Design and Verification of a Secure, Atomic Transaction Execution Protocol for 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. Various issues 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. Protocol interrruption by involved parties, passive and active attacks by an intruder, and corruption or loss of messages in the network do 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. Using a BAN-style [5] logic, we prove these properties in the presence of an intruder. We also present a protocol for electronic auctions. The auction protocol has the same properties of anonymity, security, and atomicity as the transaction protocol.

Keywords: Electronic commerce, transaction, security, atomicity, privacy, anonymity, auction, verification, validation.

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 [1, 18]. These properties are : (i)*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.) unless it has been part of the product advertisement available to all. (ii)*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. (iii)*Security*: Passive or active attacks by an intruder, abnormal termination of 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. (iv)*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). (v)*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 [8], others emphasize atomicity, but do not provide anonymity [21]. 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, many proposed protocols have not been formally analyzed for protocol correctness. Protocol flaws depend on the logic of message exchange and content, and should be distinguished from weaknesses in cryptosystems and similar mechanisms whose strength is judged in the context of computational complexity.

Furthermore, 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 [9]. 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.

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. We then extend BAN logic to model an intruder who can eavesdrop, intercept, decrypt and store parts of messages for which the intruder has a key. Next, we formally analyze the protocol in the presence of such an intruder to show that atomicity, security, privacy, and anonimity are ensured. We also argue that the protocol is secure in the presence of a more powerful intruder, incurs nominal overhead charges, and can easily be extended to transactions involving non-electronic good. 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.

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 outlines the logic we use for formal analysis of the protocol. Section 5 describes a general transaction protocol involving exchange of goods for money, formally analyzes the protocol for protocol correctness, and proves that the protocol ensures the properties enumerated above. We conclude in section 6 with a brief comparison of our protocols to other existing protocols. Appendix A details proofs for some of our theorems. Appendix B describes a protocol for auction over electronic networks.

2 Previous Work

In this section we briefly discuss some electronic commerce protocols and methods of formal verification proposed in recent literature.

2.1 Protocols for Electronic Commerce

A number of systems [2, 4, 7, 8, 13, 16, 23] 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 [8], emphasizes on privacy and anonymity [6] 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 [21]. The overhead cost involved in withdrawing ecash from the bank is fairly high [21]. 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 [21], mainly emphasizes on atomicity [26]. 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 [16], 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 [2], 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.

2.2 Methods for Formal Verification

Cryptographic protocols, if designed improperly, may have flaws that make them vulnerable to various security attacks. The use of formal methods to analyze the correctness of cryptographic protocols became prevalent with the development of the BAN logic in 1989 [5]. Since then several papers have been published reporting problems with the BAN logic [17, 22, 24] and several others proposing BAN-style logics that overcome many of these limitations [12, 25].

In analysis using BAN logic, a set of participant's final beliefs is generated from a set of initial assumptions and the protocol steps. If these beliefs satisfy the goal of the protocol, then the protocol is validated. Descendents of the BAN logic have introduced useful concepts such as correctness of the beliefs of the participants, order of the receipt of messages etc. However, the concept of an intruder is rarely included in the logic. Some tools such as model checkers and theorem provers are used to automate the proof process. Sometimes, these tools model an intruder [11]. However, such implementations are not based on formally verified models and their assumptions need to be checked for their correctness. Also, sometimes the tools make assumptions that, though correct in a general

situation, may not be applicable to the specific domain of the problem being solved. In this paper, we extend the BAN logic to introduce an intruder who can eavesdrop, intercept messages, decrypt, and store the message components encrypted with keys available to the intruder. The extension of the logic to include attacks where the intruder can introduce fake messages, constructed from previously stored message components, and plain text components, is an interesting future exercise.

3 System Model

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 addition there may be an intruder. The intruder is allowed all reasonable real world attacks that are non-cryptographic. The intruder may eavesdrop on and/or intercept message transfer between two parties; may decrypt and store parts of a message that are encrypted with its own public key or with the private key of another party; and may introduce fake messages built from old messages, their components visible to the intruder, and plain text components. Corruption or loss of messages on the network are also modelled as message interruption by this intruder. We use formal analysis to show that our protocol is secure from the first two types of attacks. We use an operational arguement to show that the protocol is secure from the third type of attack.

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 privacy or anonymity.

An auction involves a set of customers and a single 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. Since third parties are expected to act as collection agencies, they must know the customer's true identity. The customers must trust these third parties not to use a customer's identity or other information the customer reveals to the third party [14, 15].

Notation for Protocol Description

We will use the following notations to describe the protocols throughout 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 $\frac{\mathbf{1}}{\boldsymbol{b}}$.

$\boldsymbol{I}$ stands for the intruder, with public key $\boldsymbol{i}$ and private key $\frac{\mathbf{1}}{\boldsymbol{i}}$.
$\boldsymbol{T}$ 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 $\frac{\mathbf{1}}{\boldsymbol{E}}$

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

$\boldsymbol{X} \rightarrow \boldsymbol{Y}$: [$\boldsymbol{message}$]$^{\boldsymbol{y}}$ stands for $\boldsymbol{X}$ sends "message" to $\boldsymbol{Y}$ secured with $\boldsymbol{Y}$'s public key.

4 The Logic for Formal Verification

In order to formally verify various properties of our protocol (e.g., security, atomicity, anonymity, privacy), we must develop a logic. The logic we use in this paper is based on a semantics of BAN logic [3, 5]. In this section, we will state the language (or syntax), the axioms, and the semantics of this logic. We will also develop some theorems that will be used later to analyze the protocol.

