithm. Message digest algorithm is used to generate message digest of a given input message. Message digest is also called hash code or finger print of the input message. For message transfer Hash function is used after that message is digested. For signature generation private key is used then signature generation is created. This signature generated is transferred for verification. After verified valid or invalid result is obtained.

There are three algorithms that are used for digital signature generation under the DSS standard.

1) DSA (Digital Signature Algorithm)
2) RSA (Rivest Shamir Ad leman)
3) ECDSA (Elliptic Curve Digital Signature Algorithm)

Digital signature scheme is designed using two algorithms, one is asymmetric key or public key cryptographic algorithm and the other is message digest algorithm. Symmetric key and asymmetric key cryptographic algorithms cannot provide any authentication mechanisms but they provide security to the information that may be either transmitted data or stored data. Broadly authentication protocols are categorized into two, Mutual authentication and One-Way authentication. The data authentication

can be achieved by digital signature schemes. Digital signatures are used in most of the security applications and protocols and they also play an important role in every online communication which may be either personal or organizational communication.

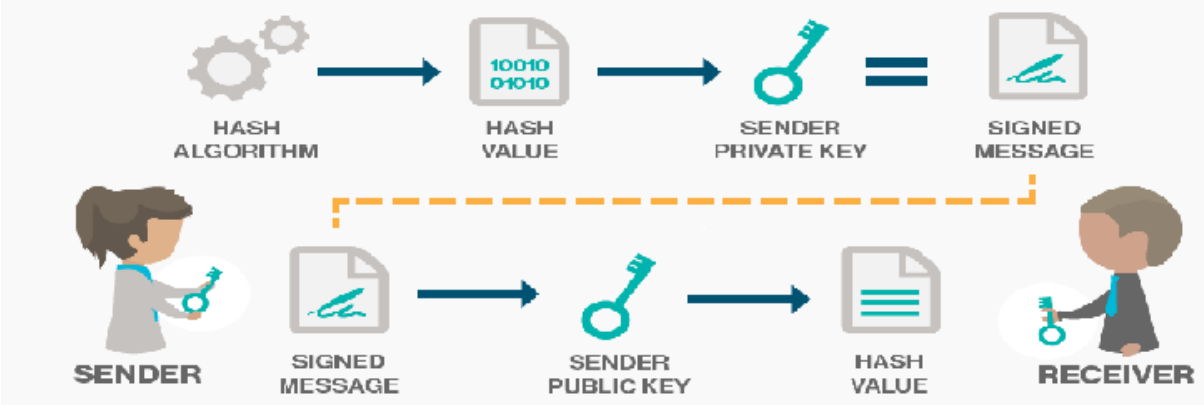


Fig 1: Process of digital signature for authentication

II. ALGORITHM APPROACH

Digital Signature Algorithm (DSA)

The **Digital Signature Algorithm** (DSA) is a Federal Information Processing Standard for **digital signatures**.

The first part of the DSA algorithm is the public key and private key generation, which can be described as:

a) Choose a prime number q, which is called the prime divisor.
b) Choose another primer number p, such that p-1 mod q = 0. p is called the prime modulus.
c) Choose an integer g, such that 1 < g < p, g**q mod p = 1 and g = h**((p–1)/q) mod p. q is also called g's multiplicative order modulo p.
d) Choose an integer, such that 0 < x < q.
e) Compute y as g**x mod p.
f) Package the public key as { p, q, g, y }.
g) Package the private key as { p, q, g, x }.

The second part of the DSA algorithm is the signature generation and signature verification, which can be described as:

Signature generation

a) Generate the message digest h, using a hash algorithm like SHA1.
b) Generate a random number k, such that 0 < k < q.
c) Compute r as (g**k mod p) mod q. If r = 0, select a different k.
d) Compute i, such that k*i mod q = 1. i is called the modular multiplicative inverse of k modulo q.
e) Compute s = i * (h + r * x) mod q. If s = 0, select a different k.
f) Package the digital signature as {r, s}.

