Anonymous Payment in a Fair E-commerce Protocol with Verifiable TTP

M. Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila,
and Llorenç Huguet-Rotger

Universitat de les Illes Balears, C. Valldemossa, Km. 7.5, 07122,
Palma de Mallorca, Spain
mpayeras@uib.es

Abstract. An electronic purchase represents an exchange between money and a digital product or the receipt of a physical product. Atomicity is a desired feature for electronic payment systems because it allows fair purchases. We present a fair payment protocol useful in electronic purchases involving electronic coins. The protocol is fair, asynchronous and efficient, and can be used with existing payment systems. Moreover we have evaluated the role of the TTP in the fair exchange protocol, showing that the incorrect behavior of the TTP can be demonstrated in all cases, so the TTP is verifiable.

Keywords: Atomicity, Verifiability, Fair exchange, Electronic coins, Anonymity.

1 Introduction

Some electronic services require the atomic exchange of elements between two or more users. The fair exchange of values always provides a fair deal to all users. Thanks to fairness, at the end of the execution of an exchange, all parties have the element they wanted to obtain. Instead, if the execution has not been successful, no party has the desired element.

Among the electronic applications that require a fair exchange of information we can find electronic contract signing, certified electronic mail and payment in exchange for a receipt (or in the event of purchase of a digital product, the exchange for a product).

An electronic purchase represents the exchange of a payment for a receipt or for a product in which the payment can be carried out by means of different types of systems. One of them is electronic cash.

In the purchase of a tangible product, a receipt can be used as a proof of the payment to demonstrate, without possible repudiation on the part of the merchant, that the user has carried out the payment. When the payment is carried out in the purchase of a digital product, the exchange of the money for the product can be carried out directly, but the need of fairness remains in the exchange, since the buyer doesn't want to take a risk paying without the security that he will receive the product, while the merchant doesn’t want to send the product before receiving the payment, neither (digital goods can be copied and therefore demanding the refund of the product doesn't make sense).

S. Katsikas, J. López, G. Pernul (Eds.): TrustBus 2005, LNCS 3592, pp. 60 – 69, 2005.

The classification of fair exchange protocols is based in the presence or absence of trusted third parties (TTP) during the execution of the protocol, and in this case, the dependence degree.

- **Protocols for fair exchange that don't require a TTP**. These protocols, by themselves, guarantee the security of the exchange and therefore they don't require the intervention of any TTP. This independence of a third party is a desirable feature, but some of these protocols require great number of interactions between the parties as good as high calculation complexity.
- **Protocols for fair exchange of values with intervention of a TTP**. Among them, the protocols that require the intervention of the TTP in each execution of the protocol can be distinguished from those where the TTP only intervenes in case the exchange doesn't conclude with success (optimistic protocols). The constant presence of the TTP has some inconveniences: the cost that the service represents for users, the possible congestion in the communication with the TTP and the additional delay caused by the communications between the user and the TTP in each execution of the protocol. In optimistic protocols the participation of the TTP is limited to some cases. They have an exchange subprotocol in which the TTP doesn’t intervene. If the exchange subprotocol concludes correctly, then it is not necessary to involve the TTP in the execution. If not, another subprotocol will be executed, involving the TTP. In function of the presented proofs, the TTP will be able to send messages or to make decisions to guarantee the fairness of the exchange.

From the previous classification it is deduced that for efficiency reasons it is desirable that a TTP exists, but that it only intervenes to solve disputes when the execution of the protocol leads to an unfair situation. In consequence, the desirable properties are formulated for optimistic exchanges [1]. These features are:

- **Fairness.** When the execution of the protocol concludes, either both parties have the wanted objects, or none of the parties has them.
- **Timeliness.** A protocol provides timeliness if all of the parties can, at any moment, conclude the exchange preserving the fairness.
- **Non repudiation.** After the exchange, each participant can prove the origin of the object that he or she has received, that is to say, a party cannot refuse the emission of the own object.
- **Verifiability of the TTP.** If the TTP intervenes in the execution of the exchange and acts incorrectly, then the fraudulent behavior should be demonstrable.
- **Efficiency.** An efficient protocol will use the smallest possible number of interactions among the users.
- **Privacy.** A protocol is confidential if it allows hiding the content of the exchange, even to the TTP if it is the case.

