Fairness in Physical Products Delivery Protocol

ABDULLAH MOHAMMED ALARAJ
Information Technology Department
Qassim University
King Abdulazziz Road, Qassim
SAUDI ARABIA
arj@qu.edu.sa

Abstract: - In an e-commerce transaction, a buyer purchases a physical product, such as a laptop, from an online seller. In an attempt to protect himself, any seller would prefer to collect payment from the buyer before he sends the product. Likewise, the buyer would prefer to have the product shipped to him before he makes a payment to the seller. Both parties have a need to take precautions to minimize their risk in case the other party proves to be untrustworthy. This paper proposes a new e-commerce fair exchange protocol based on verifiable and recoverable encryption of keys. The proposed protocol is based on offline TTP. Only seven messages are exchanged between the parties involved in the protocol. Disputes are resolved electronically in case one party evades.

Key-Words: - Fair exchange protocol; E-commerce; Physical products; Cryptographic protocols; Security protocols

1 Introduction

There has been a dramatic increase in the use of e-commerce websites for buying and selling different types of products and services. The most common e-commerce applications are Business-to-Consumer (B2C), Consumer-to-Consumer (C2C) and Business-to-Business (B2B). In B2C e-commerce, the transacting parties are businesses and individual customers. In C2C e-commerce, the transacting parties are two individual customers. In B2B e-commerce, the transacting parties are two businesses.

In B2C, C2C and B2B, making a payment for a product or service is the main transaction. The steps involved in paying for products or services are as follows [2]:

1. A buyer decides on a product to be bought from an online seller
2. The buyer makes a payment to the seller
3. Once payment is received, the seller sends the buyer the product

In the B2B domain, the buyer and the seller normally know each other and there is mutual trust between them. A problem arises, however, in the B2C and C2C domains where the buyer and the seller do not know each other and hence have no relationship of trust [15]. If there is a lack of trust, the buyer will not take step two above, i.e. he will not send payment to a seller if he cannot ensure that the seller, in turn, will send him the product. Similarly, the seller will not send the product to a buyer unless he has received payment.

On the Internet, the simultaneous exchange of items between two parties is not supported [11]. Therefore, either the seller has to deliver first, or the buyer has to produce payment first. Yet, doing so places the one party at the risk of not receiving what he is owed by the other party. The buyer is usually the one who sends his payment first. As this is the case, there might be instances where the seller fails to deliver the product or sends the wrong product. In order to achieve simultaneity in exchange over the Internet, e-commerce *fair exchange protocols* are used. This ensures either that both parties get what they are owed or neither of them gets anything.

This paper proposes a new e-commerce *fair exchange protocol* that assures the buyer gets the product and the seller gets the payment. The proposed protocol focuses on achieving a fair exchange between the buyer and the seller when buying and selling physical products online (products that need physical delivery such as phones, laptops, etc.).

The motivation for conducting this research is that most of the existing literature on the use of *fair exchange protocols* focuses on the online purchase of digital products (music, software, computer games, etc.). Most online products are physical products. Yet, in the literature consulted, there is evidence of only two cases where *fair exchange protocols* were used for the purchase of physical products online [10] and [18]. These two protocols both have the limitation that there is no discussion of a resolution of disputes in the event that one party evades. In this paper we will be exploring the protocols proposed in [10] and [18] and then attempt to propose a *fair exchange protocol* that is suitable for the online purchase of physical products. The implementation of an effective protocol of this nature will result in greater confidence in e-commerce among consumers. This in turn will lead to an increase in online trade, since buyer confidence is considered to be the most difficult obstacle in e-commerce [8].

This paper is organized as follows. Section 2 discusses the literature survey. Section 3 presents the proposed protocol. The analysis of the proposed protocol is presented in section 4. Finally, section 5 compares the proposed protocol against similar protocols in the literature consulted.

2 Literature Survey

Buyers and sellers use e-commerce websites to buy and sell products and services. The buying and selling process mainly involves an exchange between the transacting parties. That is, the buyer produces payment (money) and the seller delivers the product. As discussed in the introduction, both the buyer and the seller seek to protect themselves during the transaction against the other party's failure to produce what they owe. In conventional trade, the buyer and the seller exchange the payment and the product simultaneously because they are present at the same place. When the buyer and the seller use an e-commerce website, this simultaneous exchange is not supported [11]. To solve the problem of ensuring reliability for both the buyer and the seller, e-commerce *fair exchange protocols* are used. These protocols ensure that either both parties collect what they are owed by the other party or neither does [15].