Verification

a) Generate the message digest h, using the same hash algorithm.
b) Compute w, such that s*w mod q = 1. w is called the modular multiplicative inverse of s modulo q.
c) Compute u1 = h*w mod q.
d) Compute u2 = r*w mod q.
e) Compute v = (((g**u1)*(y**u2)) mod p) mod q.
f) If v == r, the digital signature is valid.

The Rivest-Shamir-Ad leman (RSA) digital signature scheme

The RSA signature scheme is a deterministic digital signature scheme which facilitates message verification and recovery. For the RSA public-key encryption scheme the message space M and the cipher text space C are Zn= {0, 1, 2, ..., n-1}. Key generation In RSA public key cryptosystems each user

a) Generates two large distinct random primes p and q,
b) Computes n = p q and Φ = (p-1) (q-1)
c) Selects a random integer e, 1 < e < Φ, such that gcd (e, Φ) = 1
d) Computes the unique integer d, 1 < d < Φ, such that ed ≡1 mod Φ Now the public key of Alice (sender) is (n, e) and the private key is d.

Signature generation

To sign a message m Є M, Alice
a) Identifies m with a number ~ m in Zn through a map R: M → Zn.
b) Computes the signature s = ~ md mod n.

Verification

To verify the signature of Alice, Bob (receiver)
a) Chooses the public key (e, n) of Alice.
b) Computes ~ m = se mod n.
c) Verifies that ~ m Є M'where M'denotes the set of images of R. If it does not hold, the signature is rejected else recovers the message as m = R-1(~ m).

The Elliptic Curve Digital Signature Algorithm (ECDSA)

This section describes the procedures to generate and verify the signatures using ECDSA.

Generation

To sign a message m, an entity =A' with domain parameters D= (q, FR, a, b, G, n, h) and associated key pair (d,Q) does the following:
a. Select a random or pseudorandom integer k, 1<=k<=n-1
b. Compute kG=(x1, y1) and convert x1 to an integer x11
c. Compute r= x1 mod n. If r=0 then go to step 1
d. Compute k-1 mod n
e. Compute SHA-1(m) and convert this bit string to an integer e
f. Compute s= k-1 (e+dr) mod n. If s=0 then go to step 1
Signature of A for the message m is (r, s)

Verification

To verify A's signature (r, s) on m, B obtains an authentic copy of A's domain parameters D= (q,FR, a,b,G,n,h) and associated public key Q. It is recommended that B also validates D and Q. Then B does the following:
a. Verify that r and s are integers in the interval [1, n-1]
b. Compute SHA-1 (m) and convert this bit string to an integer e
c. Compute w=s-1 mod n
d. Compute u1=ew mod n and u2=rw mod n
e. Compute x=u1G + u2Q
f. If x=O, then reject the signature. Otherwise, convert the x-coordinate x1 of X to an integer x11 and compute v= x1 mod n
Accept the signature if and only if v=r

III. DIGITAL SIGNATURE TECHNOLOGY

A. Functions of Digital Signature

Digital Signature is a method to encrypt a message which will be transferred, adopting data-exchanging protocol and data-encrypting algorithm. The abstract is like signature or seal which can be used by receiver to verify the identity of the sender. The functions of digital signature: (1) Assuring data integrity. Once the message changes a little, the abstract will change a lot for hash function's peculiarity, so that avoids the message being distorted. (2)Anti-denibility. Using public key Cryptography algorithm, the sender can't deny that he has sent the message for he has the private key. (3)Avoiding receivers forging message that is claimed to be from the sender.

