Security Providing using Blowfish, RSA and SHA-512 Algorithms

[1]G. Hanumantha Rao, [2]G. Narender, [3]T. Balaji, [4]M. Srilatha
[1,3,4]Dept. of CSE, Vasavi College of Engineering, Hyderbad, AP, India
[2]Dept. of CSE, Keshav memorial Institute of Technology, Hyderabad, AP, India

Abstract
A Computer Network is an interconnected group of autonomous computing nodes, which use a well defined, mutually agreed set of rules and conventions known as protocols, interact with one-another meaningfully and allow resource sharing preferably in a predictable and controllable manner. Communication has a major impact on today's business. It is desired to communicate data with high security. Security Attacks compromises the security and hence various Symmetric and Asymmetric cryptographic algorithms have been proposed to achieve the security services such as Authentication, Confidentiality, Integrity, Non-Repudiation and Availability. At present, various types of cryptographic algorithms provide high security to information on controlled networks. These algorithms are required to provide data security and users authenticity. To improve the strength of these security algorithms, a new security protocol for on line transaction can be designed using combination of both symmetric and asymmetric cryptographic techniques. This protocol provides three cryptographic primitives such as integrity, confidentiality and authentication. These three primitives can be achieved with the help of Secure Hash Algorithm, Blowfish algorithm and RSA algorithm. That is it uses Blowfish for encryption, RSA algorithm for authentication and SHA-512 for integrity. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.

Keywords
Network Security, Blowfish, RSA, and SHA-512

I. Introduction

The communication is major impact of today's business. The main consideration in designing an encryption algorithm has to be the security of the algorithm against undesirable attacks. The encryption technique used in the protocol is a combination of both symmetric and asymmetric cryptographic techniques [1]. The cryptographic algorithms are classified into two different types such as symmetric and asymmetric method. These algorithms are needs to protect the data, integrity and authenticity from various attacks.

In symmetric encryption method both sender and receiver share the common key value for encryption and decryption [2]. Asymmetric cryptographic 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 [3].

The authentication service is concerned with assurance that a communication is authentic. Peer entity authentication provides the corroboration of the identity of a peer entity in association. It is provided for the establishment of, or at times during the transfer phase, of a connection. Data origin authentication provides for the corroboration of the source of a data unit. It supports applications like electronic mail which there are no prior interactions between the communicating entities [4].

Data confidentiality service provides the protection of data from unauthorized disclosure. Connection Confidentiality provides the protection of all user data on a connection. Connectionless Confidentiality provides the protection of all user data in a single data block. Selective-Field Confidentiality provides the confidentiality of selected fields within the user data on a connection or in a single data block. Traffic-flow Confidentiality provides the protection of the information that might be derived from observation of traffic flows [5].

Data Integrity is the assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

II. Overview of the Algorithms

The following algorithms were chosen for implementation i.e.,

- Blowfish
- RSA
- SHA-512.

A. Blowfish

Blowfish is a symmetric block cipher developed by Bruce Schneier. Blowfish makes use of a key that ranges from 32 bits to 448 bits. The key [6] used to generate 18 32-bit subkeys and four 8×32 S-boxes containing a total of 1024 32-bit entries.

The steps in generating the P-array and S-boxes are as follows:

1. Initialize first P-array and then the four S-boxes in order using the bits of fractional part of the constant π.
2. Perform a bitwise XOR of the P-array and the K-array, reusing words from the K-array as needed.
3. Encrypt the 64-bit block of all zeros using the current P and S arrays; replace P_1 and P_2 with the output of the encryption.
4. Encrypt the output of step3 using the current P and S arrays and replace P_3 and P_4 with the resulting cipher text.
5. Continue this process to update all elements of P and then, in order, all elements of S, using at each step the output of the continuously changing Blowfish algorithm.

In the encryption process [7] the plaintext is divided into two 32-bit halves LE0 and RE0.The variables LEi and REi to refer to the left and right half of the data after round i has completed. The encryption algorithm can be defined by the following pseudocode:

```
For i=1 to 16 do
        REi=LEi-1 XOR Pi;
        LEi=F[REi] XOR REi-1;
LE17=RE16 XOR P18
RE17=LE16 XOR P17;
```

In vice versa, we were performing Decryption process.

B. RSA Algorithm

The algorithm developed by Rivest, Shamir and Adleman makes use of an expression with exponentials. The RSA algorithm involves three steps: key generation, encryption and decryption.

1. Key Generation

RSA involves a public key and a private key. The public key can be known to every one and is used for encrypting messages.

Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated in the following way:

1. choose two prime numbers p and q.
2. Compute n = p*q.
3. Compute φ(n) = (p − 1)(q − 1).
4. Choose an integer e such that 1 < e < φ(n) and gcd (e,φ(n))= 1 (i.e., e and φ(n) are coprime).
5. Determine d = e^{-1}(modφ(n)).(i.e., d is the multipli-cative inverse of e(mod φ(n))).

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the private(or decryption) exponent d which must be kept secret.

Encryption is of the following form

$c = m^e \pmod n$.

Decryption is of the following form

$m = c^d \pmod n$.

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 this a public-key algorithm with a public key of PU={e,n} and a private key of PU={d,n}.

C. SHA-512

The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993; the algorithm takes as input a message with a maximum length of less than 2128 bits and produces as output a 512-bit message digest [8]. The input is processed in 1024-bit blocks. The processing consists of the following steps.

1. Append padding bits. The message is padded so that its length is congruent to 896 modulo 1024 (length 896 mod `1024). Padding is always added, even if the message is already of the desired length. Thus the number of padding bits in the range of 1 to 1024. The padding consists of a single 1-bit followed by the necessary number of 0-bits.
2. 2. Append length. A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding). The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. The expanded message is represented as the sequence of 1024-bit blocks M_1, M_2,..., M_N, so that the total length of the expanded message is N x 1024 bits.
3. 3. Initialize hash buffer. A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers (hexadecimal values).

a = 6A09E667F3BCC908
b = BB67AE8584CAA73B
c = 3C6EF372FE94F82B
d = A54FF53A5F1D36F1
e = 510E527FADE682D1
f = 9B05688C2B3E6C1F
g = 1F83D9ABFB41BD6B
h = 5BE0CDI9137E2179

These values are stored in big-endian format, which is the most significant byte of a word in the low-address (leftmost) byte position. These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

4. The heart of the algorithm is a module that consists of 80 rounds; this module is labeled F.

Each round takes as input the 512-bit buffer value abcdefgh, and updates the contents of the buffer. At input to the first round, the buffer has the value of the intermediate hash value, H_{i-1}. Each round t makes use of a 64-bit value Wt derived from the current 1024-bit block being processed (M_i). These values are derived using a message schedule described subsequently. Each round also makes use of an additive constant Kt where $0 \le t \le 79$ indicates one of the 80 rounds. These words represent the first sixty-four bits of the fractional parts of the cube roots of the first eighty prime numbers [9]. The constants provide a "randomized" set of 64-bit patterns, which should eliminate any regularity in the input data.

The output of the eightieth round is added to the input to first round (H_{i-1}) to produce Hi. The addition is done independently for each of the eight words in the buffer with each of the corresponding words in Hi-1 using addition modulo 264.

5. Output. After all N 1024-bit blocks have been processed; the output from the nth stage is the 512-bit message digest.

We can summarize the behavior of SHA-512 as follows:

H_0	$= IV$
H_i	$= SUM_{64}(H_{i-1}, abcdefgh_i)$
MD	$= H_N$

Where IV= initial value of the adcdefgh buffer

abcdefgh=the output of the last round of processing of the ith message block

N=the number of blocks in the message (including padding and length fields).

SUM_{64}= Addition modulo 2^{64} performed separately on each word of the pair of inputs

MD=final message digest value.

III. Secure Encryption Process

In this phase, the first part sender 'A' selects common session key (Ks) value for encryption purpose. The session key is shared by both sender and receiver. By using Blowfish symmetric encryption algorithm and using the session key the plain text is encrypted for achieving confidentiality [10].

In the second part, the hash value was calculated by using the Secure Hash Algorithm (SHA-512) for the plain text and hash function to achieve message integrity. Again the hash value is encrypted using RSA with 1024-bit public key KUb of the receiver for authentication. Both encrypted information and message digest send to the receiver B.

This process was represented in the following fig. 1.