E-commerce *fair exchange protocols* are divided into different categories based on the involvement of the Trusted Third Party (TTP), the types of items to be exchanged, and the number of parties involved in the exchange (i.e. two parties or many parties) [9], [11], [15].

Early e-commerce fair exchange protocols [4] did not involve a TTP. The items to be exchanged are divided into parts and then exchanged in stages. First, one party starts the exchange by making a partial delivery to the second party. He in turn, makes a partial exchange to the first party. The first exchange is followed by a second, a third, and so forth, until both exchanges have been made in full. The exchange continues until the complete items have been exchanged between both parties. The problem with this approach is that there is a chance that the last party will not make his final exchange in full.

The other type of e-commerce *fair exchange protocols* involves a TTP to ensure reliability for all parties involved. The involvement of the TTP can be online or offline. The online TTP-based *fair exchange protocols* (such as in [5]) rely heavily on the TTP during the exchange. The online TTP-based *fair exchange protocols* generally work as follows. Both parties send what they owe to the TTP who validates what he receives. If everything is correctly validated then the TTP forwards the product and the money to the respective parties. The problem with the online TTP-based *fair exchange protocols* is that the TTP will be used in every exchange. This results in an added expense to cover the cost of the TTP – an expensive entity [1]. Equally important, is the fact that it may also lead to a bottleneck.

The offline TTP-based *fair exchange protocols* (such as in [1]-[3], [11], [13], [17]) do not rely on the TTP during the exchange of items. Rather, the parties will exchange their items directly and the TTP will be contacted in the case of dispute between the participating parties. Therefore, no additional cost is incurred for a TTP and a bottleneck is avoided.

Fair exchange protocols are used to exchange a variety of things. These include payments and digital products [1]-[2]; payments and physical products [10], [18]; email and receipt [12] and two digital signatures between contractual parties [4].

In the literature consulted, the focus for the use of *fair exchange protocols* is in the context of digital products and their payments (as in [1], [2], [11], [13], [17]); the exchange of emails and receipts [12]; and the exchange of two digital signatures on a contract [3], [4]. The use of *fair exchange protocols* for physical products has, however, not received its due attention, even though most of the products sold online are physical in nature. Only two instances were found where *fair exchange protocols* were used to ensure the fair exchange of physical products and their payment between two parties [10], [18]. More research is needed to explore the use of *fair exchange protocols* in the context of physical products - the focus of this paper.

E-commerce *fair exchange protocols* that are designed for the exchange of physical products and their payment are different from other protocols in that they need a delivery agent to be used during the exchange of the items. The delivery agent is used to deliver the physical product to the buyer. Other e-commerce *fair exchange protocols* i.e. protocols that are designed for the exchange of digital items (such as in [1], [2], [11], [13]) do not need a delivery agent because the items are sent electronically from the seller to the buyer through the protocol messages using computer networks. Hence, the design structure is different for the exchange of physical products and payments.

Zhang et al. [18] proposed an online TTP-based *fair exchange protocol*. The proposed protocol is for the exchange of payment and a physical product. The payment is sent via the protocol messages from the buyer to the seller whereas the product is delivered to the buyer via a delivery agent. The buyer starts the protocol by requesting a product from the seller. On receiving the buyer's request, the seller sends the invoice to the buyer. If the buyer is satisfied with the invoice they then send two messages. The first message is an encrypted payment to be sent to the seller and the second message is the encrypted payment to be sent to the TTP. It is assumed that the seller is able to download the TTP's encrypted payment (which the buyer sent to the TTP). The seller then compares the two encrypted payments. If they are compared correctly, the seller can be sure that the encrypted payment is correct. At this point, the seller sends the product to the delivery agent, which the buyer in turn collects from the delivery agent. Once the buyer has established that the product is as was expected, they send the decryption key to the seller who then decrypts the encrypted payment.

