

# Fair Certified E-mail Delivery\*

Aleksandra Nenadić  
Computer Science Dept.  
University of Manchester, UK  
anenadic@cs.man.ac.uk

Ning Zhang  
Computer Science Dept.  
University of Manchester, UK  
nzhang@cs.man.ac.uk

Stephen Barton  
Computer Science Dept.  
University of Manchester, UK  
s.k.barton@cs.man.ac.uk

## ABSTRACT

Communication by e-mail has become a vital part of everyday business and has replaced most of the conventional ways of communicating. Important business correspondence may require certified e-mail delivery, analogous to that provided by conventional mail service. This paper presents a novel certified e-mail delivery protocol that provides non-repudiation of origin and non-repudiation of receipt security services to protect communicating parties from each other's false denials that the e-mail has been sent and received. The protocol provides strong fairness to ensure that the recipient receives the e-mail if and only if the sender receives the receipt. The protocol makes use of an off-line and transparent trusted third party only in exceptional circumstances, i.e. when the communicating parties fail to complete the e-mail for receipt exchange due to a network failure or a party's misbehaviour. Considerations have been taken in the protocol design to reduce the use of expensive cryptographic operations for better efficiency and cost-effectiveness.

## Categories and Subject Descriptors

H.4 [Information Technology and Systems Applications]: Communications Applications—*Electronic mail*

## 1. INTRODUCTION

With the growing reliance on e-mail as the main business communication tool, there is an increasing demand for a reliable e-mail service that has embedded robust security

\*Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC '04, March 14-17, 2004, Nicosia, Cyprus  
Copyright 2004 ACM 1-58113-812-1/03/04...\$5.00

services. The basic e-mail security services include the provision of privacy (only the intended recipient can read the message) and authentication (the recipient can be assured of the identity of the sender). Cryptographic mechanisms for providing these security services, i.e. symmetric encryption and digital signatures/certificates, have been applied in Internet mail systems, such as S/MIME [21] and PGP [11].

In addition to sender authentication and message privacy, S/MIME can also provide a signed receipt service. A signed receipt from the recipient (requested by the sender) serves as a non-repudiable proof of receipt for a specific e-mail. However, the return of this receipt relies on the willingness of the recipient to honor the sender's request and provides no protection to the sender if the recipient chooses not to sign and return the acknowledgement after having read the message. In other words, this technique does not truly provide non-repudiation of the receipt security service. For a viable certified e-mail service, the following security properties are needed:

- *Non-repudiation of origin* - the recipient must have a way of proving that a specific e-mail indeed originates from the sender;
- *Non-repudiation of receipt* - the sender must have a way of proving that the recipient has indeed received a specific e-mail;
- *Strong fairness<sup>1</sup> for the exchange of an e-mail for a receipt* - the recipient should obtain a specific e-mail if and only if the sender obtains a receipt for it.

Digital signature mechanism provides the basis for the provision of non-repudiation security service. A digital signature on a message establishes the authenticity of the message and the identity of its originator. Therefore, the sender's signature on a message serves as a non-repudiable proof of its origin, and a proof of receipt is represented with the recipient's signature on the received message. A certified e-mail protocol must prevent a misbehaving recipient from refusing to return his receipt, i.e. from *selectively acknowledging* a message, after seeing its content. As the receipt depends on the message, the recipient cannot simultaneously receive the e-mail and sign the receipt. Therefore, adequate security protocols are required to simulate this pseudo-simultaneous

<sup>1</sup>An exchange protocol provides *strong fairness* [2] when, at the end of an exchange, if and only if one party has obtained the other party's item (or can obtain it without any further assistance of its originator), then the other party has obtained its expected item (or can obtain it without any further assistance of its originator).

exchange over a serial communication network and ensure that neither party suffers a disadvantage in the process.

The objective of this paper is to present a new and efficient RSA-CEMD protocol for the two communicating parties to fairly exchange an e-mail message for an RSA-based receipt. The RSA-CEMD protocol is based on a cryptographic primitive called *verifiable and recoverable encrypted signature* (VRES). VRES enables the recipient of an e-mail to encrypt his receipt (i.e. signature) in such a way that the sender can *verify* the correctness of the receipt without accessing its content, and, at the same time, the sender is convinced that an agreed trusted third party (TTP) can *recover* the receipt from its encryption, should the recipient refuse to do so. In this way the services of the TTP are invoked only in case of dispute; otherwise, the TTP is *off-line*.

