A Design of Security Protocol using Hybrid Encryption Technique (AES- Rijndael and RSA)

Dr. E. Ramaraj[1], S. Karthikeyan[2] and M. Hemalatha[3]

[1,2]Department of Computer Science and Engineering,
Alagappa University,
Karaikudi, Tamilnadu, INDIA.
[3]Department of Computer Science,
Mother Teresa Women's University,
Kodaikanal, Tamilnadu, INDIA.
E-mail: kkd_ramraj@sancharnet.in, skaarthi@ieee.org, hemalatha_sekar@yahoo.com

Abstract

This paper aims to design the new security protocol using hybrid encryption technique for on line transaction. The hybrid encryption technique is a combination of both symmetric and asymmetric cryptographic techniques. The encryption algorithms are more secured depends on the key value and its size. But, the key distribution is major problem. The various protocols are currently given the solution. The new protocol solves the key management problem using key servers. It also provides all the three cryptographic primitives - integrity, confidentiality and authentication. In this proposed design methodology, the new protocol design using Symmetric cipher (AES-Rijndael) and public key cryptography (RSA) with hash function SHA-512.

Keywords: Hybrid Encryption, Key Management, Security, Authentication and Integrity

1. Introduction

The communication is major impact of today's business. The communication device transmits the large amount of data with high security. In a business, the amount approximately worth over $1 trillion was being transacted every week on the Net. But, unfortunately, the cyber crimes nearly 97% of such crimes are undetected [3]. The security is still remains risky one. At present, the various types of cryptographic algorithms provide high security in information, computer and network-related activities. These algorithms are needs to protect the data, integrity and authenticity from various attacks [9][10]. In this paper we provide the design of new protocol for better security using key server with hybrid encryption technique.

2. Background of the Study

The cryptographic algorithms are classified into two different types such as symmetric and asymmetric method.

In symmetric encryption method both sender and receiver share the common key value for encryption and decryption. It requires that the sender find some secure way to deliver the encryption/decryption key to the receiver. The effective key distribution needs to deliver key to the receiver [8].

In [7][8], the authors described about the key distribution difficulties. Large number of protocols provides various techniques. These protocols are to provide more secure

but less performance. The public key cryptography or asymmetric cryptographic method solves the problems of key distribution. In this method, uses a pair of keys for encryption. The public key encrypts the data and corresponding private key for decryption. Each user has one pair of keys. The private key kept secret and public key knows by others. Any one wants to send some information to you they read your public key and encrypts the information. After you receive, the encrypted data using your private key to decrypt it. One issue with public key cryptosystems is that users must be constantly vigilant to ensure that they are encrypting to the correct person's key. In a public key environment you are assured that the public keys to which you are encrypting data is in fact the public key of the intended receiver. The identification of correct public key of proper person is more difficult without using any third party.

Everyone knows the cryptographic algorithms functionality. The sender sends his data using any one cryptographic algorithm with key value. The key value is more confidential. The key management is also more complex.

3. Purpose of the Study

In [6], the authors described about digital certificates and trusted third party to provide the assurance that the public key is the proper person key or not. This digital certificates basically a public key with one or two forms of ID attached and trusted information from a third party. This it will be used when it is necessary to exchange public keys with someone else. The public key infrastructure (PKI) provides the key servers to manage the keys like issue or revoke of a public key for particular person and so on. The PKI has a person or persons responsible for authenticating all the keys administered in their PKI, known as certification authority [8]. There are three forms of trust in public key cryptography. In direct trust, trust yourself by calling and validating fingerprint. In hierarchical trust, the certificate authority is competent and honest and is correctly certifying keys that are issued by your organization's PKI server. In a web trust, the various people who have signed someone's key as valid especially if some of them are people known to you directly and trusted by you.

The above discussion, the third party or person who involves in the certification is more honest. In our proposal we formulate the certification and verification of keys using key servers. The authenticity verified by using public key cryptography (RSA) and the integrity by hash functions.