The problems with Zhang et al. protocol [18] are as follow. First, an online TTP is required. This results in the extra expense of running a TTP, as it will be needed during each exchange. Second, the TTP will also be a single point of failure as the protocol will not be executed if the TTP has any failure. Third, two payments will be sent by the buyer. One payment will be sent to the TTP and the other payment will be sent to the seller. Fourth, the number of messages used in the protocol is high - eight messages. Fifth, the dispute resolution phase is not discussed. Therefore, it is not clear how fairness will be ensured if one party fails to produce what he owes.

Li et al. [10] proposed an e-commerce *fair exchange protocol* for the exchange of payment and physical product. The protocol does not involve any Trusted Third Party (TTP). Rather, it involves a bank (where all parties have their accounts) and a delivery agent who will deliver the physical product to the buyer. The protocol comprises of eight messages (steps). The protocol starts with the two parties exchanging their signatures. Then, the delivery agent will deliver the physical product to the buyer to be checked. If the buyer is satisfied with the product then they will release the key to the delivery agent. After this, the physical product is handed to the buyer. The buyer will then send a signed receipt to the delivery agent. The delivery agent will forward the key and the signed receipt to the seller. The seller will forward the key and the signed receipt to the bank. Finally, the bank transfers the amount owed from the buyer's account to the seller's account.

The problems with the Li et al. protocol [10] are as follows. First, the buyer and the seller have to have accounts at the same bank. Therefore, the protocol is not suitable if the buyer and the seller have accounts at different banks. Second, the number of messages used in the protocol is high - eight messages. Third, the dispute resolution phase is not discussed. Therefore, it is not clear how fairness will be ensured if one party fails to produce what he owes.

To overcome the problems of the Zhang et al. protocol [18] and the Li et al. protocol [10], this paper proposes a new *fair exchange protocol* for the exchange of payment and physical product between online buyers and sellers. The proposed protocol is efficient in that it has fewer messages and fewer modular exponentiations compared with similar protocols in the literature consulted [10], [18]. Moreover, the proposed protocol is based on an

offline TTP that will only be contacted if something goes wrong. Unlike with the Zhang et al. protocol [18] and the Li et al. protocol [10], the proposed protocol discusses all phases of the protocol, namely, an exchange phase and a dispute resolution phase.

3 The Proposed Protocol

3.1 General description

The proposed protocol in this paper is for the exchange of physical product and payment between a seller and a buyer. The protocol is based on offline TTP that will be passive during the exchange between parties. The protocol is based on RSA scheme [16] where RSA public keys, RSA private keys, RSA encryptions, RSA decryptions are used. The protocol is based on verifiable and recoverable encryption of keys where the buyer will encrypt the key that is used to encrypt the payment. The seller will be able to verify the encrypted key. After correctly verifying the encrypted key it is safe for the seller to release the physical product because they will be able to recover the encrypted key from the TTP in case the buyer fails to send the decryption key to decrypt the payment.

The protocol consists of two phases, the exchange phase and the dispute resolution phase. The exchange phase represents the normal execution of the protocol where the buyer and seller exchange the payment and the physical product. The dispute resolution phase is used if one party fails to produce in the exchange phase. In the dispute resolution phase the TTP will be involved to ensure fairness for both parties.

The protocol will generally work as follows. The online buyer sends an encrypted payment to the online seller. The seller will be able to verify that the encrypted payment is correct using a special certificate that will also be sent to him. If it is correct, the seller forwards the physical product to the delivery agent. The delivery agent then sends the product to the buyer. At this point, the buyer checks that the product meets the agreed specifications. If it does, the buyer signs a receipt to indicate his satisfaction. The buyer sends the decryption key to the seller to decrypt the encrypted payment. If the buyer fails to produce payment, the seller can contact the TTP to recover the decryption key.

3.2 Notations

The following represents the notations used in the proposed protocol.

