An Efficient and Secure Authentication of Short Encrypted Message over Computing Application

Jallu Venkata Dalappala Naidu[1], Behara Vineela [2]

Final M.Tech Student[1], Asst.professor[2]

[1,2]Dept of CSE, *Sarada Institute of Science, Technology and Management (SISTAM), Srikakulam, Andhra Pradesh*

Abstract:

The main goal of cryptography to provide security of transferred message to exchange message between the public channels. To provide security of transferring message we are implementing cryptography technique. In the public channel we can transfer message before we can performing checking operation for corrupting data or not. By performing corrupt process the user will retrieve transferred message and delete some content in that message. So that to overcome this problem each user will perform the encryption and decryption of transferred message. In this paper we are mainly concentrate generation of secret key, encryption and decryption of transferred message. In this paper we are implementing an efficient key generation process and send that key to all group members in the network. After completion of group key the key generation center will send that to all group members with securely. So that the only the authorized people to retrieve secret and perform the encryption, decryption of transferring message. In this paper we are implementing an efficient cryptography technique for performing the encryption and decryption process. By using binary reverse key xor encryption process for performing the encryption process for transferring message into unknown format i.e. cipher format. After completion of encryption process the sender will generate signature for cipher format data. Take those signature and cipher format data send to destination node. The destination node will retrieve cipher format data and signature again will generate signature for that cipher format. After completion of generation signature compare both signatures are equal or not. If both signatures are equal we can perform decryption process of binary reverse key xor decryption process will get original plain format data. If the signatures are not equal we can stop decryption process and jamming message. By implementing those concepts we can improve efficiency of transferring message and also provide more security of transferring message.

Keywords: *cryptography techniques, hash function, security, encryption and decryption Signature.*

.

I. Introduction

The main goal of cryptography is to preserve the integrity of messages exchanged over public channels. A message authentication code algorithm (MAC) is designed for the sole purpose of preserving message integrity. Pervasive computing is the growing trend towards embedding microprocessors in everyday objects, it means “existing everywhere”. Pervasive computing devices are completely connected and constantly available. They combine the current network technologies with wireless computing, voice recognition, internet capability and artificial intelligence is to create an environment where the connectivity of devices is embedded in such a way that the connectivity is unobtrusive and always available. Such pervasive computing and mobile computing devices rely on short messages for which MAC can be computed more efficiently. Based on their security MACs can either be unconditionally secure or computationally secure. MACs provide message integrity against the forgers with unlimited computational power. On the other hand, computationally secure MACs are only secure when forgers have limited computational power. The use of universal hash function families in the carter-wegman style is not restricted to the design on unconditionally secure authentication. Computationally secure MACs based on universal hash functions can be constructed with two rounds of computations. In the first round the message to be authenticated is compressed using a universal hashing function. Then in the second round, the compressed image is processed with a cryptographic function. Universal hashing based MACs give better performance when compared to block cipher or cryptographic hashing based MACs. One of the main differences between unconditionally secure MACs based on universal hashing is the requirement to process the compressed image with a cryptographic primitive in the latter class of MACs. Two observations to me made are:

1) They are designed independently of any other operations required to be performed on the message to be authenticated.

2) The most existing MACs are designed for the general computer communication systems, independently of the properties that messages can possess.

There have been significant efforts devoted to the design of hardware efficient implementation that suite such small devices. However, there has been little or no effort in the design of message authentication codes that can utilize other operations and the special properties of such networks. In cryptography, secure channels enable the confidential and authenticated message exchange between authorized users. A generic approach of constructing such channels is by combining an encryption primitive with an authentication primitive (MAC). In this work, we introduce the design of a new cryptographic primitive to be used in the construction of secure channels. There are two main approaches for the construction of secure cryptographic channels: a dedicated approach and a generic approach. In the dedicated approach, a cryptographic primitive is designed to achieve authenticated encryption as a standalone system (see, e.g., [1,2]. In the generic approach, an authentication primitive is combined with an encryption primitive to provide message integrity and confidentiality. Over dedicated primitives, generic compositions possess several design and analysis advantages due to their modularity and the fact that encryption and authentication schemes can be designed, analyzed, and replaced independently from each other . Further, and most important, generic compositions can allow for faster implementations of authenticated encryption when fast encryption algorithms, such as stream ciphers, are combined with fast MACs, such as universal hash functions based MACs].

