On the Security of a Certified E-Mail Scheme with Temporal Authentication

Min-Hua Shao[1,*], Jianying Zhou[2], and Guilin Wang[2]

[1] Institute of Information Management, National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu 300, Taiwan
mhshao@alumni.nccu.edu.tw
[2] Infocomm Security Department, Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{jyzhou, glwang}@i2r.a-star.edu.sg
http://www.i2r.a-star.edu.sg/icsd/

Abstract. Certified e-mail is a value-added service for standard e-mail systems, in which the intended recipient gets the mail content if and only if the mail originator receives a non-repudiation evidence that the message has been received by the recipient. As far as security is concerned, fairness is one of the most important requirements. Recently, Galdi and Giordano (2004) presented an optimistic protocol for certified e-mail with temporal authentication. In this paper, we analyze their protocol and demonstrate that it cannot achieve true fairness and has some other weaknesses. We further propose the improvements to avoid those security problems.

1 Introduction

The lack of evidence for message receipt is a missing piece of the infrastructure required for the more professional use of email [14]. Certified e-mail uses the notion of a signed receipt and strengthens the binding between the evidence and the mail being certified. In other words, the main purpose of a certified e-mail scheme is to achieve the fair exchange of a message and a receipt in the sense that either the sender obtains a receipt from the receiver and the receiver accesses the content of the e-mail simultaneously, or neither party gets the expected item. Although fairness is probably the most important one, there are other properties on the application of certified e-mail. The following security properties are defined in [12,13] and extended in [7].

- *Fairness*: The protocol should be *fair* in the sense that either each party receives the expected item or neither party receives any useful information about the other's item.
- *Non-repudiation*: Neither the sender nor the receiver of a message is able to deny the transmission.

* The author's work was done during her visit to Institute for Infocomm Research, Singapore, and funded by the National Science Council of Taiwan under the contract of NSC 93-2917-I-009-001.

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3482, pp. 701 – 710, 2005.

- *Timeliness*: Both the sender and the receiver of a message should have the ability to reach the end of a protocol run in a finite amount of time unilaterally without losing fairness.
- *Authenticity*: The players should be guaranteed of their reciprocal identity.
- *Confidentiality*: None but the intended parties can get access to the plaintext items sent during the protocol.
- *Integrity*: Message transmission should be protected against unauthorized operations in order to guaranty the correctness and authenticity of data.
- *Temporal Authentication*: The sender can obtain the evidence to prove the time at which the message was sent.
- *Sending Receipt*: The sender can obtain the evidence to prove that he/she started the process of sending a certified e-mail.

Certified e-mail has been discussed for years, and there are two major classes of schemes to address the certified mail problem: schemes that require the existence of a trusted third party (TTP), and schemes that don't require the existence of a TTP. Oppliger showed clearly that the second class, i.e., either based on a simultaneous secret exchange or trusted system is inappropriate to provide certified mail services for the Internet [14]. Therefore, the use of TTPs seems advantageous and various types of TTPs can be considered according to their involvement in the certified e-mail protocol: schemes with *in-line* TTPs [3], schemes with *on-line* TTPs [1,5,14] and schemes with *off-line* TTPs [2,6,9].

An in-line TTP, i.e. acting as a delivery authority, involves in each message's transmission during the protocol. The main advantage of in-line TTPs for certified mail is to ensure strong fairness since the TTP collects all information necessary before forwarding them to the concerned entities; and further, the in-line TTP has full control over the message flows and likely provides the sender anonymity services. However, it also implies a communication and computation bottleneck due to the heavy involvement of the TTP.

An improvement to reduce the TTP's involvement is the use of an on-line TTP. The on-line TTP is actively involved during each session of the certified e-mail protocol but not during each message's transmission. Its task may only deal with signaling information, such as cryptographic keys and/or receipts sent back to the originator [14]. In academic literature, there is often an emphasis on reducing the role and the expense of a TTP. Protocols with a light-weight TTP have been proposed. For example, Abadi et al. proposed an efficient certified e-mail scheme with a light on-line TTP [1]. A key feature of their scheme is not to deploy any public-key infrastructure; and further, Imamoto and Sakurai [8] revised their scheme in order to provide the non-repudiation of origin service.

A big step towards more efficient solutions was the introduction of off-line TTPs. That is, an off-line TTP involves in a protocol only in case of an incorrect behavior of a dishonest entity (for example, the recipient claims having not received the message or the originator claims having not received the receipt), or in case of a network error. Considering most of the time no problem will occur, this approach using an off-line TTP is also called the *optimistic* approach.