B. Public Key Encrypting Scheme

In the traditional cryptography system, the cipher code used in the process of encrypting plain text into cipher text and in the inverse process is the same. This method is called symmetric cryptography technology. Public key encrypting scheme is a kind of asymmetric Cryptography technology. It resolves the difficult problems in application. Its basic idea: the keys of the two parties are different. Every user has a key pair. One is private key which is saved by the user himself, another one is issued in public places such as internet for downloading or enquiring. Public Key algorithm is very slow (with contrast to private algorithm). It is designed for a little data, but not for much data. It is usually used together with hash function in digital

C. Hash Algorithm

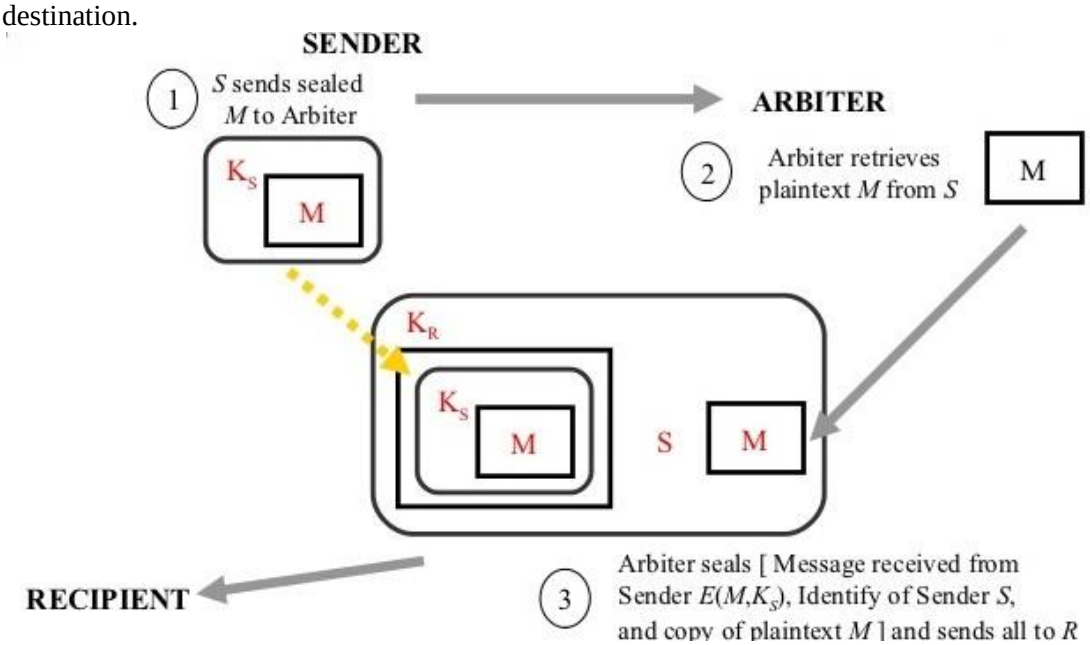
Hash algorithm is an algorithm which is used to compute a data fingerprint of a data block. It is a one-way function which satisfies the following conditions: 1) can receive data with any length; 2) can produce abstract with fixed length; 3) can compute abstract easily; 4) cannot compute message from abstract; 5) It is impossible to find two different messages which have same abstract. Hash function can make short abstract with fixed length for the binary data with any length. The popular hash algorithms are MD5, Secure Hash Algorithm (SHA, having all kinds of security level.) and so on.

IV. RELATED WORK

Here we provide comparative results of Arbitrated digital signature scheme using Authentication Protocols - Mutual authentication and one-way authentication.

The Arbitrated Digital Signature

The problems associated with Direct Digital Signature can be addressed by using an Arbiter. Implementing an arbitrated digital signature invites a third party into the process called a "trusted arbiter." The role of the trusted arbiter is usually twofold: first this independent third party verifies the integrity of the signed message or data. Second, the trusted arbiter dates, or time-stamps, the document, verifying receipt and the passing on of the signed document to its intended final destination.