- B: the Buyer
- BB: Buyer's Bank
- S: the seller
- TTP: Trusted Third Party which is a party neither S nor B. The TTP is trusted by all parties. It is assumed that the TTP will not collude with any other party
- P: payment
- DA: Delivery Agent that is responsible for delivering the product to the buyer
- h(X): a strong-collision-resistant one-way hash function, such as SHA-1 [6]
- $pk_x = (e_x, n_x)$: RSA Public Key of the party x [16], where n_x is a public RSA modulus and e_x is a public exponent
- $sk_x = (d_x, n_x)$: RSA Private Key of the party x [16], where n_x is a public RSA modulus and d_x is a private exponent
- k_x: a symmetric key generated by the party x
- C.bt: the certificate for the shared public key between B and the TTP. C.bt is issued by the TTP. A standard X.509 certificate is used to implement C.bt
- $enc.pk_x(Y)$: an RSA encryption of Y using the public key pk_x (e_x, n_x). That is, $enc.pk_x(Y) = Y^{ex} \bmod n_x = Z$
- $enc.sk_x(Z)$: an RSA decryption of Z using the private key sk_x (d_x, n_x). That is, $enc.sk_x(Z) = Z^{dx} \bmod n_x = Y$
- $enc.k_x(Y)$: encryption of Y using a symmetric key k_x (k_x can be used for decrypting $enc.k_x(Y)$)

- Sig.x (A): the RSA signature of party x on A i.e. encrypting the hash value of A using the private key sk_x [16]. That is, Sig.x (A) = $(h(A))^{dx} \bmod n_x$
- S → B: X: S sends message X to B
- P-Cert: payment's certificate that is issued by the BB. The contents of *P-Cert* are:
  - *amount:* the amount of payment
  - *payee:* the name of the party who will receive the payment
  - *hP:* hash value of payment
  - *heP:* hash value of encrypted payment with k_b
  - *heKb:* hash value of encrypted k_b with pk_{bt}
  - *Sig.BB:* BB's signature on *P-Cert*

3.3 Registration

Before the exchange phase of the protocol starts, the buyer (B) needs to contact the TTP and its bank (BB):

1. B will first contact the TTP to request sharing an RSA public key with it. The shared public key between B and TTP is denoted as pk_{bt} = (e_{bt}, n_{bt}) and its corresponding private key is denoted as sk_{bt} = (d_{bt}, n_{bt}). The TTP will have a copy of sk_{bt}. After creating the shared public key, the TTP will issue the certificate *C.bt* of the shared public key and send it to B.
2. B will then contact its bank (BB) to get the payment and its certificate P-Cert. The P-Cert is unique for each transaction.

3.4 The Exchange Phase

The proposed protocol is based on the following assumptions:

- The Delivery Agent (DA) is trusted , will not collude with any party and is known to both B and S
- All parties will use the same encryption, decryption and hash algorithms
- Communication channels are resilient, meaning that all sent messages will be received by their intended receivers
- Necessary timestamps are used in related messages to prevent the replay attacks
- Identifiers will be used to identify the sender and the receiver of the messages

The exchange phase of the protocol will start by the buyer (B) sending the first message E-M1 to the seller (S) as follows (see figure 1).

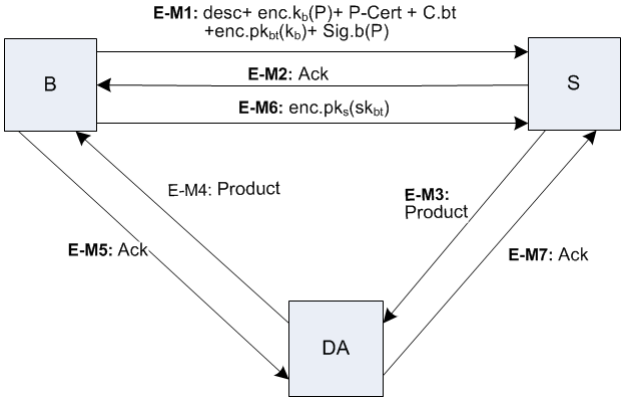

Fig.1. The Exchange Phase

[E-M1] **B → S: desc + enc.k_b(P) + P-Cert + C.bt +**

enc.pk_{bt}(k_b) + Sig.b(P)

B sends message E-M1 to S. It consists of the following.

- **desc**: specifies the description of the product that B wants from S. The description can be the product ID number. *desc* is signed by B
- **enc.k_b(P)**: encryption of the payment using k_b (k_b is generated by B)
- **P-Cert**: the payment certificate that is issued by BB
- **C.bt**: the shared public key certificate that is issued by TTP
- **enc.pk_{bt}(k_b)**: the encryption of k_b using the shared public key pk_{bt}
- **Sig.b(P)**: B's signature on the payment. This signature can serve as non-repudiation of origin which allows S to make sure that the payment is sent by B. B's signature on the payment is the encryption of the hash value of payment using B's private key sk_b

