Journal of Universal Computer Science, vol. 11, no. 1 (2005), 175-192
submitted: 15/7/04, accepted: 6/12/04, appeared: 28/1/05 © J.UCS

RSA-based Certified Delivery of E-Goods Using Verifiable and Recoverable Signature Encryption[1]

Aleksandra Nenadić, Ning Zhang, Barry Cheetham, Carole Goble
School of Computer Science, University of Manchester,
Oxford Road, M13 9PL, UK
{anenadic, nzhang, barry, carole}@cs.man.ac.uk

Abstract: Delivering electronic goods over the Internet is one of the e-commerce applications that will proliferate in the coming years. Certified e-goods delivery is a process where valuable e-goods are exchanged for an acknowledgement of their reception. This paper proposes an efficient security protocol for certified e-goods delivery with the following features: (1) it ensures strong fairness for the exchange of e-goods and proof of reception, (2) it ensures non-repudiation of origin and non-repudiation of receipt for the delivered e-goods, (3) it allows the receiver of e-goods to verify, during the exchange process, that the e-goods to be received are the one he is signing the receipt for, (4) it uses an off-line and transparent semi-trusted third party (STTP) only in cases when disputes arise, (5) it provides the confidentiality protection for the exchanged items from the STTP, and (6) achieves these features with less computational and communicational overheads than related protocols.

Keywords: certified delivery, non-repudiation, fair exchange, security protocols
Category: C.2.2 [Computer-Communication Networks]: Network Protocols, D.4.6 [Software]: Security and Protection, K.6.5 [Management of Computing and Information Systems]: Security and Protection.

1 Introduction

E-goods refer to commercial products that can be represented in electronic form and transmitted over the Internet. Examples of such e-goods are video/audio content, software, e-newspapers, e-magazines, etc. E-goods delivery over the Internet is a business process where a piece of e-goods from a merchant is exchanged for its e-payment or an acknowledgement of its receipt (should a conventional payment method such as credit cards be used) from a customer. There are other items of value that, strictly speaking, cannot be considered as commercial products, such as payments, invoices, financial reports, utility bills, but can also be represented in the electronic form and should enjoy the certified delivery as well. Therefore, throughout the paper, by e-goods we consider any e-item of value.

[1] This paper is an extended version of the paper "An RSA-based Security Protocol for Certified E-goods Delivery", Proc. IEEE International Conference on Information Technology ITCC 2004, Las Vegas, NV, USA, IEEE Computer Society (2004), 22-28.

Industry analysts have predicted that the process of goods being delivered in electronic form will explode in the coming years due to advantages such as low cost and high product penetration that the Internet brings [InfoWorld 2002]. E-goods delivery is particularly beneficial to certain business sectors such as software development, publishing, financial services and entertainment industry. Software, for example, has been sold for years in shops in the form of CDs. Converting to a digital method of distribution will not only save time and money, but also expedite the process of software updates and bug fixes. Electronic versions of newspapers can be bought and sold over the Internet unless they are available on the Web for free. On-line invoicing can cut printing, shipping and postage costs, reduce billing errors, and, consequently, companies can reach out to more customers. Licensed retailers can distribute entertainment e-goods, such as music or films, instead of mailing them to customers or selling them in shops.

E-goods delivery involves several security concerns. In addition to authenticity and integrity assurance of the contents of e-goods and their confidentiality protection while in transit, there is another security service that is important in this context and that is *certified e-goods delivery*. This security service ensures that e-goods is delivered to its intended recipient if and only if the sender can obtain a receipt proving that the recipient has indeed received the e-goods. The receipt is normally a digital signature signed by the recipient on the received e-goods, and the sender can use this receipt as a proof of delivery, should any dispute arise (for example, if the recipient falsely claims that the e-goods have never arrived). The sender typically also attaches his signature on the e-goods and transmits it together with the e-goods to ensure the receiver of the authenticity and origin of the e-goods.

Certified e-goods delivery requires two important security properties: *fairness* and *content/quality assurance.* Fairness guarantees that a recipient will receive the e-goods together with a proof of its origin from the sender if and only if the sender receives a non-repudiable proof of the reception from the recipient. Content/quality assurance ensures that the received e-goods indeed match the expected description/quality.

Achieving fairness over the Internet can prove to be a problem since the Internet, as a serial communication network, cannot support simultaneous message exchange. Back in 1980, Even and Yacobi [Even and Yacobi 1980] showed that there is no deterministic protocol by which two parties can fairly exchange their items over the Internet (see also [Pagnia and Gärter 1999]). Business partners established through the Internet may not have previous business relationships and there is a potential lack of mutual trust. For this reason, neither of the parties involved in an exchange is willing to release his item first and thereby get into a disadvantageous position. Therefore, specialised security protocols are needed to ensure fairness – either all the parties get their expected items or none of them gets anything useful.

The problem of content/quality assurance is even subtler. Parties may engage in a certified delivery process to fairly exchange an e-goods item for a receipt. However, without the content/quality assurance, the receiver may discover at the end of an exchange process that the received e-goods are not of the promised or expected quality. For example, a customer has paid for a piece of software (or a high-quality film), but the software he has received does not have the full set of features as promised (or the film copy is of low quality).

The objective of this paper is to present an efficient Certified E-Good Delivery (RSA-CEGD) protocol to support a fair exchange of e-goods for an RSA-based receipt. The protocol is designed based upon a novel scheme enabling *verification and recovery of encrypted signatures* (VRES) and *joint e-goods and key certification*. The protocol achieves both fairness and content/quality assurance.

The VRES scheme represents an encryption of an RSA signature such that its recipient can *verify* the correctness of the signature without obtaining the signature itself. If the verification is positive, the recipient is guaranteed that an agreed STTP can *recover* the original signature from the VRES, should any dispute arise. In this way, the STTP can stay *off-line*, i.e. the parties can execute the exchange protocol without any involvement of the STTP. If the parties cannot reach a fair completion of the exchange themselves, the recovery service offered by the STTP is invoked and the dispute is resolved electronically. If the STTP has no on-line presence, the parties can submit the transaction evidence, for example using CDs, and the dispute can be resolved using a conventional method such as snail post and a court of law. In addition, the involvement of the STTP in the signature recovery process is *transparent* - the signature recovered by the STTP is indistinguishable from that generated by the original signer. Both the signature and the e-goods enjoy *confidentiality* protection against unauthorized access by any third party, including the STTP.