The major contribution of the paper is the design of a novel certified e-mail protocol that offers a simple but efficient solution for the certified e-mail delivery problem. It is based on the standard RSA signature scheme [19] and satisfies all the security properties mentioned above. The protocol design attempts to minimize storage and computational requirements placed on the TTP, enabling it to serve more users and making it easier to implement. The participation of the TTP in the protocol is *transparent*, i.e. the receipt recovered by the TTP in case of dispute is indistinguishable from that generated by the original signer. The protocol is designed in such a way that only the initiator of the protocol needs to actively take part in possible dispute resolution with the TTP. This property enables the communication load on the recipient to be minimized and to safeguard the recipient from potential denial-of-service attacks by malicious senders.

The remainder of the paper is organized as follows. Next section discusses the related work in the area of certified e-mail delivery and fair exchange. Section 3 summarizes the notation and the assumptions used in the RSA-CEMD protocol design. Section 4 presents the RSA-CEMD protocol. Section 5 analyses the RSA-CEMD protocol in terms of security and performance, and compares it with the related work. Section 6 outlines our conclusions.

## 2. PREVIOUS WORK

Many solutions for the fair exchange and certified e-mail problems have been proposed in the literature. These solutions can be largely divided into two main categories depending on whether or not the protocols concerned have made the use of any TTP to assist the parties with the exchange process and achieving fairness.

Protocols that make no use of the TTP are based on the idea of simultaneous and gradual exchange of small parts of secret items (usually bit-by-bit) to achieve fairness [6, 7, 9, 16] (for more recent two-party approaches see probabilistic protocols [12, 14]). However, this category of protocols has some obvious drawbacks. Firstly, the exchanged secrets must be of the same length for the protocol to achieve fairness. Secondly, the participating parties must have the same computational power to prevent a situation when a computationally stronger party launches a brute-force attack to compute the remaining bits of his counterpart's secret be-

fore releasing all of its bits. Thirdly, the correctness of each released bit has to be checked and this creates an excessive computation and communication burden. Finally, both parties must be available on-line to engage in the protocol execution.

To overcome these weaknesses, the TTP-based protocols have been introduced. The TTP can be either *on-line*, participating in every exchange process, or *off-line*, where it does not take part in the exchange process if the parties themselves can reach a satisfactory completion. An on-line TTP can act as a *delivery authority*, which collects the exchanged items from the participating parties, checks their correctness and forwards them to the corresponding recipients [1, 6, 8]; a central server that serves as a *public message board* [20, 22]; a *notary* that receives and notarizes non-repudiation evidence to be retrieved by participating parties later on [24]; a *beacon* broadcasting randomized signals at regular time intervals [17]. The drawbacks of this approach are that the TTP has a potential to become a single point of failure, as the availability of the TTP is essential for the correct functioning of these protocols, and a communicational and computational bottleneck, as all the traffic goes through the TTP. Thus, the off-line TTP-based approach (also called the *optimistic* approach) [3, 5, 10, 13, 15, 23] remains the only practical solution for certified e-mail delivery.

Recently, a number of off-line TTP-based fair signature exchange protocols have been proposed based on a cryptographic primitive called *verifiable and recoverable encrypted signature* (VRES), some of which can be found in [3, 4]. VRES can also be applied in the design of certified e-mail protocols in the following way - the receiver of an e-mail generates the VRES of his receipt and then uses it to secure the release of the e-mail from the sender. The sender verifies the VRES, and if valid, sends the e-mail to the recipient. The possession of the recipient's VRES ensures the sender that he will definitely get the receipt, even if the recipient refuses to send it later on, as the agreed TTP can help to recover the receipt from VRES.

The main advantage of our protocol is that its novel method for VRES allows fair certified delivery in a more efficient manner in comparison with other related VRES-based certified e-mail protocols [3, 5, 13], in terms of number and length of messages required and number of expensive cryptographic operations used in formation of the messages. Protocols [3, 5] employ interactive zero-knowledge (ZK) proofs for VRES verification, which can be computationally consuming and generate an excessive communication overheads, whereas VRES verification in our protocol is performed efficiently without any on-line interactions between the sender and the recipient.