4.1 The Language

We distinguish between the following sorts: Principal, Key, Message, and Formula. Formula is subsort of Message. On Formula we have the traditional logical operators: $\wedge$, $\vee$, $\rightarrow$, and $\forall$. We also have the identity operator $=$. Furthermore, we have the following word operators:

- (*,*) : Message $\times$ Message $\longrightarrow$ Message
  (an associative, commutative, idempotent operator for message joining. Thus, if a message $\boldsymbol{Z}$ contains two message components $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$.)
- believes : Principal $\times$ Formula $\longrightarrow$ Formula
- once_said : Principal $\times$ Message $\longrightarrow$ Formula
- sees : Principal $\times$ Message $\longrightarrow$ Formula
- possesses : Principal $\times$ Key $\longrightarrow$ Formula
- public_key_of : Key $\times$ Principal $\longrightarrow$ Formula
- private_key_of : Key $\times$ Principal $\longrightarrow$ Formula
- $[*]^{*}$: Message $\times$ Key $\longrightarrow$ Formula
  (for encryption; intuitively, $[\boldsymbol{X}]^{\boldsymbol{k}}$ denotes $\boldsymbol{X}$ encrypted by $\boldsymbol{k}$)
- rightly_believes : Principal $\times$ Formula $\longrightarrow$ Formula
  Here, P rightly_believes φ := ($\boldsymbol{P}$ believes φ) $\wedge\varphi$

The word operators have higher precedence than the traditional logical operators, e.g., $\boldsymbol{P}$ believes $\varphi \wedge \psi$ must be interpreted as ($\boldsymbol{P}$ believes φ) $\wedge\psi$

4.2 Axiomatization

The axioms of our logic include the standard axioms of equational logic with modus ponens and the standard rule for universal quantification. Furthermore, we have:

1. **Rules of concatenation :** For message joining there are axioms of the forms:

   $\vdash (\boldsymbol{X}, (\boldsymbol{Y}, \boldsymbol{Z})) = ((\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{Z})$
   $\vdash (\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{Y}, \boldsymbol{X})$

2. **Rationality :** The rationality rule introduces a collection of axioms, one for each theorem of the logic:

   $\vdash \varphi \rightarrow \boldsymbol{P}$ believes φ

3. **Believing Modus Ponens :** Modus Ponens under the believes operator:

   $\vdash \boldsymbol{P}$ believes $(\varphi \rightarrow \psi) \rightarrow (\boldsymbol{P}$ believes $\varphi \rightarrow \boldsymbol{P}$ believes $\psi)$

4. **Saying parts of a joint message :** Uttering a joint message implies uttering each of its parts:

   $\vdash \boldsymbol{P}$ once_said$(\boldsymbol{X}, \boldsymbol{Y}) \rightarrow \boldsymbol{P}$ once_said $\boldsymbol{X}$

5. **Saying contents of an encrypted message :** Uttering an encrypted message, signed by yourself, while you believe that the key is good, implies uttering the encrypted message:

   $\vdash \boldsymbol{P}$ once_said $[\boldsymbol{X}]^{\boldsymbol{k}} \wedge \boldsymbol{k}$ private_key_of $\boldsymbol{P} \rightarrow \boldsymbol{P}$ once_said $\boldsymbol{X}$

6. **Seeing parts of a joint message :** Seeing parts of a joint message means seeing each part of the message separately as well:

   $\vdash \boldsymbol{P}$ sees $(\boldsymbol{X},\boldsymbol{Y}) \rightarrow \boldsymbol{P}$ sees $\boldsymbol{X}$

7. **Possessing keys of a seen key statement :** If one sees a key statement, one possesses the key that it mentioned:

   $\vdash \boldsymbol{P}$ sees $\boldsymbol{k} \rightarrow \boldsymbol{P}$ posseses $\boldsymbol{k}$

8. **Possessing believed keys :** One can only believe that a certain key is good if possessing the key:

   $\vdash \boldsymbol{P}$ believes $\boldsymbol{k}$ public_key_of $\boldsymbol{Q} \rightarrow \boldsymbol{P}$ possesses $\boldsymbol{k}$
   $\vdash \boldsymbol{P}$ believes $\boldsymbol{k}$ private_key_of $\boldsymbol{Q} \rightarrow \boldsymbol{P}$ possesses $\boldsymbol{k}$

9. **Decryption :** Seeing an encrypted message while having the key in possession, means seeing the message itself:

   $\vdash \boldsymbol{P}$ possesses $\boldsymbol{k} \wedge \boldsymbol{P}$ sees $[\boldsymbol{X}]^{\boldsymbol{k}} \rightarrow \boldsymbol{P}$ sees $\boldsymbol{X}$

10. **Seeing possessed keys:** If one possesses a key statement, one sees the key:

    $\vdash \boldsymbol{P}$ posseses $\boldsymbol{k} \rightarrow \boldsymbol{P}$ sees $\boldsymbol{k}$

11. **Good key ensures utterer :** A collection of axioms that state that if a key is good, the only ones that use it for encryption are the owners, so if somewhere, someone sees a message that contains a part encrypted with that key, that part must have been said by the key owner who encrypted it:

    $\vdash \boldsymbol{k}$ private_key_of $\boldsymbol{P} \wedge \boldsymbol{R}$ sees $[\boldsymbol{X}]^{\boldsymbol{k}} \rightarrow \boldsymbol{P}$ once_said $\boldsymbol{X}$

4.3 The model

The environment consists of a finite collection of *principals*. The local state of a principal P is defined as the tuple $(\mathcal{B}_P, \mathcal{O}_P, \mathcal{S}_P, \mathcal{K}_P)$, with the following intuitive interpretation:

- $\mathcal{B}_P$, the set of formulas that P currently believes;
- $\mathcal{O}_P$, the set of (sub-)messages P once said;
- $\mathcal{S}_P$, the set of messages that P has seen so far;
- $\mathcal{K}_P$, the set of keys P posssesses.

Definition 1 *A global state is a mapping from principals to local states for each principal in the environment. The unqualified term "state" will, from now on, mean a global state.*

Definition 2 *The symbol $\models$ is defined as a relation between a state and a formula where $s \models \phi$ means that in state s formula ϕ holds.*

Definition 3 *The function $cts[*]$ takes a message and delivers a set of messages:*

$cts[(X, Y)] := \{(X, Y) \cup cts[X] \cup cts[Y]\}$
$cts[X^k] := \{X^k \cup cts[X]\}$
$cts[X] := X$, *otherwise*

Intuitively $cts[*]$ gives all the components of a message that one could obtain if all decryption keys were available.

4.4 The Semantics

The semantics of the above logic is as follows:

$s \models True$;	
$s \models \phi \wedge \psi$	if $s \models \phi$ and $s \models \psi$;
$s \models (\phi, \psi)$	if $s \models \phi \wedge \psi$;
$s \models \phi \rightarrow \psi$	if $s \models \phi$ implies $s \models \psi$;
$s \models (\forall x : \phi)$	if $s \models \phi[x \leftarrow u]$ holds for all terms u of the appropriate kind not containing unbound variables;
$s \models P$ believes ψ	if $\phi \in \mathcal{B}_P$;
$s \models P$ believes X	if $X \in \mathcal{B}_P$;
$s \models P$ once_said X	if $X \in \mathcal{O}_P$;
$s \models P$ sees X	if $X \in \mathcal{S}_P$;
$s \models P$ Possesses k	if $k \in \mathcal{K}_P$;
$s \models k$ private_key_of P	if for all R, X, and Y such that $[X]^k \in cts[Y]$: $Y \in \mathcal{S}_R$ implies $X \in \mathcal{O}_P$;
$s \models k$ public_key_of P	if for all R : $[X]^k \in \mathcal{S}_R$ implies $X \in \mathcal{S}_P$;

It can be proved that the above logic is sound with respect to the given semantics [3].

