Protocols for Secure, Atomic Transaction Execution in Electronic Commerce

Srividhya Subramanian and Mukesh Singhal

Dept. of Computer and Information Science
The Ohio State University, Columbus, OH 43210-1277
E-mail: {ssubrama,singhal}@cis.ohio-state.edu

Abstract

Computer networks are an efficient, inexpensive, convenient and fast mode of communication and information transfer. There is widespread demand for the ability to buy and sell goods, especially electronic, using this medium. Sales over computer networks is also referred to as *electronic commerce*. A *protocol* for electronic commerce transactions is the series of steps that two parties must follow in order to transact business successfully. There are various issues that arise while designing electronic commerce protocols: how can one party be prevented from disappearing with the money/goods in the middle of a transaction? How can a party establish its identity without revealing personal information to unknown and untrusted parties? How can we prevent a third party from "stealing" money from the network? All these and other problems make designing an electronic commerce protocol both challenging and interesting.

In this paper, we first present a protocol for electronic commerce transactions between an anonymous customer and a merchant whose identity is public. Tampering with the protocol, either by the involved parties or by an intruder, does not cost the merchant or the customer more than some additional computation. Furthermore, under all circumstances, the transaction is either completed (the right amount of money is exchanged for the right goods) or aborted (there is no transfer of money or goods). In other words, the protocol ensures anonymity, security, and atomicity in business transactions. We also propose a protocol for electronic auctions. The auction protocol has the same properties of anonymity, security, and atomicity as the transaction protocol. We extend these two protocols to apply to real-time transactions where goods (such as stocks, bonds, options, etc.) must be transferred within a stipulated period of time.

Keywords: Electronic commerce, transaction, security, atomicity, privacy, anonymity, auction, real-time.

1 Introduction

There is a plethora of information and products already available to us through the World Wide Web or through email. Some of these are electronic commodities such as electronic books, papers, reports, information, typesetting services, ownership titles, share certificates, software, etc. Others are non-electronic goods that are advertised, ordered and paid for electronically. According to a projection by *Business Week*, trade worth 1,650 billion dollars will be conducted over electronic networks by the year 2000.

This growing shift towards electronic marketing can be attributed to certain factors. Electronic transfer of goods and information is easy, fast, efficient, and inexpensive. In addition, electronically advertised goods are more widely available, and easier to search for. The convenience of being able to look up various products, compare prices and features, place orders and pay, all from one's desktop computer, further favors this shift towards electronic marketing.

Buying and selling of goods over electronic networks is referred to as *electronic commerce*. An electronic commerce *transaction* is the exchange of goods for money over electronic networks. A *protocol* in electronic commerce is a series of steps required to successfully complete an electronic commerce transaction.

Several properties have been identified as either required or desirable in protocols for transactions in electronic commerce [NM95, And94]. These properties are listed below.

- *Privacy*: No third party should be able to know details about the transaction (payment amount, product details, identity of parties that wish to remain anonymous, etc.)
- *Anonymity*: Nobody (including the merchant involved in the transaction) should know the true identity of the customer involved. At the least, nobody should be able to trace all transactions made by a customer and create a profile of the customer.
- *Security*: Tampering or replay of messages by a third party, non-adherence to the protocol by involved parties, or corruption or loss of message in the network should not cost more than some additional computation to either party.
- *Atomicity*: Under all circumstances, the transaction must either go to completion (the right goods are exchanged for the right amount of money) or be aborted (there is no exchange of goods or money).
- *Low Overhead*: The cost for each transaction must be low enough to justify transactions involving a small amount of money.

We believe that, in general, security and atomicity are required for an electronic commerce protocol to be viable. Breach of security or lack of atomicity can lead to loss of money, duplication of money, or transfer of money to a third party. This may have disastrous financial consequences for an individual or an organization, especially if the amount being transferred is large. It is, therefore, very important to ensure security and atomicity of transactions. Privacy, anonymity, and low overhead cost, on the other hand, are desirable features.

While many protocols exist for electronic commerce, none has all the five features listed above. Some of them emphasize anonymity and privacy at the cost of atomicity [Dig], others emphasize atomicity, but do not provide anonymity [ST95]. Although, these protocols are suitable for specific situations in the commerce, there is a need for a general protocol that is suitable for most situations.

In addition, most existing electronic commerce protocols do not handle auctions. When they do, they allow only a very simplified version of auction with no more than one round of bidding [FR95]. Auctions are an important and common form of commerce today. A difficult aspect of auctions is that the bidder must be present at the site of the auction. This reduces the appeal of auction and restricts the number of people who would otherwise participate in it. An auction over an electronic network is, therefore, an attractive way of conducting business.

Sometimes, it is important that transactions complete within a fixed period of time (real-time transactions). This may be either because the price agreed upon changes rapidly with time (e.g., currency market, stock market, options, etc.), or because the product is useful to the customer only for a certain period of time. Since such transactions must complete almost immediately, electronic networks seem to be the ideal medium to conduct them over. However, it is not easy to detect if the good transacted was delivered on time or not. If the good was not delivered within the time specified, the customer's payment must be refunded. Existing protocols do not handle real-time transactions.