The Arbiter plays a sensitive and crucial role in this sort of scheme and all parties must have a great deal of trust that the arbitration mechanism is working properly. Every signed message from sender S to a receiver R goes first to an Arbiter A, who subjects the message and its signature to a number of tests to check its origin and content. The message is then dated and sent to R with an indication that it has been verified to the satisfaction of the arbiter. The presence of A solves the problem faced by direct signature schemes, namely that S might deny sending a message. The arbiter plays a sensitive

and crucial role in this scheme, and all parties must trust that the arbitration mechanism is working properly. There are many variations of arbitrated digital-signature schemes. Some schemes allow the arbiter to see the messages, while others don't. The particular scheme employed depends on the needs of the applications. Generally, an arbitrated digital-signature scheme has advantages over a direct digital-signature scheme such as the trust in communications between the parties provided by the trusted arbiter and in the arbitration of later disputes, if any. In the first, symmetric encryption is used. It is assumed that the sender S and the Arbiter A shares a secret key K_{XA} and that A and R share secret key K_{AR}. S constructs a message M and computes its hash value H(M). Then S transmits the message plus a signature to A. This Signature consists of an ID_S of S plus the hash value, all encrypted using K_{SA}. A decrypts the signature and checks the hash value to validate the message. Then A transmits the message to Y, encrypted with K_{AR}. The message includes ID_S, the original message from S, the signature, and Time-stamp; R can decrypt this to recover the message and the signature. The time-stamp informs R that this message is timely and not a replay. R can store M and the signature. In case of dispute, R, who claims to have received M from S, sends the following message to A.

$E(K_{AY},[ID_X||M||E(K_{SA},[ID_S||H(M)])])$

The Arbiter uses K_{AR} to recover ID_S, M, and the signature, and then uses K_{XA} to decrypt the signature and verify the hash code. In this scheme, R cannot directly checks S's signature; R considers the message from S authentic because it comes through A. In this scenario, both sides must have a high degree of trust in A:

- S must trust A not to reveal K_{SA} and not to generate false signature of the form $(K_{SA},[ID_S||H(M)])$.
- R must trust A to send $E(K_{AY},[ID_X||M||E(K_{SA},[ID_S||H(M)])||T])$ only if the hash value is correct and the signature was generated by S
- Both sides must trust A to resolve disputes fairly.

If the arbiter does live up to this trust, then S is assured that no one can forge his/her signature and R is assured that S cannot disavow his/her signature.

Arbitrated Digital Signature Techniques:

(1) S ⟶ A: $M||E(K_{SA},[ID_S||H(M)]$

(2) A ⟶ R: $E(K_{AY},[ID_S||M||E(K_{SA},[ID_S||H(M)])||T])$

(1) S ⟶ A: $ID_S||E(K_{SR},M)||E(K_{SA},[ID_S||H(E(K_{SR},M))])$

(2) A ⟶ R: $E(K_{AR},[ID_S||E(K_{SR},M)])||E(K_{SA},[ID_S||H(E(K_{SR},M))||T])$

(b) Conventional Encryption, Arbiter does not see message

(1) S ⟶ A: $ID_S||E(PR_S,[ID_S||E(PU_R,E(PR_S, M))])$

(2) A ⟶ Y: $E(PR_A,[ID_X||E(PU_R,E(PR_S, M))||T])$

(c)PUBLIC-KEY Encryption, Arbiter does not see message

Notation:

S-Sender A-Arbiter M-Message R-Receiver T-Timestamp

The preceding scenario also implies that A is able to read messages from S to R and, indeed, that any eavesdropper is able to do so. Table (b) shows a scenario the arbitration as before but also asuures confidentiality. In this case it is assumed that S and R share secret key K_{SR} . now, S transmits a identifier, a copy of the message encrypted with K_{SR}, and signature to A. The signature consists of the identifier plus the hash value of the encrypted message, all encrypted using K_{SA}. As before, A decrypts the signature and checks the hash value to validate the message. In this case, A is working

only with the encrypted version of the message and is prevented from reading it. A then transmits everything that it received from S, plus a timestamp, all encrypted with K_{AR}, to R.