4.5 Proving properties of the protocol

Definition 4 *The logic translation of an action a, $\mathcal{T}(a)$ is defined as:*

$\mathcal{T}(P \rightarrow Q : X) := (P$ *once_said* $X \wedge Q$ *sees* $X \wedge I$ *sees* $X)$
(here, I refers to the intruder)
$\mathcal{T}(\emptyset) := \emptyset$
$\mathcal{T}(P_1; P_2) := \mathcal{T}(P_1) \cup \mathcal{T}(P_2)$

Intuitively, $\mathcal{T}(P)$ is the minimal set of predicates required to compute the transition from a state to a new state on execution of a. The new state is given by the set of predicates:
$\{X : \{\mathcal{A} \cup \mathcal{T}(a) \vdash X\}$

Definition 5 *For a collection of predicates $\mathcal{A}$ and a protocol step $a = P \rightarrow Q : X$, the predicate "$\mathcal{A}$ allows a" is defined recursively with respect to the structure of message X:*

$\mathcal{A}$ *allows* $P \rightarrow Q : X^k$, $k \in \mathcal{K}_P$	:= *for some* S: $\mathcal{A}$ *allows* $P \rightarrow S : X$
$\mathcal{A}$ *allows* $P \rightarrow Q : X^k$, $k \notin \mathcal{K}_P$	:= $\mathcal{A} \vdash P$ *sees* X^k
$\mathcal{A}$ *allows* $P \rightarrow Q : (X, Y)$	:= $\mathcal{A}$ *allows* $P \rightarrow Q : X$ *and* $\mathcal{A}$ *allows* $P \rightarrow Q : Y$
$\mathcal{A}$ *allows* $P \rightarrow Q : \phi$	:= $\mathcal{A} \vdash P$ *believes* ϕ
$\mathcal{A}$ *allows* $P \rightarrow Q : X$	:= $True$ *(all other cases)*

Intuitively, $\mathcal{A}$ *allows* a means that action a is a valid action in any state where $\mathcal{A}$ holds. Any action where the message sent is built from plaintext, seen message components and known keys is an allowed action.

Definition 6 *Positive formulas are the least set such that:*

ϕ	*is positive if* $\vdash \phi$;
k *private_key_of* P	*is positive;*
k *public_key_of* P	*is positive;*
P *believes* ϕ	*is positive if* ϕ *is positive;*
P *sees* X	*is positive;*
P *once_said* X	*is positive;*
$\phi \wedge \psi$	*is positive if* ϕ *is positive and* ψ *is positive;*
$\phi \vee \psi$	*is positive if* ϕ *is positive or* ψ *is positive;*
$(\forall x : \phi)$	*is positive if* $\phi[x \leftarrow u]$ *is positive for all terms* u *of the appropriate kind not containing unbound variables.*
A finite set of formulas $\mathcal{F}$	*is positive if the logical AND of all the formulas in* $\mathcal{F}$ *is positive.*

Intuitively, a *positive formula* is a formula such that if it is $True$ in a state, it remains $True$ after execution of any *allowed* action in that state. Note that formulas where a principal sees or once_said are always positive.

Definition 7 *A specification of a protocol is given by two sets of formulas, $\mathcal{A}$ (the assumptions) and $\mathcal{C}$ (the conclusions). Protocol P satisfies such a specification (notation: $\{\mathcal{A}\}P\{\mathcal{C}\}$) when "$\mathcal{A}$ holds initially" guarantees that "$\mathcal{C}$ holds finally", i.e., after executing P.*

Definition 8 *The rectify operation $\mathcal{R}[*]$ maps formulas to formulas. In particular, it maps formulas of the form "P believes ϕ" to "P rightly_believes ϕ". It is defined as follows:*

$\mathcal{R}[P$ *believes* $Q]$	:= P *rightly_believes* ϕ
$\mathcal{R}[\phi \wedge \psi]$	:= $\mathcal{R}[\phi] \wedge \mathcal{R}[\psi]$
$\mathcal{R}[\phi \vee \psi]$	:= $\mathcal{R}[\phi] \vee \mathcal{R}[\psi]$
$\mathcal{R}[\phi \rightarrow \psi]$	:= $\mathcal{R}[\phi] \rightarrow \mathcal{R}[\psi]$
$\mathcal{R}[\forall x : \phi]$	:= $(\forall x : \mathcal{R}[\phi])$
$\mathcal{R}[\phi]$	:= ϕ, *other cases*

The *rectify* operation is important in being able to express statements such as Theorem 1 below.

Theorem 1 *If* $\mathcal{A} \cup \mathcal{T}(P)$ *is positive,* $\mathcal{A}$ *allows* P*, and* $\mathcal{A} \cup \mathcal{T}(P) \vdash C$*, then* $\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$*. For a proof, see [3].*

Corollary 1 *If* $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$ *is positive,* $\mathcal{A}$ *allows* $(P_1;P_2;\ldots;P_{n-1})$*,* $\mathcal{T}(P_n)$ *is positive,* $\mathcal{A} \cup \mathcal{T}(P_1) \cup \mathcal{T}(P_2) \cup \cdots \cup \mathcal{T}(P_{n-1})$ *allows* P_n*, and* $\mathcal{A} \cup \mathcal{T}(P_1) \cup \mathcal{T}(P_2) \cup \cdots \cup \mathcal{T}(P_{n-1}) \vdash C$*, then* $\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$*. For a proof, see Appendix A.1.*

Theorem 2 *If* $\mathcal{A} \vdash Y \notin \mathcal{S}_R$*,* $a = P \rightarrow Q : X$*,* $Y \notin cts[X]$*, and* $\{\mathcal{A}\}a\{\mathcal{C}\}$*, then* $\mathcal{C} \vdash Y \notin \mathcal{S}_R$*. For a proof, see Appendix A.2.*

Intuitively, R cannot see a previously unseen message if it is not part of the message(s) transmitted in the system.