In this paper, we first propose a protocol for a general electronic commerce transaction between an anonymous customer and a merchant whose identity is public. The protocol is atomic, secure, private, and incurs nominal overhead charges. Second, we propose a protocol for auction between anonymous customers and a known merchant. The auction protocol also is atomic, secure, private, and incurs nominal overhead charges. Third, we extend these protocols to handle real-time transactions.

The rest of the paper is organized as follows. Section 2 outlines the previous work in this area. Section 3 gives models for transaction execution and auction, and notations used in this paper. Section 4 describes a general transaction protocol involving exchange of goods for money. Section 5 describes a protocol for auction over electronic networks. Section 6 extends these protocols to handle real-time transactions. We conclude in section 7 with a brief comparison of our protocols to other existing protocols.

2 Previous Work

A number of systems [Cyb, Dig, SSBR94, Man95, BGH$^+$95, Hic95, B$^+$94] have been developed that implement some of the important features listed in Section 1. A representative set of such systems is described below.

Ecash: Ecash, a system designed by DigiCash [Dig], emphasizes on privacy and anonymity [Cha92] of the customer. The protocol works as follows: the customer withdraws an ecash coin from the bank. The customer pays a merchant by sending a function of the coin and an arbitrary number provided by the merchant. This function acts as a "proof" that the customer has the coin without revealing the actual coin to the merchant. The merchant sends the customer the goods. The merchant collects such "proofs" from various customers and deposits them in the bank at the end of the day. The bank checks to see that the coins deposited haven't been deposited previously (double spent). If a coin has been double spent, the identity of the customer is revealed to the bank (it is impossible to find the customer's identity otherwise). The network police is notified and the customer is charged with double spending. The only time anyone (including the bank) knows the identity of a customer is when the customer spends the same coin more than once.

This protocol ensures anonymity, privacy, and security. It, however, fails to be atomic in the event of a message loss. The overhead cost involved in withdrawing ecash from the bank is fairly high. There is also a small probability that a customer charged with double spending is not guilty of the crime.

NetBill: NetBill, a protocol designed at the Carnegie Melon University [ST95], mainly emphasizes on atomicity [Tyg96]. The protocol works as follows: the customer asks the merchant for the price of a product. The merchant responds with a price and the customer accepts it. Now, the merchant sends the customer goods encrypted by some key K. The customer sends the merchant an electronic payment order (EPO) with a digitally signed value for the 3-tuple $<$ *price*, *cryptographic checksum of encrypted goods*, *timeout* $>$. The merchant counter signs the EPO and sends it to the NetBill server along with a signed copy of the key K. The NetBill server checks both signatures on the EPO, checks if the customer has enough funds, stores a copy of the cryptographic checksum of the encrypted goods and the key K, and sends a signed receipt with the key K to the merchant. The merchant forwards this key to the customer.

This protocol ensures atomicity, privacy, and security and incurs a nominal overhead cost. It, however, performs a lot of centralized computation at the NetBill server. It also requires the customer to have an account with some funds at the NetBill server. In addition, it does not ensure anonymity, since at least the NetBill server will be aware of all transactions made by the customer.

DEC's Millicent: Millicent, a protocol designed at DEC [Man95], emphasizes on low overhead cost. The protocol works as follows: a piece of "scrip" represents an account the customer has established with a vendor. At any given time, a vendor has outstanding scrip (open accounts) with the recently active customers. The balance of the account is kept as the value of the scrip. When a customer makes a purchase with a scrip, the cost of the purchase is deducted from the scrip's value and a new scrip (with the new value/account balance) is returned to the customer as change. After completing a series of transactions, the customer can "cash in" the remaining value of the scrip (close the account). Brokers serve as accounting intermediaries between customers and vendors. Customers enter into long-term relationships with brokers, in much the same way as they would enter into an agreement with a bank, credit card company, or Internet service provider. Brokers buy and sell vendor scrip as a service to customers and vendors. Broker scrip serves as a common currency for customers to use when buying vendor scrip, and for vendors to give as a refund for unspent scrip.

The protocol has very low overhead, since complex payment messages (such as establishing an account with the vendor, withdrawing money from a bank, etc.) are amortized over many transactions. Security is limited and atomicity is not ensured. Instead, it is believed that the need to maintain a good clientele will be more profitable to a vendor/broker than the gain accrued by not delivering goods, creating scrips with lesser value, or other fraudulent schemes. Privacy is enforced by use of a customer "secret" (like a PIN). If this secret is compromised, so is the privacy. There is a limited form of anonymity.

iKP: This family of protocols, designed by IBM [BGH$^+$95], emphasizes on the security of transactions. The protocol works as follows: the merchant and customer agree upon the price and the product to be exchanged. The customer digitally signs a purchase request with the price, secures it with the bank's public key, and sends it to the bank. The merchant digitally signs a sales request with the price, secures it with the bank's public key, and sends it to the bank. The bank compares the prices. If they match, it charges the customer's account and instructs the merchant to complete the transaction.