4. Overview of Hybrid Encryption Approach

The various cryptographic algorithms are available for network security. The symmetric cryptographic algorithms are high speed compared than asymmetric cryptographic algorithms or public key cryptographic systems like RSA, Elliptic Curve Cryptography. The public key cryptographic algorithms are more secure than symmetric algorithms. Because, it has two keys one for encryption and another one for decryption. In this hybrid encryption technique we propose symmetric encryption for encryption/decryption and using public key cryptosystems for authentication [8].

4.1 AES-Algorithm

4.1.1 Overview of AES Cipher

The Advanced Encryption Standard (AES) is a computer security standard that became effective on May 26, 2002 by NIST to replace DES. The cryptography scheme is a symmetric block cipher that encrypts and

decrypts 128-bit blocks of data. Lengths of 128, 192, and 256 bits are standard key lengths used by AES [4].

4.1.2 Overview of AES-Rijndael

The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192 and 256 bits. A use of three key size alternatives but limits the block length to 128 bits.

The algorithm was designed to have the following characteristics:

- Resistance against all known attacks
- Speed and code compactness on a wide range of platforms
- Design simplicity
- Input to the encryption algorithm, decryption algorithm in a single 128 bit block

In AES, four different stages are used

i. Substitution bytes
   Use S-box to perform byte-to-byte substitution of the block

ii. Shift rows
   A simple permutation

iii. Mix columns
   A substitution that makes use of arithmetic

iv. Add round key
   A simple bit wise XOR of the current block with the portion of the expanded key

In add round key stage makes use of the key. Any other stage applied at the beginning or end is reversible without knowledge of the key, this scheme is more efficient and secure.

Figure 1 illustrates the AES encryption and Decryption process.

Each stage is easily reversible. For the substitute byte, shift row, mix column stages, as inverse function used in the decryption algorithm. For add round key stage, the inverse is achieved by XOR the same round key to the block. The decryption algorithm is not identical for the encryption algorithm. This is a consequence of the particular structure of the AES.

4.2 Overview of RSA

The RSA scheme is a block cipher in which the plain text and cipher texts are integers between 0 and n-1 for some n. We examine RSA in this section in some detail, beginning with an explanation of the algorithm.

4.2.1 Description of the Algorithm

The plain text is encrypted in blocks, with each block having a binary value less than some number n. That is, the block size must be less than or equal to $\log_2(n)$; in practice, the block size is 2^k bits, where $2^k<n<=2^k+1$. Encryption and decryption are of the following form, for some plain text block M and cipher text block C: [8]

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, the public key encryption algorithm with a public key of KU = {e, n} and private key of KR = {d, n}. For this algorithm to be satisfactory for public key encryption, the following requirements must be met:

1. It is possible to find values of e, d, n such that $M^{ed} = M \bmod n$ for all M<n.
2. It is relatively easy to calculate M^e and C^d for all values of M<n.
3. It is infeasible to determine d given e and n.