The *joint e-goods and key certification* concept is used in the protocol design to allow the recipient of e-goods to verify the correctness of the encrypted e-goods and its decryption key during a protocol execution. In this way, content/quality assurance of the e-goods can be achieved guaranteeing the recipient that the e-goods obtained after decryption at the end of the exchange will indeed match with what is expected. The linkage between the original e-goods, e-goods' encryption and the corresponding encryption/decryption key is validated and certified by a certification authority (CA) prior to the exchange. Although the concept of digital content validation/certification is not entirely new in the Internet arena [see Microsoft Authenticode] and has been utilised in fair e-purchase protocols [Franklin and Reiter 1997, Ray and Ray 2000] and fair document exchange protocols [Shi et al. 2003, Zhang et al. 2000], this paper is the first, to our best knowledge, to have incorporated the concept of joint e-goods and key certification to achieve certified e-goods delivery.

The rest of the paper is organised as follows. [Section 2] critically analyses recent work on fair exchange and certified delivery. [Section 3] gives the assumptions on which our protocol is designed and the security requirements it aims to satisfy. [Section 4] presents the VRES scheme, before the detailed description of the protocol is given in [Section 5]. The analysis and evaluation of the protocol, and comparison with related work are provided in [Section 6] and [Section 7], respectively. Finally, [Section 8] outlines our conclusions.

2 Background

Over the past few years, researchers have been working on the design of fair non-repudiation protocols for achieving certified delivery [Asokan et al. 1998, Ateniese and Nita-Rotaru 2002, Ateniese 2004, Deng 1996, Ferrer-Gomila et al. 2000, Kremer and Markowitch 2001, Markowitch and Roggeman 1999, Markowitch and Saeednia

2001, Schneier and Riordan 1998, Zhang and Shi 1996, Zhou et al. 1999, Zhou et al. 2000, etc.]. These works have mainly been focused on certified delivery of e-mails, for which certification of contents is not so much an issue. With certified e-mail delivery, the emphasis is on the generation of a piece of evidence that can prove the receipt of a specific e-mail by the recipient, and whether or not the content of the e-mail matches with the recipient's expectation is not important. For certified e-goods delivery, on the other hand, a signed receipt should guarantee not only the reception, but also the content/quality of the e-goods delivered. A mismatch between the expected and actual contents of the e-goods may have financial implications to the recipient and should be detected before the exchange process is over.

Certified delivery protocols most relevant to ours are those designed by [Asokan et al. 2000, Ateniese and Nita-Rotaru 2001, Ateniese 2004, Markowitch and Saeednia 2001, Nenadic et al. 2004]. These protocols are all based on the concept of verifiable and recoverable signature encryption and make the use of an off-line and transparent TTP for signature recovery. However, the verifiable and recoverable signature encryption schemes proposed in [Ateniese and Nita-Rotaru 2002, Asokan et al. 2000] are interactive, i.e. the verification of an encrypted signature is performed using a zero-knowledge (ZK) protocol that requires several srounds of on-line interactions between the sender and the recipient, which makes the solution computationally expensive. On the contrary, our protocol employs a non-interactive VRES scheme by which the verification can be performed efficiently off-line without any on-line interactions between the two exchanging parties.

The protocols by [Ateniese and Nita-Rotaru 2002, Markowitch and Saeednia 2001] do not protect the confidentiality of the exchanged items from the TTP. In addition, they impose higher security and storage requirements on the TTP. These protocols, unlike the protocol proposed in this paper, require the third party to be fully trustworthy. The e-goods delivery protocol by [Markowitch and Saeednia 2001] requires the TTP to verify the contents of the e-goods during protocol execution allowing the disclosure of the e-goods' contents to the TTP. In addition, sometimes it may be difficult for a third party to automatically validate the content, quality or features of the e-goods, as the TTP may not be specialized to do this. Therefore, there may be limitations in the practical application of this solution. In our protocol, a specialised CA (which may be different from the STTP) performs the e-goods validation prior to an exchange and issues a special certificate for the encrypted e-goods as well as its decryption key. The only task expected from the STTP is to recover an encrypted signature to ensure fairness, should a dispute arise.

3 Preliminaries

This section outlines notation used in the protocol description and the assumptions used in the protocol design. It also summarizes the security requirements the protocol is designed to satisfy.

3.1 Notation and Assumptions

The following notation has been used in the remaining part of the paper.

- $h(x)$ denotes a strong-collision-resistant one-way hash function, such as MD5 [Rivest 1992].
- $E_k(x)$ is used to express the ciphertext of a data item x encrypted with a key k. $E_k(x)$ is computed using a symmetric cryptosystem if the corresponding decryption key is the same as k, or RSA public-key cryptosystem [Rivest et al. 1978] otherwise.
- $pk = (e, n)$ and $sk = (d, n)$ denote RSA public and private key, respectively, where n is a public RSA modulus, and e and d are public and private key exponents.
- $E_{pk}(x) = x^e \ mod \ n = c$ is used to express RSA encryption of a data item x using public key $pk = (e, n)$, and $E_{sk}(c) = c^d \ mod \ n = x$ is used to express RSA decryption of ciphertext c using private key $sk = (d, n)$.
- $Sign(x) = E_{sk}(h(x)) = (h(x))^d \ mod \ n$ denotes RSA signature on a data item x generated with private key $sk = (d, n)$.
- Notation (x, y) denotes the concatenation of data items x and y.

Party P_a has valuable e-goods, denoted as D_a and a symmetric key k_a for the encryption and decryption of D_a. P_a is to send D_a to party P_b in exchange for party P_b's receipt for D_a. They have agreed to employ an off-line STTP P_t to help them with the exchange process if they cannot reach a fair completion themselves. It is assumed that P_t may misbehave on its own, e.g. trying to access the e-goods or receipt unfairly, but P_t does not conspire with either of P_a and P_b.

A CA has certified the content of the e-goods D_a and has issued certificate $CertD_a$ to P_a. This certificate is defined as follows:

$CertD_a = (desc_a, he_a, hd_a, hk_a, sign_{CA})$, where
$desc_a$ is the content summary of D_a,
$he_a = h(E_{k_a}(D_a))$ is the hash value of the encryption of D_a with the key k_a,
$hd_a = h(D_a)$ is the hash value of D_a,
$hk_a = h(k_a)$ is the hash value of the key k_a, and
$sign_{CA}$ is the CA's signature on the items $(desc_a, hd_a, hd_a, hk_a)$.