This protocol is secure, and it is atomic to the extent that credit card transactions are. It, however, does not assure anonymity from the payment system or any form of privacy. The overhead cost is low.

3 System Models

A general transaction protocol involves two parties: the customer and the merchant. The customer must have a certain amount of money, say M, that the customer is ready to pay in return for a product, P. The merchant must have the product P that the merchant is willing to sell for money M. The sale is completed when the customer receives the product P from the merchant and the merchant receives money M from the customer.

In electronic commerce transactions, these parties may be physically located anywhere. Each party has a computer connected to an electronic network. The parties communicate by sending messages to each other over the network. We assume the existence of some acceptable electronic form of payment such as a credit card number, an electronic coin withdrawn from a bank, or any other electronic entity that does not compromise the customer's anonymity. In electronic commerce, we also assume the existence of trusted third parties [KGM96, Ket95]. The third parties facilitate anonymity by issuing pseudo-identities to customers. They may also issue credit cards that can be used by the customer for payment. The customers must trust these third parties not to use a customer's pseudo-identity.

An auction involves a set of customers and a single merchant merchant. The merchant must have a product that he is willing to sell for the highest price offered. A round of bidding consists of some customers sending the price they are willing to offer the merchant. The round of bidding is bounded by a cut-off time. At this time, the merchant collects all the offers and computes the highest offer. If no higher offer has been made since the previous round of bidding, the product is sold to the highest bidder and all other customers are notified about the sale. If a higher bid has been made, all customers are notified about this price and another round of bidding starts.

Here again, the involved parties may be physically located anywhere. Each of the parties has access to a networked computer, and can communicate with other parties by sending messages over this network. Payments are electronic. We also assume that there are third parties that issue pseudo-identities to the customers. Apart from facilitating anonymity, these third parties are expected to act as collection agencies if their customer does not pay for goods bought at an auction. Third parties have this extra role because, unlike in normal transactions, the customer cannot refuse to buy a product after making the highest bid in an auction.

In both the above systems, messages may be lost or corrupted on the network, customers may try to run away with goods without paying for them, merchants may try to abscond with payments without delivering any product, and intruders may try to steal the product or the money from the network or otherwise disrupt the transaction.

Notations

We will use the following notations throughout the remaining parts of the paper.

$\boldsymbol{A}$ stands for the customer, Alice, with public key $\boldsymbol{a}$ and private key $\frac{\mathbf{1}}{\boldsymbol{a}}$.

$\boldsymbol{B}$ stands for the merchant, Bob, with public key $\boldsymbol{b}$ and private key $\frac{\mathbf{1}}{\boldsymbol{b}}$.

$\boldsymbol{T}$ stands for the third party, Tom, with public key $\boldsymbol{t}$ and private key $\frac{\mathbf{1}}{\boldsymbol{t}}$.

$\boldsymbol{e}$ and $\boldsymbol{E}$ are encryption keys with corresponding decryption keys $\frac{\mathbf{1}}{\boldsymbol{e}}$ and $\frac{\mathbf{1}}{\boldsymbol{E}}$

$\boldsymbol{X} \Longrightarrow \boldsymbol{Y}\ [\,\boldsymbol{message\ content}\,]^{\frac{1}{x}}$ stands for
$\boldsymbol{X}$ sends "message content" to $\boldsymbol{Y}$ signed with $\boldsymbol{X}$'s private key.

$\boldsymbol{X} \Longrightarrow \boldsymbol{Y}\ [\,\boldsymbol{message\ content}\,]^{\frac{y}{x}}$ stands for
$\boldsymbol{X}$ sends "message content" to $\boldsymbol{Y}$ signed with $\boldsymbol{X}$'s private key and secured with $\boldsymbol{Y}$'s public key.

4 An Atomic, Secure Protocol for Transaction Execution

A transaction in electronic commerce involves a merchant, say Bob, and a customer, say Alice. The transaction can be broken into three stages. In the first stage, Bob advertises that he has a product he wishes to sell. In the second stage, the interested buyer (Alice) and Bob agree on a price. In the third stage, the actual exchange of goods and payment occurs. All this must occur electronically.

Electronic fraud is both easier and more likely. Therefore, it is important to assure the strictest security and complete atomicity to attract organizations and individuals to use this form of commerce. The aggregation of profiles of customers in recent years, based on electronic billing information, has led to a desire for privacy and anonymity.

4.1 The Protocol

Our protocol is based on public key cryptography [RSA78, Sch94]. Every person is issued a unique and universal identity (like the Social Security Number in the US). This universal identity constitutes a public and a private key. The public key is revealed to others and is used to trace a person, if necessary. It is also used to secure confidential messages sent to the person. The private key is used as a signature, for authentication and to read secure messages.