Galdi and Giordano proposed an improved optimistic protocol for certified e-mail at TrustBus 2004 [7]. Their effort is to introduce a feature of "temporal authentica-

tion" into certified e-mail along a four-message optimistic protocol. Galdi-Giordano's certified e-mail scheme (GG scheme, for short) is effective against misbehavior of one of the players in some cases. However, we demonstrate in this paper that it suffers from a few severe security problems, and some of the security properties mentioned above cannot be satisfied. For example, the receiver can get the e-mail content by *replay attacks* even though he/she did not give the sender a receipt of the message. In this paper, we give a thorough security analysis of the GG scheme and further propose the improvements to avoid these problems.

The rest of this paper is organized as follows. We introduce the notation in Section 2, and briefly review the GG scheme in Section 3. We point out the vulnerabilities and propose solutions in Section 4. We end the paper with conclusions in Section 5.

2 Notation

In this paper, we use the same notation used in the original paper [7]. For completeness and readability, we summarize the model and all cryptographic symbols below.

–*Alice*, *Bob*, *Ted*, *Sam*: four different participating entities in which Alice is the message sender, Bob is the message receiver, Ted acts as an off-line *Trusted Third Party* (TTP), and Sam plays as an online *Trusted Stamping Server* (TSS).
–m_{subj}: the message subject associated with the message m.
–$PK_X(m)$: the encryption of the message m using the public key of the player X, where $X \in \{\mathrm{A}(lice),\mathrm{B}(ob),\mathrm{T}(ed),\mathrm{S}(am)\}$.
–$Sig_X(m)$: the signature of player X on message m.
– $\overline{PK}_x(m,r)$: the encryption of the message m, obtained by using the public key of the Player X and random string r.
–$X \rightarrow Y$: m: player X sends the message m to player Y.
–$x||y$: the concatenation of strings x and y.
–$h(\cdot)$: a collision resistant one-way hash function.

3 The GG Scheme

We first sketch the GG scheme proposed by Galdi and Giordano in [7]. In this scheme, the *basic protocol* is the core of the certified e-mail scheme that ensures timelines and message verifiability. It consists of three messages exchanged between the sender Alice and the receiver Bob in the normal situation. The *extension of the basic protocol* is provided that introduces an on-line time stamping server and add a single message due to the temporal authentication. In addition, the *recovery procedures* are launched in the abnormal situation to achieve fairness for all participants under the help of the TTP's involvement. We now review these three protocols in more detail below.

(1) The basic protocol. Assume that the sender Alice wants to deliver a message m to the receiver Bob with a guarantee that Bob can access the message m if and only if

Alice obtains a receipt from Bob. To this end, they run the following three-message optimistic protocol for fair exchange of certified e-mail.

(b1). Alice → Bob: m_1

where $m_1 = \langle env, Sig_A(env) \rangle$,

$env = \langle ID_A, ID_B, PK_B(\langle m_{subj}, h(\langle m, r \rangle) \rangle), \overline{PK}_T(PK_B(msg), r) \rangle$, $msg = \langle m_{subj}, m \rangle$

(b2). Bob → Alice: m_2

where $m_2 = Sig_B(m_1)$

(b3). Alice → Bob: m_3

where $m_3 = \langle \langle PK_B(msg), r \rangle, Sig_A(\langle PK_B(msg), r \rangle) \rangle$

Clearly, the key idea in the basic protocol is to use an electronic envelope to lock the message. In the first message flow (b1), m_1 is composed of two parts besides the players' identities. The hash value $h(\langle m, r \rangle)$ in the first part will be used to verify that the message received corresponds to the one for which the receipt has been sent. The second part that is a cipher of the actual content of the email by Ted's public key for recovery procedures. In the second message flow (b2), m_2 is the message receipt for *m* that ensures the non-repudiation of receipt for Alice. Finally, in the third message flow (b3), Bob must make sure that the receipt sent to Alice corresponds to the received message. That is, Bob needs to confirm the integrity between *m* in m_3 and *m* of $h(\langle m, r \rangle)$ in m_1. There are three items needed to verify: (1) m_{subj} received in the first message m_1 matches the one received in m_3; (2) the new hash value of $\langle m, r \rangle$ retrieved in m_3 is in correspondence with the one in m_1; (3) the ciphertext $\overline{PK}_T(PK_B(msg), r)$ drawn from m_1 and the new one produced from m_3 by using Ted's public key are the same.