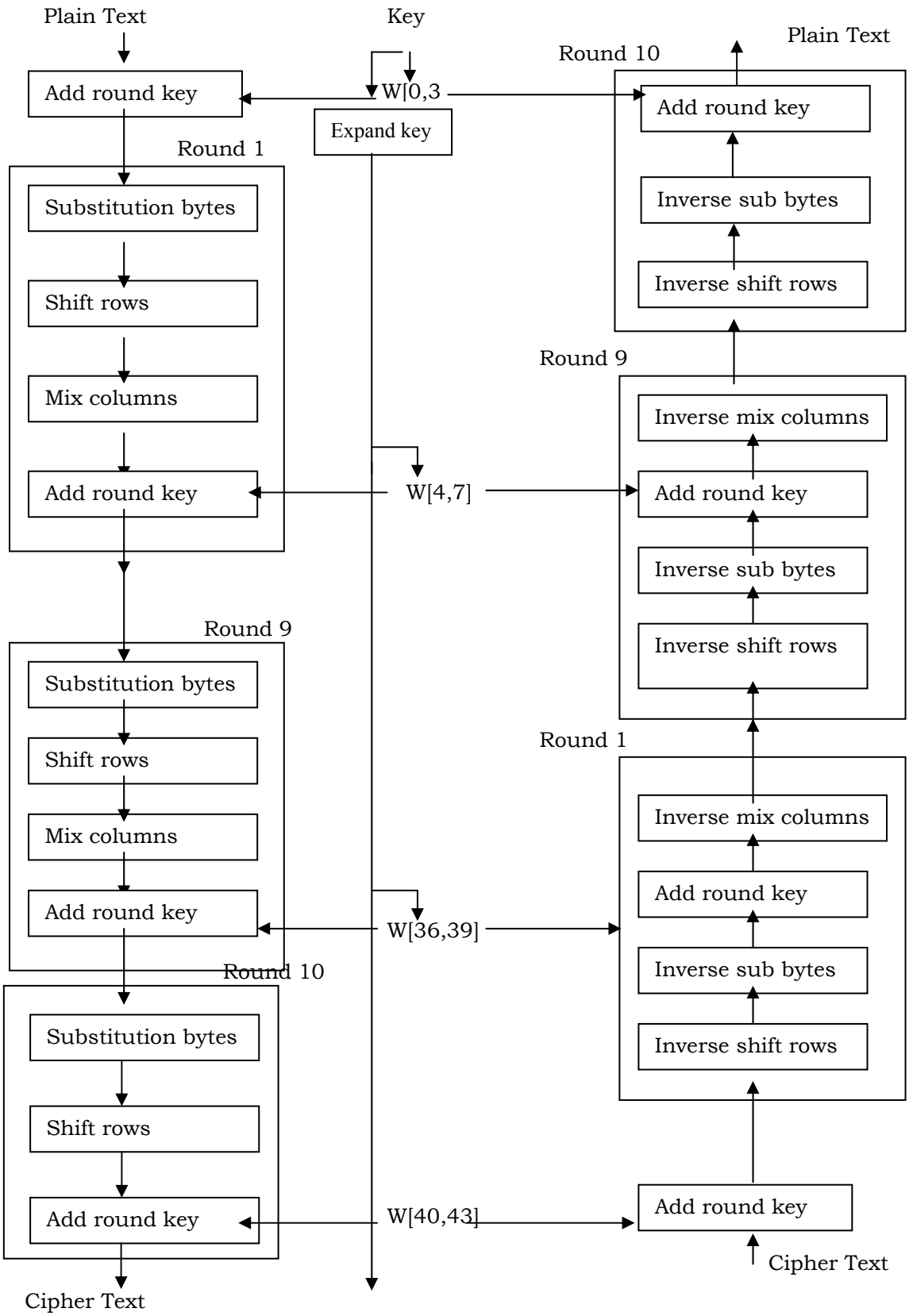

Figure 1. AES- Rijndael Encryption and Decryption process

The following states will be followed in RSA scheme.

p, q, two prime numbers (private, chosen)
n = pq (public, calculated)
e, with gcd(φ(n),e)=1; 1<e< φ(n)(public, chosen)
$d = e^{-1} \bmod \varphi(n)$ (private, calculated)

The private key consists of {d, n} and the public key consists of {e, n}. Suppose that user A has published its public key and that user B wishes to send the message M to A. Then B calculates $C = M^e \pmod n$ and transmits C. On receipt of this cipher text, user A decrypts by calculating $M = C^d \pmod n$.

So, $M^{ed} \equiv M \bmod n$. Now,
$C = M^e \bmod n$
$M = C^d \bmod n \equiv (M^e)^d \bmod n$
$\equiv M^{ed} \bmod n \equiv M \bmod n$

4.3 Hybrid Encryption Technique

In this hybrid encryption approach, sender side using 128-bit session key value with AES-Rijndael to encrypt the message. The hash value of message was encrypted using RSA algorithm with 1028 bit public key of the receiver. In the receiver side the decryption done for the encrypted message using AES-Rijndael with 128-bit session key value. To calculate the hash value using hash function SHA-512 for the original message. Using RSA with 1028 bit private key of the receiver to decrypt the encrypted hash value. To ensure the integrity the comparison performed between calculated and decrypted hash values. The following figure 2 and figure 3 explain this process.