***[E-M2]* S → B: Ack**

On receiving the first message E-M1, S will do the following verifications:

1. S will check the correctness of the encrypted payment $enc.k_b(P)$. To do so, S computes the hash value of $enc.k_b(P)$. Then, S will compare it with the hash value of encrypted payment with k_b i.e. *heP* that is included in *P-Cert*. If they match then S can be sure that B encrypted the payment using k_b.
2. S will check the correctness of k_b that is used to encrypt the *payment* (*P*). Therefore, S will compute the hash value of $enc.pk_{bt}(k_b)$ and then compare it with *heKb* that is included in P-Cert. If they match then S can be sure that the encrypted key is k_b.
3. *P-Cert*: the correctness of *P-Cert* can be checked by verifying BB's signature on *P-Cert*.
4. *C.bt*: the correctness of C.bt can be checked by verifying TTP's signature on *C.bt*.
5. Sig.b(P): to verify the signature, S will decrypt *Sig.b(P)* using B's public key pk_b to get the hash value of payment. Then, S will compare it with hash value of *payment* (*hP*) included in *P-Cert.* If they match then S can be sure that a correct payment was signed by the buyer.

If all verifications are correct, the seller can be certain that he will get the decryption key from the buyer and if the buyer fails to deliver this, the TTP will be able to recover the decryption key. Therefore, if all verifications are correct then the seller will send a signed acknowledgement to the buyer. The signed acknowledgement indicates to the buyer that the first message is correct and that they can expect the product to be delivered to them by the delivery agent (DA).

***[E-M3]* S → DA: product**

If the first message is correct then S will send the product to the delivery agent (DA).

***[E-M4]* DA → B: product**

Upon the receipt of the product, DA will verify the identity of the buyer and then send the product to him.

***[E-M5]* B → DA: Ack**

Upon receipt of the product, B will check if it meets the *desc* that was specified in E-M1. If it meets the *desc* then B will sign a receipt (the receipt may include the product ID, seller ID, buyer ID, time of the transaction). The signed receipt indicates that B is satisfied with the product. If B is not satisfied with the product then DA will return the product back to the seller.

***[E-M6]* B → S: $enc.pk_s(sk_{bt})$**

If B finds that the received product is the one specified in *desc* then B will send message E-M6 to S. Using E-M6, S will be able to get the payment from the encrypted payment (note, the encrypted payment was received by S in E-M1). So, on receiving E-M6, S will do the following operations:

1. Decrypt $enc.pk_s(sk_{bt})$ using S's private key sk_s to get sk_{bt}
2. Use the derived sk_{bt} to decrypt $enc.pk_{bt}(k_b)$ to get k_b
3. Use k_b to decrypt $enc.k_b(P)$ to get the payment. The derived payment may then be sent to the bank for withdrawal.

***[E-M7]* DA → S: Ack**

After DA receives the signed receipt from B (i.e. in E-M5), DA forwards the signed receipt to S. S will use the signed receipt in case of dispute to assure TTP that B has received the product.

At this point, both S and B receive what they are owed. That is, B gets the product and S gets the payment. If, however, B fails to send the decryption key to S or by sending incorrect decryption key to S (i.e. in message E-M6) then S can contact the TTP to resolve the dispute as will be described in the next section.

3.5 Dispute Resolution Phase

In case of dispute where B sends an incorrect decryption key in E-M6 or B does not send the decryption key at all, S will be able to recover the decryption key from TTP (see figure 2). To do so, S will send the message DR-M1 to the TTP. DR-M1 includes *"Ack"* that was sent to S in E-M7.

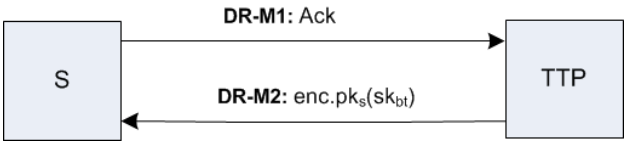

Fig.2. Dispute Resolution Phase

***[DR-M1]* S → TTP: Ack**

On receiving DR-M1, TTP checks B's signature on *Ack*. If the signature is correctly verified then this means that B has received the product and is satisfied with it. Therefore, TTP retrieves the decryption key "sk_{bt}" from its database and sends it to S. The decryption key "sk_{bt}" will be used to decrypt the key "k_b" that will then be used to decrypt the payment (note, the encrypted payment was sent to S in E-M1). If, however, the signature is not correctly verified then TTP will reject S's request.