Certificate $CertD_a$ links the encrypted D_a, D_a's description $desc_a$ and the decryption key k_a. The CA's signature $sign_{CA}$ represents its guarantee that if the encrypted e-goods $E_{k_a}(D_a)$ and the key k_a meet the conditions $h(E_{k_a}(D_a)) = he_a$ and $h(k_a) = hk_a$, then decrypting $E_{k_a}(D_a)$ using the key k_a will recover D_a with its contents matching the description in $desc_a$. A single certificate $CertD_a$ for e-goods D_a can be used by party P_a for multiple exchanges.

The e-goods certification allows P_b to verify the correctness of the encrypted e-goods and the decryption key during an exchange. Without this certification, P_b cannot get content/quality assurance, as nothing would stop a dishonest P_a from using a worthless data in exchange for P_b's receipt, and this dishonesty could not be detected until the exchange is over. The CA can be a trusted independent entity or an entity closely associated to the product chain, e.g. the producer of the e-goods.

To illustrate this, consider an example of on-line film purchase. Let D_a represent a film produced and certified by a film producer and P_a an on-line merchant contracted by the producer to sell the film over the Internet. The film producer plays the role of the CA and issues the certificate $CertD_a$ to P_a, who can then sell it as many times as he can. A customer P_b, who represents a major cinema chain, is to purchase the film over the Internet. The customer and the merchant negotiate the deal and then fairly exchange the film for the receipt. The receipt proves that the customer has received the film. The certificate from the producer guarantees that the film is indeed an original copy that meets the description $desc_a$.

It is assumed that each party P_i ($i \in \{a, b, t\}$) has an RSA public and private key pair, expressed as $pk_i = (e_i, n_i)$ and $sk_i = (d_i, n_i)$. Public key pk_i ($i \in \{a, b, t\}$) had been certified by a recognised certification authority and is known by all the other parties. Party P_b's receipt for D_a, denoted as rec_b, is represented by P_b's RSA signature on D_a. That is,

$$rec_b = (h(D_a))^{d_b} \bmod n_b = (hd_a)^{d_b} \bmod n_b.$$

Party P_b has an additional certificate C_{bt} for his special RSA public key $pk_{bt} = (e_{bt}, n_{bt})$, issued by P_t prior to the exchange. n_{bt} is public RSA modulus chosen by P_t and e_{bt} is required to be the same as e_b in P_b's public key pk_b, i.e. $e_b = e_{bt}$. The private key corresponding to pk_{bt} is denoted as $sk_{bt} = (d_{bt}, n_{bt})$ and is shared by P_b and P_t. Certificate C_{bt} is defined as:

$C_{bt} = (pk_{bt}, w_{bt}, s_{bt})$, where
$w_{bt} = (h(sk_t, pk_{bt})^{-1} \times d_{bt}) \bmod n_{bt}$, and sk_t is P_t's own private key, and
s_{bt} is P_t's RSA signature on the items (pk_{bt}, w_{bt}).

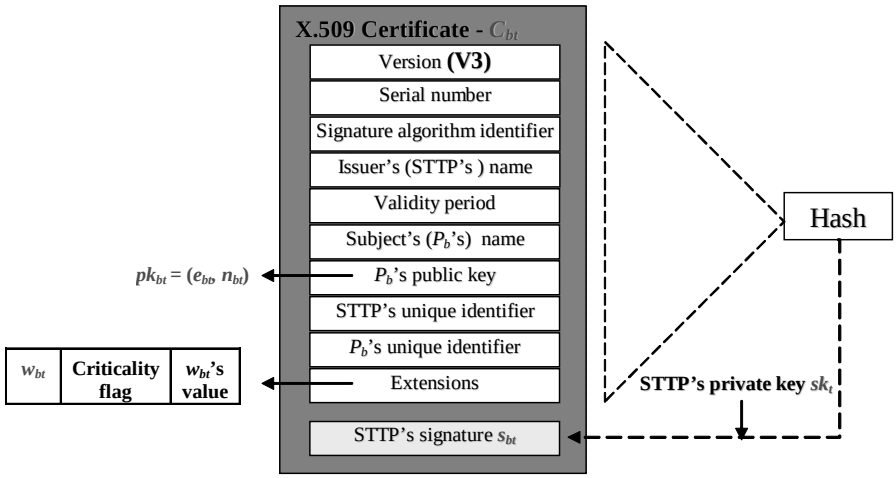

Figure 1: An example implementation of P_b's special public key certificate C_{bt}

By including w_{bt} in the certificate P_t has no need to store or safe-keep the shared private key sk_{bt}, as it can be computed from w_{bt} as $d_{bt} = (h(sk_t, pk_{bt}) \times w_{bt}) \bmod n_{bt}$. Certificate C_{bt} can be issued when party P_b registers with P_t initially, and can be

reused by P_b for multiple exchanges. The key pair (pk_{bt}, sk_{bt}) will be used in the design of the VRES scheme that is to be described in [Section 4] next. Certificate C_{bt} can be implemented using standard X.509 Version 3 certificate [X.509], which allows the use of extension field for e.g. w_{bt}, as shown in [Fig. 1].

Party P_a initiates the RSA-CEGD protocol. Communication channels to/from P_t are *resilient*, i.e. all messages inserted into such channels will eventually reach the intended recipients. No reliability assumptions are made about the channel between P_a and P_b. In other words, the channel may be *unreliable* and messages may be lost. Channels between all the parties are authenticated and confidential.

An overview of the RSA-CEGD protocol is as follows. The e-goods D_a is first encrypted with the symmetric key k_a and the CA certifies the content/quality of D_a together with the key k_a by issuing certificate $CertD_a$. The sender P_a transmits the encrypted e-goods and certificate $CertD_a$ to the recipient P_b. Upon verifying these items, P_b responds by sending his VRES containing his encrypted RSA signature (i.e. his receipt). If P_a is satisfied with the outcome of the VRES verification, it is secure for P_a to release the e-goods first by letting P_b have the decryption key k_a. If satisfied with the key received, P_b sends P_a an item needed for decrypting the VRES so P_a can obtain the receipt inside. Should this item fail to arrive, e.g. due to P_b's misbehaviour or a network failure, P_a may initiate the recovery process with the STTP P_t. P_t recovers this item for P_a from P_b's VRES using the shared private key sk_{bt}, thereby achieving the fair completion of the exchange.