(2) The extension protocol. Due to temporal authentication the time stamping server, Sam, is involved in the protocol to provide a time certification of the message *m*, notated as *t(m)*. More specifically, Sam sends a copy of the message m_2 to Alice during the second message flow (e2) in order to obtain a sender's receipt.

(e1). Alice → Sam: m_1

where $m_1 = \langle env, Sig_A(env) \rangle$,

$env = \langle ID_A, ID_B, PK_B(\langle m_{subj}, h(\langle m, r \rangle) \rangle), \overline{PK}_T(PK_B(msg), r) \rangle$, $msg = \langle m_{subj}, m \rangle$

(e2). Sam → Bob and/or Alice: m_2

where $m_2 = \langle \langle m_1, t(m_1) \rangle, Sig_S(\langle m_1, t(m_1) \rangle) \rangle$

(e3). Bob → Alice: m_3

where $m_3 = \langle m_2, Sig_B(m_2) \rangle$

(e4). Alice → Bob: m_4

where $m_4 = \langle \langle PK_B(msg), r \rangle, Sig_A(\langle PK_B(msg), r \rangle) \rangle$

(3) The recovery procedures. The recovery procedures will be launched by one of the players once the other player P misbehaves on the message m_i, notated as P failed on m_i. There are two failures discussed in the GG scheme: Alice failed on m_3 and Bob failed on m_2. Regarding Alice's failures, i.e., by sending m_3 to Bob with wrong information or by not sending m_3 to Bob, Bob can obtain the message for which he issued a receipt by the involvement of the TTP. The detailed treatment of the recovery procedures is shown below:

(r1). Bob → Ted: m_1, m_2
(r2). If (ID_A in m_1 = Alice and ID_B in m_1 = Bob) and
(verify Alice's signature $Sig_A(env)$ and Bob's signature $Sig_B(m_1)$)
then Ted → Bob: $enc = PK_B(\langle m_{subj}, m\rangle)$
Ted → Alice: m_2

As for Bob's failures, Bob can only fail to reply m_2 to Alice. In this case the following recovery procedures (r3) and (r4) are used for the basic protocol; others are used for the extension protocol.

(r3). Alice → Ted: m_1
(r4). Ted → Bob: m_1
Ted's verification:
If (no response from Bob)
then Ted → Alice: $rej = \langle\langle REJ, m_1\rangle, Sig_T(\langle REJ, m_1\rangle)\rangle$
If (m_2 from Bob) then Ted → Alice: m_2

(r5). Alice → Sam: m_1
(r6). Sam → Ted: $\langle m_s, t(m_s)\rangle$
where $m_s = \langle m_1, t(m_1)\rangle$
(r7). Ted → Bob: m_s
Ted's verification:
If (no response from Bob)
then Ted → Alice: $rej = \langle\langle REJ, m_t\rangle, Sig_T(\langle REJ, m_t\rangle)\rangle$
If (m_2 from Bob) then Ted → Alice: m_2

4 Security Analysis of the GG Scheme

4.1 Vulnerabilities

V-1. Replay Attack
Replay attack is one of active attacks that an adversary records a communication session and replays the entire session, or a portion thereof, at some later point in time. Unfortunately, the sender in the GG scheme may suffer severely from a replay attack. That is, the receiver can collude with some party Cindy to obtain the content of the

message from the TTP without providing a valid receipt to the original sender. Our attack scenario on the basic protocol of the GG scheme is illustrated in Figure 1. The detailed treatment of replay attack on Figure 1 is described below.