A protocol for atomic, secure, private electronic transactions between an anonymous customer, Alice, and a merchant, Bob, is illustrated in Figure 1.

$$\textbf{(1)} \quad \boldsymbol{B \Longrightarrow everybody\ [\,product\ description,\ price\,]^{\frac{1}{b}}}$$

$$\textbf{(2)} \quad \boldsymbol{A \Longrightarrow B\ [\,[\,product\ description,\ price\,]^{\frac{1}{b}},\ a,\ [\,price\,]^{\frac{1}{a}},\ payment^{e}\,]^{b}}$$

$$\textbf{(3)} \quad \boldsymbol{B \Longrightarrow A\ [\,[\,payment^{e}\,]^{\frac{1}{a}},\ b,\ [\,product^{E}\,]^{\frac{1}{b}}\,]^{a}}$$

$$\textbf{(4)} \quad \boldsymbol{A \Longrightarrow B\ [\,[\,product^{E}\,]^{\frac{1}{b}},\ a,\ [\,\tfrac{1}{e}\,]^{\frac{1}{a}}\,]^{b}}$$

$$\textbf{(5)} \quad \boldsymbol{B \Longrightarrow A\ [\,[\,\tfrac{1}{e}\,]^{\frac{1}{a}},\ b,\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a}}$$

Figure 1: Transaction Protocol between the merchant, Bob, and the customer, Alice.

In Step 1 of our protocol, Bob advertises his product by broadcasting both the product description and the expected price. He signs the advertisement with his private key, both for authentication by a prospective customer and for non-repudiation.

In Step 2, Alice responds to the advertisement by sending a copy of Bob's advertisement (for reference), her public key (for further communication), a signed value of price (for authentication) and also an encrypted payment. The payment is encrypted with encryption key $\boldsymbol{e}$ to prevent a situation where Bob malignantly claims her payment was lost or stolen in the network, but cashes it anyway. In this situation, he can demand another payment, thus getting paid twice for the same good. With encrypted payments, he can only cash the single payment that Alice sends him the key for. She also secures the message with Bob's public key. This ensures that only Bob can read the message.

In Step 3, Bob acknowledges Alice's payment and also sends her the encrypted product. Bob encrypts the product to prevent Alice from disappearing without paying for the product, or malignantly claiming that the product was lost or stolen in the network, and receiving two products for a single payment. He signs the encrypted product with his private key and secures the message with Alice's public key. This ensures that only Alice can get the encrypted product and she can also verify that the message is indeed from Bob.

In Step 4, Alice acknowledges receiving the encrypted product and sends a signed copy of the payment decryption key. Since only Bob has the encrypted payment, only he can use this key to get the payment. Further, since the message is secure, only he can read this message. The encryption key is signed by Alice so that Bob can verify the source of the message.

In Step 5 of the protocol, Bob acknowledges receipt of the payment decryption key from Alice and also sends her a signed copy of the product decryption key. He further secures this message with Alice's public key so that only she can read the message.

4.2 Properties

We will show below that the transaction protocol is secure, atomic, private, anonymous, and incurs nominal overhead charges.

Security

On electronic networks, an intruder may prevent a message delivery, forge a message, or replay old messages as if they are new. We consider all these three situations below, and show how the protocol is secure from security attacks.

Since both the product and the payment are encrypted, if a message with either the encrypted product or the encrypted payment is not delivered, the sender can resend the message using a different encryption key. This will only add some additional computation cost for the sender. All other messages, if lost, can just be resent without any additional cost to either party. Thus, by blocking a few messages, the intruder cannot cause the transaction to abort, reach an inconsistent state, or result in a financial loss to the involved parties. Messages genuinely lost or corrupted in the network are dealt with in the same fashion.

Since all messages are signed with the sender's private key, an intruder cannot forge a message. Hence, forgery does not pose a security threat to our protocol.

An intruder can try replaying old messages as if they are new. Let us first assume that the intruder replays the message in Step 1 from Bob. Alice responds with an encrypted payment. Bob, even if he receives this payment will respond with a note saying he is not selling the product anymore. The intruder cannot acknowledge the payment, since he cannot read and parse Alice's message. The transaction will simply abort. Let us now assume that the intruder responds to an advertisement by Bob, with an old response from Alice. Bob will now send Alice an encrypted product which Alice must acknowledge. Since Alice has not initiated a new transaction with Bob, she will ignore Bob's message even if she receives it. The intruder cannot acknowledge with an old acknowledgement either. (We assume that Bob uses different encryption keys at least when selling the same product again to the same person.) The transaction will not continue, and no product or payment will have exchanged hands. Let us next assume that the intruder tries to replay a message during a valid transaction between Bob and Alice. Step 3 and Step 4 include a copy of the encrypted product. As shown earlier, old messages will not have products encrypted with the same key and hence cannot be used. In Step 5, Bob includes a copy of the payment encryption key that the intruder cannot reproduce from old messages. Thus, the intruder cannot cause a transaction to occur, or tamper an ongoing transaction between Bob and Alice, by just replaying old messages.

We conclude that our protocol is secure from all three security threats - message blocking, forgery and replay attacks.

Atomicity

In the above protocol, Alice sends the payment decryption key, $\boldsymbol{e}$, before receiving the product decryption key, $\boldsymbol{E}$. Therefore, she cannot disappear with the product without paying for it. Bob, on the other hand, can cheat by not sending the product key in Step 5 or by sending the wrong product in Step 3. However, Bob's true identity is known. Alice also has the product advertisement, acknowledgement of her payment, and the encrypted product. Therefore, she can prove Bob's wrongdoing in front of an arbitrator [CS96] and demand that the right product be delivered. This ensures that goods are delivered if and only if the payment is made. In other words, the protocol is atomic.