Corollary 2 *If* $\mathcal{A} \vdash k \notin \mathcal{K}_R$*,* $a = P \rightarrow Q : X$*,* $k \notin cts[X]$*, and* $\{\mathcal{A}\}a\{\mathcal{C}\}$*, then* $\mathcal{C} \vdash k \notin \mathcal{K}_R$*. For a proof, see Appendix A.3.*

Intuitively, R cannot aquire a key not sent as a part of the message(s) transmitted in the system.

Theorem 3 *If* $\mathcal{A} \vdash Y \notin \mathcal{S}_R$*,* $a = P \rightarrow Q : X^k$*,* $\mathcal{A} \vdash \frac{1}{k} \notin \mathcal{K}_R$*,* $Y \neq X^k$*, and* $\{\mathcal{A}\}a\{\mathcal{C}\}$*, then* $\mathcal{C} \vdash Y \notin \mathcal{S}_R$*.*
For a proof, see Appendix A.4.

Intuitively, R cannot see a previously unseen message if the message(s) transmitted in the system cannot be decrypted by R.

5 An Atomic, Secure Protocol for Transaction Execution

In this section, we describe an atomic and secure protocol for electronic commerce. Using the logic developed in Section 4, we formally show that the protocol is atomic, private, anonymous, and secure in the presence of an intruder who can eavesdrop and intercept messages. We illustrate in Subsection 5.3 that the protocol is also secure in the presence of an intruder who can introduce fake messages (created from heard components, plain text, and known keys) into the system. We also show that the protocol incurs nominal overhead charges and can easily be adapted for electronic commerce transactions involving non-electronic goods.

5.1 The Protocol

Our protocol is based on public key cryptography [19, 20]. Every person can procure unique pairs of public and private keys. 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.

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 signed copy of Bob's advertisement (for authentication), her public key (for further communication), and also an encrypted payment. The payment is encrypted with encryption key e to prevent a situation where Bob malignantly claims her

(1) $B \rightarrow everybody\ :\ [product\ description,\ price]^{\frac{1}{b}}$

(2) $A \rightarrow B\ :\ [[[product\ description,\ price]^{\frac{1}{b}}]^{\frac{1}{a}},\ a,\ payment^{e}]^{b}$

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

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

(5) $B \rightarrow A\ :\ [[\frac{1}{e}]^{\frac{1}{b}},\ [\frac{1}{E}]^{\frac{1}{b}}]^{a}$

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

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 absconding 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, she can verify that the product is indeed from Bob, and she also has proof that Bob has received the encrypted payment.

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 key is indeed from Alice.

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.

5.2 Formal Analysis of the Protocol

We now formally analyze the protocol in the presence of an intruder using the logic presented in Section 4.

5.2.1 Initial State

The state of the system at the start of the protocol is given below:
$\mathcal{O}_A = \emptyset$; $\mathcal{O}_B = \emptyset$; $\mathcal{O}_I = \emptyset$;
$\mathcal{S}_A = \emptyset$; $\mathcal{S}_B = \emptyset$; $\mathcal{S}_I = \emptyset$;
$\mathcal{B}_A = \{\phi : \vdash \phi\} \cup \{\ b$ public_key_of B, i public_key_of I, a public_key_of A, $\frac{1}{a}$ private_key_of $a\}$;
$\mathcal{B}_B = \{\phi : \vdash \phi\} \cup \{\ b$ public_key_of B, i public_key_of I, $\frac{1}{b}$ private_key_of $B\}$;
$\mathcal{B}_I = \{\phi : \vdash \phi\} \cup \{\ b$ public_key_of B, i public_key_of I, $\frac{1}{i}$ private_key_of $I\}$;
$\mathcal{K}_A = \{\frac{1}{a}, a, b, i\}$;
$\mathcal{K}_B = \{\frac{1}{b}, b, i\}$;
$\mathcal{K}_I = \{\frac{1}{i}, b, i\}$;

5.2.2 Properties

The properties we want to verify are security, anonymity, privacy, and atomicity. These translate logically into:

Security:	$\frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge \mathit{payment} \notin \mathcal{S}_I \wedge \mathit{product} \notin \mathcal{S}_I$ (the intruder must not find out the private keys of the parties involved or get the payment or the product being transacted)
Anonymity, Privacy:	$a \notin \mathcal{S}_I$ (the intruder must also not find out the public key (i.e., psuedonym) that the customer is using for the transaction)
Atomicity:	$(\mathit{product} \in \mathcal{S}_A \wedge \mathit{payment} \in \mathcal{S}_B) \vee$ $(\mathit{product} \notin \mathcal{S}_A \wedge \mathit{payment} \notin \mathcal{S}_B) \vee$ $(\mathit{product}^E \in \mathcal{S}_A \wedge [\mathit{payment}^e]^{\frac{1}{b}} \in \mathcal{S}_A)$ (the product and payment should have been exchanged or neither the product nor the payment exchanged or the customer must have proof that the encrypted product and the encrypted payment have been exchanged)

5.2.3 Verification

Let $\mathcal{A}$ be the set of assumptions or the initial state we start with (where all the beliefs of principals are true), P the protocol, and $\mathcal{C}$ the properties we want to prove. Since the protocol may be interrupted by either the customer or the merchant failing to send the next message, we must show that $\mathcal{C}$ holds after *every step* of the protocol. That is, if P_n is step n of the protocol, then we must show $\{\mathcal{R}[\mathcal{A}]\}$ $P_1;P_2;\ldots;P_n$ $\{\mathcal{R}[\mathcal{C}]\}$ for every step in the protocol.

Step 1: $B \rightarrow \mathit{everybody}$: $[\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}}$

$$\begin{aligned}\mathcal{T}(P_1) = \quad & B \text{ once_said } [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \\ & A \text{ sees } [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \\ & B \text{ sees } [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \\ & I \text{ sees } [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}}\end{aligned}$$

(Since the only principals involved are A, B, and I, *everybody* is treated as A, B, and I.)