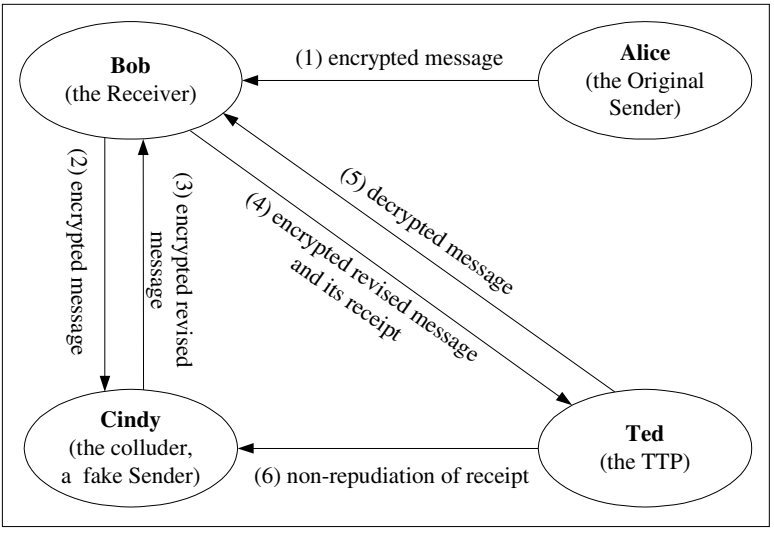

Fig. 1. Replay attack scenario on the basic protocol

(1). Alice → Bob: m_1

where $m_1 = \langle env, Sig_A(env) \rangle$,

$env = \langle ID_A, ID_B, PK_B(\langle m_{subj}, h(\langle m, r \rangle) \rangle), \overline{PK}_T(PK_B(msg), r) \rangle$, $msg = \langle m_{subj}, m \rangle$

(2-3). Bob colludes with Cindy. Cindy creates a revised version $\hat{m}_1$ of the message m_1 where ID_A is replaced with ID_C, and then generates her signature on the revised message env' in order to disguise as the sender.

Bob → Cindy: m_1

Cindy → Bob: $\hat{m}_1$

where $\hat{m}_1 = \langle env', \underline{Sig_C(env')} \rangle$,

$env' = \langle \underline{ID_C}, ID_B, PK_B(\langle m_{subj}, h(\langle m, r \rangle) \rangle), \overline{PK}_T(PK_B(msg), r) \rangle$

(4). Bob produces a false receipt $\hat{m}_2$ on the message $\hat{m}_1$ and then make a claim "Cindy failed on $\hat{m}_1$" to Ted.

Bob → Ted: $\hat{m}_1$, $\hat{m}_2$

Where $\hat{m}_2 = Sig_B(\hat{m}_1)$

(5-6). Ted follows the recovery procedure (r2). Bob will obtain the content of the message and Cindy, the conspirator, can get the receipt.

Ted → Bob: $enc = PK_B(\langle m_{subj}, m\rangle)$

Ted → Cindy: $\hat{m}_2$

Similarly, the same weakness towards replay attack is appeared in the extension protocol in the GG scheme. The attacking scenario is depicted in Figure 2 and the description of the figure is as follows.

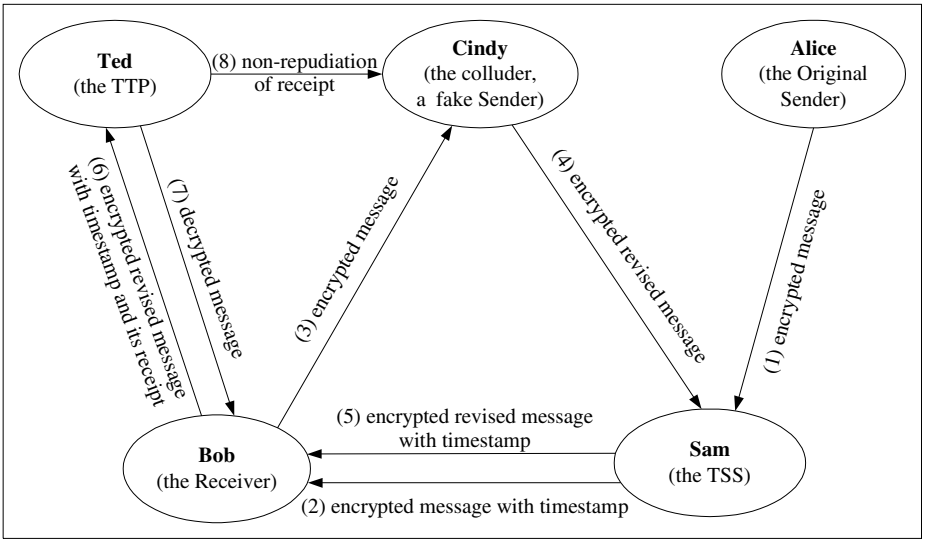

Fig. 2. Replay attack scenario on the extension protocol

(1). Alice → Sam: m_1

where $m_1 = \langle env, Sig_A(env)\rangle$,

$env = \langle ID_A, ID_B, PK_B(\langle m_{subj}, h(\langle m, r\rangle)\rangle), \overline{PK}_T(PK_B(msg), r)\rangle$, $msg = \langle m_{subj}, m\rangle$

(2). Sam → Bob: m_2

where $m_2 = \langle\langle m_1, t(m_1)\rangle, Sig_S(\langle m_1, t(m_1)\rangle)\rangle$

(3-4). Bob colludes with Cindy. Cindy creates a revised version $\hat{m}_1$ of the message m_1 in order to disguise as the sender.