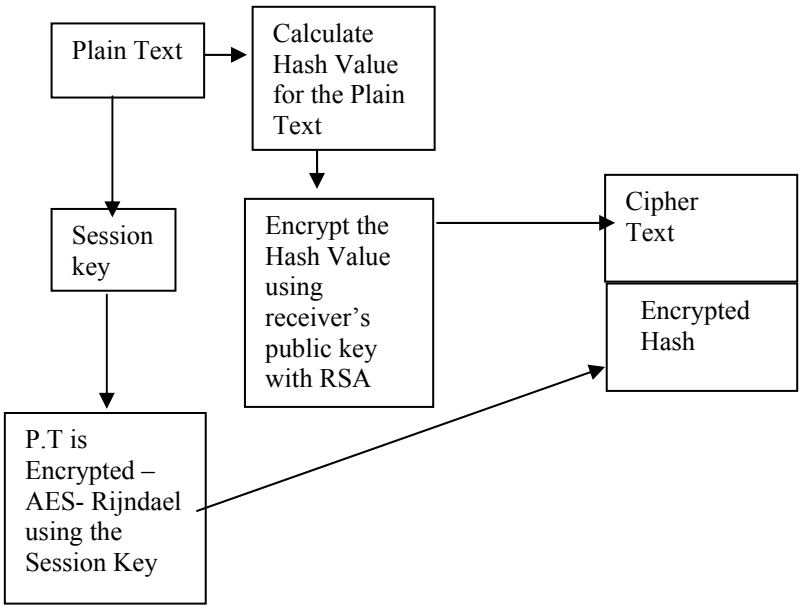

Figure-2: Encryption Process

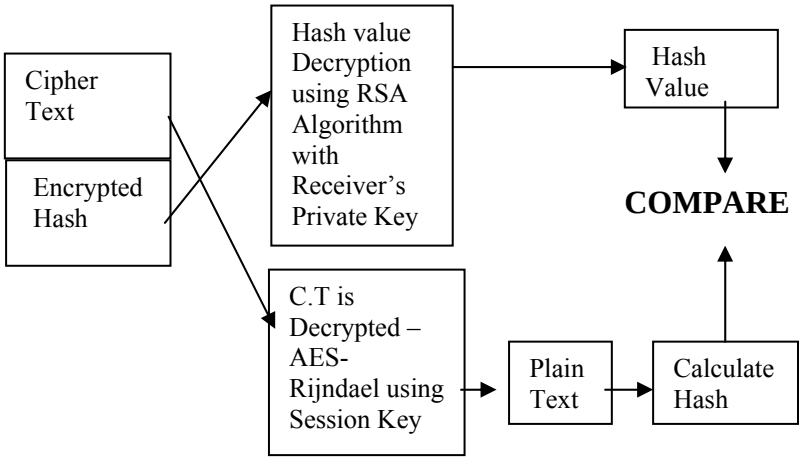

Figure-3: Decryption Process

5. Proposed Approach

In our proposed approach, to manage the keys between the users using remote key servers. The key management and secure transaction will be done by hybrid encryption technique. In this approach, three steps will be followed.

i. Session Key Establishment Phase

The identification of correct public key of proper person is more difficult without using any third party. In this phase the identification of correct public key of proper person using key servers.

The following protocol proposed by Woo and Lam [ ], makes use of nonce.

1. A->KS:$ID_A||ID_B$
2. KS->A:$E_{KRauth}[ID_B||KU_b]$
3. A->B :$E_{KUb}[N_a||ID_A]$
4. B->KS:$ID_A||ID_B||E_{KUauth}[N_a]$
5. KS>B:$E_{KRauth}[ID_A||KU_a]||E_{KUb}[E_{KRauth}[N_a||K_S||ID_B]]$
6. B->A :$E_{KUa}[E_{KRauth}[N_a||K_S||ID_A||ID_B]||N_b]$
7. A->B:$E_{KS}[N_b]$

In the above protocol, in step 1, A informs the Key Server (KS) of its intention to establish a secure connection with B. The KS returns to 'A' copy of B's public key certificate. Using B's public key, A informs B of its desire to communicate and sends a nonce N_a (step 3). In step 4, B asks the KS for A's public key certificate and requests a session key; B includes A's nonce so that the KS can stamp the session key with that nonce. The nonce is protected using KS public key. In step 5, the KS returns to B a copy of A's public key certificate, plus the information $\{N_a, K_S, ID_B, ID_A\}$. This information basically says K_S is the secret key generated by the KS on behalf of B and tied to N_a. This information is encrypted using KS private key, to allow B to verify that information from the KS. It also encrypted using B's public key, so that no other entity may use the information in an attempt to establish a fraudulent connection with A. In step 6, the information $\{N_a, K_S, ID_B, ID_A\}$ still encrypted with KS private key, is relayed with A together with a nonce N_b, generated by B. All foregoing are encrypted using A's public key. A retrieves the session key and uses it to encrypt N_b and return it to B. The last message assures B of A's knowledge of the session key. Thus it is the pair $\{N_a, ID_A\}$ that uniquely identifies the connection request of A.

