FAIRNESS IN CERTIFIED ELECTRONIC MAIL

Olivier Cailloux
Département d'Informatique
Université Libre de Bruxelles
Brussels, Belgium
email: olivier.cailloux@ulb.ac.be

Nicolás González-Deleito
Département d'Informatique
Université Libre de Bruxelles
Brussels, Belgium
email: ngonzale@ulb.ac.be

Olivier Markowitch
Département d'Informatique
Université Libre de Bruxelles
Brussels, Belgium
email: olivier.markowitch@ulb.ac.be

ABSTRACT
The growing use of the Internet promotes the replacement of traditional manual transactions by equivalent electronic services. Research was carried out to investigate enhanced services related to electronic mail. This paper points out that a certified email protocol has to provide the sender of a certified email with an evidence that this email has been either received or refused by its recipient, and proposes a new definition of the fairness property, specific to the certified email field. Finally, a new efficient certified email protocol respecting this property is presented.

KEY WORDS
computer security, certified email, security protocols, network security

1 Introduction

The development of the Internet is increasingly promoting the replacement of traditional manual transactions by equivalent electronic services. In the traditional certified mail service, a postman physically ensures that a message is exchanged for a signed receipt of this message. The postman delivers the mail to the recipient only after the latter has signed a receipt, and the recipient is sure that once he has signed the receipt he will receive the mail. In an electronic environment, this exchange process is not trivial. If we use techniques classically used in the traditional mail environment, we may face one of the following problems. On the one hand, if Bob receives an email from Alice, after having read it he would have the ability to decide whether to acknowledge the receipt of the email or not. On the other hand, if we ask Bob to send a receipt before Alice sends the email, Alice would get the receipt allowing her to claim that Bob received the email and would have thus the ability to decide whether to send the email or not.

In order to overcome these two problems, several specialised exchange protocols have been proposed during the last decade. Informally, in these two-party exchange protocols two entities want to exchange one or several items for one or several other items in such a way that, at the end of the protocol, either both parties get their expected items, or neither get any valuable information [1, 2, 3]. This property is called fairness. Until now, certified email protocols were considered as a particular instance of two-party fair exchange protocols, in which an email is exchanged for an evidence of its correct receipt. However, we suggest here to require that, even in the presence of dishonest recipients, an exchange must always take place, allowing therefore the sender to receive either an evidence attesting that the email was received, or an evidence attesting that the email was refused.

There are many ways to realise a fair exchange protocol. The most practical solutions make use of a trusted third party (TTP) that helps the protocol to always end in a fair way. Depending on its involvement, a TTP can be classified as follows [4]. If it is used in each step of the protocol [5, 6, 7], the TTP is said to be inline. If the TTP is used in each protocol run but not necessarily in each step [8, 9, 1, 10], the TTP is said to be online. Finally, if the TTP intervenes in the protocol only in case of problems between Alice and Bob (for example if Alice or Bob tries to cheat, or if a communication fails at a crucial moment) [11, 12, 2, 13, 14], the TTP is said to be offline.

In this paper, we discuss in section 2 the existing properties related to certified email protocols. We point out that this kind of protocols have to provide the sender of a certified email with an evidence that this email has been either received or refused by its receipient, and we propose a new definition of fairness dealing with the case (curiously neglected in the literature) where Bob is reluctant to accept an incoming certified email. In section 3 we discuss on the required quality of the underlying communication channels, and in section 4 we compare some well-known certified email protocols on the basis of the discussion provided in section 2. Finally, we propose in section 5 a realistic and efficient protocol that respects all the properties that we show a certified email protocol should satisfy.

2 Properties

2.1 Classical properties

Before reviewing and discussing the different properties that a certified email protocol should respect, let us first define the main types of evidences that are commonly exchanged during a successful protocol run [7]. Through the paper we will call Alice the sender of a certified email, and Bob the intended recipient. Moreover, we will denote by m the email that Alice aims to send to Bob.

An *evidence of submission* is a proof that Alice has tried to send m to Bob. An *evidence of receipt* is a proof that Bob has received m. An *evidence of origin* is a proof that Alice is at the origin of m. Providing this last evidence is sometimes considered as optional, since it is rarely provided in the traditional certified postal mail system.

The exchanged evidences can be used during a dispute resolution protocol that can possibly take place after the execution of the certified email protocol, if an entity claims that the exchange succeeded while the other entity does not agree. During this dispute resolution protocol, an adjudicator evaluates the evidences produced throughout the certified email protocol and, depending on the result, accepts or rejects the claim [1, 15].