3.2 Security Requirements

Before formally describing the RSA-CEGD protocol, its security requirements are first specified as follows:

(S1) *Non-repudiation of origin* – P_b is provided with a proof that P_a is indeed the originator of the e-goods.

(S2) *Non-repudiation of receipt* - P_a is provided with a proof that P_b has indeed received the e-goods.

(S3) *Strong fairness* - P_b has obtained P_a's e-goods or can obtain it with the assistance of P_t if and only if P_a has obtained P_b's receipt or can obtain it with the assistance of P_t.

(S4) *E-goods content/quality assurance* – P_b can verify, during the protocol execution, that the e-goods to be received are the same one he is signing the receipt for.

(S5) *E-goods and receipt confidentiality* – the e-goods to be delivered and the corresponding receipt are not disclosed to any external party, including P_t.

(S6) *Transparency of the STTP* - the participation of P_t in the protocol is transparent, i.e. the signature recovered by P_t is indistinguishable from the one sent by P_b.

4 Verifiable and Recoverable Encrypted Signature (VRES)

In this section we describe the generation, verification and recovery of the VRES. As previously explained, the VRES represents an encryption of an RSA signature that allows its recipient to *verify* the correctness of the signature inside without having the

plaintext signature, and is *recoverable* by a designated third party. The VRES scheme is designed based on the following theorem (for the proof the reader is referred to [Ray and Ray 2000]).

Theorem of cross-decryption. Let n_1 and n_2 be relatively prime moduli 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:

$$(m^e \bmod (n_1 \times n_2)) \bmod n_1 = (m')^e \bmod n_1 \text{ if and only if } m = m',$$
$$(m^e \bmod (n_1 \times n_2)) \bmod n_2 = (m')^e \bmod n_2 \text{ if and only if } m = m'.$$

This theorem states that, for two RSA cryptosystems with the same public exponents ($e_1 = e_2 = e$), either of the private exponents d_1 or d_2 can be used to decrypt the number $m^e \bmod (n_1 \times n_2)$ to obtain m.

4.1 VRES Generation

To generate verifiable and recoverable encryption of his signature rec_b, denoted as (y_b, x_b, xx_b), party P_b chooses a random prime number $0 < r_b < n_b$ and computes:

$$y_b = r_b^{e_b} \bmod (n_b \times n_{bt}),$$
$$x_b = (r_b \times (h(D_a))^{d_b}) \bmod n_b = (r_b \times rec_b) \bmod n_b,$$
$$xx_b = (r_b \times (h(y_b))^{d_{bt}}) \bmod n_{bt} = (r_b \times E_{sk_{bt}}(h(y_b))) \bmod n_{bt}.$$

Here, e_b and d_{bt} are public and private exponents of P_b's public key pk_b and private key sk_{bt}, respectively, y_b is a slightly modified RSA encryption of random number r_b, x_b represents the "encryption" of signature rec_b with random number r_b, and $E_{sk_{bt}}(h(y_b))$ is RSA encryption of $h(y_b)$ with the key sk_{bt}. Note that according to the theorem of cross-decryption we have:

$$y_b \bmod n_b = (r_b^{e_b} \bmod (n_b \times n_{bt})) \bmod n_b = r_b^{e_b} \bmod n_b = E_{pk_b}(r_b). \text{ Similarly,}$$
$$y_b \bmod n_{bt} = (r_b^{e_b} \bmod (n_b \times n_{bt})) \bmod n_{bt} = r_b^{e_{bt}} \bmod n_{bt} = E_{pk_{bt}}(r_b).$$

So, the number r_b can be recovered from y_b using either of the private keys sk_b or sk_{bt}.

4.2 VRES Verification

In order to verify P_b's VRES (y_b, x_b, xx_b), P_a does the following:

(*a*) Checks the correctness of P_t's signature s_{bt} in certificate C_{bt}.

(*b*) Confirms that $x_b^{e_b} \bmod n_b = [(r_b \times (h(D_a))^{d_b})^{e_b} \bmod n_b] = (y_b \times h(D_a)) \bmod n_b$.

(*c*) Confirms that $xx_b^{e_b} \bmod n_{bt} = [(r_b \times (h(y_b))^{d_{bt}})^{e_b} \bmod n_{bt} = (r_b^{e_b} \times (h(y_b))^{d_{bt} \times e_{bt}}) \bmod n_{bt}] = (y_b \times h(y_b)) \bmod n_{bt}$.

In detail, verification (*a*) makes sure that C_{bt} is a valid certificate issued by P_t. This guarantees the correctness of P_b's public key pk_{bt} and that P_t can recover private key sk_{bt} related to the public key pk_{bt} in C_{bt}. Verification (*b*) confirms that item x_b contains P_b's correct signature rec_b on e-goods D_a. Verification (*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 the

modulus operation in y_b is based on $n_b \times n_{bt}$, so that P_t can decrypt y_b with key sk_{bt} to obtain r_b for the recovery of P_b's signature rec_b from x_b.

4.3 VRES Recovery

To recover rec_b from the VRES (y_b, x_b, xx_b), P_t first derives the private key sk_{bt} from certificate C_{bt} using its private key sk_t (as described in [Section 3.1]). P_t then uses the derived sk_{bt} to decrypt $y_b \ mod \ n_{bt} = E_{pk_{bt}}(r_b)$ to recover r_b. Number r_b can then be used to compute P_b's signature from x_b. That is:

$$rec_b = (r_b^{-1} \times x_b) \ mod \ n_b.$$

5 The RSA-CEGD Protocol

In this section we formally present the protocol designed based on the VRES scheme described in [Section 4]. The protocol comprises 2 sub-protocols: the exchange sub-protocol and the recovery sub-protocol. Parties P_a and P_b execute the exchange sub-protocol in an attempt to exchange the items fairly without any involvement of P_t. If this attempt is not successful, the recovery sub-protocol is invoked during which P_t recovers the disputed signature. Definitions of all the items used in the protocol are given in [Tab. 1].

