# A Content Certified E-mail Protocol with a Public Mailbox

Tak-Ming Law

Hong Kong Institute of Vocational Education (Morrison Hill)
Department of Computing, 6 Oi Kwan Road, Wan Chai, Hong Kong.
Email: tmlaw@vtc.edu.hk

## Abstract

This short note presents some ideas of the *content* certified e-mail protocol with a public *mailbox*. By applying the Diffie-Hellman key exchange style to reflect the *agreement* between the sender and receiver.

The notion of certified e-mail just similar to the certification procedures for regular mails in the real-life post offices. Yet the post office only certifies whether the mail has been sent or received by the appropriate parties, but not its *content*. This insufficient verification in paper authentication protocols can be easily solved by digital signature schemes.

A certified e-mail protocol must have the following security properties:

1. The sender must have some way of proving that the receiver received the mail, should the receiver later try to deny it.
2. The receiver must have some way of proving that the sender did not send the mail, should the sender later try to claim that she did.

## 1 Introduction

E-mail almost involved in everyone's daily communications in the current and coming century. The advantages of communicate through *direct-written-notes* by the senders (which cannot be performed by some other communication tools such as long distant or local telephone call, pagers and etc) make e-mail communications so popular along decades. Recently, e-mail communications even utilized in wireless mobile communication devices, such as mobile phone or notepads. Due to the above reasons, a *secure* e-mail communication becomes a hot research area. One specific topic in that area is to implement a certification service in e-mail system while sending and receiving messages. The notion of certified e-mail just similar to the certification procedures for regular mails in the real-life post offices. Yet the post office only certifies whether the mail has been sent or received by the appropriate parties, but not its *content*. To terminate this problem in electronic communications, this short note presents some ideas of the *content* certified e-mail protocol with a public *mailbox*. The rest of this paper will be organized as follows. In section 1.1, preliminary concepts and objectives of certified email is described. Our protocol is presented in section 2. Section 3 analysis the sender-receiver protection issues of the protocol we have outlined in section 2. Section 4 discusses and lists the security characteristics of our protocol; and finally the related works and discussions as well as future developments are included in section 5; in here, we briefly compare the previous work that its objective is closely similar to ours.

### 1.1 Preliminaries

In order for e-mail to be used for important communications, some notion of certified delivery must be provided for users. A certified e-mail protocol must have the following security properties:

3. Alice (the sender) must have some way of proving that Bob (the receiver) received the mail, should Bob later try to deny it.
4. Bob must have some way of proving that Alice did not send the mail, should Alice later try to claim that she did.

Certified paper mail uses the notion of a signed receipt. When Alice sends Bob certified mail, the Post Office will not release the mail to Bob unless he signs a receipt. This signed receipt is returned to Alice, and acts as a proof of delivery. If Alice does not have this receipt, Bob can claim that the certified mail was never sent.

Of course, this protocol only certifies that Alice sent Bob some *piece* of mail but not the content. This insufficient verification in paper authentication protocols, on the other hand, can be easily solved by digital signature protocols.

The notion of certified e-mail is previously proposed in [1], however, we have an alternative way by using the notion of *public mailbox* and to further eliminate the risks that:

(1) Bob claims that he did not collect the symmetric key, sent by Alice, in the public *mailbox* on time; and,

(2) The symmetric decryption keys being exposed on the public *mailbox* so that everyone can obtain the key.

## 2 The Protocol

Alice wishes to send Bob a certified message. Bob wants to receive a certified message. A protocol must be built to facilitate this exchange. The protocol is designed to allow Alice to be able to prove to a judge that Bob has received her message if and only if he did receive it.

It is assumed that:

(1) Bob has a public key [2,3] in some commonly recognized format and there exists some public-key infrastructure.

(2) Both Bob and Alice can use the public keys to verify and decrypt the message from the other side.

(3) The judge can verify the key was valid at the time of the transaction using the public key infrastructure.

(4) There exists a public *mailbox*, which is maintained by a trusted agent for a group (membership).

Let ***M*** be the message and ***H*** a message digest, or hash function [4,5].

**Step 1**. A →B ***{notice, H(M)}pub-***$K_B$

Alice sends a notice (challenge) to Bob and stated that " I am going to send Bob a message on *date* and *time* at the public mailbox **P** and the hash of the message is ***H(M)***. --/s/ Alice". The notice accompanies with the hash ***H(M)***, digital signature [6] of Alice and encrypted by the public key of Bob, ***pub-***$K_B$.

**Step 2**. B →A ***{c, H(c)}pub-***$K_A$

The ***c*** denotes confirmation from Bob to Alice and stated that " I have received Alice's notice and be ready to collect the message on *date* and *time* in **P** || Nonce. --/s/ Bob" and concatenated, denoted as ||, with the Nonce. The confirmation accompanies with a hash of ***c***, ***H(c)***, digital signature [6] of Bob and encrypted by the public key of Alice, ***pub-***$K_A$.

**Step 3**. A →**P** $E_{H(H(M)\oplus H(c))}(M)$

Alice sends the message ***M*** encrypted by the symmetric key ***H(H(M)⊕H(c))⊕H(n)*** with a symmetric cipher [7] to the public *mailbox* **P** on the specific *date* and *time*, where ⊕ denotes bit-wise exclusive-or. Bob collects the message from **P** and decrypt it by the key ***H(H(M)⊕H(c))⊕H(n)***, which is composed of exclusive-oring and hashing the ***H(M)*** (which he received in step 1 from Alice) and ***H(c)*** (which he makes up by his own in step 2), then exclusive-or the hash of the notice, ***H(n)***, (which is sent by Alice in step 1).