Privacy

It is commonly believed that even with the knowledge of the pubic key, a malignant intruder cannot compute the corresponding private key. Thus, the intruder cannot read messages secured with the recipient's public key or forge messages signed with the sender's private key. The intruder's inability to read secured messages ensures privacy in our protocols.

Anonymity

In the above protocol, the merchant's true identity is revealed to the customer, Alice. Alice, however, can hide her identity from the merchant, Bob. To do so, Alice registers with a third party (whose true identity is known) and gets a pseudonym (a new set of public and private keys) that the customer can operate with. If Alice registers with different third parties, the transactions Alice conducts under these different pseudonyms cannot be traced back to the her. This guarantees anonymity.

Overhead Cost

The above protocol does not require any complex computation. Furthermore, there are only five messages exchanged between the customer and the merchant. Therefore, the cost incurred is nominal [GS96].

Non-Electronic Goods

With some minor modifications, the protocol can be extended to non-electronic goods. In Step 3, instead of sending an encrypted product, Bob only acknowledges the receipt of the encrypted payment. Then in the Step 5, instead of sending an encryption key for the product, Bob acknowledges receipt of the payment encryption key and sends the product via some other medium (e.g. certified post).

5 Auction Protocol

An auction involves a merchant, say Bob, and many customers. An auction can be broken into three stages similar to the transaction in the previous section. In the first stage, Bob advertises his product. In the second stage, the customers bid for the product until no customer wishes to make a higher bid. The highest bidder, say Alice, and the merchant, Bob, now exchange the product and the payment. The important difference between a normal transaction and an auction is that once Alice makes the highest bid for a product she must buy it. In the transaction protocol described in the previous section, the customer can bail out of a transaction at any time before paying for the product (sending the payment encryption key). In an auction, however, Alice must pay if she has placed the highest bid.

5.1 The Protocol

As in the transaction protocol described in Section 4, the auction protocol presented here is also based on public key cryptography.

$$(1a)\quad B \Longrightarrow \textit{everybody}\ [\textit{product description},\ \textit{list of recognized third parties}]^{\frac{1}{b}}$$

$$(1b)\quad B \Longrightarrow \textit{everybody}\ [\textit{product identifier},\ \textit{product}^{E}]^{\frac{1}{b}}$$

$$(2)\quad A \Longrightarrow B[\textit{message}\ (1a),\ a,\ \textit{third party}\ t,\ \textit{price},\ \textit{payment}^{e},\ [\textit{price}]^{\frac{1}{a}}]^{b}$$

$$(3)\quad B \Longrightarrow \textit{everybody}\ [\textit{product description},\ \textit{maximum price offered}]^{\frac{1}{b}}$$

$$(4)\quad B \Longrightarrow \textit{everybody}\ [\textit{product description},\ \textit{sold for price}]^{\frac{1}{b}}$$

$$(5)\quad A \Longrightarrow B\ [[\textit{product description},\ \textit{sold for price}]^{\frac{1}{b}},\ \tfrac{1}{e},\ [\tfrac{1}{e}]^{\frac{1}{a}}]^{b}$$

$$(6)\quad B \Longrightarrow A\ [[\tfrac{1}{e}]^{\frac{1}{a}},\ \tfrac{1}{E},\ [\tfrac{1}{E}]^{\frac{1}{b}}]^{a}$$

Figure 2: Auction Protocol between the merchant, Bob, and a representative customer, Alice.

A protocol for auction in electronic commerce is illustrated in Figure 2. Here, we only show the highest bidder, Alice. There may, however, be any number of customers involved until the product is sold to Alice.

In Step 1, Bob advertises for the product by broadcasting the product description and a list of third parties that he trusts. In an auction, once the highest bid is made, the transaction must occur. To ensure this, the seller, Bob, must be able to reach the buyer, Alice, if she absconds. A third party helps achieve this without revealing Alice's true identity to Bob. Bob includes in his advertisement the names of third parties he is willing to interact with. He also broadcasts the product encrypted with key $\boldsymbol{E}$. Both messages have the product identifier (may be a part of the product description) to prevent mix-ups with messages regarding sales of other products. Bob signs both messages so that customers can verify that they are from him.

In Step 2, Alice (and other interested customers) responds to the advertisement with a reference to the advertisement, her pseudo-identity (i.e., Alice's public key and name of the third party that issued the public key), price offered, payment encrypted with key $\boldsymbol{e}$ and the signed value of the price offered. Bob can look up the third party's directory to check that Alice's public key is indeed issued by the third party involved. Furthermore, he can verify the message is from Alice by decrypting the signed value of the price offered. Since the message is secured with Bob's public key, only Bob can read it.