## 3. PRELIMINARIES

### 3.1 Notation

The following notation is used throughout the paper.

- $E_{pk}(x)$  and  $E_{sk}(x)$  express the ciphertexts of a data item  $x$  created with a public key  $pk$  and a private key  $sk$ , respectively, using RSA public-key cryptosystem [19].
- $h(x)$  is a strong-collision-resistant one-way hash function, i.e.  $h(x)$  has the following properties:

- (a) for any  $x$ , it is easy to compute  $h(x)$ ;
- (b) given  $h(x)$ , it is hard to compute  $x$ ;
- (c) given  $x$ , it is hard to find  $x' (\neq x)$  such that  $h(x) = h(x')$ .

- $x, y$  denotes the concatenation of data items  $x$  and  $y$ .

### 3.2 Assumptions

The following assumptions are used in the design of the RSA-CEMD protocol.

- Party  $P_a$  wishes to send an e-mail message  $M$  to party  $P_b$  in exchange for  $P_b$ 's receipt for  $M$ .
- $P_a$  and  $P_b$  have agreed to employ an off-line TTP  $P_t$  to help them with the exchange if they cannot reach a fair completion of the exchange themselves.
- Every party  $P_i (i \in a, b, t)$  has a pair of public and private RSA keys, expressed as  $pk_i = (e_i, n_i)$  and  $sk_i = (d_i, n_i)$ , where  $n_i$  is a product of two distinct large primes  $p_i$  and  $q_i$ , and  $(e_i \times d_i) = 1 \pmod{((p_i - 1) \times (q_i - 1))}$ . Public key  $pk_i (i \in a, b, t)$  is certified by a Certification Authority and known by all the other parties.
- Party  $P_b$ 's receipt for message  $M$ , denoted as  $receipt_b$ , is represented by  $P_b$ 's RSA signature on the message  $M$ :

$$receipt_b = (h(M))^{d_b} \pmod{n_b} = E_{sk_b}(h(M)).$$

- Party  $P_b$  has obtained a certificate  $C_{bt} = (pk_{bt}, w_{bt}, s_{bt})$ , issued by  $P_t$  prior to the exchange, for an additional RSA-based public key  $pk_{bt}$ . The public key  $pk_{bt}$  and its corresponding private key  $sk_{bt}$  are denoted as  $pk_{bt} = (e_{bt}, n_{bt})$  and  $sk_{bt} = (d_{bt}, n_{bt})$ , respectively, where  $n_{bt}$  is a product of two distinct large primes chosen by  $P_t$  and is approximately the same size as  $n_b$ , and  $e_{bt}$  is required to be the same as  $e_b$ , i.e.  $e_b = e_{bt}$ . The purpose of this certificate is to establish a secret key between  $P_b$  and  $P_t$ , which is to be used by  $P_b$  for the generation of verifiable and recoverable encryption of receipt for message  $M$ , and by  $P_t$  for the recovery of  $P_b$ 's receipt, should it be necessary.  $w_{bt}$  in  $C_{bt}$  is defined as  $w_{bt} = (h(sk_t, pk_{bt})^{-1} \times d_{bt}) \pmod{n_{bt}}$ , where  $sk_t$  is  $P_t$ 's private key. Note that  $P_t$  has no need to store the secret key  $sk_{bt}$  as it can compute it from  $w_{bt}$  using its private key  $sk_t$ , i.e.  $d_{bt} = (h(sk_t, pk_{bt}) \times w_{bt}) \pmod{n_{bt}}$ .  $s_{bt}$  in certificate  $C_{bt}$  is  $P_t$ 's RSA signature on  $h(pk_{bt}, w_{bt})$ , i.e.  $s_{bt} = E_{sk_t}(h(pk_{bt}, w_{bt}))$ .  $P_t$  only needs to issue one certificate  $C_{bt}$  for  $P_b$ , e.g. when party  $P_b$  registers with  $P_t$ .
- The VRES is based on the following theory [18].

