Anonymity and Fair-Exchange in e-Commerce Protocol for Physical Products Delivery

Cătălin V. Bîrjoveanu
Department of Computer Science, "Al.I.Cuza" University of Iaşi, Iaşi, Romania

Keywords: Electronic Commerce Security, Anonymity, Fair-Exchange, Security Protocols.

Abstract: Fair exchange and customer's and merchant's anonymity are two important properties of e-commerce transactions. There is to date a variety of proposed e-commerce protocols to achieve fair exchange and customer's anonymity for transactions involving digital products. For physical products delivery there is no e-commerce protocol to provide fair exchange and customer's and merchant's anonymity. In this paper, we propose the first e-commerce protocol for physical products delivery that will provide fair exchange in all circumstances, anonymity of customer and merchant for any collusion that can be formed, non-repudiation, integrity and confidentiality of data exchanged between the parties.

1 INTRODUCTION

The electronic commerce (e-commerce) has grown rapidly and dynamically, becoming a part of everyday life. It is difficult for the customer and the merchant to trust each other in the online environment. There are many proposed solutions in which the customer sends the payment first, but in this case the customer may have losses if the merchant behaves dishonest and does not send the product to customer. Also, if the merchant sends the product first, then the customer might not send the payment to merchant, and in this way the merchant is prejudiced. Therefore, in order to solve the situations like the ones mentioned above, is necessary to design the *fair-exchange e-commerce protocols*. The fair-exchange ensures that either the customer gets the product and the merchant gets the payment for product, or none do.

Fair-exchange protocols are used in different contexts to exchange payments and digital products (as computer software, digital books, etc.) (Li et al., 2006); payments and physical products (as laptops, phones, etc.) (Alaraj, 2012),(Li et al., 2006),(Zhang et al., 2006); email and receipt; and two digital signatures on a document.

To achieve fair exchange, most of the proposed protocols are based on a Trusted Third Party (TTP) that acts like item validator or to solve the disputes.

Anonymity of customer and merchant is also a key property that must be considered in a fair-exchange e-commerce protocol. The customer may want not to reveal sensitive data of his identity (as credit card number, information about customer's bank, customer's account number) so that this information can not be used by merchant in commercial purpose to build spending habits of the customer. An e-commerce protocol provides the customer's anonymity if no party and no coalition between parties can make a link between the true identity of the customer and actions taken by him. Also, the merchant may want to remain anonymous in e-commerce transactions. For example, a merchant who has business in many areas may want that his customers can not link transactions where the merchant is involved in all these areas.

Among the protocols proposed to date for physical products delivery (Aimeur et al., 2006), (Alaraj, 2012), (Li et al., 2006), (Zhang et al., 2006), none of them provide fair exchange and customer's and merchant's anonymity. In (Zhang et al., 2006), the authors claims that the proposed e-commerce protocol for physical products delivery guarantees fair exchange and anonymity of both the client and the merchant. There are several problems with the protocol proposed in (Zhang et al., 2006). First, in the physical delivery of the product to cabinet is not take into consideration a delivery agent. Thus, the protocol works only in certain particular scenarios in which the merchant can directly access the cabinet from which the client collects the product. This is difficult to achieve in an e-commerce environment in which the customer and the merchant are not lo-

V. Bîrjoveanu C..
Anonymity and Fair-Exchange in e-Commerce Protocol for Physical Products Delivery.
DOI: 10.5220/0005508801700177
In *Proceedings of the 12th International Conference on Security and Cryptography* (SECRYPT-2015), pages 170-177
ISBN: 978-989-758-117-5

cated in the same geographical area. Secondly, it does not provide anonymity of customer and merchant. There are collisions between two parties (such as customer and merchant's bank) that destroy the merchant's anonymity and, also there are collisions between three parties (such as merchant, merchant's bank and customer's bank) that destroy the customer's anonymity. In (Aimeur et al., 2006) is proposed a protocol for physical product delivery that ensures the customer's anonymity, but does not discuss how to make the payment and how it ensures fair-exchange. In (Alaraj, 2012) and (Li et al., 2006) are proposed e-commerce protocols that ensure fair-exchange for physical product delivery, but they do not take into consideration the anonymity issue.

In this paper, we propose the first protocol for physical products delivery that will provide fair exchange in all circumstances, anonymity of customer and merchant for any collusion that can be formed, non-repudiation, integrity and confidentiality of data exchanged between the parties.