Once the identity of the customer is established, Bob broadcasts the maximum price offered. This constitutes Step 3. The message is signed by Bob so that the customers can verify that it is from him. Bob can hike the maximum price announced in Step 3. This is an accepted practice in auctioneering. Most auctions have their agents among the bidders. These agents artificially hike the maximum bid, goading other bidders to offer more. This method may also be used to resolve ties between bidders. Step 2 and 3 are repeated till no higher price is offered.

When the no higher bid is made, the product is advertised to be sold for the highest price quoted (Step 4). This announcement is again signed by Bob so that it can be verified that he sent it.

In Step 5, the highest bidder, Alice, responds with the decryption key for her payment. (If she does not respond, Bob approaches the third party with her bid and collects the payment.) Alice encloses the old notice for reference and a signed copy of the decryption key so that Bob can verify that the message is from her.

In Step 6, Bob acknowledges the payment and sends the signed product encryption key.

5.2 Properties

We will show below how the five properties of security, atomicity, privacy, anonymity, and low overhead cost, are ensured.

Security

Security of the protocol can be compromised by preventing message delivery, forging messages, or by replaying old messages. We will examine each of these cases separately.

As in the previous protocol, loss of a message can be easily handled by resending the lost message. If the message includes the encrypted payment or the encrypted product, the encryption key used is changed. An intruder may, however, prevent a higher bid from reaching the merchant. To do this, the intruder must either know beforehand the identity of the customer planning to make a higher bid, or must block all messages reaching the merchant. The former is impossible to know, and the later can be easily detected by having the merchant, Bob, send a message to himself during every round of bidding. Hence, preventing delivery is not a significant threat to the security of our protocol.

Since we use private key signatures, an intruder cannot forge any messages. Forgery, therefore, does not pose a security threat to our protocol.

Consider a situation where an intruder tries to replay old messages. The intruder can replay an old advertisement by Bob as if it were new, but cannot respond to the bids by the customers as the response must include the customer's encrypted payment from the secure message they send to Bob. If the intruder, instead tries to respond to an advertisement as if he is a customer, he will have to find an old response by the customer to the same product. This has to be from a previous round of bidding, as the same product cannot be auctioned off twice. Customers make higher offers in each round of bidding. The old replayed message will be for a price equal to one of previous rounds of bidding and will hence be ignored by Bob. Same holds for Step 3. In Steps 4 and 5, the intruder cannot find a corresponding old message for the product being sold now, and hence cannot replay the massage. In Step

6, the intruder cannot acknowledge the payment encryption key sent by the customer. The intruder cannot, therefore, replay messages to either conduct an auction in someone else's name or bid in an auction with another person's identity.We conclude that our protocol is secure from all external attacks.

Atomiciy

As shown for the electronic commerce transactions in Section 4, the auction protocol is atomic. The analysis in the previous section holds because, in the auction protocol, the encryption keys are exchanged at the end of the protocol and in the same fashion as in the transaction protocol.

Privacy

As discussed for the electronic commerce protocol in Section 4, the auction protocol described here is also based on public key cryptography and an intruder cannot read secured messages. This ensures privacy.

Anonymity

In the above protocol, the merchant's true identity is revealed to the customers. The customers, however, can hide their identity from the merchant. To do so, a customer registers with a third party (whose true identity is known) and gets a pseudonym (a new set of public and private keys) that the customer can operate with. If the customer registers with different third parties, the transactions the customer conducts under these different pseudonyms cannot be traced back to the same person. Even when the merchant wishes to contact a cutomer (for not paying), the merchant has to do so via the third party. Thus, the customer's true identity is never revealed to the merchant and the customer's transactions are not revealed to any one third party. This guarantees anonymity.

Overhead Cost

The above protocol does not require any complex computation. Furthermore, goods being auctioned are likely to be of sufficiently high value to justify the cost incurred by multiple bidding rounds. If the bidding continues through many rounds, the merchant may also decide to sell the goods to the currently highest bidder. Hence, the cost incurred remains low.

Non-Electronic Goods

The protocol can be modified to handle non-electronic goods as follows: the merchant, Bob, does not send the message (1b) with the encrypted product. In Step 6, Bob acknowledges the receipt of the encryption key from Alice and sends the product to her through another medium (e.g., certified post).

6 Transactions in Real-time

In a variety of situations, it might be essential to complete a transaction within a certain period of time. We refer to such transactions as real-time transactions. As noted in Section 1, real-time transactions are important when the price of the product fluctuates rapidly with the time at which the transaction occurs or the product is useful to the customer only for a limited period of time.

For real-time transactions, it must be established if the product was indeed delivered within the stipulated period of time. If the product was not delivered within this period of time, the customer must be refunded his payment. We verify the time of delivery by timestamping [ST94] messages between the merchant and the customer.

The parties involved, the resources at their disposal, their location are all exactly the same as in the situation with no time restrictions. We further assume that every customer is associated with a local trusted processor [YT95]. All messages are sent and received by the customer via its trusted processor. The message transmission between a customer and its trusted processor is FIFO and the transmission time is negligible. To every incoming and outgoing message, the trusted processor stamps the time (as per its local clock) when the message was received by it. It also signs the timestamped message to prevent timestamp forgery. Now, at the trusted processor, all messages may be totally ordered based on the time at which they arrive at that processor. The trusted processor also holds copies of all timestamped messages for a certain period of time specified by the customer.