**Theory of cross-decryption.** Let  $n_1$  and  $n_2$  be relatively prime and bases of two RSA cryptosystems, and  $e_1 = e_2 = e$  the corresponding public-key exponents. For any two messages  $m$  and  $m'$ , such that  $m < \min(n_1, n_2)$  and  $m' < \min(n_1, n_2)$ , the following holds:

$$(m^e \pmod{(n_1 \times n_2)}) \pmod{n_1} = m'^e \pmod{n_1} \Leftrightarrow m = m',$$

$$(m^e \pmod{(n_1 \times n_2)}) \pmod{n_2} = m'^e \pmod{n_2} \Leftrightarrow m = m'.$$

In simple terms, this theory states that for two RSA cryptosystems with the same public keys, either of the private keys  $d_1$  or  $d_2$  can be used to decrypt  $m^e \pmod{(n_1 \times n_2)}$  to recover  $m$ .

- Party  $P_a$  initiates the RSA-CEMD protocol. The communication between all the protocol parties is carried out through confidential and authenticated channels [11, 21].

## 4. RSA-CEMD PROTOCOL

The RSA-CEMD protocol comprises two protocols - the exchange protocol and the receipt recovery protocol.

### 4.1 The Exchange Protocol

In the exchange protocol, parties  $P_a$  and  $P_b$  attempt to exchange message  $M$  for its receipt. The exchange protocol (Figure 1) comprises steps (E1)-(E4).

(E1):  $P_a \rightarrow P_b : h(M), E_{sk_a}(h(M))$   
(E2):  $P_b \rightarrow P_a : x_b, xx_b, y_b, C_{bt}$   
(E3):  $P_a \rightarrow P_b : M$   
(E4):  $P_b \rightarrow P_a : r_b$

**Figure 1: The exchange protocol**

**(E1):**  $P_a$  transfers to  $P_b$  the hash value  $h(M)$  and its digital signature  $E_{sk_a}(h(M))$  on  $M$ . The signature will serve as a non-repudiable proof of origin of  $M$ .

**(E2):** Upon receipt of the two items,  $P_b$  verifies  $P_a$ 's signature using Verification 1.

#### Verification 1:

Decrypt the signature  $E_{sk_a}(h(M))$  with  $P_a$ 's public key  $pk_a$  to gain a hash value  $h(M)'$ , and confirm that  $h(M)' = h(M)$ .

If Verification 1 is negative,  $P_b$  may either ask  $P_a$  to re-send message (E1) or terminate the protocol execution. Otherwise,  $P_b$  produces verifiable and recoverable encryption of its receipt for message  $M$ , denoted as  $(y_b, x_b, xx_b)$ . To do so,  $P_b$  chooses a random prime number  $r_b < n_b$ , and computes:

$$y_b = r_b^{e_b} \pmod{(n_b \times n_{bt})},$$

$$x_b = r_b \times (h(M))^{d_b} \pmod{n_b} = (r_b \times receipt_b) \pmod{n_b},$$

$$xx_b = r_b \times E_{sk_{bt}}(h(y_b)) \pmod{n_{bt}} = (r_b \times (h(y_b))^{d_{bt}}) \pmod{n_{bt}}$$

$P_b$  then transfers his VRES  $(y_b, x_b, xx_b)$  and certificate  $C_{bt}$  to  $P_a$ .

**(E3):** Upon receipt of these items,  $P_a$  performs Verification 2 to check the correctness of  $P_b$ 's VRES.

#### Verification 2:

- (a) Check the correctness of signature  $s_{bt}$  in certificate  $C_{bt} = (pk_{bt}, w_{bt}, s_{bt})$ , i.e. decrypt  $s_{bt}$  with  $P_t$ 's public key  $pk_t$  to gain a hash value  $h v'$ , and confirm that  $h(pk_{bt}, w_{bt}) = h v'$ .
- (b) Confirm that

$$x_b^{e_b} \pmod{n_b} = (r_b \times (h(M))^{d_b})^{e_b} \pmod{n_b}$$

$$= (y_b \times h(M)) \pmod{n_b}.$$

- (c) Confirm that

$$xx_b^{e_b} \pmod{n_{bt}} = (r_b \times E_{sk_{bt}}(h(y_b)))^{e_b} \pmod{n_{bt}}$$

$$= (r_b \times (h(y_b))^{d_{bt}})^{e_b} \pmod{n_{bt}}$$

$$= (r_b^{e_b} \times (h(y_b))^{d_{bt} \times e_{bt}}) \pmod{n_{bt}}$$

$$= (y_b \times h(y_b)) \pmod{n_{bt}}.$$

Here,  $e_b = e_{bt}$ , and  $y_b \bmod n_b = (r_b^{e_b} \bmod (n_b \times n_{bt})) \bmod n_b = r_b^{e_b} \bmod n_b = E_{pk_b}(r_b)$  according to the theory of cross-decryption, and similarly  $y_b \bmod n_{bt} = E_{pk_{bt}}(r_b)$ , so  $y_b$  can be decrypted using either private key  $sk_b$  or  $sk_{bt}$  to recover  $r_b$ .