k_a: symmetric key for encryption/decryption of e-goods D_a generated by P_a;
$CertD_a = (desc_a, he_a, hd_a, hk_a, sign_{CA})$: certificate for D_a issued by the CA, where $he_a = h(E_{k_a}(D_a))$ is the hash value of the encryption of D_a with the key k_a, $hd_a = h(D_a)$ is the hash value of D_a, $hk_a = h(k_a)$ is the hash value of the key k_a;
$E_{sk_a}(hd_a)$: P_a's RSA signature on D_a serving as a proof of origin of D_a;
$rec_b = (hd_a)^{d_b} \ mod \ n_b$: P_b's receipt for P_a's e-goods D_a, i.e. P_b's RSA signature on D_a;
r_b: random prime generated by P_b for the generation of the VRES (y_b, x_b, xx_b);
(y_b, x_b, xx_b): P_b's VRES, where $y_b = r_b^{e_b} \ mod \ (n_b \times n_{bt})$ encryption of r_b with P_b's public key pk_b, also recoverable by P_t, $x_b = (r_b \times (hd_a)^{d_b}) \ mod \ n_b = (r_b \times rec_b) \ mod \ n_b$ is encryption of rec_b with r_b, $xx_b = (r_b \times E_{sk_{bt}}(h(y_b))) \ mod \ n_{bt}$ is a control number that confirms the correct use of r_b;
C_{bt}: P_b's RSA public-key certificate issued by P_t, $pk_{bt} = (e_{bt}, n_{bt})$, $sk_{bt} = (d_{bt}, n_{bt})$: public and private RSA keys related to C_{bt} with $e_{bt} = e_b$;
$s_b = E_{sk_b}(h(C_{bt}, y_b, hk_a, P_a))$: P_b's recovery authorisation token;

Table 1: Definitions of the protocol's items

5.1 The Exchange Sub-Protocol

The exchange sub-protocol comprises steps (E1)-(E4), as shown in [Fig. 2], and is executed as follows.

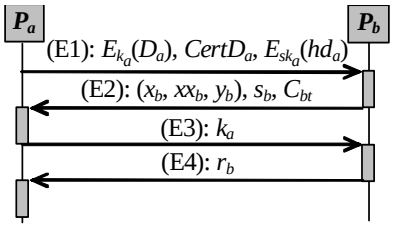

Figure 2: The exchange sub-protocol

(E1): P_a initiates the exchange by transmitting the encrypted e-goods $E_k(D_a)$, the e-goods' certificate $CertD_a$ and P_a's signature $E_{sk_a}(hd_a)$ on D_a to P_b. The signature will serve as a non-repudiable proof of origin of D_a.

$$P_a \rightarrow P_b: E_{k_a}(D_a), CertD_a, E_{sk_a}(hd_a)$$

(E2): P_b verifies the correctness of $CertD_a$ and $E_{k_a}(D_a)$. P_b also verifies P_a's signature $E_{sk_a}(hd_a)$ by decrypting it with P_a's public key pk_a to obtain a hash value hd_a' and comparing it with the hash value hd_a contained in certificate $CertD_a$. If the two hash values do not match, P_b may ask P_a to re-send message (E1) or terminate the protocol execution. Otherwise, P_b generates the VRES for his signature rec_b, i.e. (y_b, x_b, xx_b). P_b also produces a *recovery authorisation* token s_b that is defined as P_b's RSA signature on the items (C_{bt}, y_b, hk_a, P_a). The authorization token s_b authorises P_a to invoke the recovery process with P_t under certain conditions (controlled by P_b). Actually, s_b states that "P_t can recover r_b for P_a from y_b, if and only if P_a provides P_t with a key k_a, such that $h(k_a) = hk_a$". P_b then transfers the items, x_b, xx_b, y_b, s_b, C_{bt}, to P_a.

$$P_b \rightarrow P_a: x_b, xx_b, y_b, s_b, C_{bt}$$

(E3): P_a verifies the correctness of P_b's VRES (y_b, x_b, xx_b) as explained in [Section 4.2], and the correctness of P_b's recovery authorisation token s_b by verifying P_b's signature in s_b. If both of these verifications are positive, P_a sends the key k_a to P_b.

$$P_a \rightarrow P_b: k_a$$

Otherwise, P_a may ask P_b to re-send message (E2) or abort the protocol run.

(E4): P_b verifies the correctness of the received key k_a by confirming that $h(k_a) = hk_a$. If this verification fails, P_b may ask P_a to re-send message (E3) or abort the protocol run.

$$P_b \rightarrow P_a: r_b$$

Otherwise, P_b uses k_a to decrypt $E_{k_a}(D_a)$ to obtain D_a, and sends the random number r_b to P_a.

Having received r_b from P_b, P_a uses r_b to compute the signature rec_b from x_b as $rec_b = (r_b^{-1} \times x_b)\ mod\ n_b$ and verifies its correctness by confirming that $E_{pk_b}(rec_b) = hd_a$. If this verification is positive, the exchange process is completed successfully. If this verification is negative or P_a fails to receive r_b from P_b, P_a may initiate the recovery sub-protocol with P_t.

5.2 The Recovery Sub-Protocol

The recovery sub-protocol comprises steps (R1)-(R3), as shown in [Fig. 3], and is executed as follows.

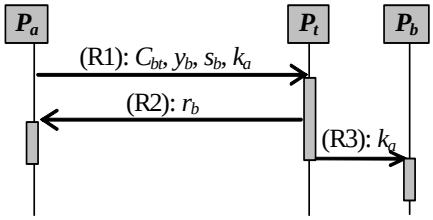

Figure 3: The recovery sub-protocol

(R1): P_a transfers the items C_{bt}, y_b, s_b and k_a to P_t to request the receipt (i.e. signature) recovery.

$$P_a \rightarrow P_t: C_{bt}, y_b, s_b, k_a$$

(R2): P_t verifies the correctness of P_a's recovery request by verifying P_b's authorisation token s_b to ensure that P_a has presented the correct key k_a. If this verification fails, P_t rejects P_a's request. Otherwise, P_t recovers r_b from y_b and sends it to P_a.

$$P_t \rightarrow P_a: r_b$$

(R3): P_t also forwards the key k_a to P_b.

$$P_t \rightarrow P_b: k_a$$

6 Security Analysis

In this section, we analyse the security of the cryptographic building blocks used in the protocol design, discuss possible ways in which the protocol can be attacked and justify that the protocol satisfies the requirements specified in [Section 3.2].