6.1 The Protocols

For an electronic commerce transaction with real-time restrictions, if Alice, the customer, does not receive the decryption key for the product before the stipulated period of time after she has sent her payment encryption key, she sends a demand for refund immediately. If the product decryption key reaches the trusted local processor before Alice's demand for refund, the timestamp on the demand will indicate that and the demand will be ignored. If, on the other hand, the product hasn't reached the local trusted processor when the demand is timestamped, Bob needs to verify that Alice hasn't sent the demand too early, before he refunds Alice's money. This can be done by checking the difference in clock time between the payment message and the demand for refund message. If this difference is greater than or equal to the time stipulated at the time of the advertisement of the product, Bob refunds the payment.

An electronic commerce transaction protocol with real-time restrictions is shown in Figure 3. The message contents are similar to the protocol without time restrictions. The

$$(1)\quad B \Longrightarrow everybody\,[\,product\ description,\ stipulated\ time,\ price\,]^{\frac{1}{b}}$$

$$(2)\quad A \Longrightarrow B\,[\,[\,product\ description,\ stipulated\ time,\ price\,]^{\frac{1}{b}},\ a,\ [\,price\,]^{\frac{1}{a}},\ payment^{e}\,]^{b}$$

$$(3)\quad B \Longrightarrow A\,[\,[\,payment^{e}\,]^{\frac{1}{a}},\ b,\ [\,product^{E}\,]^{\frac{1}{b}}\,]^{a}$$

$$(4)\quad A \Longrightarrow B\,[\,[\,product^{E}\,]^{\frac{1}{b}},\ a,\ [\,\tfrac{1}{e}\,]^{\frac{1}{a}}\,]^{b}$$

$$(5)\quad B \Longrightarrow A\,[\,[\,\tfrac{1}{e}\,]^{\frac{1}{a}},\ b,\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a}$$

Figure 3: Transaction protocol with real-time restrictions.

only difference is that the time difference between payment and delivery is agreed upon along with the price.

An auction protocol with real-time restrictions is a similar extension of the auction protocol without any time restrictions. If Alice does not receive either the maximum price notice or the sold notice within the time specified at during the advertisement of the product, she sends a void notice to Bob. This means that her bid is now voided. Now, if the product was sold to Alice, Bob cannot hold her to the contract. Also, if Alice does not receive the product decryption key within a stipulated period of time after her payment (following the sold notice), she can demand a refund.

$$(1a)\quad B \Longrightarrow everybody\,[\,product\ description,\ stipulated\ time,\ list\ of\ recognized\ third\ parties\,]^{\frac{1}{b}}$$

$$(1b)\quad B \Longrightarrow everybody\,[\,product\ identifier,\ product^{E}]^{\frac{1}{b}}$$

$$(2)\quad A \Longrightarrow B[\,message\ (1a)\ a,\ third\ party\ t,\ price,\ payment^{e},\ price^{\frac{1}{a}}\,]^{b}$$

$$(3)\quad B \Longrightarrow everybody\,[\,product\ description,\ maximum\ price\ offered\,]^{\frac{1}{b}}$$

$$(4)\quad B \Longrightarrow everybody\,[\,product\ description,\ sold\ for\ price\,]^{\frac{1}{b}}$$

$$(5)\quad A \Longrightarrow B\,[\,[\,product\ description,\ sold\ for\ price\,]^{\frac{1}{b}},\ [\,\tfrac{1}{e}\,]^{\frac{1}{a}}\,]^{b}$$

$$(6)\quad B \Longrightarrow A\,[\,[\,\tfrac{1}{e}\,]^{\frac{1}{a}},\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a}$$

Figure 4: Auction protocol with real-time restrictions

The auction protocol with real time restrictions is shown in Figure 4. The message contents are similar to those in the auction protocol without time restrictions. The difference is that the stipulated time for response after the bid is sent or for delivery after the payment is made is agreed upon along with the price.

6.2 Properties

The arguements for privacy, security, anonymity and low overhead cost remain the same as in the previous protocols. The only change in the protocols is that the transaction may be voided after payment (or for auctions, after the sold notice is sent). This raises issues in atomicity. What if the payment is refunded even though the product reached on time? What if the payment is collected even though the product did not reach on time? We discuss these issues here.

In both the above protocols, the customer, Alice, cannot change the timestamp on the messages since Alice cannot tamper with the trusted processor's clock, nor can she forge its signature. She can, however, have the product key rerouted to her through a secret channel and prevent it from reaching her trusted processor. That way, she gets the product and can also demand the refund. To prevent this, the merchants will secure all outgoing messages with the public key of the receiving trusted processor. Now, Alice cannot get the product key without first having it deciphered by her trusted processor.

The merchant, Bob, can deny that he received a demand for refund or this demand may genuinely be lost in the network. Such a denial will be futile since the timestamp records on Alice's trusted processor will show that either the product key has still not been received or that it was received after a demand for refund was sent.