In our new protocol design, we revise the above protocol design to protect various attacks.

1. A->KS:$ID_A||ID_B||E_{KUauth}[V_a]$
2. KS->A:$E_{KRauth}[ID_B||KU_b||V_a]$
3. A->B:$E_{KUb}[N_a||ID_A]$
4. B->KS:$ID_A||ID_B||E_{KUauth}[N_a||V_b]$
5. KS->B:$E_{KRauth}[ID_A||KU_a||ID_B]||E_{KUb}[E_{KRauth}[N_a]]$
6. B->A:$E_{KUa}[E_{KRb}[N_a||K_S||ID_A||ID_B]||N_b]$
7. A->B:$E_{KS}[N_b]$

In our revised protocol, in step 1, A informs the Key Server (KS) of its intention to establish a secure connection with B. The A sends ID_A, ID_B and $E_{KUauth}[V_a]$. The V_a is a shared common value only knows both A and KS. The KS maintains unique values for each user, it checks the user ID_A and value V_a is correct or not. If it is correct the KS assures that the information requisite is the correct person. If any unauthorized person as a member in KS, he/she sends the encrypted request using public key of the key server, but they don't know about the value V_a. In step 2, the KS returns to 'A' copy of B's public key certificate and value V_a. Using B's public key, A informs B of its desire to communicate and sends a nonce N_a (step 3). In step 4, B asks the KS for A's public key certificate using the information $ID_A||ID_B||E_{KUauth}[N_a||V_b]$. In this step we include V_b, so KS checks V_b and assures that the B is a correct person or not. The nonce also protected using KS public key. In step 5, the KS returns to B a copy of A's public key certificate, plus the information $\{N_a, ID_B, ID_A\}$. In step 6, the session key K_S fixed by B and the information $\{N_a, K_S, ID_B, ID_A\}$ still encrypted with B's private key and again encrypted with A's public key, is relayed with A together with a nonce N_b, generated by B. A retrieves the session key and uses it to encrypt N_b and return it to B.

In this protocol, the KS assures the requester is a correct person or not. The session key value KS only knew by A and B.

ii. Secure Transmission Phase

The figure-4 illustrates this. In this phase, the first part sender 'A' selects common 128-bit session key value (KS) for encryption purpose. The AES-Rijndael symmetric encryption algorithm-using key value KS to encrypt the plain text. In the second part, the hash value was calculated by using the plain text and hash function. Again the hash value is encrypted using RSA with 1024-bit public key KU_b. Both encrypted information send to the receiver B. The SHA-512 hash function used to perform hash calculation.

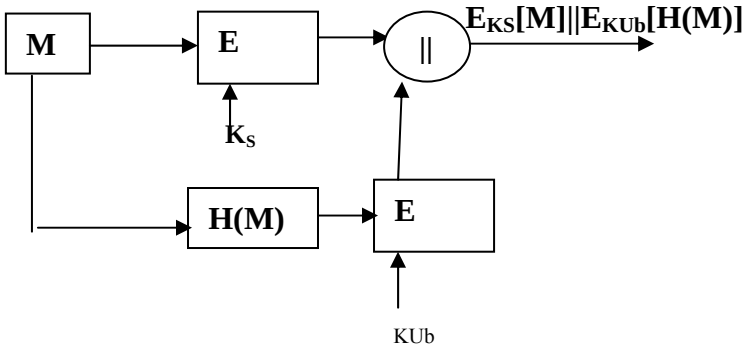

Figure-4: Secure Encryption Process

iii. Secure Decryption Phase

In this phase, the first part sender B selects the session key KS for decryption purpose. The AES-Rijndael symmetric encryption algorithm using session key value KS to decrypt the plain text and calculate the hash value using hash algorithm SHA-512. In the second part, using receiver private key value KR_b with RSA algorithm decrypts the encrypted hash value. Then, the decrypted hash value is compared with calculated hash value. If the hash value is equal, the receiver assures that integrity of the message good. The figure-5 illustrates this.