***[DR-M2]* TTP → S: enc.pk$_s$(sk$_{bt}$)**

On receiving DR-M2, S will do the following operations:

1. Decrypt $enc.pk_s(sk_{bt})$ using S's private key sk_s to get sk_{bt}
2. Use the derived sk_{bt} to decrypt $enc.pk_{bt}(k_b)$ to get k_b
3. Use k_b to decrypt $enc.k_b(P)$ to get the payment. The derived payment may then be sent to the bank for withdrawal

At this point, both S and B receive what they are owed. That is, B gets the product (in the exchange phase) and S gets the payment (either in the exchange phase or in the dispute resolution phase). Therefore, fairness is ensured for both B and S.

Note that B is not involved in the dispute resolution phase. Rather, TTP will resolve the dispute without contacting B.

4 Analysis

In this section, the fairness, non-repudiation and security of the proposed protocol will be discussed.

Fairness:

The fairness property of the proposed protocol will be studied by analyzing all possible scenarios in both the exchange and dispute resolution phases.

The scenarios of executing the exchange phase of the protocol will be studied as follows.

1. **Scenario:** B sends E-M1 but S does not send E-M2
   Result: no one loses anything because the payment is encrypted and S does not have the key to decrypt it. Also, the product is not yet revealed by S
2. **Scenario:** B sends E-M1 to S and S sends E-M2 to B
   Result: no one loses anything because the payment is encrypted and S does not have the key to decrypt it. Also, the product is not revealed by S
3. **Scenario:** B sends E-M1 to S, S sends E-M2 to B, and S sends E-M3 to DA
   Result: no one loses anything because the payment is encrypted and S does not have the key to decrypt it. Also, the product is sent to DA (it is assumed that DA will not collude with any one of the parties). So, B did not get the product yet
4. **Scenario:** B sends E-M1 to S, S sends E-M2 to B, S sends E-M3 to DA, and DA sends E-M4 to B
   Result: in this case DA will send the product to B but B will not get full control of the product unless B is satisfied with it. If B is not satisfied with the product then DA will get the product back from B. DA will then return it back to S. if, however, B is satisfied with the product, then DA will ask B to sign a receipt (that is in message E-M5). Therefore, in this case where B is satisfied, B gets the product but S has not yet received the decryption key to decrypt the payment. If B sends correct E-M6 to S then both B and S get each other's items. If, however, B does not send E-M6 or sends incorrect E-M6 to S then S can recover the decryption key from the TTP by providing them with the receipt that B signed. When TTP receives the signed receipt, TTP will

validate it and if it is correct then TTP will send the decryption key to S. Therefore, fairness is ensured for both B and S either through the exchange phase or dispute resolution phase.

Scenarios in the dispute resolution phase are studied as follows:

1. S sends DR-M1 to TTP. There are two possibilities:
   a. DR-M1 is incorrect:
      Result: TTP will check B's signature on the receipt included in the acknowledgement. If TTP finds the signature is incorrect then TTP will reject S's request. Therefore, S has not gained any privilege
   b. DR-M1 is correct:
      Result: TTP will check B's signature on the receipt included in the acknowledgement. If the TTP finds the signature is correct then TTP will resolve the dispute by sending DR-M2 to S. In this case, B has received the product (in the exchange phase) and S has received the decryption key to decrypt the payment in the dispute phase. Therefore, the fairness is ensured for both B and S

Non-repudiation:

At the end of executing the protocol, the buyer will not be able to deny receiving the product, nor will the seller be able to deny receiving payment. After receiving the first message in the exchange phase, if the seller finds the first message to be correct, he signs the receipt indicating that the encrypted payment is in order. Similarly, when the buyer finds that the physical product matches his expectations in message E-M4 he signs for receiving the product. Although the seller signs for receiving the encrypted payment rather than the payment itself, the seller still receives the decryption key (for decrypting the encrypted payment) from the buyer, or, if the buyer fails to send the decryption key, from the TTP.

