Digital Signature Extension to Authenticate Mail Receivers

Somnuk Puangproonpitag and Wilailak Kaprasit
Faculty of Informatics, Mahasarakham University
E-mail: {somnuk.p, ka.wilailak}@msu.ac.th

Abstract - Digital signature is a very useful technique to authenticate sender and content of an e-mail. We have pointed out (in our previous work) that receiver authentication cannot be supported by the traditional digital signature software. Yet, in some cases, the receiver authentication is very crucial for a non-repudiation purpose. This paper proposes an extended module for digital signature by adding Receiver Authenticity Tag (RAT) for receivers, who are sent to, cc and bcc. In addition, this work has modified Gnu Privacy Guard (GPG) software as a prototyped program to demonstrate an effectiveness of the extended design.

Keywords: Digital Signature Extension, Receiver Authentication, Mail Security, Non-repudiation

I. INTRODUCTION

Digital signature is a technique to authenticate e-mails. It is very useful to ensure the authenticity of sender and content of an e-mail. Quite a few programs (such as Gnu Privacy Guard (GnuPG) [4], Pretty Good Privacy (PGP) [6]) can help sign and verify e-mails using digital signature. Several people believe that using digital signature can also enforce “non-repudiation”. So, senders cannot reject the responsibilities to the e-mails that they have sent with their digital signature.

However, our previous work in [5] have pointed out several situations that the digital signature may not be helpful for non-repudiation. The key problem is that the traditional digital signature can not authenticate receivers. The digital signature can only confirm sender and content authenticity. As a result, in this paper, we propose a new design of Receiver Authenticity Tag (RAT) as an extension to the traditional digital signature. This design covers all receivers from three fields (To, Cc and Bcc). We have also implemented a prototyped program according to our design. Moreover, several experiments have done to test and compare our program with the traditional digital signature programs (such as PGP, GPG). The results show that our design and program outperform the old design of digital signature. They can solve the problem of fake receivers. By integrating our extension into digital signature software, it would enhance mail authentication efficiently.

The remainder of this paper is organized as follows. We begin in Section II by discussion of background and related work. In Section III, we describe the rationale of our new design. The framework and algorithms of the new design are proposed in Section IV. Section V shows the experimental results in testing our new design. In Section VI, we conclude and discuss the future work.

II. BACKGROUND AND RELATED WORK

A. Digital Signature and Mail Authentication

Digital signature is a well-known technique to authenticate mail. It is firstly proposed in [1]. To create a digital signature, a variable-length message is first hashed using a message digesting algorithm to be a fixed-size output. The output is then encrypted using the sender’s private key to generate the digital signature tag. After that, the signature tag is appended to the message. This digital signature can only be created by the owner of the private key, but can be verified by anyone. To verify the signature, it is decrypted using the sender’s public key and compared with the hash of the received message. If they match, then the message and the sender are authentic. A lot of people understand that sender and content authentication of mail would be enough to non-repudiation purpose. However, we point out a case (in section III) that may need an extension of digital signature to authenticate also receivers.

B. OpenPGP and GPG

OpenPGP [2] is an IETF specification (in RFC 2440) for a standard, completely open PGP. It uses a combination of strong asymmetric and symmetric cryptographies for electronic communications and data storage. It provides authentication and confidentiality services for e-mail and file storage applications.

GnuPG or GPG [4] is open-source software according to OpenPGP format. It is used to sign and verify digital signature. In this work, we modify GPG in order to support our extension module of the standard OpenPGP digital signature.

C. Receiver Authentication Problems

In [5], we have pointed out that receiver authenticity is a crucial thing that missing from the digital signature design. Without the receiver
authenticity, non-repudiation may not be able to be enforced. The detailed experiment can be found in [5]. We have also proposed an initial design to fix the digital signature system using Receiver Authenticity Tag (RAT). However, the design covered only receivers in fields “To” and “Cc” (not bcc). In this paper, we propose a new design with better details. The new design also covers an authentication mechanism for receivers who are “Bcc”.

III. RATIONALES OF THE NEW DESIGN

An e-mail can be forged even with a digital signature.

Jack has received an e-mail with a digital signature from Wilailak. Due to the digital signature attached at the end of mail, he can verify that the e-mail is sent from Wilailak, and the content is exactly what Wilailak sent. Several people think that Wilailak would not be able to non-repudiate the responsibilities to this e-mail.

However, we argue that the non-repudiation by such as a digital signature may not be claimed in some cases. For example, Jack can forge an e-mail with exactly the same content and then send to Somnuk.