The paper is structured as follows: section 2 gives an informal description of our protocol, section 3 presents the protocol, section 4 provides an analysis of the proposed protocol and section 5 contains the conclusion.

2 INFORMAL DESCRIPTION

The goals of our protocol are to obtain the fair exchange of physical product and electronic payment between a customer and a merchant and also to provide anonymity for both of them. The protocol uses an online Trusted Third Party (TTP) that will validate the coins of the customer and will provide fair exchange of items if any party misbehaves or prematurely aborts.

Both customer and merchant may choose to remain anonymous during the protocol execution. To ensure anonymity in the payment phase, our protocol uses the electronic cash payment mechanism based on group blind digital signatures on behalf of the banks proposed in (Lysyanskaya and Ramzan, 1998). This mechanism provides anonymity of the customer and anonymity of the bank that issues the electronic cash. Moreover, after the way it is used in our protocol, it provides anonymity of the merchant in the payment phase.

To ensure anonymity of the customer and the merchant in the physical product delivery phase, our protocol is based on existence of a delivery agent whose role is to take the product from a source cabinet and provide it to a destination cabinet. Both source cabinet and destination cabinet provides access to the physical products by passwords to conceal true identity of the customer and the merchant.

Informally, the protocol works as follows:

The customer decides the product he wants to buy and in what follows the customer buys a digital coin of appropriate value from his bank and validates this coin to TTP. The customer sends to the merchant the purchase order and the digital signature of TTP on the encrypted coin. The merchant uses a delivery agent to send the product to the customer. After the product is posted to the destination cabinet, the customer collects the product and he provides to the destination cabinet an evidence of the product collection and sends to the merchant the decryption key of the coin and his bank's signature on the coin. The merchant sends the coin to his bank for redemption. The merchant's bank verifies the validity of the coin and then transfers the coin value in the merchant's account.

3 THE PROTOCOL

In the Table 1 are presented the notations used in the description of the proposed protocol.

3.1 Assumptions

The following assumptions are made for our protocol: (1) All parties use the same algorithms for encryption, hash, digital signature and the same group blind digital signature protocol mentioned in the Table 1. (2) Cryptographic algorithms are strong enough. (3) TTP is the group manager, namely Central Bank, that is known by all parties implied in protocol. TTP does not misbehave or collude with any of parties to provide benefits to another party. (4) C and M each have an account to their bank. (5) All banks from group and group manager share a commit-buffer in that the transaction value is stored until the transaction is completed successfully or aborted. (6) All banks from group and group manager maintain a global list of coin's serial already spent, validated but unspent, or canceled, to allow any bank to check a digital coin for double-spending or double-canceling. Each record in the list includes besides the coin's serial, a *spent* flag. The value of this flag corresponds to the current state of the coin. The *spent* flag has three possible values: $spent = 0$ means that the coin is validated by TTP but not yet spent, $spent = 1$ means that the coin has already been spent, $spent = 2$ means that the coin has already been canceled. (7) A source cabinet SC exists, where the physical product is placed by M, and DA can take the product from SC only by knowing

Table 1: Notations used in the protocol description.

Symbol	Interpretation
C/C', M/M'	True identity/pseudo identity of the customer and the merchant
CB, MB, DA, TTP	Customer's Bank, Merchant's Bank, Delivery Agent and Trusted Third Party
SC/DC	The source/destination cabinet from which the product must be taken/posted by DA
SC_{addr}/DC_{addr}	The mailing address of SC/DC
C_{acct}/M_{acct}	Customer's/Merchant's bank account with CB/MB
Pid	The identifier of the product that C wants to buy from M in the current transaction t_i
Pr, $Quantity$, Po	Price, quantity, purchase order used to order the product with Pid identifier
$A \rightarrow B : m$	A sends the message m to B
$DC \rightarrow DA \rightarrow M' \rightarrow C' : m$	DC sends to DA the message m that is forwarded by DA to M' and by M' to C'
A_{pub}, A_{prv}	Public/private key pair of A
A'_{ipub}, A'_{iprv}	One time public/private key pair of A used only in the transaction t_i
$\{m\}_{K'}$	The message m encrypted with the key K'
$h(m)$	The digest of m obtained by applying of a hash function h, such as SHA-1
$sig_A(m)$	(RSA) Digital Signature with the A's private key A_{prv} on $h(m)$
T_{TTP}, L	Timestamp generated by TTP, lifetime of encrypted digital coin's validity
N_A, n, K	Nonce generated by A, digital coin generated by C, AES symmetric key that encrypts n
$sig_{CB}(n)$	CB's signature on n obtained by running the Group Blind Digital Signature Protocol