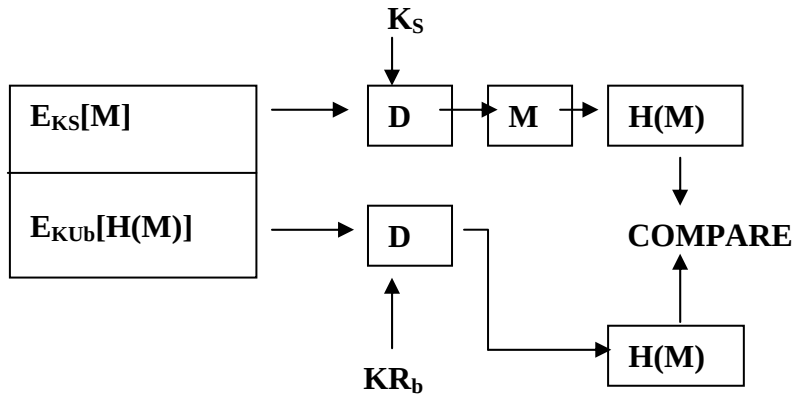

Figure-5: Secure Decryption Process

6. Conclusion

In this paper we give the protocol design for secure key agreement using hybrid encryption technique. The public keys were not freely available. This protocol provides the way to select the session key by the receiver not key server. This hybrid encryption method also surely will increase the performance of cryptographic algorithms. This protocol will ensure the confidentiality, integrity and authentication. The AES-Rijndael algorithm provides confidentiality, the hash function provides the integrity and RSA will ensure the authentication.

Acknowledgement

The authors wish to thank the Chairman and Managing Trustee, Administrative Officer, Principal and Faculty for their kind support for doing research in Karpagam Arts and Science College, Coimbatore, Tamilnadu, INDIA.

References

1. Chein HY, Jan JK, Tseng YM. (2002), "An efficient and practical to remote authentication: Smart Card Security", *ELSEVIER-Computers & Security Journal*, 21(4), pp. 372-375.
2. Eun-Jun Yoon, Eun-Kyung Ryu, Kee-Young Yoo (2005), "An improvement of Hwang-Lee-Tang's simple remote user authentication scheme", *ELSEVIER-Computers & Security Journal*, 24(1), pp. 50-56.
3. Hwang MS, Lf LH. (2000), "A new remote user authentication scheme using Smart Cards", *IEEE Transactions*, 46(1). pp. 110-120
4. James Nechvatal, Elaine Barker and Lawrence Bassham, "Report on the Development of the Advanced Encryption Standard (AES)", Computer and Security Division, National Institute of Standards and Technology (NIST), US Dept. of Commerce.

5. Mayer R. Thompson, Abdelilah and Srilekha Mudumbai (2003), "Certificate-Based Authorization Policy in a PKI Environment", *ACM Transactions on Information and System Security*, Vol. 6; No. 4, 566-588
6. Srdjan Capkun, Levente Buttyan and Jean-Pierre Hubaux (2003), "Self organized public key management for mobile ad-hoc networks", *IEEE Transactions,* 2(1), pp. 51-63.
7. Tianjie Cao, Dongdai Lin and Rui Xue (2005), "A randomized RSA-based partially blind signature scheme for electronic cash", *ELSEVIER-Computers & Security Journal*, 24(1), pp. 44-49.
8. William Stallings (2003), *Cryptography and Network Security-Principles and Practices,* 3rd Edition, Pearson Education Asia.
9. Wu ST, Chieu BC (2003), "A user friendly remote authentication scheme with smart cards:, *ELSEVIER-Computers & Security Journal*, 22(6), pp. 547-597.
10. Yang WH, Shieh SP (1999), "Password Authentication Schemes with Smart Card", *ELSEVIER-Computers & Security Journal*, 18(8), pp. 727-760.