Security:

In the event that a message between the buyer and seller gets intercepted, there will be no loss for either the buyer or the seller. For the first message, E-M1, the payment will be encrypted with k_b and k_b will be encrypted with the shared public key pk_{bt} (pk_{bt} is shared between the buyer and the TTP). Therefore, for anyone to claim the payment, they will need to get the shared private key sk_{bt}. However, sk_{bt} is sent to the seller encrypted with the seller's public key in message E-M6. Therefore, only the seller will be able to decrypt it. No one will be able to intercept the physical product because it is sent physically to the buyer by the delivery agent who will verify the identity of the buyer.

5 Comparisons

In this section, the proposed protocol will only be compared against e-commerce *fair exchange protocols* that are designed for the exchange of physical products and their payment between a buyer and a seller. The protocols in the literature of this type are Zhang et al. protocol [18] (will be referred to hereafter as the Zhang Protocol) and Li et al. protocol [10] (will be referred to hereafter as the Li Protocol).

The criteria of comparisons are: (1) type of TTP used, (2) number of messages in the exchange phase of the protocol, (3) number of messages in the dispute resolution phase of the protocol, (4) parties involved in the dispute resolution phase, (5) number of modular exponentiations (which are considered to be the most expensive operations [11]) in the exchange phase, (6) number of modular exponentiations in the dispute resolution phase.

Table 1: Protocols Comparisons

	Zhang Protocol [18]	Li Protocol [10]	Our Protocol
Type of TTP	Online	No TTP is involved	Offline
# messages (exchange	8	8	7

phase)			
# messages (dispute resolution)	Not discussed	Not discussed	2
Both parties are involved in dispute resolution	Not discussed	Not discussed	No
# modular exponentiations (exchange phase)	25	10	10
# modular exponentiations (dispute resolution phase)	Not discussed	Not discussed	4

As can be seen in Table 1, the Li protocol does not involve a TTP in the exchange phase of the protocol (apart from the bank and the delivery agent) whereas the Zhang protocol involves an online TTP that has to be available during the exchange phase of the protocol. Our protocol involves an offline TTP that will only be involved in case of disputes.

Our protocol has the lowest number of messages in the exchange phase of all protocols. Only two messages will be used in the dispute resolution phase of our protocol. The other two protocols do not discuss the dispute resolution phase in their protocols. Therefore, it is not clear how their protocols will work in the event that one party fails to follow through on the exchange.

Regarding the number of modular exponentiations, both our protocol and the Li protocol have 10 modular exponentiations whereas the Zhang protocol has 25. Because the Zhang protocol and the Li protocol have not discussed the dispute resolution phase in their protocols, the number of modular exponentiations in their protocols remains unclear. Our suggested protocol has only 4 modular exponentiations in the dispute resolution phase.

In our protocol, the buyer is not involved in the dispute resolution phase. The seller is the party who raises disputes to TTP as the seller delivers its product first. That is, the seller will send its product to DA and DA will deliver it to the buyer before the buyer sends its decryption key to the seller. Therefore, there is a chance that the buyer might either fail to send the decryption key to the seller, or send an incorrect decryption key. In case of a dispute, the seller will contact TTP for resolution. TTP does not need to contact the buyer for validating the seller's request. Instead, TTP will validate the seller's request by verifying the buyer's signature on the receipt of the product. Therefore, if TTP finds that the buyer's signature is correctly verified then there is no need to contact the buyer because the correct signature indicates that the buyer has received the product and is satisfied with it. (Otherwise, the buyer would not have signed the receipt.)

It is clear that our protocol covers all phases i.e. the exchange phase and the dispute resolution phase. Therefore, parties involved in our protocol will be able to recover the key in case one party evades. It is not clear how disputes will be resolved in the Zhang and Li protocols.
The proposed protocol is efficient and practical in many respects. First, due to the direct exchange of payment and product between the buyer and seller, there is no cost associated with the exchange of payment and product. Hence the TTP is not involved in the normal execution of the exchange. Second, the number of messages in the exchange protocol is the lowest of all similar protocols. Third, the number of modular exponentiations is low in both the exchange and dispute resolution protocols. Forth, the simplicity of the design on which our approach is based, to use the verifiable and recoverable encryption of key, makes the proposed protocol very simple and easy to integrate with an existing e-commerce website (an area of research that will be investigated more deeply in future work).