In detail, verification 2(a) makes sure that certificate  $C_{bt}$  is valid so that  $P_t$  can recover private key  $sk_{bt}$  related to public key  $pk_{bt}$  in  $C_{bt}$ . Verification 2(b) confirms that item  $x_b$  contains  $P_b$ 's receipt related to message  $M$ . Verification 2(c) together with (b) ensures that the same number  $r_b$  is used in the computations of  $y_b$ ,  $x_b$  and  $xx_b$ , and that the modulus operation in  $y_b$  is based on  $n_b \times n_{bt}$ , so that  $P_t$  can decrypt  $y_b$  with the private key  $sk_{bt}$  to obtain  $r_b$  for the recovery of  $P_b$ 's receipt, to be detailed below. If Verification 2 is negative,  $P_a$  may either ask  $P_b$  to re-send message (E2) or terminate the protocol execution. Otherwise,  $P_a$  transfers the message  $M$  to  $P_b$ .

**(E4):** Upon receipt of  $M$ ,  $P_b$  performs Verification 3 to ensure the correct message  $M$  was received.

#### Verification 3:

Confirm that message  $M$  received generates the hash value identical to that received in step (E1), i.e. calculate the fresh hash value  $h(M)''$  of the received message  $M$  and compare it with the hash value  $h(M)$  received from  $P_a$  in step (E1).

If Verification 3 is negative,  $P_b$  may either ask  $P_a$  to re-send message (E3) or terminate the protocol execution. Otherwise,  $P_b$  transfers  $r_b$  to  $P_a$ .

Upon receipt of  $r_b$ ,  $P_a$  uses it to derive  $receipt_b$  from  $x_b$  received earlier:

$$receipt_b = r_b^{-1} \times x_b \bmod n_b.$$

$P_a$  verifies the correctness of  $receipt_b$  using Verification 4.

#### Verification 4:

Confirm that  $E_{pk_b}(receipt_b) = h(M)$ , i.e. decrypt  $receipt_b$  with  $P_b$ 's public key  $pk_b$  to gain the hash value  $h(M)'''$ , and confirm that  $h(M)''' = h(M)$ .

If Verification 4 is positive, the certified e-mail delivery is completed successfully, i.e.  $P_a$  has obtained  $P_b$ 's receipt and  $P_b$  has obtained  $P_a$ 's message  $M$  together with its proof of origin  $E_{sk_a}(h(M))$ .

## 4.2 The Receipt Recovery Protocol

In case when  $P_a$  fails to obtain  $P_b$ 's correct receipt after handing over  $M$  to  $P_b$ ,  $P_a$  may request  $P_t$  for the receipt recovery by invoking the recovery protocol (Figure 2). The steps (R1)-(R3) of the recovery protocol are performed as follows.

(R1):  $P_a \longrightarrow P_t : M, C_{bt}, y_b, x_b$   
(R2):  $P_t \longrightarrow P_a : r_b$   
(R3):  $P_t \longrightarrow P_b : M$

**Figure 2: The receipt recovery protocol**

**(R1):**  $P_a$  transfers the items  $M$ ,  $C_{bt}$ ,  $y_b$  and  $x_b$  to  $P_t$ , which performs the following verification.

#### Verification 5:

Confirm that  $x_b^{e_b} \bmod n_b = (r_b \times (h(M))^{d_b})^{e_b} \bmod n_b = (y_b \times h(M)) \bmod n_b$ .