On the other hand, several properties are usually considered as mandatory in the literature on certified email protocols. The first one is fairness [1, 2, 3].

Property 1 A certified email protocol is said to be *fair* if and only if at the end of a protocol execution either (1) Alice has received her expected evidence of receipt and Bob has received m (as well as the possibly corresponding evidence of origin), or (2) Alice has not received her evidence of receipt and Bob has not received any information regarding m (nor the possible evidence of origin). □

Another essential property, sometimes silently neglected, is timeliness [3].

Property 2 A certified email protocol respects the *timeliness* property if and only if each honest participant is always able to reach, in a finite amount of time, a point in the protocol where he can stop the protocol without losing fairness. □

The viability property [3] guarantees that the considered protocol achieves its goal.

Property 3 A certified email protocol is *viable* if there exists at least one possible protocol execution in which the exchange of an email (and the possibly corresponding evidence of origin) for an evidence of receipt of this email succeeds. □

Finally, an important property, not always considered mandatory, is the dated receipt property (also known as *temporal authentication* [16]). It allows Alice to prove that Bob received a certified email at a given moment in time, preventing therefore a dishonest Bob from claiming not to have received this email at that time.

Property 4 A certified email protocol provides the *dated receipt* property if and only if at the end of a protocol execution Alice obtains a time evidence allowing her to prove (with a bounded imprecision) the moment at which the receipt (or denial, cf. subsection 2.2) of m by Bob took place (or nothing if Bob did not have the possibility to receive m). □

2.2 Evidence of denial and fairness

When we compare the certified mail service provided by the post to existing certified email protocols, we notice that the latter do not offer Alice any strong evidence if Bob refuses to receive an email. Some protocols deliver an *evidence of submission* [5] issued by the TTP, proving that Alice has *tried* to send the email. Other protocols use evidences such as a *report of delivery* [7], which is only delivered if Bob accepts the email. In all cases, they are not intended to allow Alice to prove that the email has not been delivered because of Bob's unwillingness to receive this email. To the best of our knowledge, this important problem has never been considered before in the literature [5, 11, 8, 7, 10, 13, 14, 16, 17].

Consider the scenario where a landlady Alice sends a certified email to a tenant Bob not paying the rent of his flat. Bob could refuse any email from Alice and, during a dispute where she claims to have reminded Bob to pay the rent, Bob could claim to not have received any notification from her, due, for example, to a network failure. Usually, contrary to the traditional certified mail service, Alice will not receive here an evidence that Bob refused the email and will not be able to prove Bob's misbehaviour during a dispute. We think that certified email protocols should offer such kind of evidences when Bob refuses to receive an email.

This behaviour is not considered in the existing literature because certified email protocols are traditionally considered as being fair exchange protocols. Fair exchange protocols are often used to exchange a purchase against a payment or to exchange digital signatures on a contract. As all the participants are assumed to be willing to perform the exchange, it usually does not make sense in this context for a participant to refuse an item. Therefore, in addition to the evidence of receipt, we need another evidence that will allow a more precise understanding of how a protocol execution has ended.

We define thus an *evidence of denial* as a proof that Bob has refused to receive m. Consequently, the definition of fairness needs to be modified in order to take into account the need for this new evidence.

Property 5 A certified email protocol is said to be *fair* if and only if at the end of a protocol execution either (1) Alice has received her expected evidence of receipt and Bob has received m (as well as the possibly corresponding evidence of origin), or (2) Alice has received an evidence of denial and Bob has not received any information about m (nor the possibly corresponding evidence of origin) because he did not want to take part in the protocol. □

The no author-based selective receipt property defined by Kremer and Markowitch [14] prevents Bob from deciding whether to accept or not an email from Alice on the basis of her identity. We point out that this property can be useful in a protocol respecting our new fairness definition, in order to increase the chance of obtaining an evidence of receipt instead of an evidence of denial.

3 Network quality

Some minimal hypothesis need to be made on the network quality to ensure the correct execution of any certified email protocol. The following communication channels are usually considered in the literature on fair exchange protocols: *unreliable*, for which no assumptions are made (the data sent through it may be lost); *resilient*, which correctly delivers the sent data after a finite, but unknown, amount of time (the sent data may be delayed, but will eventually arrive); and *operational*, in which the sent data correctly arrive before a known finite constant amount of time.