Bob → Cindy: m_1

Cindy → Sam: $\hat{m}_1$

where $\hat{m}_1 = \langle env', \underline{Sig_C(env')}\rangle$,

$env' = \langle \underline{ID_C}, ID_B, PK_B(\langle m_{subj}, h(\langle m, r\rangle)\rangle), \overline{PK}_T(PK_B(msg), r)\rangle$, $msg = \langle m_{subj}, m\rangle$

(5). Sam → Bob: $\hat{m}_2$

where $\hat{m}_2 = \langle\langle \hat{m}_1, t(\hat{m}_1)\rangle, Sig_S(\langle \hat{m}_1, t(\hat{m}_1)\rangle)\rangle$

(6). Bob produces a false receipt $\hat{m}_3$ and then gives $\hat{m}_1$ and $\hat{m}_3$ to Ted for recovery.

Bob → Ted: $\hat{m}_1$, $\hat{m}_3$

where $\hat{m}_3 = \langle \hat{m}_2, Sig_B(\hat{m}_2)\rangle$

(7-8). Ted follows the recovery procedure (r2) after successful verification. Bob can get the content of the message without providing the valid receipt to Alice.

Ted → Bob: $enc = PK_B(\langle m_{subj}, m\rangle)$

Ted → Cindy: $\hat{m}_3$

The above replay attack demonstrates that fairness cannot be preserved in the GG protocol.

V-2. Incomplete Recovery Data on "Alice Failed on m_3"

According to the recovery procedures (r1 and r2) of the GG scheme, in case Alice fails on message m_3 Ted will compute the message *enc* from m_1 and send *enc* to Bob and m_2 to Alice after verifying the correctness of the message m_1 and m_2 provided by Bob. Here the problem is how to prove that the message *enc* is consistent with the non-repudiation receipt m_2. The receiver Bob in the GG scheme is designate to take the responsibility. However, it is beyond Bob's capability due to insufficient data. That is, Bob is short of the random string "*r*", and thus he cannot generate the hash of $\langle m, r\rangle$ from *enc* to compare it with the hash value $h(\langle m, r\rangle)$ from m_1.

4.2 Improvements

I-1. Protection Against Replay Attacks

A basic mechanism to prevent replay attacks is the *challenge-response* technique, in which, one entity (the claimant) proves its identity to another entity (the verifier) by demonstrating knowledge of a secret known to be associated with that entity [11]. This can be done by providing a response to a time-variant challenge that consists of three main classes of time-variant parameters: random numbers, sequence numbers, and timestamps. The weakness of the GG scheme against replay attacks is due to the inability of the TTP in detection of the real initiator of the message m_1 in the recovery procedures. Therefore, the identities of the involved parties (Alice, Bob, and TTP) and timestamp should be considered. The revised *env* in the message m_1 is $\langle ID_A, ID_B, ID_T, PK_B(\langle m_{subj}, h(\langle m, r\rangle)\rangle), \overline{PK_T}(\langle ID_A, ID_B, ID_T\rangle, t_d, PK_B(msg), r)\rangle$. Here, the identities of the involved parties $\langle ID_A, ID_B, ID_T\rangle$ will be effective against such an attack; and further, the timestamp t_d is used to provide the TTP with the deadline for dealing with the recovery procedures.

I-2. Provision of Complete Recovery Data on "Alice Failed on m_3"

The key to verify whether the delivered message corresponds to the non-repudiation of receipt m_2 and is also the promised one in m_1 is the hash value $h(\langle m, r\rangle)$. That

means, both *m* and *r* are required in verification. Therefore, a supplement to recovery data on "Alice failed on m_3" is the random string *r* contained in the recovery message *enc*. That is, Ted will compute the message $enc = \langle\langle PK_B(msg), r\rangle, Sig_T(\langle PK_B(msg), r\rangle)\rangle$ from m_1 and send *enc* to Bob and m_2 to Alice after the correct verification. Then, Bob will be able to verify that the receipt sent to Alice corresponds to the received message.