a password that is set by M when he puts the product in. In the physical delivery of the product phase, we use SC to replace the correspondence's address of M. So, the true identity of M remains hidden if it wants. Also, a destination cabinet DC exists, where the physical product is provided by DA, and C can collect the product from DC only by knowing a password that is set by M. The purpose of using DC is to hide the true identity of C if he wants. SC and DC have the ability to digitally sign messages, verify digital signatures on messages and to check if the password entered by DA, respectively C corresponds to the barcode set on product. After DA/C provides the correct password, SC/DC opens a hatch where packed product is available to DA/C. DC has a video camera mounted that records the moment when C unwraps the packed product and check if the product is the ordered one. DC has a device that allows to C, by pushing a button, to send the encrypted recording to TTP. C uses this feature only in the case is not satisfied with the product as an evidence of wrong product reception. Otherwise, the recording is automatically deleted. (8)Communication channels that are set between parties provides anonymity, except the cases in that the parties choose to reveal their true identities.

3.2 Prelude

We assume that before the starting of the protocol, the following system setup steps are executed: (1) TTP generates a public/private key pair, (TTP_{pub}, TTP_{prv}) and provides the public key TTP_{pub} to C and M. (2) When C and M create accounts to their banks, each of them generates a public/private key pair, (C_{pub}, C_{prv}) and (M_{pub}, M_{prv}), respectively. C provides his public key C_{pub} to CB and M provides his public key M_{pub} to MB. The banks maintain databases with public keys of their clients associated to their accounts. (3) C, respectively M, generates a one time public/private key pair, (C'_{ipub}, C'_{iprv}), respectively (M'_{ipub}, M'_{iprv}) that each of them will use it only in the current transaction. (4) The *Setup* and *Join* phases of the Group Blind Digital Signature Protocol (GBDS Protocol) (Lysyanskaya and Ramzan, 1998) are executed. Briefly, this means that the group manager TTP generates a secret key for group manager and the group's public key. CB and MB obtain from TTP the group membership certificate.

3.3 Protocol Description

In the following we describe our protocol, splitting it in four phases. The messages exchanged in the protocol are shown in the Figure 1.

Phase 1: Buying and Validating Digital Coins. After C finds the physical product he wants to acquire from M, he will contact his bank to buy a digital coin with the value that he must pay to M. C generates a new digital coin that is a number n of 256 bits consisting of a unique coin serial number represented on the first 224 bits and the coin value represented in the last

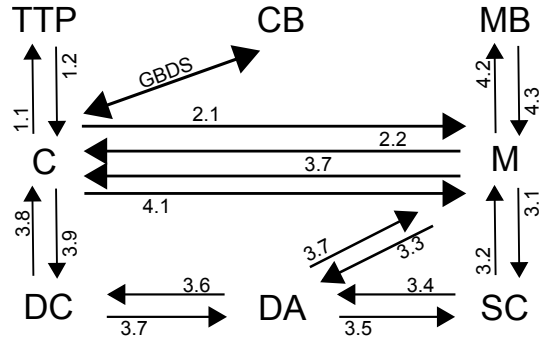

Figure 1: Messages exchanged in protocol.

32 bits. The protocol starts with running the GBDS Protocol between C and CB on the digital coin n. The coin's value is sent by C to CB by signing it. CB transfers the coin value from C_{acct} to the commit-buffer, and after running all steps of the protocol, C obtains $sig_{CB}(n)$-the signature of his bank on the digital coin n on behalf of the bank's group. In this phase, CB knows the identity of the customer and the value of some digital coin purchased by him, but it doesn't know the serial number of the coin because his signature on the coin is blind.

Message 1.1: $C' \rightarrow TTP$:
$\{C', M', C'_{ipub}, K\}_{TTP_{pub}}, \{n\}_K,$
$sig_{CB}(n), sig_{C'}(sig_{CB}(n))$

After C gets the group signature on the digital coin, he validates at TTP an encrypted version of the digital coin. For this, C generates a symmetric key K and sends to TTP a message that contains the following informations: C', M', the one time public key C'_{ipub} that the customer will use only in the current transaction t_i, the key K, which are encrypted with TTP's public key to provide confidentiality; the digital coin n encrypted with the key K; his bank's signature on the digital coin - $sig_{CB}(n)$; and his signature on the $sig_{CB}(n)$.