6.1 VRES Security

The security of the protocol is dependent on the security of the cryptographic primitive, namely, P_b's VRES (y_b, x_b, xx_b). The security of the VRES can be analysed

in relation to two issues. Firstly, is it sufficiently difficult to convert the VRES into the original signature rec_b without knowing random number r_b, and, secondly, is it sufficiently difficult for P_b to forge his VRES and trick P_a into accepting it?

y_b (= $r_b^{e_b}$ *mod* ($n_b \times n_{bt}$)) represents a minor variation of an RSA encryption, so it is hard for any other party P_o ($\notin \{P_b, P_t\}$) to decrypt y_b to obtain r_b without knowing private key sk_b or sk_{bt}. In order to obtain rec_b from x_b (= ($r_b \times rec_b$) *mod* n_b), party P_o ($\neq\{P_b\}$) has to guess the random number r_b or to factor x_b, which is considered computationally difficult provided that r_b and n_b are sufficiently large. With the current state-of-the-art in factoring algorithms [Stern 2001], r_b and n_b should be at least 1024 bits each. Similar discussion can be applied to xx_b (= ($r_b \times (h(y_b))^{d_{bt}}$) *mod* n_{bt}). Therefore, it is difficult for P_o to illegitimately obtain rec_b from the VRES.

In an attempt to cheat P_a, P_b may send a forged VRES (y_b', x_b', xx_b') to trick P_a to release the e-goods D_a. For example, P_b may try to generate a false signature $rec_b' \neq rec_b$ on a different D_a' and then use it to generate the VRES (y_b', x_b', xx_b'), i.e.

$$y_b' = r_b'^{e_b} \text{ mod } (n_b \times n_{bt}),$$
$$x_b' = (r_b' \times h(D_a')^{d_b}) \text{ mod } n_b,$$
$$xx_b' = (r_b' \times (h(y_b'))^{d_{bt}}) \text{ mod } n_{bt}.$$

For the forgery to be successful the VRES (y_b', x_b', xx_b') must pass the VRES verification, which means that the following must hold:

$$x_b'^{e_b} \text{ mod } n_b = (y_b' \times h(D_a)) \text{ mod } n_b,$$
$$xx_b'^{e_b} \text{ mod } n_{bt} = (y_b' \times h(y_b')) \text{ mod } n_{bt}.$$

This can, however, be truth if and only if $h(D_a) = h(D_a')$. As long as $h(x)$ is a strong collision-resistant hash function, this cheating is unlikely to succeed.

6.2 Attack Analysis

Cheating by party P_a. P_a may attempt to cheat P_b by sending an incorrect e-goods encryption $E_{k_a}(D_a)$ or an incorrect certificate $CertD_a$ in step (E1). However, P_b can detect these deception attempts through certificate verification, and, consequently, P_b may terminate the protocol after step (E1). Therefore P_a gains no benefit from this misbehaviour.

P_a may attempt to send an incorrect key k_a to P_b in step (E3). This deception can easily be detected by comparing the computed hash value of the key received and the one contained in the certificate $CertD_a$, so P_a cannot gain any advantage over P_b.

P_a may also attempt to cheat by requesting P_t to recover P_b's signature prior to step (E3), or by sending an incorrect k_a to P_t in step (R1). In both cases, for P_t to recover r_b, P_a must provide the correct key k_a (i.e. the one that can pass verification performed by P_t). If the key k_a cannot pass this verification, P_t will reject P_a's request. Once the request is accepted, P_t recovers P_b's r_b for P_a, but also forwards P_a's k_a to P_b. Therefore, P_a cannot gain any advantage over P_b, as the correct key k_a will ultimately reach P_b through P_t.

Cheating by party P_b. P_b may attempt to cheat P_a by using different random numbers instead of a single number r_b or an incorrect signature rec_b' to generate the

VRES (y_b, x_b, xx_b). As previously discussed, the VRES verification (*b*) will fail if an incorrect signature is used to produce x_b, and verification (*c*) or (*b*) will fail if different numbers are used to generate the VRES. Consequently, P_a may terminate the exchange process after step (E2), so P_b gains no benefit from this misbehaviour.

P_b may also attempt to cheat P_a by refusing to send r_b or sending an incorrect r_b in step (E4). P_b cannot benefit from this misbehaviour because, by step (E4), P_a must have already received P_b's correct VRES that guarantees the recovery of r_b, and consequently rec_b, by P_t.

Cheating by party P_t. P_t may attempt to obtain the signature rec_b or e-goods D_a during the recovery process. Without colluding with either P_a or P_b, it is difficult for P_t to be successful in these attacks. P_t only deals with y_b and the random number r_b, which are not sufficient for the disclosure of the signature rec_b. Also, e-goods D_a is not sent to P_t during the recovery process. Therefore, P_t cannot obtain any information about the exchanged items.

6.3 Satisfaction of Security Requirements

Non-repudiation and fairness. Suppose that P_b has obtained P_a's D_a, i.e. P_b has received the key k_a for the decryption of $E_{k_a}(D_a)$ in step (E3) or in step (R3). In this case, P_a has certainly got the correct items from P_b in step (E2). Consequently, P_a can obtain r_b from P_b in step (E4), or from P_t 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 rec_b, i.e. P_a has received the correct items from P_b in step (E2), and r_b in step (E4) or from P_t in step (R2). This implies that P_b has received the correct key k_a from P_a in step (E3) or from P_t in step (R3). Therefore, at the end of a protocol execution, either party P_b will receive D_a and party P_a will receive rec_b, or neither of them will receive anything useful. Party P_a must also provide the correct signature on D_a in step (E1), as a proof of origin of D_a. Therefore the protocol meets the non-repudiation of origin (S1), non-repudiation of receipt (S2), and the strong fairness (S3) requirements.

E-goods content/quality assurance: Based on the certificate $CertD_a$ issued by a CA, party P_b can verify the correctness of the decryption key k_a during the protocol execution, and he trusts the CA to perform the certification correctly. In this way, P_b is assured that the e-goods D_a, to be obtained at the end of the protocol execution (by decryption with the key k_a) will indeed match with the description given in $CertD_a$ (S4).

E-goods and receipt confidentiality: During the recovery process P_t deals only with the random number r_b and key k_a, while e-goods D_a, rec_b, and the number x_b (which can be used to derive rec_b) are not disclosed to P_t. Thus, the privacy of the exchanged e-goods and the corresponding receipt is preserved (S5).

Transparency of the STTP: It is evident that the structure of the receipt received from P_b in the exchange protocol is identical to that recovered by P_t during the recovery protocol, i.e. the received receipt does not reveal whether or not P_t has been involved in the exchange process (S6).

