International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Special Issue 1; March 2018; Page No. 39-42

Implementation of double encrypted MAC based on Diffie-Hellman key exchange for improved integrity and authentication

[1] Dr. Richa Purohit, [2] Savita Vivek Mohurle, [3] Manisha Patil, [4] Dr. Deepshikha Bhargava
[1, 2, 3] MIT ACSC, Alandi (D) Pune, Maharashtra, India
[4] Amity University Rajasthan, Jaipur, Rajasthan, India

Abstract
Network security is the major factor of importance while communicating a message between authenticated parties using internet. Confidentiality, integrity and availability are three major goals of network security. There are many methods for ensuring, that these goals have been achieved, like digital signature, hash functions, public/private encryption etc. and similarly, many algorithms exist for implementing each of these methods. This paper proposes one such method in the area of network security where, using secret key along with public and private key for creating encrypted hash, assures message integrity, sender and receiver authentication and non-repudiation between both involved parties.

Keywords: authentication, integrity, network security, hash, secret key, KDC

1. Introduction

In current scenario, where e-communication has become a need, Digital Signatures, similar to hand written signatures provide a mechanism to authorize the information being transferred [1]. Network Security revolves around three main goals or objectives [2]:

a) **Confidentiality: -** No one, except sender and receiver, should be able to view, copy, print or modify the data and moreover, meaning of message should not be understandable by any unauthorized party. Third party should not even be able to get the information that any message is being transferred from some sender to some receiver.
b) **Integrity: -** Once the message has been transferred, there should be no mechanism to modify its contents by any other user of the network.
c) **Availability: -** The original and authenticated data should be visible to corresponding authenticated receiver and to no one else in any case.

These three goals are depicted in Figure 1.

Confidentiality
Integrity
Availability
Security

Fig 1: Three Goals of Security

2. Current Scenario

To a certain extent, confidentiality can be maintained by using suitable type of encryption techniques and preventing intruder from analyzing traffic. Keyed and unkeyed Hash Functions (in the form of MDC and MAC) are used for maintaining integrity of transferred message [3, 4]. MD5 [5] and SHA-1 are widely used and most popular world-wide established hash functions. MD4 is the basis for both of these hash functions [6]. Since it became popular, many successful attacks have been performed on MD4, so all hash functions that are based upon its structure may also have common weaknesses [4]. Digital Signatures provide a mechanism for authorization of sender and receiver and thus can prevent non-repudiation later on. DSS is known acceptable Digital Signature standard, widely used in public and private domains [7].

3. Proposed Mechanism:

This paper proposes a mechanism for generating double encrypted hash value of the message based on Diffie-Hellman key exchange algorithm. Diffie Hellman is a symmetric key exchange algorithm [11]. Using this algorithm for encoding final digest value can ensure that two parties that are involved in the communication have previously authenticate each other while exchanging their secret key.

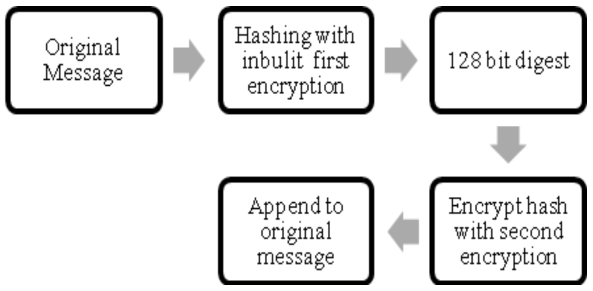
Original Message
Hashing with inbulit first encryption
128 bit digest
Encrypt hash with second encryption
Append to original message

Fig 2: Overview of proposed method of securing data

We also propose use of a keyed algorithm while implementing Hash function for original message. Thus, after the hashing, the output digest will be in an encrypted form so that no one can trace it and get a meaning out of it. The overview of the process can be shown as given flowchart (Figure 2):
The proposed system implements a twofold encryption, first while hashing and second after hashing. Second encryption is performed on calculated digest value of hash function. At the end, this encrypted hash value is either transmitted to receiver or passed to signature function for further authentication. Basic structure of the process takes from Merkle-Damgard Construction [8, 9], Advanced Encryption Standard [10] and Diffie-Hellman Key Exchange algorithms [11]. The detailed steps of the mechanism are as follows:

1. First of all, divide the whole message into blocks of 512 bit. If the last block is lesser than 512 bit, than pad the last block so that its length is congruent 448 modulo 512. i.e. length $\cong$ 448 mod 512. For padding, use first bit 1 and all remaining bits 0.
2. At the end of last block, append length of original message in 64 bit pattern so that it completes last block also.
3. Processing for hashing is done in 512 bit blocks and in loop. Each 512 bit block is treated as sixteen 32 bit words. Each block is processed with a 128 bit CV (Cumulative vector). For processing the first block, CV is initialized with a fixed 128 bit value and known as IV (Initialization vector).
4. The CV (or IV for first block) is first divided into four 32 bit blocks- A, B, C and D. The processing of input blocks consists of four similar stages, called rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Figure 3 illustrates one operation within a round. There are four possible functions; a different one is used in each round:

   F (B,C,D) = (B AND C) OR (NOT(B) AND D)
   G (B,C,D) = (B AND D) OR (C AND NOT(D))
   H (B,C,D) = B XOR C XOR D
   I (B,C,D) = C XOR (B OR NOT(D))

5. The output of processing of each 512 bit block with 128 bit CV is a 128 bit value.
6. As first encryption, this 128 bit value is passed to an iterative substitution-permutation encryption function. This function computes on bytes, instead of bits. Thus, these 128 bit input to this encryption function, is treated as 16 bytes input for further processing. These 16 bytes are arranged in four columns and four rows for processing as a matrix in 10 similar rounds of operations. Here the key length is also 128 bit for each round of operation. This is similar to AES. One round of processing can be shown in Figure 3:

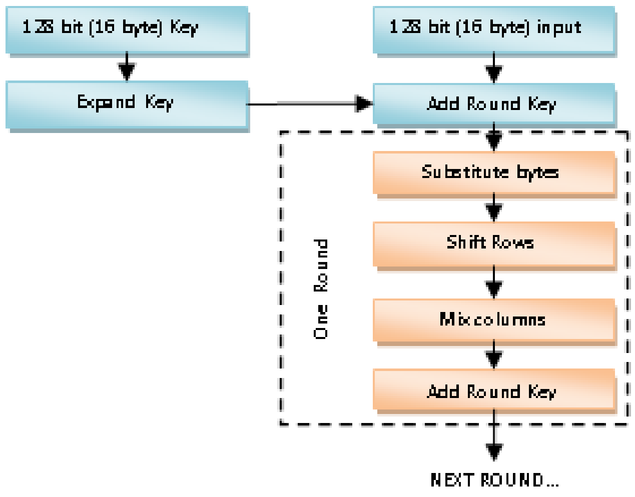

Fig 3: One Round of First Encryption

7. First encryption produces a 128 bit encrypted value, which is used as cumulative vector (CV) for next block of message.
8. 128 bit final output of encryption of last block's processed value is then passed to second encryption function.
9. This second encryption function is symmetric encryption function, where both involved parties (sender and receiver) share a secret key using Diffie-Helman Key exchange scheme based on discrete logarithm. The key exchange for second encryption is done as follows:
   a. Both parties work on a certain prime number P and its primitive root R.
   b. Sender selects its private key S_{pr}, where $S_{pr} < P$ and calculates its public key S_{pb}, where $S_{pb} = R^{Spr}$ mod P.
   c. Receiver selects its private key R_{pr}, where $R_{pr} < P$ and calculates its public key R_{pb}, where $R_{pb} = R^{Rpr}$ mod P.
   d. Sender calculates secret key K, where K= (R_{pb}) mod P.
   e. Receiver calculates secret key K, where K= (S_{pb}) mod P.
10. Sender uses secret key K to encrypt 128 bit final value (as discussed in point viii), and appends this to the encrypted message before transmission.
11. At the receiver end, receiver traces the encrypted hash that is appended to the message and calculates hash in the similar fashion and gets encrypted hash. If calculated and received hash are same than it proves that both involved

parties have authenticated each other for this communication.

Complete processing can be shown as given figure 4.

4. Strength of the Proposed Algorithm

Proposed algorithm uses two different keys for two different encryption functions. First encryption is symmetric encryption technique that uses 128 bit key in 10 rounds of operation to work upon 128 bit input and produces 128 bit CV. For its implementation, a trusted third party plays a role of key distribution center (KDC) [12] and securely passes 128 bit key to both sender and receiver. Similarly for second encryption, the sender and receiver share their public keys, prime number and its primitive root, and then they form their secret keys using this common information and their secret key. Again for distribution of public keys, prime number and its primitive root, trusted third party KDC comes into the picture and passes this information to both involved parties via trusted medium.

If any of the key is missing from any of the party then secure communication is not possible and message digest verification will not happen. Such a message is resend with all trusted communication between KDC and authorized parties.

Thus, this solution proposes sharing of secret keys twice during one message transfer and this results in twice authentication of sender and receiver while communication. This dual authentication enhances chances of avoiding fraud and thus avoids threat to integrity because no party can modify, create or send the data if it has not received key from KDC.