On reception of the message 1.1, TTP decrypts first encrypted component of the message and obtains the one time public key C'_{ipub} of C and the key K. TTP obtains the digital coin n by decrypting the ciphertext $\{n\}_K$ with the key K and uses the C'_{ipub} to verify the signature of the customer on the signed coin. TTP checks the validity of the digital coin n by verifying the signature $sig_{CB}(n)$ using the group public key, and checks whether this coin has already been spent, or validated by TTP (in a previous request) but not yet spent or canceled, by verifying the *spent* flag of the coin's serial in the global list of coin's serial. If all checks out, TTP add the coin in the list setting the coin's *spent* flag on the value 0, and sends to the customer the following message:

Message 1.2: $TTP \rightarrow C' : T_{TTP}, L, N_{TTP},$
$sig_{TTP}(C', M', \{n\}_K, Val, T_{TTP}, L, N_{TTP})$

The message 1.2 contains a timestamp T_{TTP}, a lifetime of encrypted coin's validity L and a nonce N_{TTP}, all to avoid replay attacks. C checks if T_{TTP} and L are recently enough, and then verifies the TTP's signature that will be used by customer to confirm to the other party (the merchant) that $\{n\}_K$ represents the encryption of a valid coin n (that was signed by a bank from the group, and its lifetime has not expired), the coin is fresh and of the value Val, but without exposing the digital coin n to the merchant.

Phase 2: Agreement on the Transaction's Terms. The agreement phase is initiated by the customer that sends a message to the merchant that represents his intention to buy a product from him.

Message 2.1: $C' \rightarrow M'$:
$Po, C'_{ipub}, sig_{C'}(Po, C'_{ipub}),$
$\{n\}_K, Val, T_{TTP}, L, N_{TTP},$
$sig_{TTP}(C', M', \{n\}_K, Val, T_{TTP}, L, N_{TTP})$

where
$Po = C', M', Pid, Pr, Quantity, Val, DC_{addr}, h(N_C)$

Po contains $h(N_C)$ whose goal is to be used as a barcode on the product and is set by M such that only who knows the password N_C can collect the product from DC_{addr}; N_C is kept secret by C, while M receives a digest of him.

When the message 2.1 is received, M checks the terms of the transaction initiated by C. M verifies if T_{TTP} and L are recently enough, the informations from Po, the customer's signature on Po and the signature of TTP. If M is not satisfied, he sends an abort message to the customer and aborts the transaction. If M is satisfied, the signature of TTP assures him that $\{n\}_K$ represents the encryption of a valid digital coin of value Val from Po. The new nonce N_{TTP} ensures the merchant about freshness of the digital coin. Even if M does not know the digital coin and can't redeem it in this phase, he knows that could redeem it after it will post the product with Pid identifier to DC_{addr}.

Message 2.2: $M' \rightarrow C' : sig_{M'}(sig_{C'}(Po)),$
$\{M'_{ipub}\}_{C'_{ipub}}$

If M is satisfied by the conditions of the message it received, he sends to C a message to ensure C by M's agreement on the terms of transaction specified in the message 2.1. After receiving message 2.2, if C receives an abort message from M, then he aborts the transaction. Otherwise, C obtains M's public key M'_{ipub}, by decrypting $\{M'_{ipub}\}_{C'_{ipub}}$ with his one time private key C'_{iprv} and checks the merchant's signature.

Phase 3: Physical Delivery of the Product. If the phase 2 is successful, M posts the product to SC from

where the product must be taken by *DA*.

Message 3.1: $M' \rightarrow SC : product$

The product has two barcodes set by M: $h(N_M)$ to control the access of *DA* to *SC*, and $h(N_C)$ to control the access of *C* to *DC*. We consider that the product with both barcodes is placed by *M* on a shelf of *SC* that is located at SC_{addr}. We use SC_{addr} to conceal the true identity of the merchant and such to ensure his anonymity.

Message 3.2: $SC \rightarrow M' : sig_{SC}(M', Pid, DA)$

Upon receiving the product from the merchant, *SC* confirms to him by a signed acknowledgment.