2 Payment per Receipt Protocols

In a payment using a credit card, the purchase order that includes the number of the card (and the signature) is exchanged for the receipt of the payment or for the product.

The exchange of a signed purchase order in exchange for a receipt of the payment can be considered an application of the contract signing protocols. On the other hand, in payments with electronic cash, the coin becomes the element to exchange on the part of the buyer, and the exchange cannot be considered solved with contract signing protocols, since there are specific situations that origin the interruption of the exchange and could cause the loss of the coin for the two parties or the loss of anonymity to some of them. For example, when in an off-line electronic cash system an error causes that a payer doesn't know if the receiver has received or not the coin, the payer cannot take the risk to use the coin again since if the payment had been concluded, the payer not only would be identified but also he or she would be accused of reutilization.

Besides providing fairness, it is desirable that the exchange protocols allow demonstrating which object the other party has received, and therefore, in the event of later dispute, they can present proofs of the exchange. Purchase protocols can provide atomic exchange, certified delivery for some or all the parties involved in the exchange, or both.

In [9], atomicity of the money is defined as the feature that avoids the creation or destruction of money during its transfer. Therefore, these protocols don't provide fair exchange. Atomicity of goods is defined also in [9] and is applied to the protocols that not only present atomicity of the money but also allow the fair exchange between the product and the coin.

Certified delivery [9] provides coin and goods atomicity, and also provides evidences to both parties of what they have sent and of what the other party has received. This certified delivery can be unilateral or bilateral [4], in function of how many parties possess reception proofs. Certified and atomic delivery [7] provides atomicity of both the good and the coin and the parties have come to an agreement in the initial negotiation and the exchange provides proofs that the goods and the coin have been received. It has, at the same time, atomicity of the good and certified delivery. Finally, distributed atomic purchase [7] provides atomicity of the money and of the good when more than one merchant is involved in the purchase.

The solution adopted in [9] is useful in the event of shortcomings of the system, but it is not useful in case of fraud intent. The system uses a coordinator that knows the identity of all the parts, so the system doesn't allow anonymous payments. The protocols described in [4, 6, 9, 10] carry out the exchange with an on-line TTP. In [6], the active TTP is a blackboard where all users can read and write. [4] provides unilateral certified delivery, and the bank that acts as a TTP is involved in the payment. [9] presents an on-line payment where the bank also acts as a TTP and it guarantees the fair exchange during the payment. Similar solutions are [7] and [8] where a coordinator of on-line payment is used.

Other solutions, as [5], don't need a TTP. In this case the authors opted to divide the coin in two parts that will be sent before and after the reception of the good. The merchant is not protected; he can't contact a TTP if he doesn't receive the second part of the coin. Coins can have an ambiguous state if the buyer doesn't take the risk of being identified in case of reutilization. As a conclusion, it doesn't provide atomicity, and it only provides little protection to the payer. [12] doesn't satisfy the ideal features, if the exchange doesn't finish in a satisfactory way, the client won't be able to get the good, he will only be able to recover the money, that is to say, the exchange can be cancelled, but not finished. [11] doesn't include the analysis of the payment system that would be used in the exchange. The purchase is not certified; the merchant cannot demonstrate that the client has received the good.

According to the ideal features, the objective is a certified and atomic purchase in a protocol that provides anonymity, at least to the payer, maintaining the anonymity that provides the payment system.