From Definition 6,
$\mathcal{A}$ is *positive* (1)
$\mathcal{T}(P_1)$ is *positive* (2)
From Definition 5,
$\mathcal{A}$ *allows* P_1 (3)
We know that,

$$\mathcal{A} \vdash \tfrac{1}{a} \notin \mathcal{S}_I \wedge \tfrac{1}{b} \notin \mathcal{S}_I \wedge \mathit{payment} \notin \mathcal{S}_I \wedge \mathit{product} \notin \mathcal{S}_I \wedge a \notin \mathcal{S}_I \wedge \mathit{product} \notin \mathcal{S}_A \wedge \mathit{payment} \notin \mathcal{S}_B \quad (4)$$

$$\tfrac{1}{a} \notin [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \tfrac{1}{b} \notin [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \mathit{payment} \notin [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge \mathit{product} \notin [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \wedge a \notin [\mathit{product\ description},\ \mathit{price}]^{\frac{1}{b}} \quad (5)$$

From Theorem 2, (4) and (5) above,

$$\mathcal{A} \cup \mathcal{T}(P_1) \vdash \tfrac{1}{a} \notin \mathcal{S}_I \wedge \tfrac{1}{b} \notin \mathcal{S}_I \wedge \mathit{payment} \notin \mathcal{S}_I \wedge \mathit{product} \notin \mathcal{S}_I \text{ (security)} \quad (6)$$

$$\mathcal{A} \cup \mathcal{T}(P_1) \vdash a \notin \mathcal{S}_I \text{ (anonymity, privacy)} \quad (7)$$

$$\mathcal{A} \cup \mathcal{T}(P_1) \vdash \mathit{product} \notin \mathcal{S}_A \wedge \mathit{payment} \notin \mathcal{S}_B \text{ (atomicity)} \quad (8)$$

From (6), (7), and (8),

$$\mathcal{A} \cup \mathcal{T}(P_1) \vdash \mathcal{C} \quad (9)$$

From Theorem 1, (1), (2), (3), (9), and the fact that all initial beliefs of all principals are valid,

$$\{\mathcal{R}[\mathcal{A}]\}\ P_1\ \{\mathcal{R}[\mathcal{C}]\} \quad (10)$$

Step 2: $\boldsymbol{A \rightarrow B}$ **:** $[\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b$

$$\begin{aligned}\mathcal{T}(\boldsymbol{P_2}) = \quad & \boldsymbol{B}\ \mathsf{once_said}\ [\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b \wedge \\ & \boldsymbol{A}\ \mathsf{sees}\ [\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b \wedge \\ & \boldsymbol{B}\ \mathsf{sees}\ [\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b \wedge \\ & \boldsymbol{I}\ \mathsf{sees}\ [\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b\end{aligned}$$

From (1) and (2) and Definition 6,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1})$ is *positive* (11)
(3) states,
$\mathcal{A}$ *allows* $(\boldsymbol{P_1})$ (12)
From Definition 6,
$\mathcal{T}(\boldsymbol{P_2})$ is *positive.* (13)
From $\mathcal{T}(\boldsymbol{P_2})$ above, $\boldsymbol{A}$ sees $[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}$.
Hence, from Definition 5,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1})$ *allows* $\boldsymbol{A \rightarrow B}$ **:** $[\,[\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]$. (14)
From (14) and Definition 5,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1})$ *allows* $\boldsymbol{A \rightarrow B}$ **:** $[\,[\,[\,\boldsymbol{product\ description},\ \boldsymbol{price}\,]^{\frac{1}{b}}\,]^{\frac{1}{a}},\ \boldsymbol{a},\ \boldsymbol{payment^e}\,]^b$ (15)
That is,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1})$ *allows* $\boldsymbol{P_2}$ (16)
We know that,
$\mathcal{A} \vdash \frac{1}{b} \notin \mathcal{K}_{\boldsymbol{I}}$ (17)
$\frac{1}{b} \notin \boldsymbol{cts}[\boldsymbol{P_1}]$ (18)
From Theorem 2, (17) and (18) above,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1}) \vdash \frac{1}{b} \notin \mathcal{K}_{\boldsymbol{I}}$ (19)
From Theorem 3, (6), (7), (8), and (19) we can conclude,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2}) \vdash \frac{1}{a} \notin \mathcal{S}_{\boldsymbol{I}} \wedge \frac{1}{b} \notin \mathcal{S}_{\boldsymbol{I}} \wedge \boldsymbol{payment} \notin \mathcal{S}_{\boldsymbol{I}} \wedge \boldsymbol{product} \notin \mathcal{S}_{\boldsymbol{I}}$ (security) (20)
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1}) \vdash \boldsymbol{a} \notin \mathcal{S}_{\boldsymbol{I}}$ (anonymity, privacy) (21)
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1}) \vdash \boldsymbol{product} \notin \mathcal{S}_{\boldsymbol{A}} \wedge \boldsymbol{payment} \notin \mathcal{S}_{\boldsymbol{B}}$ (atomicity) (22)
From (20), (21), and (22) we can conclude that,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1}) \vdash \mathcal{C}$ (23)
From Theorem 1, (11), (12), (13), (16), and (23) we can conclude that,
$\{\mathcal{R}[\mathcal{A}]\}\ \boldsymbol{P_1};\boldsymbol{P_2}\ \{\mathcal{R}[\mathcal{C}]\}$ (24)

Step 3: $\boldsymbol{B \rightarrow A}$ **:** $[\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a$

$$\begin{aligned}\mathcal{T}(\boldsymbol{P_3}) = \quad & \boldsymbol{B}\ \mathsf{once_said}\ [\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a \wedge \\ & \boldsymbol{A}\ \mathsf{sees}\ [\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a \wedge \\ & \boldsymbol{B}\ \mathsf{sees}\ [\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a \wedge \\ & \boldsymbol{I}\ \mathsf{sees}\ [\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a\end{aligned}$$

From (11), (13) and Definition 6,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2})$ *positive* (25)
From (12) and (16) and Definition 5,
$\mathcal{A}$ *allows* $(\boldsymbol{P_1};\boldsymbol{P_2})$ (26)
From Definition 6,
$\mathcal{T}(\boldsymbol{P_3})$ is *positive.* (27)
From $\mathcal{T}(\boldsymbol{P_2})$ above and Axiom 6 and Axiom 9, $\boldsymbol{A}$ sees $[\,\boldsymbol{payment^e}\,]$ (28)
From Definition 5 and (28),
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2})$ *allows* $\boldsymbol{B \rightarrow A}$ **:** $[\,[\boldsymbol{payment^e}\,]^{\frac{1}{b}},\ [\boldsymbol{product^E}\,]^{\frac{1}{b}}\,]^a$