Message 3.3: $M' \rightarrow DA : Pid, SC_{addr}, DC_{addr}, \{M', M'_{ipub}, N_M\}_{DA_{pub}}, sig_{M'}(M', Pid, SC_{addr}, DC_{addr}, N_M)$

M sends to *DA* a delivery request message 3.3. N_M has the same goal as N_C, but in this case the password N_M is shared only between *M* and *DA*. *DA* recovers the public key M'_{ipub} of *M* and N_M by decrypting $\{M', M'_{ipub}, N_M\}_{DA_{pub}}$ with his private key, and checks the signature of *M*. If the signature is successfully verified, *DA* is ensured by message's authenticity.

Message 3.4: $SC \rightarrow DA : product$

DA collects the product from *SC* by proving he knows the password N_M.

Message 3.5: $DA \rightarrow SC : sig_{DA}(M', Pid, DA, SC_{addr}, DC_{addr}, N_M)$

To confirm the collection of the product, *DA* sends to *SC* an acknowledgment in the message 3.5.

Message 3.6: $DA \rightarrow DC : product$

Further, *DA* posts the product to *DC* as is specified by merchant in the delivery request.

Message 3.7: $DC \rightarrow DA \rightarrow M' \rightarrow C' : sig_{DC}(M', Pid, DA, DC_{addr})$

Upon receiving the product from *DA*, *DC* confirms him by a signed acknowledgment which *DA* forwards it to the merchant. Thus, the merchant has the proof of posting the product to DC_{addr} and thereafter he forwards the proof to the customer.

Message 3.8: $DC \rightarrow C' : product$

The customer collects the product from *DC* using the password N_C.

Message 3.9: $C' \rightarrow DC : sig_{C'}(M', Pid, DC_{addr}, C', N_C)$

The customer checks if the collected product meets the specifications from *Po*. If the customer is satisfied with the product, he sends a signed acknowledgment to *DC*.

Phase 4: Payment. If the customer collects the product and is satisfied, then he sends to the merchant the message 4.1. **Message 4.1:** $C' \rightarrow M' : \{K\}_{M'_{ipub}}, sig_{C'}(K), sig_{CB}(n)$

The merchant obtains the key *K* and verifies the customer's signature on *K*, then he uses *K* to decrypt the encrypted coin received in the message 2.1 from the customer. *M* verifies the validity of the $sig_{CB}(n)$ using the group public key. If the coin is valid, *M* sends it to *MB* for redemption in the message 4.2.

Message 4.2: $M \rightarrow MB : \{n, sig_{CB}(n), sig_M(n, sig_{CB}(n)), M, M_{acct}\}_{MB_{pub}}$

MB decrypts the received message, checks *M*'s signature, checks that $sig_{CB}(n)$ is a valid signature of some bank from bank's group, using the group public key, without knowing who is the bank that signed the coin. *MB* checks if the coin has already been spent or canceled by checking the value of the *spent* flag of the coin, using the global list of the coins. If all checks are satisfied, *MB* updates the global list by setting the *spent* flag of the coin *n* to the value 1, transfers the coin value from commit-buffer to M_{acct}, and sends to *M* a signed acknowledgment of successfully redemption of the coin. Otherwise, if some check is not satisfied, *MB* sends to *M* an suitable error message.

Message 4.3: $MB \rightarrow M : sig_{MB}(ack)$

4 ANALYSIS OF THE PROTOCOL

4.1 Ensuring Fair-Exchange

In an e-commerce protocol the fair exchange assures that two parties exchange items of value such that either both parties obtain each other's item or none do. Our protocol assures fair exchange if either *C* gets the physical product and *M* gets the payment for product, or none do. If *C* and *M* behave honestly, the proposed protocol assures fair exchange. We will consider all possible scenarios in which *M* or *C* behave dishonest or prematurely abort the protocol. To ensure fair exchange in all this scenarios, extensions of the basic protocol are necessary as we will see below.

If *M* behaves dishonest, then the following scenarios are possible:

1. *M* receives from C' a correct message 2.1, but he doesn't continue the protocol. Such behavior brings no benefit to *M* because he is in possession of an encrypted coin with a key that does not know, so he can't redeem the coin and get the payment. But *C* has bought a coin from his bank, which can not be used by him. In this scenario, *C* initiates the extended protocol providing to TTP the message 2.1 he sent it to *M*. TTP checks the information received from *C* and ask *M* for his agreement on the terms of transaction. If *M* responds to the TTP's request by sending to *C* the

message 2.2, then the basic protocol can continue with the phase 3. If M doesn't respond, then TTP sends to C a cancellation request of the coin in the first message below that further C sends encrypted together with his account information to CB.