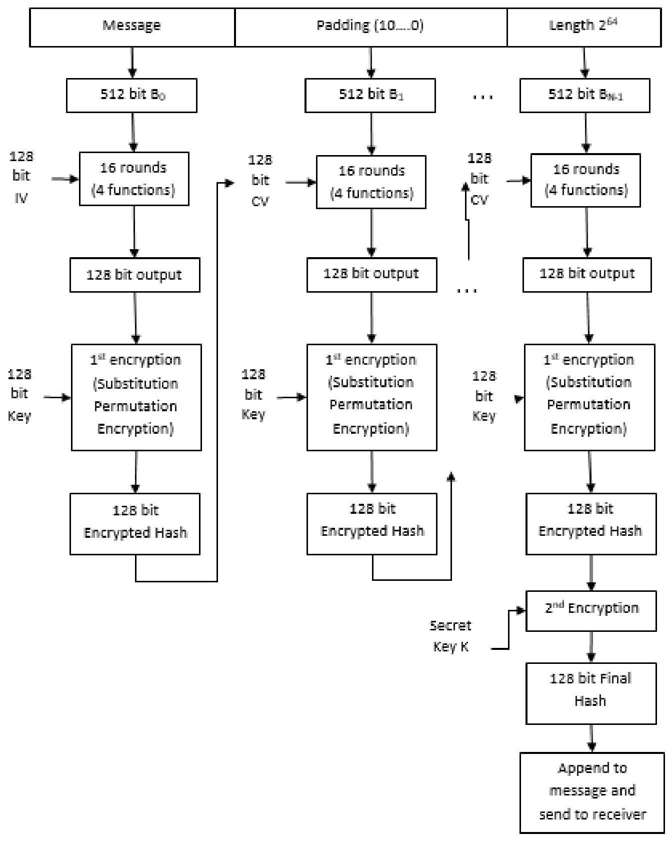

Fig 4: Flowchart for Double Encrypted MAC Algorithm

Moreover, non-repudiation can also be avoided because if a party receives authenticated data, then it means the sender was authenticated via using key from KDC.

Key distribution for both encryption processes using KDC is shown in figure 5 and figure 6.

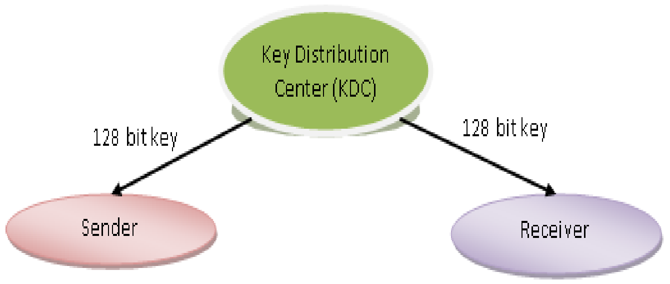

Fig 5: Key Distribution in First Encryption

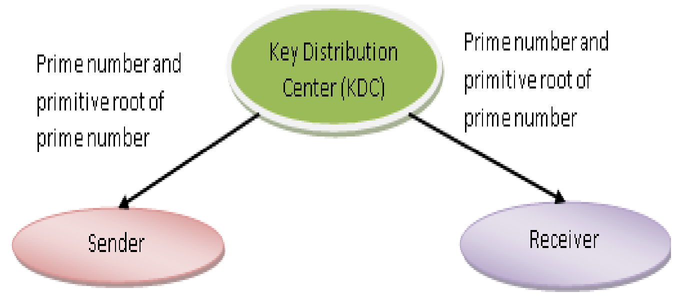
Key Distribution Center (KDC)
Prime number and primitive root of prime number
Prime number and primitive root of prime number
Sender
Receiver

Fig 6: Key Distribution in Second Encryption

5. Conclusion

Secure network communication along with proper authentication and integrity is most demanding feature while interchanging information using internet. Hash function and digital signature serve this purpose. This paper proposes modified hash function that uses keys, generated by Diffie-Helman Key exchange, for generating final digest value in two different keyed functions; one private and one public. Use of key makes this algorithm suitable for ideal MDC (Message Detection Code) and helps in preventing repudiation by involved authorized parties and modification by any unauthorized party.

6. References

1. Gauravram P. Cryptographic Hash Functions: Cryptanalysis, design and Applications, Ph.D. thesis, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, 2003.
2. Stallings W. Cryptography and Network Security Principles and Practices, Prentice Hall Press Upper Saddle River, 2010.
3. Ilya M. Hash Functions: Theory, Attacks, and Applications, in Proceedings of Microsoft Research, Silicon Valley Campus, 2005, 1-22.
4. Purohit R, *et al*. Integrtation of Encryption and Hash Function for Improved Message Authenticity.
5. International Journal of Engineering Research and Applications. 2012; 2(5):2137-2142
6. Rivest R. The MD5 Message-Digest Algorithm. RFC, 1992, 1321.

7. Rivest RL. MD4 Message Digest Algorithm. RFC, 1990, 1186.
8. National Institute of Science and Technology, Implementing Cryptography, NIST SP 800. Available at: http://csrc.nist.gov/publications/ nistpubs/800-21-1/sp800-21-1_Dec2005.pdf.
9. Damgård I. A Design Principle for Hash Functions, in CRYPTO, 1989, 416-427.
10. A Survey on Recent Cryptographic Hash Function Design. International Journal of Emerging Trends & Technology in Computer Science. 2013; 2(1):117-122.
11. Dandekar AK, Pradhan S, Ghormade S. Design of AES-512 Algorithm for Communication Network". International Research Journal of Engineering and Technology. 2016; 3(5):438-443.
12. Eun-Jun Yoon, Kee-Young Yoo. An Efficient Diffie-HellmanMAC Key Exchange Scheme. Fourth International Conference on Innovative Computing, Information and Control, 2009.
13. Design and Analysis of a New Hash Algorithm with Key Integration. International Journal of Computer Applications Published by Foundation of Computer Science, New York, USA. 2013; 81(1):33-38. DOI: 10.5120/13978-1974.