Our protocol assumes that the channels used between the TTP and Alice and between the TTP and Bob are operational. Indeed, if the message is sent using a channel that is not operational, then Bob can always argue, in case of a trial or a dispute resolution, that he has still not received the message that Alice sent him. Alice will not be able to prove that Bob has received her message, while a dishonest Bob has probably received it, thus resulting in an unfair situation. Although the Internet does not formally provide operational channels, we can assume, in the context of certified email protocols, that its bound on the messages' transmission delay is just much larger than what is usually considered to be an operational channel. In consequence, protocols may be based on an operational channel that does not require messages to reach their destination in less than a few seconds or a few minutes, but within hours or days. Therefore we propose in section 5 a new protocol that respects all the required properties, and that takes the day-based bound on the transmission time into account to satisfy the dated receipt property.

4 Protocols comparison

Several papers in the literature address specifically the certified email problem, either with an inline [5, 7], online [8, 10] or offline [11, 13, 14] trusted third party. We review in this section some relevant and well-known certified email protocols, by studying them in light of the properties defined in section 2. The choice has been made in order to focus on the most representative protocols out of the vast existing diversity.

A basic inline protocol with six message exchanges, and providing an evidence of submission to Alice, is presented by Bahreman and Tygar [5]. Deng et al. [8] use a protocol with an online TTP and four messages, where Bob has to obtain the expected email from a public directory managed by the TTP (instead of having the latter directly sending the email to him). Riordan and Schneier [10] use a public board, also on a four messages model. Zhou and Gollmann [7] propose a protocol with several trusted third parties, with the email going from one third party to the following in a chain. Micali [11] is the first to propose an optimistic protocol, and this idea is modified by Ateniese et al. [13] to build a hybrid model, having some characteristics of the online model and some of the offline one. Kremer and Markowitch [14] introduce the no author-based selective receipt property, preventing Bob from learning the sender's identity before sending the evidence of receipt, and propose an optimistic protocol respecting this property. Blundo et al. [16] rely on an online time stamping server to provide the dated receipt property, and this protocol is improved by Galdi and Giordano [17].

As explained in section 2, most certified email protocols do not provide any evidence to Alice when Bob refuses the email. Only the Galdi and Giordano's protocol has been found to satisfy fairness (property 5). We are the first to define this property, but that protocol satisfies it as a side effect (to ensure timeliness). However, Galdi and Giordano use an online TTP (called a timestamping server, but that needs to be trusted), which is less efficient than the pure offline scheme that we propose in the next section.

The table presented below shows which properties these classical certified email protocols satisfy. As the authors do not always specify a sufficient network quality in order to provide fairness (or do not specify any network quality at all), we consider in this comparison that all the used communication channels are operational.

The properties considered in this table are timeliness (T), the delivery of an evidence of submission (EOS), the delivery of an evidence of origin (EOO), fairness in the sense of property 5 (F), dated receipt (DR), and the minimal number of messages needed for the exchange to succeed (Ex). ✓ indicates that the protocol satisfies the given property, × that the protocol does not satisfy it, and in, on and off denote respectively an inline, online and offline TTP. Every protocol considered in this table respects the viability property.

Protocol	**TTP**	**T**	**EOS**	**EOO**	**F**	**DR**	**Ex**
BT [5]	in	×	✓	✓	×	×	6
M [11]	off	✓	×	✓	×	×	3
DGLW [8]	on	✓	×	✓	×	×	4
ZG [7]	in	✓	✓	×	×	×	6
RS[a] [10]	on	✓	×	×	×	✓	4
AMG [13]	off / on [b]	×	×	✓	×	×	5
KM [14]	off / on [c]	✓	×	✓	×	×	4
BCDP [16]	off / on [d]	✓	✓	✓	×	✓	6
GG [17]	off / on [d]	✓	×	✓	✓	✓	4
Ours	off	✓	✓	✓	✓	✓	3

[a] The non-optimistic version.

[b] This protocol is hard to classify because it uses two types of TTPs, one of them being offline and the other being inline or online.

[c] This protocol uses a structure ensuring anonymity that must be online.

[d] This protocol uses two types of TTPs, one of them being offline and the other being online.

The protocol that we propose fulfils all the properties mentioned here and is efficient.