$TTP \rightarrow C' : n, sig_{CB}(n), sig_{TTP}(sig_{CB}(n))$

$C \rightarrow CB :$

$\{n, sig_{CB}(n), sig_{TTP}(n, sig_{CB}(n)), C, C_{acct}\}_{CB_{pub}}$

CB checks if $sig_{CB}(n)$ is a valid signature, TTP's signature, if the coin n has not already been spent or canceled by checking the global list of coin's serial. If all checks are satisfied, CB sets the *spent* flag of the coin n to the value 2, transfers the coin value from commit-buffer to C_{acct}, and sends to C a signed acknowledgment of successfully cancellation of the coin n. In this way the coin's value is redeemed by C. Otherwise, if some check is not satisfied, CB sends to C an suitable error message.

2. M receives from C' a correct message 2.1 and sends the message 2.2 to C', but he doesn't posts the product or posts a product that doesn't comply with the specifications from Po. Similarly to the scenario 1, M does not have any benefit from this behavior. If C is not satisfied with the collected product, he pushes the button of the DC's device that allows sending to TTP the recording of the moment when C unwraps the packed product, proving to TTP that the received product is wrong. Also, C sends to TTP all the messages received/sent from/to M. TTP checks the information received from C and send to M all evidence received from C and ask M to post the correct product. If M responds to TTP by sending such a proof, then C can continue the basic protocol with collecting the product from DC. If M doesn't respond to the TTP, then similarly to the scenario 1, TTP will cancel the customer's coin used in the current transaction.
3. M sends to CB many times the same message 4.2 for multiple redemption of the same coin. This scenario is solved in the basic protocol, because MB checks if the coin received in the message 4.2 has already been spent by checking the value of the *spent* flag of the coin.

If C behaves dishonest, then the following scenarios are possible:

1. C collects the product in the message 3.8, but does not send to M' the decryption key of the encrypted coin or sends to M' in the message 4.1 a wrong decryption key. In this scenario, M initiates the extended protocol providing to TTP all the messages received/sent from/to C. TTP checks the current transaction's messages and ask C for digital coin decryption key. According to the response of C, there are three possible scenarios:
   (a) If C responds to the TTP's request by sending to M' the digital coin decryption key K in a message 4.1, then M' can continue the basic protocol with coin redemption.
   (b) If C doesn't respond to TTP, then TTP sends to M' the digital coin decryption key K that is in possession of TTP from phase 1:
   $TTP \rightarrow M' : \{K\}_{M'_{ipub}}, sig_{TTP}(K), sig_{CB}(n).$
   M can decrypt the digital coin using the key K received from TTP, checks the validity of $sig_{CB}(n)$ and can continue the basic protocol with the message 4.2 for coin redemption.
   (c) If C falsely claims that he doesn't provide the decryption key because M' didn't posted the product or M' posted another product than the ordered one. To claim this, C must submit to TTP the proof of wrong product reception: the recording of the moment when C unwraps the packed product. This proof is sent on a secure channel from DC's device to TTP, so TTP can not be fooled. Further, this scenario is solved similarly with the previous (b) scenario.
2. C sends the same digital coin to TTP (in the message 1.1) in two different sessions of validating encrypted digital coins, to initiate two different buying transactions with two distinct merchants. This scenario is solved in the basic protocol because all banks and TTP maintain a global list of coin's serial already spent, validated but unspent, or canceled. On reception from C of the first request for validating the coin, TTP adds the coin to the global list of coins, and therefore any new validation request of the same coin from C is detected by TTP. Thus, TTP detects double spending from C and aborts the second transaction.
3. C sends to M' in the message 2.1, an encrypted coin already spent. This scenario is solved in the basic protocol. If the coin wasn't used to buy from M', then M' detects this by verifying the TTP's signature that validated the encrypted coin. Otherwise, if the coin was already used to buy from M', then M' can check this by verifying N_{TTP}. So, M detects double spending from C and aborts the transaction.
4. C sends to M' in the message 2.1, an encrypted coin of insufficient value. This scenario is solved in the basic protocol, because M checks if the value from Po corresponds with the encrypted coin's value validated by TTP. If these values are not equal, then M aborts the transaction.