That is,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2)$ *allows* P_3 (29)
We know that,
$\frac{1}{a} \notin cts[P_2; P_2]$ (30)
$\mathcal{A} \vdash \frac{1}{a} \notin \mathcal{S}_I$ (31)
From Theorem 2, Axiom 10, and (30),
$\mathcal{A} \cup \mathcal{T}(P_1;P_2) \vdash \frac{1}{a} \notin \mathcal{K}_I$ (32)
From Theorem 3, ((32), 20), (20), and (22) ,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3) \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I \wedge a \notin \mathcal{S}_I$ (33)
From $\mathcal{T}(P_3)$, Axiom 6, and Axiom 9 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3) \vdash product^E \in \mathcal{S}_A \wedge [payment^e]^{\frac{1}{b}} \in \mathcal{S}_A$ (34)
From (33) and (34), $\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3) \vdash \mathcal{C}$ (35)
From (25), (26), (27),(29), (35), and Theorem 1, we can conclude that,
$\{\mathcal{R}[\mathcal{A}]\}\ P_1;P_2;P_3\ \{\mathcal{R}[\mathcal{C}]\}$ (36)

Step 4: $A \rightarrow B\ :\ [\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b$

$\mathcal{T}(P_4) =\ B$ once_said $[\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b \wedge$
A sees $[\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b \wedge$
B sees $[\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b \wedge$
I sees $[\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b$

From Definition 6, (25) and (27),
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3)$ is *positive* (37)
From Definition 5, (26) and (29)
$\mathcal{A}$ *allows* $(P_1;P_2;P_3)$ (38)
From Definition 6,
$\mathcal{T}(P_4)$ is *positive.* (39)
From $\mathcal{T}(P_4)$ above and Axiom 6 and Axiom 9, A sees $[\,product^E\,]$ (40)
From Definition 5 and (40),
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3)$ *allows* $A \rightarrow B\ :\ [\,[\,product^E\,]^{\frac{1}{a}},\ [\,\frac{1}{e}\,]^{\frac{1}{a}}\,]^b$
That is,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3)$ *allows* P_4 (41)
We also know that,
$\frac{1}{b} \notin cts[P_2; P_3]$ (42)
From Theorem 2, (42) and (19),
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3) \vdash \frac{1}{b} \notin \mathcal{K}_I$ (43)
From Theorem 3, (43), (33), and (34), we can conclude,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3;P_4) \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I \wedge$
$a \notin \mathcal{S}_I \wedge product^E \in \mathcal{S}_A \wedge [payment^e]^{\frac{1}{b}} \in \mathcal{S}_A$ (44)
That is,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3;P_4)\ \mathcal{C}$ (45)
From Theorem 1, (37), (38), (39), (41), and (45), we can conclude that,
$\{\mathcal{R}[\mathcal{A}]\}\ P_1;P_2;P_3;P_4\ \{\mathcal{R}[\mathcal{C}]\}$ (46)

Step 5: $B \rightarrow A\ :\ [\,[\,\frac{1}{e}\,]^{\frac{1}{b}},\ [\,\frac{1}{E}\,]^{\frac{1}{b}}\,]^a$

$$\begin{aligned}\mathcal{T}(\boldsymbol{P_5}) = \quad & \boldsymbol{B} \text{ once_said } [\,[\,\tfrac{1}{e}\,]^{\frac{1}{b}},\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a} \wedge \\ & \boldsymbol{A} \text{ sees } [\,[\,\tfrac{1}{e}\,]^{\frac{1}{b}},\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a} \wedge \\ & \boldsymbol{B} \text{ sees } [\,[\,\tfrac{1}{e}\,]^{\frac{1}{b}},\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a} \wedge \\ & \boldsymbol{I} \text{ sees } [\,[\,\tfrac{1}{e}\,]^{\frac{1}{b}},\ [\,\tfrac{1}{E}\,]^{\frac{1}{b}}\,]^{a}\end{aligned}$$

From (37), (39), and Definition 6,
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4})$ is *positive* (47)
From $\mathcal{T}(\boldsymbol{P_5})$, Axiom 6 and Axiom 9, $\boldsymbol{A}$ sees $[\,\frac{1}{e}\,]$ (48)
From (48) and Definition 5,
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4})$ *allows* $\boldsymbol{B} \rightarrow \boldsymbol{A}\ :\ [\,[\,\frac{1}{e}\,]^{\frac{1}{b}},\ [\,\frac{1}{E}\,]^{\frac{1}{b}}\,]^{a}$ (49)
From Definition 6,
$\mathcal{T}(\boldsymbol{P_5})$ is *positive* (50)
From $\mathcal{T}(\boldsymbol{P_4})$, $\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4}) \vdash \boldsymbol{B}$ sees $[\,\frac{1}{e}\,]^{\frac{1}{b}}$ (51)
From (51) and Definition 5,
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4})$ *allows* $\boldsymbol{B} \rightarrow \boldsymbol{A}\ :\ [\,[\,\frac{1}{e}\,]^{\frac{1}{b}},\ [\,\frac{1}{E}\,]^{\frac{1}{b}}\,]^{a}$
That is,
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4})$ *allows* $\boldsymbol{P_5}$ (52)
We know that,
$\frac{1}{a} \notin \boldsymbol{cts}[\boldsymbol{P_3};\boldsymbol{P_4}]$ (53)
From Theorem 2, (53) and (31),
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4}) \vdash \frac{1}{a} \notin \boldsymbol{\mathcal{K}_I}$ (54)
From Theorem 3 and (44) we can conclude:
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4};\boldsymbol{P_5}) \vdash \frac{1}{a} \notin \boldsymbol{\mathcal{S}_I} \wedge \frac{1}{b} \notin \boldsymbol{\mathcal{S}_I} \wedge \boldsymbol{payment} \notin \boldsymbol{\mathcal{S}_I} \wedge \boldsymbol{product} \notin \boldsymbol{\mathcal{S}_I} \wedge \boldsymbol{a} \notin \boldsymbol{\mathcal{S}_I}$ (55)
Further, from Axiom 6, Axiom 9, $\mathcal{T}(\boldsymbol{P_2})$, $\mathcal{T}(\boldsymbol{P_3})$, $\mathcal{T}(\boldsymbol{P_4})$, and $\mathcal{T}(\boldsymbol{P_5})$,
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4};\boldsymbol{P_5}) \vdash \boldsymbol{product} \in \boldsymbol{\mathcal{S}_A} \wedge \boldsymbol{payment} \in \boldsymbol{\mathcal{S}_B}$ (56)
From (55) and (56),
$\boldsymbol{\mathcal{A}} \cup \mathcal{T}(\boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4};\boldsymbol{P_5}) \vdash \boldsymbol{\mathcal{C}}$ (57)
From Theorem 1, (47), (49), (50), (52), and (57), we can conclude that,
$\{\boldsymbol{\mathcal{R}}[\boldsymbol{\mathcal{A}}]\}\ \boldsymbol{P_1};\boldsymbol{P_2};\boldsymbol{P_3};\boldsymbol{P_4};\boldsymbol{P_5}\ \{\boldsymbol{\mathcal{R}}[\boldsymbol{\mathcal{C}}]\}$ (58)

5.3 Intelligent Replay Attacks, Cost, and Non-Electronic Goods

In the formal analysis, we did not prove security in the case of intelligent replay attacks. Intelligent replay attacks are attacks where the intruder may also introduce fake messages built from message components the intruder has seen and plain text components. We also did not discuss transactions involving non electronic goods or the cost of the protocol execution. Next, we discuss these issues briefly.