5 A certified email protocol

Before describing our protocol, let us first indicate the notations that we will use through this section. $X \rightarrow Y : \quad i$

Protocol 1 Main protocol

1. $A \to B: \quad f_{\mathrm{EOO}}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m), \mathrm{EOO}$
2. $B \to A: \quad f_{\mathrm{EOR}}, \mathit{label}, \mathrm{EOR}$
3. $A \to B: \quad \mathrm{key}$

Protocol 2 Recovery protocol for Bob

1. $B \overset{o}{\to} TTP: \quad f_{r_B}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m), \mathrm{EOO}, \mathrm{EOR}$
2. $TTP \overset{o}{\to} A: \quad f_{\mathrm{EOR}}, \mathit{label}, \mathrm{EOR}$

   $TTP \overset{o}{\to} B: \quad \mathrm{key}$

denotes entity X sending information i to entity Y using an unreliable channel; $X \overset{o}{\to} Y: \quad i$ denotes X sending information i to Y using an operational channel; $h(i)$ is the result of applying i to a one-way collision-resistant hash function h; $E_k(p)$ is the result of applying a symmetric encryption algorithm E to the plaintext p with the secret key k; $E_X(p)$ is the result of applying a (deterministic) asymmetric encryption algorithm E to the plaintext p with X's public key[1]; $S_X(i)$ denotes the digital signature (with appendix) of X on information i; and f_x is a flag indicating the purpose of a message in a given protocol, where x identifies the corresponding message in that protocol.

Moreover, we assume that the correct public keys needed to verify the digital signatures produced during the protocol, and to asymmetrically encrypt sensitive information, have been obtained by the participants.

The protocol allows Alice and Bob to exchange the certified email for the corresponding evidence of receipt by themselves during a main protocol. If a problem occurs during this main protocol (due to a dishonest participant or a poor communication channel quality), the participants will be able to ask the help of the TTP in order to fairly end the protocol, by running their corresponding recovery protocol. We assume that the communication channel used between the TTP and Alice, as well as between the TTP and Bob, is operational, and we denote by $delay_o$ the maximal transmission delay on these channels.

When considering time bounds, we will use a particular value $\mathrm{T\_MAX}$ representing the time allowed to complete the main protocol. For example, a reasonable value for this constant could be five hours. We will also assume that there is a bound on the imprecision between a perfectly synchronised clock and the clock of each participant. Taking for example one hour as a bound ensures that it is possible for the participants to synchronise their clocks without requiring any particular channel quality, e.g. listening to the radio is sufficient. For a better readability, we will not take into account in the protocol description these supplementary delays nor the different processing times, but it is easy to modify it accordingly.

Through all the three protocols a particular information, the label, is used to identify the protocol run. We define it [18] as $\mathit{label} = h(A, B, TTP, h(m), h(k), t)$.

The main protocol (which is summarised by protocol 1) begins with Alice sending to Bob the email m ciphered with a session key k of her choice (at this step, Bob does not know yet this session key), the information $\mathrm{key} = (f_{key}, \mathit{label}, k, S_A(f_{key}, A, B, \mathit{label}, k))$ ciphered with the TTP's public key (needed in case of recovery), the hash values of the email and of the session key (needed to verify the label), the current date and time t, and her digital signature on these information, $\mathrm{EOO} = S_A(f_{\mathrm{EOO}}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m))$.

If the label is coherent with the information present in the message, if Alice's signature EOO is correct, and if $t \leq$ current time $< t + \mathrm{T\_MAX}$, then Bob answers with his digital signature on the received information, $\mathrm{EOR} = S_B(f_{\mathrm{EOR}}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m))$. By doing so, he acknowledges having received the ciphered email from Alice.

Finally, if Bob's signature is correct, Alice sends him the session key k as well as her signature on it. Since Bob has the ciphered email and the corresponding session key, he is now able to derive the email m from $E_k(m)$.

If Bob does not receive the third message from Alice before time $t + \mathrm{T\_MAX}$ or receives a message not allowing him to reconstruct the label from the values of k and m, Bob has to invoke the following recovery protocol in order to obtain the expected session key k.

In his recovery protocol (protocol 2), Bob provides the TTP with all the information that should have been ex-

[1] The algorithms used to encrypt information are supposed to be deterministic. Although they are not formally provably secure, such algorithms are widely used in practice.