7 Comparisons and Performance Evaluation

In this section, we compare our RSA-CEGD protocol with the related proposals seen in literature. To the best of authors' knowledge, this paper presents the first protocol for certified e-goods delivery with an embedded e-goods content/quality assurance. As off-line TTP-based certified delivery protocols are the most related to ours, so we focus our comparison to this class of protocols. For a more extensive survey of fair non-repudiation protocols see [Kremer et al. 2002].

In certified e-mail protocols [Asokan et al. 1998, Ferrer-Gomila et al. 2000, Kremer and Markowitch 2001, Zhou et al. 1999, Zhou et al. 2000], a proof of receipt is represented by a token consisted of several items, i.e. it is not a standard signature on the e-mail. In addition, in these protocols, a signature recovered by the TTP is structurally different from the one produced by the original signer. This means that the third party in these protocols is not transparent and that these receipts always have to be interpreted in the context of the protocol that has generated it. In contrast to this, receipts received in the RSA-CEGD protocol are standard RSA-based signatures, and the STTP's participation is transparent.

The RSA-CEGD protocol is designed in such a way that only the sender is actively involved in the receipt recovery, while the recipient only takes a passive role in the process. This reduces communication load on the recipient and safeguards him from potential denial-of-service attacks from malicious senders. Protocols [Asokan et al. 1998, Ferrer-Gomila et al. 2000, Kremer and Markowitch 2001, Markowitch and Saeednia 2001, Zhou et al. 1999, Zhou et al. 2000] require both the sender and the recipient to actively participate in dispute resolution, and, in order to prematurely terminate the normal exchange protocol they have to contact the third party and execute an abort protocol. In other words, these protocols comprise 3 or 4 sub-protocols, which increases the communication overheads.

Verifiable and recoverable signature encryptions have been so far mainly utilised in fair signature exchange protocols [see Asokan et al. 2000, Ateniese 1999, Bao et al. 1998, Chen 1998], and have only recently been applied in certified delivery. Here, we compare the RSA-CEGD protocol with certified delivery protocols based on VRES-like schemes, namely, the Ateniese and Nita-Rotaru's protocol [Ateniese and Nita-Rotaru 2002], denoted as A-NR (for an extended version see [Ateniese 2004]), the Markowitch and Saeednia's protocol [Markowitch and Saeednia 2001], denoted as M-S, and the Nenadic et al.'s protocols [Nenadic et al. 2004], denoted as N-Z-B. The comparisons are performed in terms of the computational costs incurred for the generation, verification and recovery of the respective VRES schemes and for the protocol executions, and the results are shown in [Tab. 2]. Computational costs refer to the number of modular exponentiations used, as they are the most expensive computations.

A-NR is a certified e-mail delivery protocol where no quality/content assurance is required, whereas RSA-CEGD protocol is designed for certified e-goods delivery and therefore needs to meet this additional security requirement. A-NR employs an interactive verifiable and recoverable signature encryption scheme based on RSA signatures, which is less efficient than our non-interactive VRES scheme. Both RSA-CEGD and A-NR protocols have an initialisation stage for parties P_b and P_t to agree

on a shared secret that is subsequently used in the verifiable and recoverable encrypted signature schemes. However, in A-NR, the third party is required to store and safe-keep this secret, whereas in our protocol it can be computed from P_b's certificate C_{bt} and the third party need not store anything. Therefore, the security and storage requirements placed on the third party in RSA-CEGD are reduced. In addition, no confidentiality protection is provided for the e-mail and the receipt in A-NR, as they have to be disclosed to the third party to allow correct recovery.

M-S is a certified e-goods delivery protocol with a non-interactive verifiable and recoverable signature encryption scheme based on Girault-Poupard-Stern signatures [Girault 1991, Poupard and Stern 1998]. However, these signatures are not very often used in practise. Also, during the recovery process in M-S, the third party has to verify the contents of the e-goods, but the authors do not fully clarify how this verification is performed, which would additionally increase the computational cost for this protocol. In RSA-CEGD protocol, this is done by joint e-goods and key certification though certificate $CertD_a$. RSA-CEGD protocol also satisfies the confidentiality requirement for the exchanged items, whereas M-S does not.

The N-Z-B protocol uses the same VRES scheme as the RSA-CEGD protocol. However, the former is slightly more efficient than RSA-CEGD, as no e-goods certification and verification is necessary for certified e-email delivery applications that are aimed at by N-Z-B.

	RSA-CEGD	A-NR	M-S	N-Z-B
Receipt type	RSA	RSA	GPS	RSA
VRES type	non-inter.	inter.	non-inter.	non-inter.
VRES generation	3	3	2	3
VRES verification	3	6	2	3
VRES recovery	1	5	1	1
Exchange protocol	11	15	16	9
Recovery protocol	2	6	4/4*	2
# mess. (exchange protocol)	4	7	4	4
# mess. (recovery protocol)	3	3	3/3*	3
Items' confidentiality	Yes	No	No	Yes

*Numbers shown are for P_a's/ P_b's recovery protocols

Table 2: Comparison of computational costs

8 Conclusions

This paper has presented a fair non-repudiation protocol for achieving certified e-goods delivery in e-commerce. The protocol is based on two important concepts - the VRES scheme, which enables the protocol to achieve strong fairness, and the joint e-goods and key certification, which prevents a dishonest party from using some junk data in exchange for a receipt. The e-goods and the receipt exchanged enjoy the confidentiality protection against any third party including the STTP. The protocol places weak security requirements on the STTP, which simplifies its implementation

and management. Our protocol has been compared against related work in the field and we have demonstrated that it achieves more security properties that related protocols and with less computational costs. Our future work will include the formal verification and prototyping of the protocol presented.

Acknowledgements

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 (EPSRC) and the Department of Trade and Industry (DTI).

References

[Asokan et al. 1998] Asokan, N., Shoup, V., Waidner, M.: "Asynchronous Protocols for Optimistic Fair Exchange", Proc. IEEE Symposium on Security and Privacy, Oakland, CA (1998), 86-100.

[Asokan et al. 2000] Asokan, N., Schunter, M., Waidner, M.: "Optimistic Fair Exchange of Digital Signatures", IEEE Journal on Selected Areas in Communications, 18 (2000), 593-610.

[Ateniese 1999] Ateniese, G.: "Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures", Proc. ACM Conference on Computer and Communications Security, ACM Press, New York, USA (1999), 138-146.

[Ateniese and Nita-Rotaru 2002] Ateniese, G., Nita-Rotaru, C.: "Stateless-recipient Certified E-mail System Based on Verifiable Encryption", Proc. RSA Conference 2002, LNCS, 2271 (2002), Springer-Verlag, Berlin, Germany, 182-199.