5.3.1 Intelligent Replay Attacks

Since atleast one component of all messages is signed with the sender's private key, an intruder cannot forge that component. An intruder can try constructing a message with that component replaced from an old message. Again, all messages except for the message in Step 1 are secured with the intended recepient's public key. Thus, from Theorem 3, we conclude that the intruder cannot see the signed components except for the product advertisement (in Step 1). The only attack possible is a plain replay attack where the intruder replays a whole message from an earlier transaction.

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.

5.3.2 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 [10].

5.3.3 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).

6 Conclusion

In this paper, we presented a secure, atomic protocol for electronic commerce transaction execution. The advantages of our protocol over other previously proposed protocols are its simplicity (and hence, ease in implementation), minimal third party role (third party is only required to arbitrate a conflict), privacy and anonymity from all (including any central server/authority), absence of any special account in which the customer must keep money, and nominal cost. 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.

We also developed a logic based on the BAN logic that models an intruder who can eavesdrop, intercept, decrypt, and store components of messages encrypted with keys available to the intruder. We formally proved atomicity, security, anonymity, and privacy in the presence of such an intruder. We also showed that our protocol is secure in the presence of intelligent replay attacks where intruders may introduce fake messages composed of previously stored message components, plain text components, and known keys. Furthermore, we showed that our protocol incurs nominal charges and can easily be extended to transactions involving non-electronic goods.

Very little work has been done towards the design of secure, atomic auction protocols [9]. To our best knowledge, there is no published protocol that supports more than one round of bidding. We presented (in Appendix B) a protocol for electronic auction with multiple rounds of bidding. This protocol is also secure, atomic, anonymous, private, and nominal in cost.

Future areas of interest include extending the logic developed to model a more powerful intruder and formal analysis of the auction protocol presented in Appendix B.

References

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

[2] 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.

[3] A. Bleeker and L. Meertens. A Semantics for BAN Logic. In *DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

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

[5] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. In *ACM Transactions on Computer Systems*, 1990.

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

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

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

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

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

[11] F. Germeau and G. Leduc. Model-based Design and Verification of Security Protocols using LOTOS. In *DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[12] L. Gong, R. Needham, and R. Yahalom. Reasoning about Beliefs in Cryptographic Protocols. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1990.

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

[14] 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.

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

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

[17] C. Meadows. Analyzing the Needham-Schroeder Public Key Protocol: A Comparison of Two Approaches. In *Proceedings of ESORICS*, 1996.

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

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

[20] B. Schneider. Applied Cryptography. 1994.

[21] 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.

[22] E. Snekkenes. Exploring the BAN Approach to Protocol Analysis. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1991.

[23] 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.

[24] P. Syverson. Adding Time to a Logic of Authentication. In *Proceedings of First ACM Conference on Computer and Communications Security*, 1993.

[25] P. Syverson and P. van Oorschot. On Unifying some Cryptographic Protocol Logics. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1994.

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

A Proofs

A.1 Proof of Corollary 1

Theorem 1 states,
If $\mathcal{A} \cup \mathcal{T}(P)$ is positive, $\mathcal{A}$ *allows* P, and $\mathcal{A} \cup \mathcal{T}(P) \vdash C$, then $\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$
Hence, for each step P_n :
If $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_n)$ is positive, $\mathcal{A}$ *allows* $(P_1;P_2;\ldots;P_n)$, and $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_n) \vdash C$, then
$\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$ (59)
From Definition 4 and (59) we get,
If $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1}) \cup \mathcal{T}(P_n)$ is positive, $\mathcal{A}$ *allows* $(P_1;P_2;\ldots;P_n)$, and $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$
$\cup \mathcal{T}(P_n) \vdash C$, then, $\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$ (60)
From Definition 5 and (60) we get,
If $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1}) \cup \mathcal{T}(P_n)$ is positive, $\mathcal{A}$ *allows* $(P_1;P_2;\ldots;P_{n-1})$, $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$
allows P_n, and $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1}) \cup \mathcal{T}(P_n) \vdash C$, then, $\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$ (61)
From Definition 6 and (61 we get,
If $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$ is positive,
$\mathcal{T}(P_n)$ is positive,
$\mathcal{A}$ *allows* $(P_1;P_2;\ldots;P_{n-1})$,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$ allows P_n, and
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1}) \cup \mathcal{T}(P_n) \vdash C$, then,
$\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$ (62)
From Definition 4 and (62) we get,
If $\mathcal{A} \cup \mathcal{T}(P_1;P_2;\ldots;P_{n-1})$ is positive,
$\mathcal{T}(P_n)$ is positive,
$\mathcal{A}$ *allows*$(P_1;P_2;\ldots;P_{n-1})$,
$\mathcal{A} \cup \mathcal{T}(P_1) \cup \mathcal{T}(P_2) \cup \cdots \cup \mathcal{T}(P_{n-1})$ allows P_n, and
$\mathcal{A} \cup \mathcal{T}(P_1) \cup \mathcal{T}(P_2) \cup \cdots \cup \mathcal{T}(P_{n-1}) \vdash C$, then
$\{\mathcal{R}[\mathcal{A}]\}\ P\ \{\mathcal{R}[\mathcal{C}]\}$ (63)
■