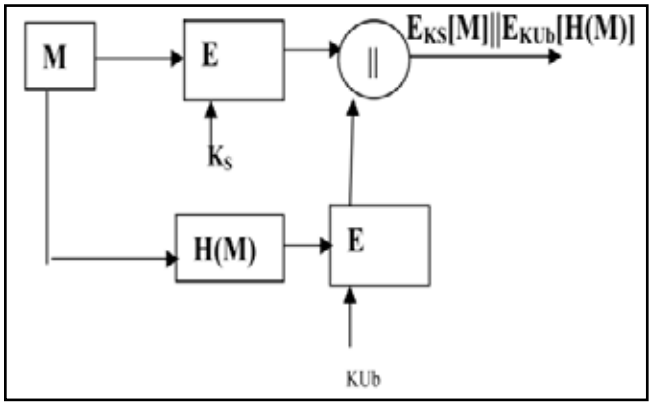

Fig. 1: Secure Encryption Process

IV. Secure Decryption Process

In this phase, the first part sender B selects the session key Ks for decryption purpose. This session key is known for both the sender and receiver. By using the Blowfish symmetric encryption algorithm and using session key value to decrypt the plain text. After decryption (getting plain text) the decrypted text was used to calculate the hash value using Secure Hash Algorithm (SHA-512) [11].

In the second part, using receiver private key value KRb with RSA algorithm decrypts the encrypted hash value. Then, the decrypted hash value is compared with calculated hash value. If the hash values are equal, the receiver assures that integrity of the message good else receiver identifying that message is not good.

This process was represented in the fig. 2.

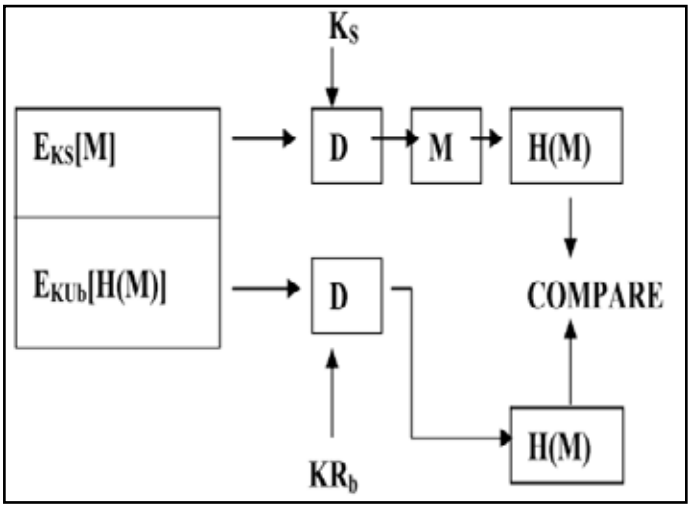

Fig. 2: Secure Decryption Process

V. Conclusion

A new security protocol has been designed for better security. It is a combination of both symmetric and asymmetric cryptographic techniques. It provides three cryptographic primitives– integrity, confidentiality and authentication. It makes use of Blowfish to encrypt, RSA to authenticate and SHA-512 to check for integrity. To enhance the Encryption technique, security aspects of key management need to be looked into more detail for its robustness.

References

[1] Ramaraj, E, Karthikeyan, S,"A Design of Enhanced Security Protocol for Wireless Communication using Hybrid Encryption Technique (AES – Rijndael and RSA)", Indian Journal of Computing Technology, pp. 22-29,May, 2006.

[2] William Stallings,"Cryptography and Network Security – Principles and Practices", 3rd Edition, Pearson Education Asia – 2003.

[3] Hung-Min Sun, et al.,"Dual RSA and its Security Analysis", IEEE Transaction on Information Theory, Aug 2007, pp. 2922 – 2933,2007.

[4] S. D. Galbraith, C. Heneghan, J. F. McKee,"Tunable balancing of RSA", 2005. Updated version of ACISP 2005.

[5] Wu ST, Chieu BC,"A user friendly remote authentication scheme with smart cards", ELSEVIER-Computers & Security Journal, 22(6), pp. 547-597, 2003.

[6] Yang WH, Shieh SP,"Password Authentication Schemes with Smart Card", ELSEVIER-Computers & Security Journal, 18(8), pp. 727-760, 1999.

[7] [Online] Available: http://www.csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[8] [Online] Available: http://www.vocal.com/data_sheets/SHA1.pdf

[9] Arjen K. Lenstra,"Selecting Cryptographic Key Sizes", Vol. 14, Issue 4, Journal of Cryptography, [Online] Available: http://www.springerlink.com/content/6d8hb94aenemfm5g, pp. 255-293, 2002

[10] Ravindra Kumar Chahar and et.al.,"Design of a new Security Protocol", IEEE International Conference on Computational Intelligence and Multimedia Applications, pp. 132 – 134, 2007

[11] Ramaraj, E, Karthikeyan, S,"A Design of Enhanced Security Protocol for Wireless Communication using Hybrid Encryption Technique", Indian Journal of Computing Technology, pp. 22-29, May, 2006.