3 Features of the Proposed Protocol

The proposed protocol presents the following features:

- **Bilateral certified delivery:** The merchant can demonstrate that the buyer has received the product or receipt. On the other hand the buyer can demonstrate that the merchant has received the payment, as well as which element he has received.
- **Anonymity:** The buyer will be anonymous if the payment system used in the exchange is an anonymous one, and will remain anonymous although he contacts the TTP. If the exchange concludes with the participation of the TTP and the client uses the coin again, the reutilization is detected as usual and the buyer is identified.
- **Exchange:** The payment is carried out in **two stages.** In the first one, a part of the coin is sent to the merchant while in the second stage a secret proof related with the coin, only known by the payer, is revealed. The payee cannot deposit the coin if he doesn't receive the second part of the payment. However, with the first part of the coin the payee can contact the TTP to finish the exchange.
- **Security of the payment:** The exchange protocol keeps the security of the payment system used in the exchange: it can detect double spending, identify double spenders and prevent overspending.
- **Off-line TTP:** The TTP is involved only to solve conflicts when the exchange has not been completed or some party has acted maliciously.
- **Efficient and functional with habitual payment systems:** The exchange protocol is appropriate for the use with various electronic cash systems. The features that these systems must satisfy are:

  - Coin created by the bank (debit system).
  - The bank cannot relate the coins with the payer's identity: anonymous coins.
  - The merchant can verify the coin when he receives it. He cannot prevent double spending.
  - The payment has a challenge-response stage.
  - Double spenders are identified a posteriori.
  - The payer remains anonymous if he behaves correctly.

  These features are given in numerous electronic cash protocols, like [2] and [3] that have been adapted and used to prove the applicability of the exchange protocol.
- **Exchange finalization:** once the purchase commitment is established (2 steps), the protocol allows to finish the exchange, not only to cancel it.

4 Description of the Fair E-commerce Protocol

Three parties are involved in the protocol: the buyer (or payer), the merchant (or payee) and the TTP. The buyer wants to buy a product identified as *Product_code*. The notation used in the description of the protocol is included in table 1.

Table 1. Notation

C	Anonymous buyer	PR_x	*x*'s private key
M	Merchant	PU_x	*x*'s public key
T	TTP	Sign[x,y]	Signature on *x* that proves the knowledge of a secret element, *y*
H[]	Hash Function	a	First part of the coin
E_k[]	Ciphering with secret key *k*	b	Secret element, second part of the coin
D_k[]	Deciphering with secret key *k*	CANCELLED FINISHED	Boolean variables, false by default.
Id	Exchange identifier	Product_code	H[Product_description]

The protocol is formed by three subprotocols: exchange, cancellation and finalization. The exchange subprotocol, described in table 2, is formed by the following steps:

- **Step 0. Product selection and purchase order.** *C* sends the purchase order referencing *Product_code* and the first part of the coin that will be involved in the payment to *M*.
- **Step 1. First part of the purchase commitment: challenge.** *M* generates a challenge for the payment (*pc*). This will be the challenge used in the electronic payment system. *M* encrypts the requested product or receipt using the session secret key *k*, then the secret session key using *T*'s public key. Finally, *M* signs the relationship between both elements and sends them to *C*.
- **Step 2. Second part of the purchase commitment: response to the challenge.** *C* responds to the payment challenge and signs the relationship between *a* (first part of the coin) and the ciphered product or receipt (*c*), proving that he knows the second part of the coin (secret element *b*). The response to the payment challenge, *rpc*, can be used to identify the client in case of double spending. Once this message is received, both parties can request the finalization of the exchange.
- **Step 3. *M* sends the session key.** After the reception of the message sent in step 2, *M* verifies the answers received from *C*: *Sign(d, b)* and *rpc* and sends the key *k* for the deciphering of the product or receipt.
- **Step 4. *C* sends the secret proof.** *C* sends the secret proof that will allow the deposit of the coin.