The purpose of Verification 5 is to prevent  $P_a$  from sending an incorrect message  $M'$  and getting  $P_t$  to unlawfully recover  $P_b$ 's receipt for him. In other words,  $h(M)$  received by  $P_b$  in step (E1) and used by  $P_b$  for the computation of  $x_b$  in step (E2) must be identical to the hash of the message  $M$  computed by  $P_t$  in step (R1). If Verification 5 is negative,  $P_t$  rejects  $P_a$ 's request. Otherwise,  $P_t$  derives key  $sk_{bt} = (d_{bt}, n_{bt})$  from  $C_{bt} = (pk_{bt}, w_{bt}, s_{bt})$  using its private key  $sk_t$ , as described in section 3.2, and uses key  $sk_{bt}$  to decrypt  $y_b \bmod n_{bt} = E_{pk_{bt}}(r_b)$  to recover  $r_b$ .

**(R2):**  $P_t$  then sends  $r_b$  to  $P_a$ , who uses it to compute  $P_b$ 's receipt as follows:

$$receipt_b = r_b^{-1} \times x_b \bmod n_b.$$

**(R3):**  $P_t$  forwards  $M$  to  $P_b$ .

## 5. RSA-CEMD PROTOCOL ANALYSIS

### 5.1 Security Analysis

Here, we first analyse the security of verifiable and recoverable signature encryption of  $P_b$ 's receipt  $(y_b, x_b, xx_b)$ .  $y_b = r_b^{e_b} \bmod (n_b \times n_{bt})$  is a minor variation of RSA encryption, so it is hard for any other party  $P_o (\notin \{P_b, P_t\})$  to decrypt  $y_b$  to recover  $r_b$  without knowing private key  $sk_b$  or  $sk_{bt}$ . It is also hard for  $P_o (\neq P_b)$  to factor  $x_b = (r_b \times receipt_b) \bmod n_b$  to gain receipt without knowing  $r_b$ . Similarly, it is hard for  $P_o$  to obtain  $r_b$  from  $xx_b = (r_b \times E_{sk_{bt}}(h(y_b))) \bmod n_{bt}$ . Therefore, it is hard for  $P_o$  to obtain receipt from  $(y_b, x_b, xx_b)$ .

We now consider various attempts of cheating by either  $P_a$  or  $P_b$ , and analyse how the protocol contends these attempts.

- $P_b$  attempts to cheat by using different random numbers instead of a single number  $r_b$  or an incorrect receipt' to generate  $(y_b, x_b, xx_b)$ . Verification 2(b) performed  $P_a$  will fail if the incorrect receipt' has been used to produce  $x_b$ , and Verification 2(c) or 2(b) will fail if different numbers have been used to generate  $(y_b, x_b, xx_b)$ . Consequently,  $P_a$  will terminate the protocol, so  $P_b$  gains no benefit from this misbehaviour.
- $P_b$  attempts to cheat by refusing to send  $r_b$  or sending an incorrect  $r_b'$  in step (E4).  $P_a$  can detect this deception after a timeout or by Verification 4. However, as  $P_a$  must have received  $P_b$ 's correct VRES in step (E2) (otherwise  $P_a$  would have terminated the protocol earlier),  $P_a$  can ask  $P_t$  for the recovery of  $r_b$  to obtain receipt.
- $P_a$  attempts to cheat by refusing to send  $M$  or sending an incorrect  $M'$  in step (E3). If  $P_b$  does not receive  $M$  before a timeout or detects the incorrect message  $M'$  through Verification 3,  $P_b$  will consequently terminate the protocol. This means that  $P_a$  will not receive  $r_b$  in step (E4), which is needed to compute  $P_b$ 's receipt, so  $P_a$  gains no benefit from this misbehaviour.
- $P_a$  attempts to cheat by requesting  $P_t$  to recover  $P_b$ 's receipt after step (E2) without sending  $M$  to  $P_b$  in step (E3). One of the conditions for  $P_t$  to accept  $P_a$ 's request is that  $P_a$  must provide message  $M$  that can

pass Verification 5. If Verification 5 is positive,  $P_t$  forwards  $P_a$ 's message  $M$  to  $P_b$  while passing  $r_b$  to  $P_a$ . Thus,  $P_a$  cannot benefit from this misbehaviour, as message  $M$  will ultimately be delivered to  $P_b$  by  $P_t$ .

The following discussion demonstrates that the protocol can meet the non-repudiation and strong fairness requirements.