[Ateniese 2004] Ateniese G.: "Verifiable Encryption of Digital Signatures and Applications", ACM Transactions on Information and System Security, 7, 1 (2004), 1-20.

[Bao et al. 1998] Bao, F., Deng, R., Mao, W.: "Efficient and Practical Fair Exchange Protocols with Off-line TTP", Proc. IEEE Symposium on Security and Privacy, Oakland, USA (1998), 77-85.

[Chen 1998] Chen, L.: "Efficient Fair Exchange with Verifiable Confirmation of Signatures", Proc. Advances in Cryptology - ASIACRYPT '98, LNCS, 1514 (1998), Springer-Verlag, Berlin, Germany, 286-299.

[Deng et al. 1996] Deng, R. H., Gong, L., Lazar, A.A., Wang, W.: "Practical Protocols for Certified Electronic Mail", Journal of Network and System Management, 4, 3 (1996), 279-297.

[Even and Yacobi 1980] Even, S., Yacobi Y.: "Relations Among Public-Key Signature Systems", Technical Report 175, Technion, Haifa, Israel (1980).

[Franklin and Reiter 1997] Franklin, M. K., Reiter, M.: "Fair Exchange with a Semi-trusted Third Party" (extended abstract), Proc. ACM Conference on Computer and Communications Security, Zurich, Switzerland (1997), 1-5.

[Ferrer-Gomila et al. 2000][9] Ferrer-Gomila, J. L., Payeras-Capella, M., Huguet i Rotger, L.: "An Efficient Protocol for Certified Electronic Mail", Proc. International Information Security Workshop 2000, LNCS, 1975 (2000), Springer-Verlag, Germany, 237-248.

[Girault 1991] Girault, M.: "Self-certified Public Keys", Proc. of Advances in Cryptology – EUROCRYPT '91, LNCS, 547 (1991), Springer-Verlag, 490-497.

[InfoWorld 2002] April, Carolyn A.: "Delivering the goods – digitally", InfoWorld (2002), available on-line at http://www.infoworld.com/article/02/11/08/021111ctdigital_1.html.

[Kremer and Markowitch 2001] Kremer, S. Markowitch, O.: "Selective Receipt in Certified E-mail", Proc. INDOCRYPT 2001, LNCS, 2247 (2001), Springer-Verlag, 136-149.

[Kremer et al. 2002] Kremer, S., Markowitch, O., Zhou, J.: "An Intensive Survey of Fair Non-Repudiation Protocols", Computer Communications, 25, 17 (2002), Elsevier, 1606-1621.

[Markowitch and Roggeman 1999] Markowitch, O., Roggeman, Y.: "Probabilistic Non-repudiation without Trusted Third Party", Proc. Conference on Security in Communication Networks (1999).

[Markowitch and Saeednia 2001] Markowitch, O. Saeednia, S.: "Optimistic Fair Exchange with Transparent Signature Recovery", Proc. International Conference on Financial Cryptography, LNCS, 2339 (2001), Springer-Verlag, 339-350.

[Microsoft Authenticode] Microsoft Authenticode, available at http://msdn.microsoft.com/library/default.asp?url=/workshop/security/authcode/authenticode_node_entry.asp

[Nenadic et al. 2004] Nenadic A., Zhang N., Barton S.: "Fair Certified E-mail Delivery", Proc. ACM Symposium on Applied Computing - SAC 2004, Nicosia, Cyprus (2004), 391-396.

[Pagnia and Gärtner 1999] Pagnia, H., Gärtner, F.: "On the Impossibility of Fair Exchange without a Trusted Third Party", Technical Report TUD-BS-1999-02, University of Darmstadt, Germany (1999).

[Poupard and Stern 1998] Poupard, G., Stern, J.: "Security Analysis of a Practical 'On the Fly' Authentication and Signature Generation", Proc. of Advances in Cryptology - EUROCRYPT '98, LNCS, 1403 (1998), Springer-Verlag, 422-436.

[Ray and Ray 2000] Ray, I., and Ray, I.: "An Optimistic Fair Exchange E-commerce Protocol with Automated Dispute Resolution", Proc. International Conference on E-Commerce and Web Technologies EC-Web 2000, LNCS, 1875 (2000), Springer-Verlag, Berlin, Germany, 84-93.

[Rivest et al. 1978] Rivest, R., Shamir, A., Adleman, L.: "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, ACM Press, 21, 2 (1978), 120-126.

[Rivest 1992] Rivest R.: "RFC 1321 - The MD5 Message-Digest Algorithm", MIT Laboratory for Computer Science and RSA Data Security, Inc. (1992), www.faqs.org/rfcs/rfc1321.html.

[Schneier and Riordan 1998] Schneier B., Riordan J.: "A Certified E-Mail Protocol", Proc. Annual Computer Security Applications Conference, ACM Press (1998), 347-352.

[Shi et al. 2003] Shi, Q., Zhang, N., Merabti, M.: "Signature-based Approach to Fair Document Exchange", Communications, IEE Proceedings, 150, 1 (2003), 21-27.

[Stern 2001] Stern, J.: "Evaluation Report on the Factoring Problem", (2001), 1-18.

[X.509] X.509 Certificate Specification, The Internet Engineering Task Force (IETF) - The PKIX Working Group, available at http://www.ietf.org/html.charters/pkix-charter.html.

[Zhang and Shi 1996] Zhang, N., Shi, Q.: "Achieving Non-Repudiation of Receipt", The Computer Journal, 39, 10 (1996), 844-853.

[Zhang et al. 2000] Zhang N., Shi Q., Merabti M.: "Anonymous Public-Key Certificates for Anonymous and Fair Document Exchange", Communications, IEE Proceedings, 147, 6 (2000), 345-350.

[Zhou et al. 1999] Zhou J., Deng R., Bao F.: "Evolution of Fair Non-repudiation with TTP", Proc. Australasian Conference on Information Security and Privacy ACISP '99, LNCS, 1587 (1999), Springer-Verlag, Berlin, Germany, 258-269.

[Zhou et al. 2000] Zhou J., Deng R., Bao F.: "Some Remarks on a Fair Exchange Protocol", Proc. International Workshop on Practice and Theory in Public Key Cryptography 2000, LNCS, 1751 (2000), Springer-Verlag, 46-57.