Table (c) shows, S double encrypts a message M first with S's private key, PR_S and then with R's public key, PU_R. This is a signed, secret version of the message. This signed message, together with S's identifier, is encrypted again with PR_S and, together with ID_S, is sent to A. The inner, double-encrypted message is secure from the arbiter. However, A can decrypt the outer encryption to assure that the message must have come from S.

This scheme has a number of advantages over the preceding two schemes.

1. No information is shared among the parties before communication, preventing alliances to defraud.
2. No incorrectly dated messages can be sent, even if PR_S is compromised, assuming that PR_A is not compromised. 3. Finally, the content of the message from S to R is secret from A and anyone else. However, this final scheme involves encryption of the message twice with a public-key algorithm.

Authentication Protocols

In this section we focus on two general areas – Mutual Authentication and One-way Authentication, and examine some of the implications of authentication techniques in both.

Mutual Authentication

An important application area is that of mutual authentication protocols. Such protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys. There, the focus was key distribution.

Central to the problem of authenticated key exchange are 2 issues: Confidentiality and Timeliness. To prevent masquerade and to prevent compromise of session keys, essential identification and session key information must be communicated in encrypted form. This requires the prior existence of secret or public keys that can be used for this purpose. The second issue, Timeliness, is important because of their threat of message replays. Such replays, at worst, could allow an opponent to compromise a session key or successfully impersonate another party. At minimum, a successful replay can disrupt operations by presenting parties with messages that appear but are not.

Following are the examples of replay attacks.

- *Simple replay*: The opponent simply copies a message and replays it later.
- *Repetition that can be logged:* An opponent can replay a time-stamped message within the valid time window.
- *Repetition that cannot be detected:* This situation could arise because the original message could have been suppressed and thus did not arrive at its destination; only the replay message arrives.
- *Backward replay without modification:* This is a replay back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content.

One approach to coping with replay attacks is to attach a sequence number to each message used in an authentication exchange. A new message is accepted only if its sequence number is in the proper order. The difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant it has deal with. Because of this overhead, sequence numbers are generally not used for authentication and key exchange. Instead, one of the following two general approaches is used:

- *Time-stamps:* Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time. This approach requires that clocks among the various participants be synchronized.
- *Challenge/response:* Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

Symmetric Encryption Approaches

A two-level hierarchy of symmetric encryption keys can be used to provide confidentiality for communication in a distributed environment. In general, this strategy involves the use of a trusted key distribution center (KDC). Each party in the network shares a secret key, known as a master key, with the KDC. The KDC is responsible for generating keys to be used for a short time over a connection between two parties, known as session keys, and for distributing those keys using the master keys to protect the distribution.

One-Way Authentication

One application foe which encryption is growing in popularity is electronic mail (e-mail). The very nature of electronic mail, and its chief benefit, is that it is not necessary for the sender and receiver to be online at the same time. Instead, the e-mail message is forwarded to the receiver's electronic mailbox, where it is buffered until the receiver is available to read it.

The "envelope" or header of the e-mail message must be in the clear, so that the message can be handled by the store-and-forward e-mail protocol, such as Simple Mail transfer Protocol (SMTP) or X.400. However, it is often desirable that the mail-handling protocol not require access to the plain text form of the message, because that would require trusting the mail-handling mechanism. Accordingly, the e-mail message should be encrypted such that the mail-handling system is not in possession of the decryption key.

A second requirement is that of authentication. Typically, the recipient wants some assurance that the message is from the alleged sender

1 message (A ⟶ B) to establish

- identity of A and that messages is from A
- message intended for B
- integrity & originality of message

Symmetric Encryption Approach

Using symmetric encryption, the decentralized key distribution scenario is impractical. This scheme requires the sender to issue a request to the intended recipient, await a response that includes a session key, and only then send the message. With some refinement, the KDC strategy is a candidate for encrypted electronic mail. Because we wish to avoid requiring that the recipient be online at the same time sender. This approach guarantees that only the intended recipient of a message will be able to read it. It also provides a level of authentication that the sender is A. As specified, the protocol does not protect against replays. Some measure of defence could e provided by including a timestamp with the message. However, because of the potential delays in the e-mail process, such timestamps may have limited usefulness.