Protocol 3 Recovery protocol for Alice

```
1.   $A \overset{o}{\to} TTP:$   $f_{\mathrm{EOO}}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m), \mathrm{EOO}$
2.   $TTP \overset{o}{\to} B:$   $f_{\mathrm{EOO}}, A, B, TTP, \mathit{label}, h(m), h(k), t, E_{TTP}(\mathrm{key}), E_k(m), \mathrm{EOO}$
3.   if Bob replies
  a.   $B \overset{o}{\to} TTP:$   $f_{\mathrm{EOR}}, \mathit{label}, \mathrm{EOR}$
  b.   $TTP \overset{o}{\to} A:$   $f_{\mathrm{EOR}}, \mathit{label}, \mathrm{EOR}$
       $TTP \overset{o}{\to} B:$   $\mathrm{key}$
3'.  if Bob does not reply
       $TTP \overset{o}{\to} A:$   $f_{denial}, \mathit{label}, S_{TTP}(f_{denial}, A, B, \mathit{label})$
```

changed during the first two steps of the main protocol.

If the label can be reconstructed from the received information, if EOO and EOR are correct with respect to the received information, and if $t \leq$ current time $< t + \mathrm{T\_MAX} + \mathit{delay}_o$, then the TTP deciphers $E_{TTP}(\mathrm{key})$, obtains the session key k, deciphers $E_k(m)$, and verifies whether m is coherent with $h(m)$. If this last test succeeds, then the TTP provides the session key k and Alice's signature on this key (both information are contained in key) to Bob, and the evidence EOR to Alice. Otherwise, if one of these tests fails, the TTP replies to Bob with an error message.

On the other hand, if Bob does not (properly) perform the second step of the main protocol, Alice needs to execute, before time $t + \mathrm{T\_MAX}$, the following recovery protocol with the TTP in order to obtain an evidence of receipt or an evidence of denial for the email that she is trying to send.

Alice initiates her recovery protocol (protocol 3) by sending to the TTP the first message of the main protocol[1]. Upon receipt of this message, the TTP verifies whether the label is coherent with the hash values of the message and of the key, and whether Alice's signature EOO is correct. Furthermore, it verifies that $t \leq$ current time $< t + \mathrm{T\_MAX} + \mathit{delay}_o$.

If all these tests succeed, the TTP forwards this first message to Bob. If Bob has still not replied at time $t + \mathrm{T\_MAX} + 3 \cdot \mathit{delay}_o$, the TTP provides Alice with an evidence that Bob refused to receive her email. Otherwise, it performs the same actions as in the second step of the recovery protocol for Bob.

The evidence of origin that Bob may receive at the end of the protocol is composed of EOO, m, k, the timestamp t, and the signature $S_A(f_{key}, A, B, \mathit{label}, k)$. Similarly, the evidence of receipt that Alice may receive at the end of the protocol is composed of EOR, m, k, t, and the signature $S_A(f_{key}, A, B, \mathit{label}, k)$. The evidence of denial is composed of $S_{TTP}(f_{denial}, A, B, \mathit{label})$, m, k and t.

If Alice wants to prove that she has received an evidence of receipt or an evidence of denial, she has to present the corresponding evidence, as well as the identities of A, B and TTP, to an adjudicator. The latter has first to reconstruct the label, and for an evidence of receipt, reconstruct also $E_{TTP}(\mathrm{key})$ and $E_k(m)$. The adjudicator verifies then respectively EOR or $S_{TTP}(f_{denial}, A, B, \mathit{label})$. If the verification succeeds, it settles respectively that Bob has received the email m or that Bob refused m. Similarly, if Bob wants to prove that he has received a message from Alice, he has to present the evidence of origin that he has obtained during the protocol, as well as the identities of A, B and TTP, to an adjudicator. The latter has first to reconstruct the label, $E_{TTP}(\mathrm{key})$ and $E_k(m)$. The adjudicator verifies then the signature EOO. If this verification succeeds, it settles that Alice is the author of m.

The protocol is fair in the sense of property 5 since each execution ends with Alice obtaining either an evidence of receipt (if Bob received the message), or an evidence of denial (otherwise). The same reasoning shows that it is fair in the sense of property 1.

Concerning timeliness, we note that the main protocol should be finished at $t + \mathrm{T\_MAX}$. If this is not the case, Alice and/or Bob have to run their recovery protocol. Their request has to reach the TTP no later than $t + \mathrm{T\_MAX} + \mathit{delay}_o$, otherwise the TTP will reject it. In the case of Bob's recovery protocol, the protocol ends no later than $t + \mathrm{T\_MAX} + 2 \cdot \mathit{delay}_o$. For Alice, the second message reaches Bob no later than $t + \mathrm{T\_MAX} + 2 \cdot \mathit{delay}_o$. The TTP will at most wait until $t + \mathrm{T\_MAX} + 3 \cdot \mathit{delay}_o$. Its response will reach Alice and/or Bob no later than $t + \mathrm{T\_MAX} + 4 \cdot \mathit{delay}_o$. As both Alice and Bob know this value from the beginning of the protocol, the timeliness property is respected. An important related remark is that Bob must, once he has replied to Alice in the main protocol, remain available until $t + \mathrm{T\_MAX} + 2 \cdot \mathit{delay}_o$ in case Alice would invoke her recovery protocol in order to

[1] Note that fairness is still respected if the message is different, for example with another t, since any modification results in another protocol run.

obtain an evidence of denial after having already received a valid evidence of receipt.