5. C sends to M' in the message 2.1, an encrypted coin that has already been canceled. C's intention is to buy a product without paying for it. This scenario is solved in the basic protocol, because M checks if T_{TTP} and L are recently enough. In this scenario, these values are not recently enough, and M aborts the transaction.
6. C sends to his bank many times the cancellation requests of the same coin for multiple redemption of the same canceled coin. This scenario is solved in the extended protocol, because CB checks if the coin received in a cancellation request has already been canceled.

DA, SC, and DC send signed acknowledgments of product collection in the phase 3 of the basic protocol. Moreover, these three entities have no interest not to follow the protocol steps, because their interest is to get profit from fees for such services provided in e-commerce transactions. Each party involved in the protocol must keep a record of every message sent or received in protocol including signed acknowledgments of DA, SC, and DC. If one of the parties mentioned above (DA, SC, or DC) behaves dishonest, the other parties send the records to TTP to trigger an off-line mechanisms to ensure fairness.

4.2 Analysis of Anonymity

One of the main objectives of our protocol is to ensure the anonymity of the customer and the merchant. In what follows, we show that the protocol proposed ensures anonymity of the customer and the merchant in any possible collusion scenario.

An e-commerce protocol provides customer's anonymity if no party and no coalition between parties can make a link between the true identity of the customer and actions taken by him in the e-commerce transaction. More exactly, our protocol ensures the customer's anonymity if the true identity of the customer, C, can't be linked with the pseudo identity of the customer, C', which he uses in the e-commerce transaction.

Ensuring the customer's anonymity rises two problems that must be solved in this regard: guaranteeing the customer's anonymity in the payment's phase, and guaranteeing the customer's anonymity when the customer collects the physical product. To provide the anonymity of the customer in the payment phase, we use an electronic cash payment that is based on group blind digital signatures on behalf of the banks. The only steps from our protocol in that the customer uses his true identity are the GBDS Protocol's steps, because CB must know C_{acct} to charge it with the coin's value. CB knows only that C bought a coin with a certain value, but it doesn't know the serial of the coin. Following, CB can't associate C with the coin bought by him, maintaining thus the anonymity of the customer. Another essential feature of the GBDS Protocol is the customer bank's anonymity: any party can check if $sig_{CB}(n)$ is valid, but without knowing who is the bank that signed the coin. CB is not known by any other party (except C), so, CB can't participate in no coalition with any other party to obtain sensitive information to destroy the customer's anonymity.

Table 2: Informations that each party knows after protocol execution.

	Entity						
Info	C	M	CB	MB	DA	SC	DC
C	y	n	y	n	n	n	n
M	n	y	n	y	n	n	n
CB	y	n	y	n	n	n	n
MB	n	y	n	y	n	n	n
DA	n	y	n	n	y	y	y
SC	n	y	n	n	y	y	n
DC	y	y	n	n	y	y	y
C'	y	y	n	n	n	n	y
C_{pub}	y	n	y	n	n	n	n
C'_{ipub}	y	y	n	n	n	n	y
n	y	y	n	y	n	n	n
$C\&t_i$	y	n	n	n	n	n	n
$C'\&t_i$	y	y	n	n	n	n	y
M'	y	y	n	n	y	y	y
M_{pub}	n	y	n	y	n	n	n
M'_{ipub}	y	y	n	n	y	n	n
$M\&t_i$	n	y	n	n	n	n	n
$M'\&t_i$	y	y	n	n	y	y	y

We show in the Table 2, the information that each party in the protocol knows after protocol execution. The information have the following meaning. For example, we consider the first row: y under the column C and CB means that C and CB know C - the true identity of the customer; n under the column M, MB, DA, SC and DC means that the true identity of the customer is not known to M, MB, DA, SC and DC. $C\&t_i$ means that C performs the transaction t_i. The meaning is extended for $C'\&t_i$, $M\&t_i$, $M'\&t_i$.

From Table 2 we observe that no party alone has sufficient information to link the true identity of the customer, C, with the pseudo identity C'. Only C can

disclose this information if he wants.

2-party Collusion. The 2-party collisions where *M* can occur are *M* and *MB*, *M* and *DA*, *M* and *SC*, *M* and *DC*. From this collisions, *M* doesn't get more knowledge than he already had, and the other parties get the knowledge of *M*. By colluding between *DA* and *SC*, *DA* doesn't get more knowledge than he already had. The coalitions between *DA* and *DC*, *DC* and *SC* get only information that C' performs the transaction t_i. All other 2-party collisions that could form are between *CB* and *M*, *CB* and *MB*, *CB* and *DA*, *CB* and *SC*, *CB* and *DC*, *MB* and *DA*, *MB* and *SC*, *MB* and *DC*, but they are not possible because the parties involved do not know each other.