Steps 1 and 2 of the exchange subprotocol form the purchase commitment. After step 2, *T* can finish the exchange at *C* or *M* request executing the finalization subprotocol. If the exchange is stopped before the reception of step 2, the commitment is not established, and *T* cannot conclude it. In order to invalidate the elements sent in step 1, *M* can request the cancellation of the exchange using the cancellation subprotocol. A protocol without the fourth step would be vulnerable; without the fourth step the receiver of the coin, after step 2, could lie and begin the cancellation subprotocol claiming that he hadn't received the coin. Then, if *C* uses the coin again he would lose privacy because his identity would be revealed.

Table 2. Exchange subprotocol

EXCHANGE SUBPROTOCOL	
0. C → M:	Product_code, a
1. M → C:	pc, c = E_k(product), $K_t = PU_T(k)$, $H_M = PR_M\{H[H(c), Kt], Id\}$
2. C → M:	rpc, d = H[a, c, Id], Sign(d, b)
3. M → C:	$K_M = PR_M$ (k, Id)
4. C → M:	b

The cancellation and finalization subprotocols are executed between *C* or *M* and *T*, whenever the exchange subprotocol doesn't conclude successfully. *T* can choose between concluding and canceling the exchange in function of the presented proofs, the purchase order and previous decisions.

Table 3. Cancellation subprotocol

CANCELLATION SUBPROTOCOL		
	M → T:	a, c, k_T, h_M, $h_{MT1} = PR_M(c, k_t, h_M, a)$
IF (FINISHED = TRUE)	T → M:	rpc, d, Sign(d, b), $P_{TM} = PR_T(b)$
ELSE	T → M: T:	Cancellation proof = PR_T(“cancelled”, h_M) CANCELLED = TRUE

Table 4. *C*'s finalization subprotocol

C's FINALIZATION SUBPROTOCOL		
	C → T:	a, pc, c, k_T, h_M, rpc, d, Sign(d, b), b
IF (CANCELLED = TRUE)	T → C:	Cancellation proof = PR_T(“canc.”, Sign (d, b))
ELSE	T → C: T:	PR_T(k) FINISH = TRUE

The cancellation subprotocol can only be executed by *M* in case the purchase commitment doesn’t conclude (*M* doesn’t receive the message of the second step of the exchange subprotocol). The finalization subprotocol can be executed by both parties once the purchase commitment has concluded, that is, if *C* doesn't receive the key *k* (step 3) or the merchant doesn't receive the proof of the coin, *b* (step 4). The subprotocols are described in tables 3 and 4.

Table 5. *M*’s finalization subprotocol

M’s FINALIZATION SUBPROTOCOL		
	M → T:	a, pc, c, k_T, h_M, rpc, d, Sign(d, b), h_{MT2} = PR_M(c, k_t, h_m, *a*, rrp, Sign (d, b))
IF (FINISHED = TRUE)	T → M:	P_{TM} = PR_T(b)

ELSE	T → M:
	T:
		Deposit authorization without b
		FINISHED = TRUE
		CANCELLED = FALSE

4.1 Fairness

In order to evaluate the fairness of the protocol, we will analyze all possible situations derived from the execution of the protocol, involving or not involving the TTP.

- **Concluded Exchange**. If the exchange has been carried out without problems, *C* has the product or the receipt ($D_k(c)$) and he or she can demonstrate that it is the received product or receipt ($H_M=PR_M\{H[H(c), Kt], Id\}$). Moreover, *C* can demonstrate that he carried out the payment, since he can provide the key: $K_M=PR_M(k, Id)$. *M* has both parts of the payment: *a*, *rpc*, *Sign(d, b)* and the secret proof of the coin: *b*. With the last element he can demonstrate that *C* has received the product or receipt.
- **Unfinished exchange.** If the exchange doesn’t conclude successfully, both parties can contact *T* and begin the execution of the finalization or cancellation subprotocols. The exchange can be broken up in different stages:
  - ***M* doesn't receive the message of step 2.** If either step 1 or step 2 are not executed, the purchase commitment is not settled down. *M* can request the cancellation of the exchange while *C* can request its finalization. *M* cannot request the finalization of the exchange since he or she doesn't have the element *Sign(d, b)*. In function or the request order, the following situations are possible:
    - ***C* finishes, *M* cancels:** *T* sends the key *k* to *C* and *b* to *M*.
    - ***M* cancels, *C* finishes:** *T* sends a cancellation proof to *M*. *C* won't receive the key, *k*.
  - ***C* doesn't receive the message of step 3 or *M* doesn't receive the message of step 4.** *M* can finish or cancel the exchange while *C* can only finish the exchange, so in this case there are four possible situations:
    - ***M* finishes, *C* finishes**: *M* will obtain an authorization to deposit without *b*. When *C* tries to finish, *T* sends him the key, *k*.