Public-key Encryption Approaches

We have public-key encryption approaches that are suited to electronic mail, including the straightforward encryption of the entire message for confidentiality, authentication, or both. These approaches require that either the sender know the recipient's public key (confidentiality) or the recipient know the sender's public key (authentication) or both (confidentiality plus authentication). in addition public key algorithm must be applied once or twice to what may be a long message.

If confidentiality is the primary concern, the following may be more efficient:

$A \longrightarrow B: E(K_S, M)$

In this case, the message is encrypted with a one-time secret key. A also encrypts this one-time key with B's public key. Only B will be able to use the corresponding private key to recover the one-time key and then use that key to decrypt the message. This scheme is more efficient than simply encrypting the entire message with B's public key.

If authentication is the primary concern, then a digital signature may suffice:

$A \longrightarrow B: PR_A, H(M)$

This method guarantees that A cannot later deny having sent the message. However, this technique is open to another kind of fraud.

Here, both the message and signature can be encrypted with recipient's public key:

$A \longrightarrow B: E(M||E(PR_A, H(M))])$

In addition to the message, A sends B the signature, encrypted with A's private key, and A's certificate, encrypted with the private key of the authentication server. The recipient of the message first uses the certificate to obtain the sender's public key ad verify that it is authentic and then uses the public key to verify the message itself. If confidentiality is required, then the entire message can be encrypted with B's public key. Alternatively, the entire message can be encrypted with a one-time secret key; the secret key is also transmitted, encrypted with B's public key.

V. EXPERIMENTAL APPROACH

Our Experiment supports an Internal Arbiter for processing the security mechanisms that happens between a dedicated Digital signature sender and Digital signature receiver. The Arbiter may concurrently handle multiple security mechanisms upon different digital signature Service transactions request. In this Experiment we are using software which is based on process of real phenomenon with the authentication protocols. This software provides the process that is similar to real world environment. This software designed in a manner that is very close to the real world. The software used in this experiment Arbitrated digital signature for transferring of secured messages.

The arbitrated digital signature for modelling and transferring of large amount of digital signature. It provides availability of authentication protocols, which responsible for creation and verification of multiple, independent transferring services on a signature verification and recognisation method. The security issues are caused by authentication protocols with the help of Mutual Authentication and One-way Authentication where the transfer of messages takes place using arbiter as a third party in-order to avoid hacking, losing and misusing of digital messages.

The Steps involved in Experiment using Arbitrated Digital Signature

1. Creation of a digital message
2. Creation of sender and receiver
3. Arbiter is created
4. Transaction process starts
5. Involvement of arbiter to provide security for digital message
6. Successful transmission of digital message from sender to receiver

This experiment is done using authentication protocols with the help of arbiter as a third party person who provides security and be a reason for safe transmission of digital message without any hacking and loss of data between sender and receiver.

VI. CONCLUSION

Many traditional and newer business and applications have recently been carrying out enormous amounts of electronic transactions, which have led to a critical need for protecting the information from being maliciously altered, for ensuring the authenticity, and for supporting non-repudiation. This paper describes arbitrated digital signature using authentication protocols. This experiment is done on two different protocols such as mutual authentication and One-way Authentication. Just as signatures facilitate validation and verification of the authenticity of paper documents, digital signatures serve the purpose of validation and authentication of electronic documents.

The experiment results conclude that the arbitrated of digital signature with E-authentication protocols provides highest security for encryption and Decryption of message compared to Direct digital signature from unauthorised persons. Since Authentication Protocols provides more security for the transfer of digital message.

VII. FUTURE WORK

The performance and time-delay of transmitting digital messages are very complex problems. when there exists large amount of message for communication, in order to provide security from arbiter, it need to consider some protocols which takes little-bit time to process. Thus, it makes coming messages in waiting state for providing security as the message cannot be transmitted until it has been checked for security.