3-party Collusion. From the analysis above, we observe that *M* can be involved in the following 3-party collisions: *M*, *MB*, *DA*; *M*, *MB*, *SC*; *M*, *MB*, *DC*; *M*, *DA*, *SC*; *M*, *DA*, *DC* and *M*, *SC*, *DC*. These coalitions are reduced to 2-party collisions in which *M* is involved because from these 3-party collisions *M* doesn't get more knowledge than he already had. One more 3-party collusion can be formed between *DA*, *SC* and *DC*, but it does not get more information about customer as against the 2-party collisions *DA* and *DC*, or *DC* and *SC*.

4-party Collusion. The only 4-party collisions (*M*, *MB*, *DA*, *SC*; *M*, *MB*, *DA*, *DC*; *M*, *MB*, *SC*, *DC*; *M*, *DA*, *SC*, *DC*) are reduced to 3-party collisions because *M* already knows all information known to the other parties from this collisions.

5-party Collusion. The only possible 5-party collusion *M*, *MB*, *DA*, *SC* and *DC*, is reduced to 3-party collisions by same arguments as above.

Regarding merchant's anonymity, he uses his true identity only in communication with *MB* in the coin redemption phase. *MB* knows personal information about *M* (such as M_{acct}), but it doesn' t know information about the pseudo identity M'. Moreover, *MB* is not known to any other party (except *M*), and can't participate in coalitions with any other party to destroy the merchant's anonymity. Our protocol doesn't use the merchant's correspondence address, but uses a source cabinet where the product is placed by *M*.

Is easy to see from Table 2 that no party alone has sufficient information to link the true identity of the merchant, *M*, with the pseudo identity M'.

2-party Collusion. The 2-party collisions where *C* can occur are *C* and *CB*, *C* and *DC*. From collusion between *C* and *CB*, *C* doesn't get more knowledge than he already had, and *CB* gets the knowledge of *C*. From collusion between *C* and *DC*, *C* gets as new information the identity of *DA*, and *DC* gets the knowledge of *C*. By colluding *DA* and *SC*, *DA* and *DC*, *SC* and *DC* are obtained only information about M', no information about *M*. No other 2-party collusion is possible because there is no other party to know another party.

3-party Collusion. The 3-party collisions that can be formed are *C*, *CB*, *DC*; *C*, *DA*, *DC*; *C*, *SC*, *DC*; *DA*, *SC*, *DC*, and these get only the information that M' performs the transaction t_i, without any information about *M*.

4-party Collusion. The only 4-party collisions *C*, *CB*, *DC*, *DA*; *C*, *CB*, *DC*, *SC* and *C*, *SC*, *DC*, *DA*, are reduced to 3-party collisions.

5-party Collusion. The only 5-party collusion *C*, *CB*, *DA*, *SC* and *DC*, is reduced to 3-party collusion *C*, *DA*, and *DC*.

5 CONCLUSIONS

By integrating an electronic cash payment mechanism and using a suitable mechanism for physical products delivery, the proposed protocol is the first to provide fair exchange between physical products and payments in all circumstances, and customer and merchant's anonymity in any collusion scenario. All of these makes the proposed protocol a candidate to be used effectively in practice for electronic transactions that implies buying physical products.

Future work will include formal proving of the correctness of the proposed protocol using strand spaces framework or formal verification using automated model checking tools (e.g. AVISPA).

REFERENCES

Aimeur, E., Brassard, G., and Mani Onana, F. (2006). Secure anonymous physical delivery. *IADIS International Journal on WWW/Internet*.

Alaraj, A. (2012). Fairness in physical products delivery protocol. *International Journal of Computer Networks & Communications (IJCNC)*.

Li, H., Kou, W., and Du, X. (2006). Fair e-commerce protocols without a third party. In *11th IEEE Symposium on Computers and Communications*. IEEE.

Lysyanskaya, A. and Ramzan, Z. (1998). Group blind signature: a scalable solution to electronic cash. In *Financial Cryptography*. Springer.

Zhang, Q., Markantonakis, K., and Mayes, K. (2006). A practical fair exchange e-payment protocol for anonymous purchase and physical delivery. In *4th ACS/IEEE International Conference on Computer Systems and Applications*. IEEE.