Suppose that  $P_b$  has obtained  $P_a$ 's message  $M$ , i.e.  $P_b$  has received  $M$  in step (E3) of the exchange protocol, or in step (R3) of the recovery protocol. Then  $P_a$  has certainly got the correct items from  $P_b$  in step (E2) of the exchange protocol. Consequently,  $P_a$  can obtain  $r_b$  from  $P_b$  in step (E4), or request  $P_t$  for the recovery to get  $r_b$  in step (R2). After obtaining  $r_b$ ,  $P_a$  can use it to derive  $P_b$ 's receipt from  $x_b$ .

Similarly, suppose that  $P_a$  has obtained  $receipt_b$ , i.e.  $P_a$  has received the correct items from  $P_b$  in step (E2) and  $r_b$  in step (E4) of the exchange protocol, or  $r_b$  from  $P_t$  in step (R2) of the recovery protocol. This implies that  $P_b$  has received the correct  $M$  from  $P_a$  in step (E3), or from  $P_t$  in step (R3).

The above analysis implies that either  $P_b$  will receive message  $M$  and  $P_a$  will receive  $P_b$ 's receipt, or neither of them will receive anything. Therefore, the protocol meets the non-repudiation of receipt and the strong fairness requirements. Non-repudiation of origin is achieved by having  $P_a$  send its signature  $E_{sk_a}(h(M))$  in step (E1).

## 5.2 Comparison with Related Work

As can be seen from the RSA-CEMD protocol description, the involvement of the off-line TTP in the protocol is transparent, i.e. the structure of the receipt received at the end of the exchange does not reveal whether or not the TTP has been involved in the protocol execution, and the receipt received is a standard RSA-based signature. Protocols [2, 3, 10, 23], however, impose a non-standard and protocol-dependant structure for the receipts, and the TTP's involvement in these protocols is not transparent. In protocols [2, 3, 10, 23] both the sender and the recipient actively participate in dispute resolution, whereas in our protocol only the sender is actively involved. Certified delivery protocols from [3, 5, 13] are based on the VRES similarly to our protocol. Asokan et al.'s protocol [3] can be applied to various signatures schemes, but is rather inefficient due to the underlying interactive ZK proof used for VRES verification. Markowitch and Saeednia's protocol [13] is based on the Girault-Poupard-Stern (GPS) signature scheme, which is not widely used in practise. In the following we provide a detailed comparison of our protocol with the Ateniese and Nita-Rotaru's recently proposed certified e-mail protocol [5], denoted as the ANR protocol hereafter.

It seems appropriate to compare the two protocols, as both are based on the VRES method, are capable of achieving strong fairness, and are suited for RSA-based receipts. Efficiency of the two protocols is analysed in terms of the number of protocol messages and the number of expensive computations involved in the formation and verification of all the protocol messages. Expensive computations refer to modular exponentiations. Single multiplications and hash value computations are omitted, as they are considerably faster. The results of the comparison are shown in Table 1. The following analysis demonstrates that our protocol requires

less communication and computation overheads, and places less security and storage requirements on the TTP.

1. Both protocols require an initialisation phase for a party and a TTP to agree on a shared secret, which is used by the TTP for possible receipt recovery. In both protocols the TTP issues a certificate for the shared secret. In the ANR protocol, however, the TTP needs to store and safe-keep the secret. In our protocol, there is no need for the TTP to store the shared secret as it can be computed from the party's certificate.
2. VRES generations in both protocols require 3 modular exponentiations. However, VRES verification in our protocol is performed non-interactively and requires 3 modular exponentiations, whereas VRES verification in the ANR protocol is an interactive ZK proof and requires 4 exponentiations.
3. Both exchange protocols require 4 messages. However, since VRES verification in the ANR protocol is executed interactively, it requires a ZK sub-protocol with 2 additional protocol messages, thus 6 messages in total.
4. The recovery process in the ANR protocol is more complex than that in our protocol, and requires 5 expensive exponentiations, as opposed to only 2 exponentiations in our protocol.

	Our protocol	ANR protocol
No. of exponentiations in VRES generation	3	3
No. of exponentiations in VRES verification	3	4
No. of exponentiations in the exchange protocol	9	14
No. of exponentiations in the recovery protocol	2	5
No. of messages in the exchange protocol	4	6
No. of messages in the recovery protocol	3	3

**Table 1: Comparison of our and the ANR protocol**

## 6. CONCLUSIONS

This paper has presented a secure and efficient RSA-CEMD protocol for certified e-mail delivery with standard RSA-based receipts. The proposed protocol is suited for situations where the involved parties can resolve the communication problems themselves, and rely on an off-line and transparent TTP only as a last resort. The protocol has proven non-repudiation and strong fairness properties, and requires only a small number of expensive cryptographic computations and protocol messages. For our further work, we intend to formally verify and prototype the protocol.