- ***M* cancels, *C* finishes**: *M* and *C* will obtain a cancellation proof.
- ***C* finishes, *M* cancels or *C* finishes, *M* finishes***:* *C* will obtain the key *k* and *M* will obtain *b*.

In any case, the execution of the subprotocols leads to a fair situation.

5 Verifiability of the Trusted Third Party

During the execution of the cancellation or the finalization subprotocol, *T* decides the final state of the exchange checking the values of the boolean variables and the information received in the request. If *T* doesn't follow the subprotocol and sends inadequate elements to the parties, it is acting unfairly. The parties or an external verifier have to be able to detect and prove the fraudulent behavior of *T*. If the fraud can be detected and demonstrated the protocol is verifiable.

All possible fraudulent behaviors are listed and explained below.

- If *M* doesn't receive the message of step 2, the parties can act as follow:
  - ***M* cancels**, ***C* finishes**. In this case *T* must send a cancellation proof to *M* and then the same element to *C*. However, *T* can act incorrectly, giving a cancellation proof to *M*, and revealing the key *k* to *C*. This fraudulent behavior will be called **FB1**. Another incorrect action would be to provide *b* to *M*, but *T* cannot do it since *T* ignores its value.
  - ***C* finishes**, ***M* cancels**. In this case *T* has to send $P_{RT}(k, id)$ to *C* and P_{TM} to *M*. If *T* doesn't give the key *k* to *C* (instead *T* sends a cancellation proof to *C*), saying that the exchange has already been canceled, and gives *b* to *M,* is again acting fraudulently. This situation will be called **FB2**. In this same situation, *T* can give the key *k* to *C* even though he doesn't give *b* to *M*, saying that the exchange has not concluded. This is the behavior called **FB1**, as above.
- If ***C* doesn't receive the message of step 3 or *M* doesn't receive the message of step 4** the following situations are possible:
  - ***M* finishes, *C* finishes**. *T* must send a deposit authorization to *M* and the key to *C*. If *T* doesn't give *k* to *C*, and authorizes *M* to deposit the coin without the knowledge of the secret proof, second part of the coin, *b* is acting fraudulently. This behavior will be called **FB3**.
  - ***C* finishes, *M* finishes**. *T* must send the secret proof to *M* and the key to *C*. **FB3** is again possible. Moreover, *T* can give the key *k* to *C* and authorize *M* to deposit the coin without the knowledge of *b*. This behavior will be called **FB4**.

Four different fraudulent behaviors have been described. Now we will explain how they can be detected:

- **FB1**: *M* has a cancellation proof and *C* has the key, *k*. This situation can be the result of two kinds of execution. In the first one, *C* obtains the key from *M*. Later, *M* requests the cancellation of the exchange and obtains a cancellation proof. *T* has acted correctly; *M* is the one that has acted fraudulently, since he could have requested the finalization of the exchange. *C* can reveal the identity of the fraudulent party, showing the received signature on the pair *k, id,* that is, *C* can demonstrate *M*'s fraud showing *M*'s signature on *k*, id sent in step 3 of the exchange subprotocol, while *T* has h_{MT1}. The second kind of execution is the cancellation of the exchange at *M*'s request followed by the transfer of *k* from *T* to *C*. To demonstrate the fraudulent behavior of *T*, *M* can request *C* to show the signature on *k, id.* Now *C* can't provide the element K_M. If *C* cannot provide it, but instead *C* shows $PR_T(k)$ the fraudulent behavior of *T* can be demonstrated.

