International Journal of Innovative Research and Advanced Studies (IJIRAS)
Volume 3 Issue 9, August 2016

ISSN: 2394-4404

Mechanism For Imposing E-Mail Header Security: Expectation And Solution

Dr. Richa Purohit

Assistant Professor, MIT Arts,
Commerce and Science College, Alandi, Pune

Abstract: ***Today e-mails have become an important tool for message transfer over internet. Since beginning message security is being considered as a basic requirement in this field. But various headers like To, From, Subject, Date etc. are also equally important and should be sent and received in secured manner. This paper discusses various e-mail security tools and also suggests a simple mechanism to achieve security for e-mail headers. This paper discusses existing popular e-mail security technique S/MIME in brief and also suggests a simple mechanism to achieve security for e-mail headers. This technique uses encryption, digital signature and hashing for providing security to e-mail headers.***

Keywords: e-mail, e-mail header, security, message digest, digital signature, encryption.

I. INTRODUCTION

With increased demand of fast and efficient communication, e-mails got popularity, as they are written documents and can be used for official purpose also. Ever since the inception of Computer and Computer Networks, security is the biggest area of concern for researchers. While sending crucial data through mail, the issue of security becomes even more crucial. We cannot trust a received mail, if we know that the process of creating, sending and receiving e-mail is not safe enough. Thus, with an increase of e-mails in current scenario, increased security is also required to build trust on the mechanism.

II. COMPOSITION OF E-MAIL HEADER

E-mail header is the beginning part of the e-mail [1], which contains little basic information about sender, receiver and the settings at the time e-mail was sent. In SMTP (Simple Mail Transfer Protocol), the typical mail header comprise of following parts:

- ✓ *FROM:* This part displays the sender's e-mail address so that recipient can know where is the message coming from.
- ✓ *TO:* This part shows the recipient's email address to which the e-mail is to be delivered.
- ✓ *SUBJECT:* If the sender wishes, then he/she can give the title or topic of the e-mail.
- ✓ *RETURN-PATH:* This part is useful for "Reply-to" when the recipient also wishes to send acknowledgement or some another message to the sender, as contains the e-mail address of the sender.
- ✓ *DATE:* The date and time when the e-mail was composed and sent to the recipient.
- ✓ *DELIVERY DATE:* This part shows the date and time when e-mail was received by the receiver on his computer.\
- ✓ *BCC:* A Blind Carbon Copy part contains the (list of) address(es) of recipients which will remain invisible to other recipients.
- ✓ *CC:* A list of multiple visible recipients to whom the mail will be sent, apart from main recipient.
- ✓ *MESSAGE ID:* This is an auto generated field for each mail, whose purpose is to prevent more than one delivery of the same message to the same receiver again and again.

E-MAIL HEADER SPOOFING

Spoofing means to imitate something [2]. In e-mail, all header parts can be spoofed. The reason for spoofing one part may be different from another. For example spoofing "From" field, may lead to receive a mail with the name of some party that is authorized for communication, where as the actual sender being someone else. Thus, receiver will receive and may also reply to such mail, considering that mail has come from an authorized sender only. Spoofing "Date" field may display the mail at top of unread mails, and hence may get higher priority while reading. An email can be spoofed by predated it or by postdated it, as compared to original sending date of that mail. Similarly, though message ID is auto-generated and unique for each mail, but if it is spoofed then actual mail with the same message ID will be automatically discarded by the recipient, considering it to be same as previously received.

Thus, any type of header part, if spoofed, may lead to unauthenticated and untrustworthy e-mail and can cause further problems in communication. That is why, with an increase of malicious attacks on security aspects of e-mail, the increased need of security of header part of e-mail, from spoofing, is also being considered as most vital issue now-a-days.

III. EXPECTATIONS FROM WELL DEFINED E-MAIL SECURITY FUNCTIONS

E-Mail Security function is always desired to provide [3]:

- ✓ Source Authentication- It is desirable that source of the e-mail is authenticated before receiving a mail from it. Source Authentication ensures that the sender is one from the authenticated list of senders for this particular receiver and is trusted by third party for further notarization, if required.
- ✓ Message Integrity- The message contents, received by the receiver should be exactly the same as sent by the sender. This property is called ensuring message integrity. This is required to ensure that no third unauthorized party has changed the contents before being received by the receiver.
- ✓ Source non-repudiation- This function ensures that no unauthorized third party is sending the e-mail to the receiver with the name of some authorized sender. The original sender is sending this mail with its name only.
- ✓ Data Confidentiality- This is the property of securing e-mail from being copied, printed, viewed by anyone else but the sender and the receiver. Even, the third party or intruder should not be aware of the fact that some communication has been occurred through e-mail between particular source and receiver at particular time.
- ✓ Private/Public Key Management- While sender and receiver of the e-mail are engaged with process of encryption/decryption, digital signature and hashing etc. for securing their mails, their private, public or secret keys should be distributed in secured manner. That is, private key should never be transmitted over web, so anyone, other than user, can know it. Public key should be distributed in a manner such that it has not been altered in the path and can be used for decryption purpose effectively. Secret key should be known to sender and receiver only.