The reasoning above shows that the protocol also satisfies the dated receipt property, as an adjudicator knows that Alice's email has been received by Bob after t, and before $t + \mathrm{T\_MAX} + 4 \cdot delay_o$.

6 Conclusion

We have examined the main properties that have been considered in the literature on certified email protocols. In consequence, we have proposed that a certified email protocol should provide an evidence to the sender even when the recipient is reluctant to participate in the protocol, and, as a result, we have suggested a new definition of the fairness property. We have also presented a new protocol satisfying all the required properties, being more efficient than the few existing protocols indirectly satisfying our new fairness definition.

References

[1] Jianying Zhou. *Non-repudiation*. PhD thesis, Royal Holloway, University of London, 1997.

[2] N. Asokan. *Fairness in Electronic Commerce*. PhD thesis, University of Waterloo, 1998.

[3] Olivier Markowitch, Dieter Gollmann, and Steve Kremer. On fairness in exchange protocols. In *Proceedings of the 5th International Conference on Information Security and Cryptology (ICISC 2002)*, volume 2587 of *Lecture Notes in Computer Science*, Springer-Verlag, November 2002, 451–464.

[4] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. (CRC Press, 1997).

[5] Alireza Bahreman and J. D. Tygar. Certified electronic mail. In *Proceedings of 1994 Network and Distributed System Security Symposium (NDSS 1994)*, Internet Society, February 1994, 3–19.

[6] Tom Coffey and Puneet Saidha. Non-repudiation with mandatory proof of receipt. *Computer Communication Review*, 26(1), January 1996, 6–17.

[7] Jianying Zhou and Dieter Gollmann. Certified electronic mail. In *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS'96)*, volume 1146 of *Lecture Notes in Computer Science*, Springer-Verlag, September 1996, 160–171.

[8] Robert H. Deng, Li Gong, Aurel A. Lazar, and Weiguo Wang. Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 4(3), September 1996, 279–297.

[9] Ning Zhang and Qi Shi. Achieving non-repudiation of receipt. *The Computer Journal*, 39(10), 1996, 844–853.

[10] Bruce Schneier and James Riordan. A certified e-mail protocol. In *Proceedings of the 14th Annual Computer Security Applications Conference*. ACM Press, December 1998.

[11] Silvio Micali. Simultaneous electronic transactions, March 1995. U.S. Patent No. 5,666,420.

[12] Jianying Zhou and Dieter Gollmann. An efficient non-repudiation protocol. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, IEEE Computer Society Press, June 1997, 126–132.

[13] Giuseppe Ateniese, Breno de Medeiros, and Michael T. Goodrich. TRICERT: A distributed certified e-mail scheme. In *Proceedings of 2001 Network and Distributed System Security Symposium (NDSS 2001)*. Internet Society, February 2001.

[14] Steve Kremer and Olivier Markowitch. Selective receipt in certified e-mail. In *Proceedings of the 2nd International Conference on Cryptology in India (Indocrypt 2001)*, volume 2247 of *Lecture Notes in Computer Science*, Springer-Verlag, December 2001, 136–148.

[15] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17), November 2002, 1606–1621.

[16] Carlo Blundo, Stelvio Cimato, and Roberto De Prisco. Certified email: Design and implementation of a new optimistic protocol. In *Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC 2003)*, IEEE Computer Society, 2003, 828–833.

[17] Clemente Galdi and Raffaella Giordano. Certified e-mail with temporal authentication: An improved optimistic protocol. In *Proceedings of the 1st International Conference on Trust and Privacy in Digital Business (TrustBus 2004)*, volume 3184 of *Lecture Notes in Computer Science*, Springer-Verlag, 2004, 181–190.

[18] Sigrid Gürgens, Carsten Rudolph, and Holger Vogt. On the security of fair non-repudiation protocols. In *Proceedings of the 6th International Conference on Information Security (ISC 2003)*, volume 2851 of *Lecture Notes in Computer Science*, Springer-Verlag, October 2003, 193–207.