## 7. ACKNOWLEDGMENTS

The work presented in this paper is part of the FIDES project (LINK, GR/R55177/01) funded jointly by the UK Engineering and Physical Sciences Research Council and Department of Trade and Industry.

## 8. REFERENCES

- [1] M. Abadi, N. Glew, B. Horne, and B. Pinkas. Certified email with a light on-line trusted third party: Design and implementation. In *11th International World Wide Web Conference (WWW'02)*. ACM Press, 2002.
- [2] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 86–99, 1998.
- [3] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610, 2000.
- [4] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proceedings of the 6th ACM conference on Computer and Communications security*, pages 138–146. ACM Press, 1999.
- [5] G. Ateniese and C. Nita-Rotaru. Stateless-recipient certified e-mail system based on verifiable encryption. In *Proceedings of the Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2002*, volume 2271, pages 182–199, Berlin, Germany, 2002. LNCS, Springer-Verlag.
- [6] A. Bahreman and J. Tygar. Certified electronic mail. In *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pages 3–19, 1994.
- [7] M. Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1(2):175–193, 1983.
- [8] R. H. Deng, L. Gong, A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *Journal of Network and System Management*, 4(3):279–297, 1996.
- [9] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
- [10] J. Ferrer-Gomila, M. Payeras-Capella, and L. Huguet-Rotger. An efficient protocol for certified electronic mail. In *Proceedings of the 3rd International Information Security Workshop ISW 2000*, volume 1975, pages 237–248, Berlin, Germany, 2000. LNCS, Springer-Verlag.
- [11] The Internet Engineering Task Force (IETF). *OpenPGP, An Open Specification for Pretty Good Privacy*. Available at [www.ietf.org/html.charters/openpgp-charter.html](http://www.ietf.org/html.charters/openpgp-charter.html).
- [12] O. Markowitch and Y. Roggeman. Probabilistic non-repudiation without trusted third party. In *Proceedings of 2nd Conference on Security in Communication Networks*, 1999.
- [13] O. Markowitch and S. Saeednia. Optimistic fair exchange with transparent signature recovery. In *Proceedings of 5th International Conference Financial Cryptography*, volume 2339, pages 339–350, Berlin, Germany, 2001. LNCS, Springer-Verlag.
- [14] J. Mitsianis. *A New Approach to Enforcing Non-repudiation of Receipt*, 2001. Manuscript.
- [15] M. Mut-Puigserver, J. L. Ferrer-Gomila, and L. Huguet-Rotger. Certified electronic mail protocol resistant to a minority of malicious third parties. In *Proceedings of IEEE INFOCOM 2000*, volume 3, pages 1401–1405, 2000.
- [16] T. Okamoto and K. Ohta. How to simultaneously exchange secrets by general assumptions. In *Proceedings of the 2nd ACM Conference on Computer and Communication Security*, pages 184–192, 1994.
- [17] M. Rabin. Transaction protection by beacons. *Journal of Computer and System Science*, 27:256–267, 1983.
- [18] I. Ray, I. Ray, and N. Narasimhamurthi. A fair exchange e-commerce protocol with automated dispute resolution. In *IFIP Workshop on Database Security*, pages 27–38, 2000.
- [19] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [20] B. Schneier and J. Riordan. A certified e-mail protocol. In *Proceedings of 13th Annual Computer Security Applications Conference*, pages 347–352. ACM Press, 1998.
- [21] S/MIME. *Secure Multipurpose Internet Mail Extensions*. Available at [www.rsasecurity.com/standards/smime/](http://www.rsasecurity.com/standards/smime/).
- [22] N. Zhang and Q. Shi. Achieving non-repudiation of receipt. *The Computer Journal*, 39(10):844–853, 1996.
- [23] J. Zhou, R. Deng, and F. Bao. Some remarks on a fair exchange protocol. In *Proceedings of International Workshop on Practice and Theory in Public Key Cryptography*, volume 1751, pages 46–57, Berlin, Germany, 2000. LNCS, Springer-Verlag.
- [24] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 55–61, 1996.