A.2 Proof of Theorem 2

$\mathcal{A} \cup \mathcal{T}(a) \vdash k \notin \mathcal{S}_P$
$\equiv \mathcal{A} \cup \mathcal{T}(a) \nvdash P$ sees $k \wedge$
Let, $\mathcal{A} \cup \mathcal{T}(a) \vdash P$ sees k
Since the only rule of inference is Modus Ponens,
$\exists z : (\mathcal{A} \cup \mathcal{T}(a) \vdash z \rightarrow P$ sees $Y) \wedge (\mathcal{A} \cup \mathcal{T}(a) \vdash z)$
We can extend this arguement to proof of $z \rightarrow P$ sees Y and further till we find
a suitable expression $z \ni$
$z \in \{\mathcal{A} \cup \mathcal{T}(a)\} \wedge z$ is of the form $(z_1 \rightarrow (z_2 \rightarrow (\ldots (z_n \rightarrow P$ sees $Y) \ldots)))$
The only candidates for z are Axiom 6 and Axiom 9. Hence,
$\exists X : ((P$ sees $X) \in \{\mathcal{A} \cup \mathcal{T}(a)\}) \wedge Y \in cts[X]$
We know that if,
$\exists X : ((P$ sees $X) \in \mathcal{A}) \wedge Y \in cts[X]) \Rightarrow P$ sees Y, then
$\mathcal{A} \vdash P$ sees Y
Since the implication is not true, $\exists X : ((P$ sees $X) \in \mathcal{T}(a)) \wedge Y \in cts[X]$
We know from Definition 4,
$(P$ sees $X) \in \mathcal{T}(a))$ only if X is the message transmitted in action a
But we also know that, $Y \notin cts[X]$ where X is the message transmitted in action a

Proof by contradiction. ■

A.3 Proof of Corollary 2

We know that, $\mathcal{A} \nvdash \boldsymbol{P}$ possesses $\boldsymbol{k}$
Using Axiom 7, $\mathcal{A} \nvdash \boldsymbol{P}$ sees $\boldsymbol{k}$
We know $\boldsymbol{k} \notin \boldsymbol{cts}[\boldsymbol{X}]$, where $\boldsymbol{X}$ is the message transmitted during action $\boldsymbol{a}$
From Theorem 2,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \nvdash \boldsymbol{P}$ sees $\boldsymbol{k}$
Using Axiom 10,
$\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \nvdash \boldsymbol{P}$ possesses $\boldsymbol{k}$

■

A.4 Proof of Theorem 3

$\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \vdash \boldsymbol{Y} \notin \mathcal{S}_{\boldsymbol{P}}$
$\equiv \mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \nvdash \boldsymbol{P}$ sees $\boldsymbol{Y} \wedge$
Let, $\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \vdash \boldsymbol{P}$ sees $\boldsymbol{Y}$
Since the only rule of inference is Modus Ponens,
$\exists \boldsymbol{z} : (\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \vdash \boldsymbol{z} \rightarrow \boldsymbol{P}$ sees $\boldsymbol{Y}) \wedge (\mathcal{A} \cup \mathcal{T}(\boldsymbol{a}) \vdash \boldsymbol{z})$
We can extend this arguement to proof of $\boldsymbol{z} \rightarrow \boldsymbol{P}$ sees $\boldsymbol{Y}$ and further till we find a suitable expression $\boldsymbol{y} \ni$
$\boldsymbol{y} \in \{\mathcal{A} \cup \mathcal{T}(\boldsymbol{a})\} \wedge \boldsymbol{y}$ is of the form $(\boldsymbol{z_1} \rightarrow (\boldsymbol{z_2} \rightarrow (\ldots (\boldsymbol{z_n} \rightarrow \boldsymbol{P}$ sees $\boldsymbol{Y}) \ldots)))$
The only candidates for $\boldsymbol{y}$ are Axiom 6 and Axiom 9. Hence,
$\exists \boldsymbol{X} : ((\boldsymbol{P}$ sees $\boldsymbol{X}) \in \{\mathcal{A} \cup \mathcal{T}(\boldsymbol{a})\}) \wedge \boldsymbol{Y} \in \boldsymbol{octs}[\boldsymbol{P}, \boldsymbol{X}]$
Here $\boldsymbol{octs}[\boldsymbol{X}]$ is defined as
$\boldsymbol{octs}[\boldsymbol{P}, (\boldsymbol{M}, \boldsymbol{N})] := \boldsymbol{cts}[[(\boldsymbol{M}, \boldsymbol{N})]$
$\boldsymbol{octs}[\boldsymbol{P}, \boldsymbol{M^k}] :=$ if $\boldsymbol{P}$ possesses $\boldsymbol{k}$ then $\boldsymbol{M^k} \cup \boldsymbol{cts}[\boldsymbol{M^k}]$ else $\boldsymbol{M^k}$
$\boldsymbol{octs}[\boldsymbol{P}, \boldsymbol{M}] := \boldsymbol{cts}[\boldsymbol{M}]$, otherwise
We know that if,
$(\exists \boldsymbol{X} : ((\boldsymbol{P}$ sees $\boldsymbol{X}) \in \mathcal{A}) \wedge \boldsymbol{Y} \in \boldsymbol{cts}[\boldsymbol{X}]) \Rightarrow \boldsymbol{P}$ sees $\boldsymbol{Y}$, then
$\mathcal{A} \vdash \boldsymbol{P}$ sees $\boldsymbol{Y}$
Since the implication is not true, $\exists \boldsymbol{X} : ((\boldsymbol{P}$ sees $\boldsymbol{X}) \in \mathcal{T}(\boldsymbol{a})) \wedge \boldsymbol{Y} \in \boldsymbol{cts}[\boldsymbol{X}]$
We know from Definition 4,
$(\boldsymbol{P}$ sees $\boldsymbol{X}) \in \mathcal{T}(\boldsymbol{a}))$ only if $\boldsymbol{X}$ is the message transmitted in action $\boldsymbol{a}$
We know that, $\boldsymbol{octs}[\boldsymbol{P}, \boldsymbol{X}] = \boldsymbol{X}$ where $\boldsymbol{X}$ is the message transmitted in action $\boldsymbol{a}$
Hence, $\boldsymbol{Y} \notin \boldsymbol{octs}[\boldsymbol{P}, \boldsymbol{X}]$
Proof by contradiction. ■

B 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

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

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

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

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

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

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

(6) $B \rightarrow A$: $[[\frac{1}{e}]^{\frac{1}{a}},\ \frac{1}{E},\ [\frac{1}{E}]^{\frac{1}{b}}]^{a}$

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

the payment encryption key). In an auction, however, Alice must pay if she has placed the highest bid.

B.1 The Protocol

As in the transaction protocol described in Section 5, the auction protocol presented here is also based on public key cryptography.

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.

B.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 5, 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 5, 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).