REFERENCES

[1]. Shraddha Kalbhor**,** Anita Gaikwad, Kajal Bhise, Prof. Dipmala Salunke, Varsha Bangar,"A Survey on Digital Signature", Information Technology, 2015, IJETAE 2015, Volume 5, Issue 1.

[2]. Venkateswara Rao Pallipamu, Thammi Reddy K, Suresh Varma P, "Design of RSA Digital Signature Scheme using a novel Cryptographic Hash Algorithm",2014,IJATAE 2014,Volume 4,Issue 6.

[3]. Erfaneh Noroozi, Salwani Mohd Daud, Ali Sabouhi, "Secure Digital Signature Schemes Based on Hash Functions" , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-4, March 2013
[4]. M. Bellare and P. Rogaway, "The Exact Security of Digital Signatures –Howto Sign with RSA and Rabin," Proc. Of Eurocrypt'96, Springer-Verlag, LNCS,pp.399– 416, 1996.378-379
[5]. Arvind Negi , Punit Sharma, Prasant Chaudhary, Himanshu Gupta,"New Method for Obtaining Digital Signature Certificate using Proposed RSA Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 121 – No.23, July 2015
[6]. Chiranjib Dutta, Swati Sarkar, Animesh Kar,"An Efficient Implementation of Digital Signature", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 2, February 2015
[7]. R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public key cryptosystems," Communications of the ACM, vol. 21, pp. 120- 126, 2003
[8]. D. Pointcheval and J.Stern,‖ Security arguments for digital signatures and blind signatures,‖ Journal of Cryptology, vol.13,no.3,pp.361-396,2000.
[9]. Abhishek Roy and Sunil Karforma,"A Survey on Digital Signatures and its Application", J. of Comp. and I.T. Vol. 3(1&2), 45-69 (2012).
[10]. http://www.cse.unr.edu/~bebis/CS477/Papers/DigitalSignatures.pdf
[11]. Shivendra Singh, Md. Sarfaraz Iqbal, Arunima Jaiswal,"Survey on Techniques Developed using Digital Signature: Public key Cryptography", International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 16, May 2015.
[12]. by Sandro Gerić, Tomislav Vidačić, MIPRO 2012, May 21-25,2012, Opatija, Croatia," XML Digital Signature and its Role in Information System Security".
[13]. Wang HongBin Ren Yan,"Code – Based Designated Verifier Signature Scheme". Emerging Intelligent Data and Web Technologies (EIDWT), IEEE 2013 Fourth International Conference.

AUTHORS

GyaderlaRanjith is a Post Graduate in Master of Technology from J.N.T University, in Computer Science and Engineering. Having Teaching Experience of 05 years, He is actively involved in teaching Theory of Computation, Computer programming languages, Object oriented concepts and Computer Networks. He published 5 Research papers at International Journals, Member of IACSIT and at present Working as Asst Professor & HOD, Department of Computer Science and Engineering, Warangal Institute of Technology and Science, Oorugonda (V), GudepaduX Roads, Atmakur (M), Warangal-506342.

BonaganiPrathusha persuing M.TECH in Computer Science and Engineering, specialization at Software Engineering at Warangal Institute of Technology and Science, At present working as Assistant Professor in Warangal Institute of Technological sciences, Oorugonda(V), GudepaduX Roads, Atmakur(M), Warangal, Affiliated to Kakathiya University. Interested research areas are Artificial Intelligence, Network Security, Data Mining and Data Warehousing and Cloud Computing.

PochalaSagarika working as Assistant professor at Warangal Institute of Technological sciences, worked as a software engineer at Open Origin IT Solutions, Amplus Solutions, Sholey IT Solutions. Interested areas of research are Network Security and Cloud Security, at present Persuing M.TECH at Warangal Institute of Technology and Science, Oorugonda (V), GudepaduX Roads, Atmakur (M), Warangal-506342.