II. RELATED WORK

Many standard MACs that can be used in the construction of authenticated encryption schemes have appeared in the literature. Standard MACs can be block ciphers based, cryptographic hash functions based, or universal hash functions based. CBC-MAC is one of the most known block cipher based MACs specified in FIPS publication 113 [3] and the International Organization for Standardization ISO/IEC 9797-1 [4]. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B [5], which was based on OMAC of Iwata and Kurosawa [5]. Other block cipher based MACs include, but are not limited to, XOR-MAC [6] and PMAC [7]. The security of different MACs has been exhaustively studied. HMAC is a popular example of the use of iterated cryptographic hash functions to design MACs [8], which was adopted as a standard [9]. Another cryptographic hash function based MAC is the MDx-MAC of Preneel and Oorschot [10].

HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 . Bosselaers et al. described how cryptographic hash functions can be carefully coded to take advantage of the structure of the Pentium processor to speed up the authentication process [11]. The use of universal hash families was pioneered by Wegman and Carter in the context of designing unconditionally secure authentication. The use of universal hash functions for the design of computationally secure MACs appeared in [12]. The basic concept behind the design of computationally secure universal hash functions based MACs is to compress the message using universal hash functions and then process the compressed output using a cryptographic function. The key idea is that processing messages using universal hash functions is faster than processing them block by block using block ciphers. Then, since the hashed image is typically much shorter than the message itself, processing the hashed image with a cryptographic function is faster then processing the entire message.

III.PROPOSED SYSTEM

In this section we are mainly proposed message authentication approach that is faster than the existing approach. Before performing message authentication the key generation center will generate secret key and sent to public channel members for message encryption and decryption. After generating secret key the channel member or group member will send message to specified member of the group. Before sending message the group member will encrypt message and generate signature for that message. After completion of encryption and signature generation the group member will send message and signature to specified member of the group. The specified group member will retrieve the cipher message and signature. After retrieving the group member again generate signature and compare both signature are equal the message is authenticated or not equal it will block the message. The following concepts are specifying generation of secret key, encryption and decryption of message and generate signature for encrypted message.

Users Registration:

This module explains the process computation of key and users registration. After registering users the KGC will generate id for individual users U_i and sent to users. During registration process each user will choose a random secret value S_i and send to KGC. Once user registration process is completed, KGC assigns a permanent secret id, denoted by Pi for each Member Ui in the group.

Key generation and distribution to group member:

In this module each user will request for group key, the KGC will randomly generate secret. After generating secret key the KGC will send that key to each user with in secure manner. By providing security for secret key the KGC will generate a message and send to all users. The KGC will now generation of message and its value is calculated by using following formula.

Message=$(P_1 ® S_1) * (P_2 ® S_2) * ....* (P_n ® S_n)$ + secret key.

After generating message the KGC will sent the message to all group members. The group member will retrieve the message get secret key from the message. Upon receiving the message M, the each member in the groups will generate the key in the following manner.

Secret key = M mod $(P_i ® S_i)$ for all i.

After completion of secret key each user will encrypt the message by using following algorithm.

Reverse Binary xor Encryption Algorithm:

We will be presenting the steps of the encryption algorithm of the reverse binary xor Algorithm. The following steps are as shown in Figure 1:

1. Input secret key and transferring message to encryption process.

2. Get each character from the message and convert into ASCii values.

3. After converting ascii values each value xor with key until the length of message is completed.

4. The completion of xor operation each ascii value can converted into binary format.

5. Reverse previous binary data until completion length of message.

6. After reversing binary data that data can be perform once complement.

7. The previous complement data convert into ascii format.

8. Divide each ascii value by secret key and get remainder and coefficient until completion of length of message.

9.Each character of remainder and coefficient become one point and those points send to specified group member.