## 3 Analysis

The protocol is secure against cheating attempts by either Alice or Bob.

- **Protection for Bob**: (1) After Bob decrypt the message ***M*** in step 3, he can perform his own hash on ***M*** and compare it with the ***H(M)*** received in step 1 to verify whether the content is the same as the one sent by Alice from the beginning. (2) The public *mailbox* **P** can testify that the e-mail has been received on specific *date* and *time*. (3) The email is sender *self-authenticated*. That is, after Bob has successfully decrypted the e-mail by using the key ***H(H(M)⊕H(c))⊕H(n)***, he will know that the email is definitely sent by Alice. No one has the knowledge of ***H(M)***, ***H(c)*** and ***H(n)*** besides Alice and Bob. The symmetric key itself becomes the proof of the *agreement* in sending and receiving of e-mail between Alice and Bob.
- **Protection for Alice**: It is necessary for Alice to have received ***H(c)*** from Bob before she can encrypt the message ***M***. The ***H(c)*** becomes *evidence* that Bob agrees to collect the message in the public *mailbox* **P** on the specific *date* and *time* so that Bob turns out no right to deny this confirmation whether he has or he has not collected the message. In case that Bob forgot to

collect e-mail from **P** that turns into Bob's own fault but not the liability of Alice.

## 4 Security Characteristics

- Both the notice from Alice in step 1 and confirmation from Bob in step 2 are *digital signed* respectively so that no one can disavow their authority.
- The symmetric key ***H(H(M)⊕H(c))⊕H(n)*** comes over an evidence of the *agreement* between Alice and Bob for the previous notice and confirmation. It even authenticates the message content by the key element ***H(M)***.
- Unlike the protocol in [1], privacy of e-mail content is considered and the symmetric key will not be opened to the public; and only Alice and Bob have the knowledge of it.
- The concatenation of nonce in the confirmation in step 2 improves the security against *dictionary attacks*.
- The public *mailbox* **P** only opens to group members that breaks the admission of adversaries.

## 5 Related works and discussions

The goals and objectives proposed in [1] is very similar to ours. In [1], Schneier puts his protocol to send the encrypted message to the receiver in the first step, then publish the symmetric decryption key after received the confirmation from the designated receiver. Compared to his approach, our protocol conceals the symmetric decryption key in a Diffie-Hellman [8] style and which reflects a sense of *agreement* between Alice's notice and Bob's confirmation.

One can consider to integrate another practical payment methods (like NETBILL [9], VARIETYCASH [10], CYBERCASH [11] and DIGICASH [12]) to implement the concept of this protocol. Similarly, this protocol could also be used with fair-exchange protocol [13,14] to improve the security while sending and receiving of messages.

For future developments, The notion of "sending information to the future" [15,16] can be implemented in this protocol. It enables the sender to send a encrypted message to the receiver in present but constraint it to be revealed at a specific time in the future.

## Reference

[1] B. Schneier, "A Certified E-Mail Protocol with No Trusted Third Party",

[2] R. Rivest, A. shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of ACM, v. 21, Feb 1978, pp. 120-126.

[3] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, V. IT-31, n. 4, 1985, pp. 469-472.

[4] National Institute of Standards and Technology, NIST FIPS PUV180, "Secure Hash Standard," U.S. Patent 5,509,071, 16 Apr 1996.

[5] Ross Anderson, Eli Biham, Tiger: A Fast New Hash Function, available from author:biham@cs.technion.ac.il.

[6] National Institute of Standards and Technology. FIPS 186: Federal Information Processing Standard; Digital Signature Standard (DSS). May 1994.

[7] R.L. Rivest, "The RC5 encryption Algorithm," Fast Software Encryption, 2nd International Workshop Proceedings, Springer-Verlag, 1995, pp. 86-96.

[8] W. Diffie and M. Hellman, "New Directions in Cryptography,", IEEE Trans. on Info. Theory Vol. IT-22(6) pp. 644-654 (Nov. 1976).

[9] Marvin Sirbu and J.D. Tygar. NetBill: An Internet Commerce System Optimized for Network Delivered Services. *In **IEEE** Personal Communications*, pages 6-11, August 1995.

[10] M. Bellare, J. Garay, C. Jutla and M. Yung. VarietyCash: a Multi-purpose Electronic Payment System. Proceedings of the 3rd Usenix Workshop on Electronic Commerce, 1998.

[11] CYBERCASH. The CyberCash™ System – How it Works. <http://www.cybercash.com/cybercash/cyber2.html>.

[12] DIGICASH. About ecash. <http://www.digicash.com/ecash/ecash-

home.html>.

[13] S. Ketchpel, Transaction protection for information buyers and sellers, in Proceedings of the Dartmouth Institute for Advanced Graduate Studies '95, 1995.

[14] S. Ketchpel and H. Garcia-Molina, Making trust explicit in distributed commerce transactions, Stanford Digital Library Project Working Paper SIDL-WP-1995-0018, October 12, 1995.

[15] Ronald L. Rivest, Adi Shamir, David A. Wagner, Time-lock puzzles and timed-release Crypto, 1996, Available from author:http://theory.lcs.mit.edu/~rivest/publications.html.

[16] Mihir Bellare and Shafi Goldwasser, Encapsulated key escrow, MIT Laboratory for Computer Science Technical Report 688, April 1996.