7 Conclusion

In this paper, we developed two protocols: one for atomic, secure electronic transactions and the other for electronic auctions. We extended both these protocols to real-time transactions where goods such as stocks, options, etc. must be transferred within a stipulated period of time. Our protocols are atomic, secure, anonymous, private and inexpensive.

Previously proposed protocols for electronic commerce transactions were designed for specific applications. When extended to general transactions applications they have one or more of the following drawbacks: the customer is required to keep some money in a special account, the protocol is not atomic, the protocol is not sufficiently secure, the protocol has complex computations (and hence, high overhead costs), the customer's identity and transaction patterns are revealed to some central authority (e.g., the bank), or the third party's role is extensive and expensive. We expect that there will be many customers who will not buy goods over electronic networks frequently enough to warrant establishing a special account, or wish to reveal their transactions to a central authority, or transact amounts large enough to justify the high overhead cost involved. These customers will, however, constitute a large part of the trade conducted electronically. Our general transaction protocol, caters to customers with a variety of interests and requirements and hence is more general in its application.

There is very little work previously done on auction protocols. The protocol in [FR95] is the only extensive work towards implementing electronic auctions. [FR95] ensures security and anonymity but not atomicity. Furthermore, customers may only bid once. Our protocol, on the other hand, meets all the five requirements described in Section 1 and allows multiple rounds of bidding.

There is no previous work on real-time transaction protocols. Transactions requiring real-time completion (e.g. trading in stocks, currencies, options, etc.) constitute a large portion of the potential electronic commerce. Hence, real-time protocols are important. The difficulty in designing these lies in correctly detecting whether the transaction completed on time or not. In our protocols, we detect the time of completion using some simple scalar timestamps by local trusted processors. We believe that our work in this area will help bring these important commerce transactions to the electronic front.

References

[And94] R. Anderson. Why Cryptosystems Fail. In *Communications of ACM*, volume 37, 1994.

[B+94] J. P. Boly et al. The ESPRIT Project CAFE - High Security Digital Payment Systems. In *ESORICS '94*, 1994.

[BGH+95] M. Bellare, J. A. Garay, R. Hauser, et al. iKP- A Family of Secure Electronic Payment Protocols. In *Proceedings of the First USENIX Workshop on Electronic Commerce*, 1995.

[Cha92] D. Chaum. Achieving Electronic Privacy. In *Scientific American*, 1992.

[CS96] T. Coffey and P. Saidha. Non-Repudiation with Mandatory Proof of Receipt. In *Sigcomm*, volume 26, 1996.

[Cyb] CyberCash. The CyberCash System - How it works. Available at http://www.cybercash.com/cybercash/cyber2.html.

[Dig] DigiCash. About Ecash. Available at http://www.digicash.com/ecash/ecash-home.html.

[FR95] M. Franklin and M. Reiter. The Design and Implementation of a Secure Auction Service. In *Proceedings of IEEE Symposium on Security and Privacy*, 1995.

[GS96] E. Gabber and A. Silberschatz. Agora: A Minimal Distributed Protocol for Electronic Commerce. In *2nd USENIX Workshop on Electronic Commerce*, 1996.

[Hic95] K. E. B. Hickman. Secure Socket Library. 1995. Available at http://www.mcom.com/info/SSL.html.

[Ket95] S. Ketchpel. Transaction Protection for Information Buyers and Sellers. In *DAGS95: Electronic Publishing and the Information Superhighway*, 1995. Available at http://robotics.stanford.edu/users/ketchpel/dags4.html.

[KGM96] S. P. Ketchpel and H. Garcia-Molina. Making Trust Explicit in Distributed Commerce Transactions. In *Proceedings of the 16th ICDCS*, 1996.

[Man95] M. Manasse. The Millicent Protocols for Electronic Commerce. In *Proceedings of the First USENIX Workshop on Electronic Commerce*, 1995.

[NM95] B. C. Neuman and G. Medvinsky. Requirements for Network Payment: The NetCheque Perspective. In *Proceedings of IEEE COMPCON'95*, 1995.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In *Communications of ACM*, 1978.

[Sch94] B. Schneider. . In *Applied Cryptography*, 1994.

[SSBR94] L. H. Stein, E. A. Stefferud, N. S. Borenstein, and M. T. Rose. The Green Commerce Model. 1994. Available at http://www.fv.com/tech/green-model.html.

[ST94] S. Smith and J. D. Tygar. Security and Privacy for Partial Order Time. In *Proceedings of the ISCA International Conference on Parallel and Distributed Computing Systems*, 1994.

[ST95] M. Sirbu and J. D. Tygar. NetBill: An Internet Commerce System. In *IEEE COMPCON*, 1995. Available at http://www.ini.cmu.edu/netbill/CompCon.html.

[Tyg96] J. D. Tygar. Atomicity in Electronic Commerce. In *PODC*, 1996.

[YT95] B. Yee and J. D. Tygar. Secure Coprocessors in Electronic Commerce Applications. In *Proceedings of the first USENIX Workshop on Electronic Commerce*, 1995.