As shown in Figure 2, steps to forge the e-mail are as follows:

1. Open a command shell.
   Start | Run cmd
2. Telnet to the mail server on port 25.
   C:> telnet mail.msu.ac.th 25
3. We have to identify by saying HELO.
   HELO
4. Enter the spoofed sender and the recipient of the e-mail.
   MAIL FROM: ka.wilailak@msu.ac.th
   RCPT TO: somnuk.p@msu.ac.th
5. Use the DATA command to send the message.
   Subject: Credit Card
   After that, copy and paste the message and digital signature tag from Wilailak into the content.
6. Enter a period on a separate line to send the email and “QUIT” to terminate telnet.
   QUIT

The above is only an example of forging mail. Whether or not we can forge e-mail in any mail system, it is actually up to how secure mail server has been setup.

After that, Somnuk tries to check an e-mail, as shown in Figure 1. He finds that an e-mail in Figure 1 has been sent by Wilailak. Especially, after verifying the digital signature using PGPfreeware 8.0.3, the e-mail is verified as a good mail (as shown in Figure 3). So, he misunderstand that Wilailak has sent him the forged e-mail.

This case has shown that the digital signature can only authenticate the sender and the contents of an e-mail but not the receiver. In several cases as discussed in [5], the contents may not need to be modified and mail forgery can still be effective to trick Somnuk (the victim in this case). The digital signature would not be able to protect this problem. Moreover, due to the lack of receiver authentication, Wilailak (the sender) would be able to repudiate sending any e-mail to any receivers (even having signed her digital signature on the e-mail). So, the digital signature would not be a valid proof of receiving any mail from any one. Hence, in this paper, we design an extension of the digital signature in order to authenticate receiver.

IV. FRAMEWORK AND ALGORITHMS

The solution is adding “Receiver Authenticity Tag (RAT)” into traditional digital signature concept. In this work, we extend our initial design, proposed in
by separate RAT into two parts. One is for receivers, who are “To” or “Cc”. The other one is for receivers who are “Bcc”.

To and Cc: As shown in Figure 4, the “To” or “Cc” RAT is created by converting an e-mail address into all capital characters, then hashing it using the MD5 algorithm [7], finally converting it into 20 decimal RAT.

Bcc: The “Bcc” RAT must be created with a more complex technique. As “blind” carbon copy, it is needed to ensure that the “Bcc” receivers are not revealed due to our RAT. Yet, the design would allow receivers to authenticate themselves as validated receivers.

As shown in Figure 5, the steps to create Bcc RAT are as follows: (1) converting an Bcc e-mail into all capital characters (2) hashing it by MD5 (3) converting the results into 20 decimals (4) encrypting it by Advanced Encryption Standard (AES) [3] using a random session key (5) converting session key to decimals and encrypting the session key by public key of the Bcc receiver (6) concatenating the results of the 4th and the 5th steps to be a RAT.

An Extension of Digital Signature: After getting RAT from the previous steps, the steps of generating digital signature are modified as shown in Figure 6. The key extension step is adding RAT into the message before processing other digital signature steps.

The verifying steps of digital signature are modified by adding RAT verification process before the old first step. For verifying RAT in cases of “To” and “Cc”, the receiver e-mail address is hashed and converted into 10 decimals; then searched out from the RAT to ensure receiver authenticity. In case of “Bcc”, the Bcc RAT must be decrypted using private key of the receiver before verifying RAT. The other steps of digital signature verification after that are kept the same one.

V. EXPERIMENTAL RESULTS

In order to experiment on our new design, we have modified GPG software and implemented prototyped software with GUI (shown in Figure 7-10) using JAVA. Our new digital signature supported software is called ISAN-RAT-OpenPGP version 1.1. While the prototyped software is OpenPGP compatible, it includes a RAT extension to the traditional digital signature software. We then experiment to test whether we can protect the problem in Section III by using our software. We also compare our software with PGP software. The results are summarized in Table 1.

As shown in Figures 7 and 8, when we test the software in cases of genuine and bogus receivers, the results are shown that our software can verify not only the sender and the content authenticities, but also the receiver authenticity.
VI. CONCLUSIONS AND FUTURE WORK

To summarize, this work has point out the problem of receiver authentication in e-mail system with a digital signature. Hence, an extension of the digital signature has been proposed to authenticate mail receivers (including To, Cc, Bcc receivers). A program has been implemented according to our new design using an open-source GnuPG and JAVA language. After experimenting on the software in comparison with PGFreeware, the results demonstrate that our design and program outperform the old digital signature software in terms of receiver authentication. In the future, we plan to distribute this software to public in order to receive feedbacks from the number of real users.

VII. REFERENCES