To provide these features, the e-mail security function must be considered as group of a number of algorithms [4], such as:

- ✓ Traditional encryption algorithms, either symmetric such as DES, CBC, CTR etc. or asymmetric, such as RSA etc. These algorithms provide data confidentiality.
- ✓ A digital signature algorithm that is based upon public key algorithms, such as DSA for ensuring source authentication
- ✓ An algorithm for generating public keys or private keys such as Pseudorandom Number Generation
- ✓ A mechanism for storing these keys and distributing them to relevant parties.
- ✓ A hashing algorithm for maintaining message integrity while mail transfers from one end to another over the network.

IV. RELATED WORK

Though after a long journey since e-mails are in use, many algorithms have been designed to give more and more security to e-mails, there is not any of the single secure algorithm or solution. But still many organizations, like SecretAgent, HyperSend, Popmail, Privacy X etc. including the big giants like Microsoft and Netscape, are working on developing a secure solution for e-mails.

Presently S/MIME is an extensively used standard which provides end-to-end security services such as source authentication, non-repudiation, message integrity and data confidentiality by defining a data encapsulation format. Although S/MIME does not secure header part of the e-mail as a whole, yet S/MIME Certificate Handling specification [RFC5750] provides a few methods for some partial header protection [5]. For example, The receiver must check the sender's information, in the FROM part of e-mail, to see if it is a valid internet mail address and if it is present in the provided certificate. This helps in achieving integrity of FROM header value.

DKIM [6] is another method for this. This is a domain level Authentication framework for e-mails. It usually checks for integrity and source authentication on message header for a domain actor, that is the SMTP service or its equivalent.

V. PROPOSED SOLUTION

The most suitable technique for securing e-mail headers is to use signed attributes. The one possible and easy applicable alternative is to simply make a structure of all e-mail headers (including sender, receiver, date, tie, subject etc.) and give them a feature of encapsulation. The structure is secured with digital signature, thus all encapsulated header fields will get a common digital signature as part of structure. After this process, calculate message digest on the encrypted encapsulated header part of the e-mail. The message digest

process is a process of calculating hash, which will work on contents together with signature.

Verification of digital signature and digest value is done at the receiver's end. At the end of the verification process, the secure header fields and respective message header fields are compared together. If they get matched, then it is assumed that signature is valid and header integrity is ensured. Otherwise, signature is considered to be invalid one and the e-mail with such signature is discarded at the receiver's end.

The secured e-mail headers help providing various aspects of security services such as non-repudiation, data confidentiality, and message integrity.

The signature of this encapsulated message can be further encrypted either by sender or by any other trusted third party. At the receiver's end, either the receiver or the same trusted third party decrypts the message, verifies the signature and at last fetches the e-mail header contents from the encapsulated part [7].

The process at sender's end can be expressed as shown in Figure 1:

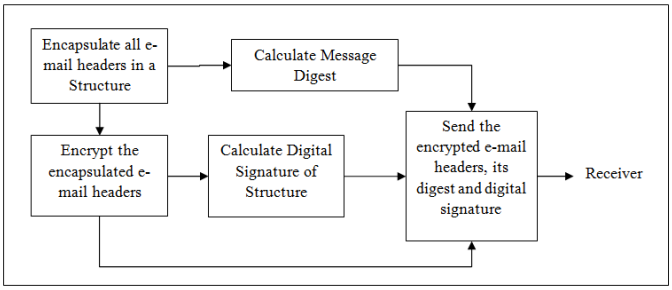

Figure 1: An overview of steps at sender's end

The Process at receiver's end will be reverse of that of sender's end and can be expressed as shown in Figure 2:

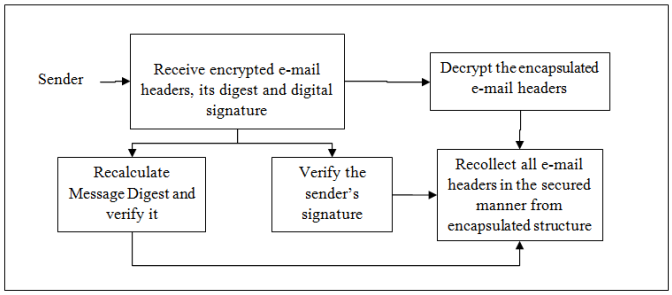

Figure 2: An overview of steps at receiver's end

Calculation of message digest over encrypted form of encapsulated e-mail headers can be done using keyed hash function. This proposed hash function makes use of a secret key which is shared between sender and receiver for the purpose of hashing only. A trusted public key manager can be held responsible for managing distribution of this secret key between sender and receiver. The keyed hash function works as follows:

The hash function takes a block of 512 bits for processing as input, thus all encapsulated header parts are broken into blocks of 512 bits. A 64 bit key is used throughout the processing for each block. A 128 bit IV (Initialization Value) is used for first block. The output of processing each block is of 128 bit. This 128 bit output is further divided into two parts: left 64 bits and right 64 bits. Both left and right parts are then treated with a 64 bit key for encryption. This 64 bit key contains 8 parity bits (1 bit for each byte). The encryption consists of 16 rounds of operation and each round works with 48 bit key from the original 56 bit key (64 bit – 8 parity bit = 56 bit effective key length). After processing both left and right parts for encryption the 64 bit outputs are again combined to form a 128 bit final encrypted output. This output is accepted as CV (Current Value) and used for processing for next block of 512 bits [8]. The final output of last block is taken as digest for the encapsulated headers of the e-mail, and transmitted along with the digital signature and encrypted headers.

This whole process can be shown as Figure 3:

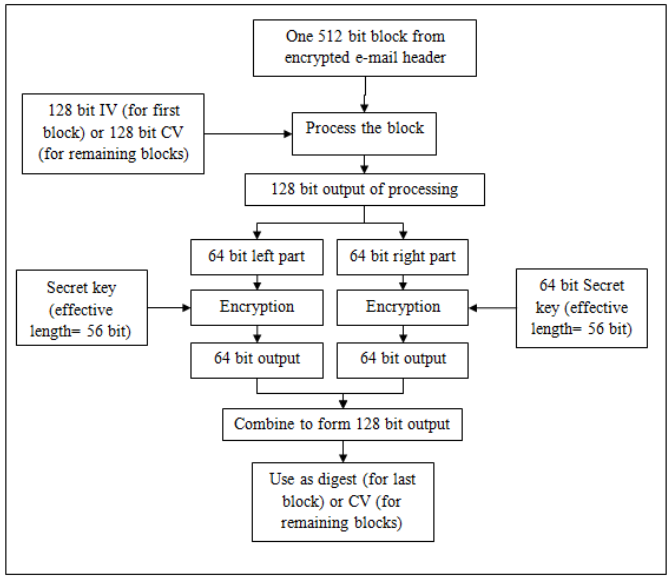

Figure 3: Processing steps of calculating keyed hash function on encrypted e-mail headers

At the receiving end, the receiver again performs same steps and calculated the digest before decryption. If the calculated digest and received digest value are same, then such a condition indicates that header values are not modified in the path by any other party. But if these two values are different then it indicates that any one ore values from header parts have been modified after it has been sent by sender. It ensures loss of header integrity, and such an e-mail should be discarded by receiver before any further processing on it.

VI. CONCLUSION

E-mail, being an important tool of communication in today's era, its security is also an important requirement. Not only the contents of e-mail, but the header of the mail also needs to be kept secure from intruders and malicious attackers. The paper discusses required security functions and e-mail headers in brief. Further few existing e-mail header security protocols such as S/MIME and DKIM are also discussed. Following this, a simple concept of securing e-mail headers is given which uses encryption, calculating digest and digital signature for e-mail headers altogether considering all header contents in encapsulated fashion. This provides integrity, source authentication and data confidentiality to sender and receiver for the e-mail headers.

REFERENCES

[1] Marwan Al Zarouni. (2004). Tracing E-mail Headers. *Proceedings of Australian Computer, Network & Information Forensics Conference on 25th November, 2004, School of Computer and Information Science,* Edith Cowan University Western Australia, pp. 16-30.

[2] Banday, M. T. (2011). Design and Development of Efficient Techniques for Securing E-mail System from threats. *PhD Thesis,* University of Kashmir, India.

[3] M. Tariq Banday. (2011). Analysing E-Mail Headers for Forensic Investigation. *Journal of Digital Forensics, Security and Law*, Vol. 6(2), pp- 49-64.

[4] z. Guo, L. Xu. (2015). Research of security structure model for web application systems based on the relational database. *International Journal of Security and Networks*. Vol. 10(4), pp- 207-213.

[5] L. Cailleux, C. Bonatti. (2015). Securing Header Fields with S/MIME. *Independent Experimental Submission*. Request for Comments: 7508. ISSN: 2070-1721.

[6] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed. (2011). Domain Keys Identified Mail (DKIM) Signatures. STD 76, RFC 637. Available at: http://www.rfc-editor.org/info/rfc6376

[7] J. Katz, M. Yung. (2006). Characterization of Security Notions for Probabilistic Private-Key Encryption. *Journal of Ctyptology*, vol 19(1), pp- 67- 96.

[8] [8] R. Purohit, U. Mishra, A. Bansal. (2013). Design and Analysis of a New Hash Algorithm with Key Integration. *International Journal of Computer Applications*. 81(1):33-38. Published by Foundation of Computer Science, New York, USA. DOI: 10.5120/13978-1974.