- **FB2**: Once *C* has obtained a cancellation proof, he can contact with the bank to deposit the coin or to request its change for a coin not used in any purchase attempt, without any risk. But now the bank detects that the coin has been deposited previously by *M*, and therefore, the bank suspects that when providing a cancellation proof to *C*, *T* acted incorrectly. However, *T* can defend itself providing H_{MT1} if *T* has it, since in this case *T* demonstrates that it was *M* who acted incorrectly when he requested the cancellation of the exchange.
- **FB3**: This case can be demonstrated as FB2.
- **FB4**: *M* can deposit the coin, but he can also prove that, although without damages, *T* has acted incorrectly giving *k* to *C* if *C* didn't have K_M and on the other hand *C* has $PR_T(k)$, since the TTP should have sent *b* to *M*, instead of giving him an authorization to deposit the coin.

Anyway, the incorrect behavior of *T* can be demonstrated, and therefore the TTP is verifiable.

6 Conclusions

The electronic purchase of a product requires a fair exchange. One of the values to exchange is an electronic coin and the other is the product or its receipt, depending on the kind of product (digital or physical). This paper presents a fair exchange protocol that can be used with existing payment systems. The buyer and the merchant can exchange their elements in only 4 steps without the intervention of the TTP. In this protocol, however, a TTP can be invoked for dispute resolution. For this reason, a fraudulent behavior of the third party would lead to an unfair exchange. We explain how the protocol allows the detection of fraud attempts. As a conclusion the TTP is verifiable and the exchange is always fair.

References

1. Asokan, N., Shoup, V., Waidner, M.: "Asynchronous protocols for optimistic fair exchange", IEEE Symposium on Research in Security and Privacy, pages 86-99, 1998.
2. Brands, S.: "Untraceable off-line cash in wallet with observers", Crypto'93, LNCS 773, pages 302-318, Springer Verlag, 1994.
3. Chaum, D., Fiat, A., Naor, M.: "Untraceable electronic cash", Crypto'88, LNCS 403, pages 319-327. Springer Verlag, 1988.
4. Camp, J., Harkavy, M., Tygar, J.D., Yee, B.: "Anonymous atomic transactions", 2^{nd} USENIX workshop on electronic commerce, pages 123-133, 1996.

5. Jakobsson, M.: “Ripping coins for a fair exchange”, Eurocrypt’95, LNCS 921, pages 220-230, Springer Verlag, 1995.
6. Pagnia, H., Jansen, R.: “Towards multiple payment schemes for digital money”, Financial Cryptography’ 97, LNCS 1318, pages 203-216, Springer Verlag, 1997.
7. Schuldt, H., Popovivi, A., Schek, H.: “Execution guarantees in electronic commerce payments”, 8th international workshop on foundations of models and languages for data and objects (TDD’99), LNCS 1773, Springer Verlag, 1999.
8. Su, J., Tygar, J.D.: “Building blocs for atomicity in electronic commerce”, 6th USENIX security symposium, 1996.
9. Tang, L.: “Verifiable transaction atomicity for electronic payment protocols”, IEEE ICDCS’96, pages 261-269, 1996.
10. Tygar, J.D.: “Atomicity in electronic commerce”, 15th ACM symposium on distributed computing”, pages 8-26, 1996.
11. Vogt, H., Pagnia, H., Gärtner, F.C.: “Modular fair exchange protocols for electronic commerce”, 15th Annual Computer Security Applications Conference’99, pages 3-11, 1999.
12. Xu, S., Yung, M., Zhang, G., Zhu, H. “Money conservation via atomicity in fair off-line e-cash”, International security workshop ISW’99, LNCS 1729, pages 14-31, Springer Verlag, 1999.