Before sending the cipher data to specified group member the user will generate signature for encrypted message by using MD5 algorithm. After generating signature the user will send cipher message and signature to specified group member. The specified group member will retrieve the cipher message and signature and again will generate signature for cipher message. The group member will compare both signatures are equal the message is authenticated otherwise the message will corrupt and block the message. If the message is authenticated then specified group member will retrieve cipher message and get the original message by performing decryption process of reverse binary xor encryption algorithm. The Decryption process of Reverse binary xor encryption algorithm as follows.

1. Retrieve the point from the sender group member.
2. Get the single ascii value from the point by using following formula.
   Ascii val= quotient * secretkey + remainder.

3. The previous ascii val will be convert into binary format.

4. The previous binary data can be perform the once complement.

5. After performing once complement that binary data will be reverse until the completion message length.

6. After reversing that binary data will convert into ascii format.

7. The previous ascii values will be xor with secret key until completion of message length.

8. After performing xor operation that ascii values can be converted characters get the original message.

IV. Conclusions

In this paper we propose new technique for authenticated transferring shorted encrypted message in a public channels. The fact is that message is authenticated must also be encrypted is to deliver the specified group member. Before performing authentication of encrypted message the key generation center will generate secret key and send to all group member. Each user will use the secret key for encryption and decryption process. After generation of group key each user encrypt the message using binary reverse key xor encryption algorithm of encryption process. After encrypt message the user will send cipher message to specified group member in the group. The group member will retrieve the message and again generate signature. After generating signature compare both signature are equal the transferring message authenticated or the message not authenticated. The transferring message is authenticated then the group member will decrypt message by using decryption process of binary reverse key xor encryption algorithm. By implementing those concepts we can improve efficiency and security of transferring message.

REFERENCES

[1]. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, and T. Kohno, "Helix: Fast encryption and authentication in a single cryptographic primitive," in Proceedings of Fast Software Encryption–FSE'03, vol. 2887, Lecture notes in computer science. Springer, 2003, pp. 330–346

[2]. J. Carter and M. Wegman, "Universal classes of hash functions," in Proceedings of the ninth annual ACM symposium on Theory of computing–STOC'77. ACM, 1977, pp. 106–112.

[3]. FIPS 113. Computer Data Authentication. Federal Information Processing Standards Publication, 113, 1985.

[4]. ISO/IEC 9797-1. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999.

[5]. M. Dworkin. Recommendation for block cipher modes of operation: The CMAC mode for authentication, 2005.

[6]. T. Iwata and K. Kurosawa. omac: One-key cbc mac. In Fast.

[7]. M. Bellare, R. Guerin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In Advances in Cryptology–CRYPTO'95, volume 963, pages 15–28. Lecture Notes in Computer Science, Springer, 1995.

[8]. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In Advances in Cryptology–CRYPTO'96, volume 96, pages 1–15. Lecture Notes in Computer Science, Springer, 1996.

[9]. FIPS 198. The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication, 198, 2002.

[10]. B. Preneel and P. Van Oorschot. MDx-MAC and building fast MACs from hash functions. In Advances in Cryptology-CRYPTO'95, volume 963, pages 1–14. Lecture Notes in Computer Science, Springer, 1995.

[11]. A. Bosselaers, R. Govaerts, and J. Vandewalle. Fast hashing on the Pentium. In Advances in CryptologyCRYPTO'96, volume 1109, pages 298–312. Lecture Notes in Computer Science, Springer, 1996.

[12]. . D. McGrew and J. Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. In Progress in Cryptology-INDOCRYPT'04, volume 3348, pages 343–355. Lecture notes in computer science, Springer, 2004.

BIOGRAPHIES:

Jallu Venkata Dalappala Naidu is student in M.tech (CSE) in Sarada Institute of Science Technology and Management, Srikakulam. He has received her B.tech (CSE) from Aditya Institute of Technology and Management, Tekkali. His interesting areas are data mining and network security

Behara Vineela is working as Asst.professorin Sarada Institute of Science, Technology And Management, Srikakulam, Andhra Pradesh. She received her M.Tech (CSE) from AITAM, Tekkali, Srikakulam, Andhra Pradesh. JNTU Kakinada Andhra Pradesh. Her research areas include Net-work Security