6 Conclusion

This paper presented a new e-commerce *fair exchange protocol* for exchanging physical products and payments between online sellers and buyers. One strong point of the proposed protocol is that it consists of only seven messages that are exchanged between the parties involved in the protocol. This represents the lowest number of messages of all

similar protocols. Another strong point of the proposed protocol is that it includes the dispute resolution phase that will be implemented in case of disputes between parties. Similar protocols, in the literature consulted, do not discuss a dispute resolution phase; and hence it is unclear how the parties involved will handle disputes.

Future work will include the formally proofing of the fairness property of the proposed protocol using formal methods [7]. It will also include the implementation of the proposed protocol and its integration with an e-commerce application for buying and selling physical products.

References:

[1] A. Alaraj and M. Munro, An e-commerce fair exchange protocol that enforces the customer to be honest, International Journal of Product Lifecycle Management, IJPLM, vol. 3, no. 2/3, pp. 114-131, 2008

[2] A. Alaraj and M. Munro, An efficient e-commerce fair exchange protocol that encourages customer and merchant to be honest, in Proceedings of the 27th International Conference on Computer Safety, Reliability and Security (SafeComp), New Castle, UK, Lecture Notes in Computer Science, vol. 5219, 2008, pp. 193-206.

[3] N. Asokan, M. Schunter and M. Waidner, Optimistic protocols for fair exchange, in Proceedings of the Fourth ACM Conference on Computer and Communication Security, Zurich, 1997, pp. 8-17.

[4] M. Ben-Or, O. Goldreich, S. Micali and R. Rivest, A fair protocol for signing contracts, IEEE Transactions on Information Theory, vol. 36, no. 1, pp. 40-46, 1990.

[5] S. Devane, M. Chatterjee and D. Phatak, Secure e-commerce protocol for purchase of e-goods - using smart card, In the 3d IEEE International Symposium on Information Assurance and Security, Manchester, 2007, pp. 9-14.

[6] N. Ferguson and B. Schneier, Practical Cryptography. Indianpolis, Indiana: Wiley, 2003

[7] N. Heintze, J. Tygar, J. Wing and H. Wong, Model checking electronic commerce protocols, in Proceedings of the 2nd USENIX Workshop in Electronic Commerce, Oakland, California, 1996, pp. 146–164.

[8] J. Hernandez-Ardieta, A. Gonzalez-Tablas and B. Alvarez, An optimistic fair exchange protocol based on signature policies, Computers Security, vol. 27, no. 7-8, pp. 309-322, 2008.

[9] S. Kremer, O. Markowitch and J. Zhou, An intensive survey of fair non-repudiation protocols, Computer Communications, vol. 25, no. 17, pp. 1606-1621, 2002.

[10] H. Li, W. Kou and X. Du, Fair E-Commerce Protocols without a Third Party, in Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06), Italy, 2006, pp. 324-327.

[11] A. Nenadic, N. Zhang, B. Cheetham and C. Goble, RSA-based certified delivery of e-goods using verifiable and recoverable signature encryption, Journal of Universal Computer Science, vol. 11,no. 1, pp. 175-192, 2005.

[12] A. Nenadic, N. Zhang and Q. Shi, RSA-based verifiable and recoverable encryption of signatures and its application in certified e-mail delivery, Journal of Computer Security, vol. 13, no. 5, pp. 757-777, 2005.

[13] I. Ray, I. Ray and N. Narasimhamurthy, An anonymous and failure resilient fair-exchange e-commerce protocol, Decision Support Systems, vol. 39, pp. 267– 292, 2005.

[14] I. Ray and I. Ray, An Optimistic Fair Exchange E-Commerce Protocol with Automated Dispute Resolution, in Proceedings of 1st Electronic Commerce and Web Technologies Conference, London, Lecture Notes in Computer Science, vol. 1875, 2000, pp. 84-93.

[15] I. Ray and I. Ray, Fair exchange in e-commerce, ACM SIGecom Exchange, vol. 3, no. 2, pp. 9–17, 2002.

[16] R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120 –126, 1978.

[17] N. Zhang, Q. Shi, M. Merabti and R. Askwith, Practical and efficient fair document exchange over networks, the Journal of Network and Computer Applications, vol. 29, no. 1, pp. 46-61, 2006.

[18] Q. Zhang, K. Markantonakis and K. Mayes, A practical fair exchange e-payment protocol for anonymous purchase and physical delivery, in Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications, UAE, 2006, pp. 851-858.