I-3. Specification of Encryption Algorithms

In the GG scheme, there is no clear specification on the public encryption algorithms PK_B and PK_T, except the authors stressed that PK_T is required to be a randomized encryption algorithm. So it seems the GG scheme works well with (a) any secure randomized encryption algorithm PK_T and (b) any secure encryption algorithm PK_B. However, this is not the fact. First, the random number *r* is needed to check $h(\langle m, r\rangle)$. In order to guarantee the TTP can recover *r*, it is required that from the ciphertext $\overline{PK}_T(PK_B(msg), r)$, the TTP can recover not only the message *m* but also *r*. This requirement is satisfied by the OAEP series of encryption schemes [4], but not by the Cramer-Shoup cryptosystem and the ElGamal encryption scheme. Similarly, we also need to assume PK_B is a deterministic encryption algorithm or a randomized encryption algorithm with the above mentioned property. If this is not true, a verifier (e.g., a judge) cannot verify the non-repudiation evidences.

5 Conclusion

The binding between the irrefutable *evidence* and the electronic mail being delivered is the purpose of certified email. The evidence will be a proof-of-delivery that the message was delivered to the recipient. A desirable requirement for a certified e-mail protocol is fairness.

In this paper, we briefly reviewed an optimistic scheme for certified e-mail proposed by Galdi and Giordano. Their scheme is effective against the failures of the participants in most cases. However, we found that it cannot achieve true fairness, i.e., in case of collusion. We further proposed the improvements to avoid such an attack.

References

1. M. Abadi, N. Glew, B. Horne, and B. Pinkas. Certified email with a light on-line trusted third party: Design and implementation. *Proceedings of 2002 International World Wide Web Conference*, pp. 387-395, ACM Press, 2002.
2. G. Ateniese, B.de Medeiros, and M.T. Goodrich. TRICERT: A distributed certified e-mail scheme. *Proceedings of 2001 Symposium on Network and Distributed Systems Security*, Internet Society, 2001.
3. Bahreman and J.D. Tygar. Certified electronic mail. *Proceedings of 1994 Symposium on Network and Distributed System Security*, pp. 3-19, Internet Society, 1994.
4. F. Bao, G. Wang, J. Zhou, and H. Zhu. Analysis and improvement of Micali's fair contract signing protocol. *Proceedings of 2004 Australasian Conference on Information Security and Privacy*, LNCS 3108, pp. 176-187, Springer-Verlag, 2004.

5. R. Deng, L. Gong, A. Lazar, and W. Wang. Practical protocol for certified electronic mail. *Journal of Network and Systems Management*, Vol. 4, No. 3, pp. 279-297, 1996.
6. J.L. Ferrer-Gomila, M. Payeras-Capella, and L. Huguet-Rotger. An efficient protocol for certified electronic mail. *Proceedings of 2000 Information Security Workshop*, LNCS 1975, pp. 237-248, Springer-Verlag, 2000.
7. C. Galdi and R. Giordano. Certified e-mail with temporal authentication: An improved optimistic protocol. *Proceedings of 2004 International Conference on Trust and Privacy in Digital Business*, LNCS 3184, pp. 181-190, Springer-Verlag, 2004.
8. K. Imamoto and K. Sakurai. A certified e-mail system with receiver's selective usage of delivery authority. *Proceedings of Indocrypt 2002*, LNCS 2551, pp. 326-338, Springer-Verlag, 2002.
9. S. Kremer and O. Markowitch. Selective receipt in certified e-mail. *Proceedings of Indocrypt 2001*, LNCS 2247, pp. 136-148, Springer-Verlag, 2001.
10. S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of fair non-repudiation protocol. *Computer Communications*, Vol. 25, No.17, pp. 1606-1621, Elsevier, 2002.
11. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC Press, ISBN: 0-8493-8523-7, October 1996.
12. J.R.M. Monteiro and R. Dahab. An attack on a protocol for certified delivery. *Proceedings of 2002 Information Security Conference*, LNCS 2851, pp. 428-426, Springer-Verlag, 2002.
13. J.A. Onieva, J. Zhou, and J. Lopez. Enhancing certified email service for timeliness and multicast. *Proceedings of 4th International Network Conference*, pp. 327-336, Plymouth, UK, 2004.
14. R. Oppliger. Certified mail: The next challenge for secure messaging. *Communications of the ACM*, Vol. 47, No. 8, August 2004.
15. B. Schneier and J. Riordan. A certified e-mail protocol. *Proceedings of 1997 Annual Computer Security Applications Conference*, pp. 232-238, IEEE computer Society Press, 1997.