Design and Verification of Protocols for Secure Transaction Execution in Electronic Commerce (Dissertation Proposal)

Srividhya Subramanian
Dept. of Computer and Information Science
The Ohio State University, Columbus, OH 43210-1277
Tel: (614)-292-5813, Fax: (614)-292-2911
E-mail: ssubrama@cis.ohio-state.edu

Abstract: Computer networks present an efficient, inexpensive, convenient, and fast mode of communication and information transfer. There is widespread demand for the ability to buy and sell goods 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 execute in order to transact business successfully. There are various issues that arise while designing electronic commerce protocols such as preventing participating parties from absconding with the money/goods during the transaction, establishing a party's identity without compromising on anonymity or privacy, and preventing intruders from "stealing" money from the network. All these and other problems make designing secure electronic commerce protocol both challenging and interesting.

The focus of this dissertation is on the design, implementation, and validation of electronic commerce protocols that address the above issues. We identify three important applications of electronic commerce: (a) General transactions, (b) Auctions, (c) Real-time transactions. We further identify five useful properties - security, atomicity, anonymity, privacy, and low overhead cost - as either essential or desirable in electronic commerce transactions. For each of the above applications, we propose a protocol that ensures all these five properties. We further propose a methodology to convert protocols designed for specific applications into protocols for real-time transactions while preserving the useful properties of the previous protocol. We develop a logic based on the semantics of the well known BAN logic to formally verify that the protocols and the methodology ensure all the useful properties. We plan to experimentally analyze the performance of our protocols. We also address the problem of providing a viable and inexpensive payment scheme for mobile customers.

Keywords: Electronic commerce, transaction, security, atomicity, privacy, anonymity, auction, real-time, 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 such as electronic books, papers, reports, information, typesetting services, ownership titles, share certificates, software, etc., are in elecronic form. Others are non-electronically-stored 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 many 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 trend 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.

- *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).
- *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.
- *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.).
- *Low Overhead*: The cost for each transaction must be low enough to justify transactions involving small amounts 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.

We propose to develop electronic commerce protocols for general transactions, auctions, and real-time transactions, formally prove the above properties for these protocols, and analyze their

performance characteristics. We also plan to design a viable and inexpensive payment scheme for a mobile customers.

2 Existing Work

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

2.1 Protocols for General Electronic Commerce Transactions

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 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[ST95]. The overhead cost involved in withdrawing ecash from the bank is fairly high.

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 complete 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. Privacy is enforced by use of a customer "secret" (like a PIN). 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.

2.2 Protocols for Electronic Auctions

While there exist a number of auction sites on the internet such as Onsale (`http://www.onsale.com`), FirstAuction (`http://www.firstauction.com`), ZAuction (`http://www.zauction.com`), Dealdeal (`http://www.dealdeal.com`), and Ubid (`http://www.ubid.com`), that are fully/partially automated, few of them ensure some of the important properties such as security, privacy, anonymity, atomicity, etc. This is due to the inherent difficulties in automated price negotiation [BSS96]. While electronic auctions are complex, they are also equally popular and desirable. Consequently, a lot of research has is being done in the are of electronic auctions and particularly in electronic negotiations. Negotiation Support Systems (NSS) is a class of computer software systems geared specifically towards situations requiring negotiations. While these are powerful tools, they require near-constant human input and are far from being fully automated negotiation engines [BSS96]. A number of auction protocols [FR95, BSS96, YWL98, WW98, KHT98, PPLB98] have been proposed in the literature. These protocols are each directed at specific types of auction applications: single round auctions, real-time auctions, distributed auctions, etc. There still exists a need for a general purpose auction protocol that is efficient and satisfies all the five useful properties identified earlier.

2.3 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 [BAN90]. Since then several papers have been published reporting problems with the BAN logic [Mea96, Sne91, Syv93] and several others proposing BAN-style logics that overcome many of these limitations [GNY90, SvO94].

In analyses 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 enhanced the logic by introducing useful concepts such as correctness of the beliefs of the participants, order of the receipt of messages etc. Some tools such as model checkers and theorem provers are also used to automate the proof process.

Traditionally formal logic has been used to validate a protocol's correctness - but not to verify its security in the presence of intruders and other outside factors. Some theorem provers model an intruder [GL97]. However, these theorem provers are not based on formally verified models and their assumptions need to be checked for correctness.

3 Problem Statement and Motivation

The problem of providing secure electronic commerce transactions in untrusted domains such as the Internet may be divided into three layers as shown in Figure 1. The bottom layer involves designing new cryptographic algorithms, such as signature algorithms for authentication and non-repudiation, and encryption algorithms to prevent unauthorized parties from finding the contents of a ciphertext. The middle layer involves designing various payment schemes. Payment schemes can be offline or online, in the form of digital coins or accounts. The topmost layer involves designing protocols for electronic commerce transactions using these payment schemes. At each layer, the useful security properties of the layer(s) below it must be preserved and the other relevant security properties must be ensured. The bottom layer and some work in the middle layer largely constitute research in cryptography. The topmost protocol layer and a part of the payment scheme layer fall under the framework of research relating to electronic commerce applications. This thesis proposes to contribute to the top layer by designing protocols with some important properties that have not been provided by previous protocols. These additional properties promise to bring many important commerce applications to the electronic front. The thesis also proposes to contribute to the middle layer by designing new payment schemes that are not as expensive as the known offline schemes and are not as restrictive as the existing account based online schemes. The specific objectives of this dissertation are enumerated below.

- **General Transactions**: While many electronic commerce protocols already exist, none of these ensures all the five important properties listed in Section 1. Furthermore, these protocols may be used only in conjunction with specific payment schemes. Customers must sometimes set up a special account to use each of these payment schemes. One objective of this dissertation is to design a secure protocol for general electronic commerce transactions,

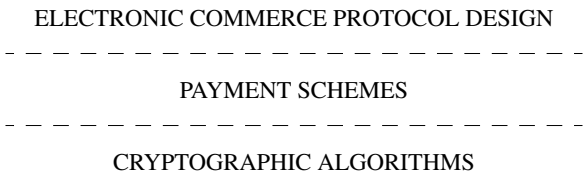

Figure 1: Layers involved in providing secure electronic commerce transactions.

which provides all the five useful properties and can be used with any payment scheme. The scheme must be simple (to enable easy implementation).

- **Auctions**:Auctions are an important and common form of commerce today. In traditional auctions 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, particularly attractive. However, electronic negotiation is very difficult to design [BSS96], and hence, 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]. We propose to design a secure protocol for auctions that allows multiple rounds of bidding and ensures the five important properties of electronic commerce mentioned before and that can used with any known payment scheme.

- **Real-Time Transactions**: 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 (e.g., news updates, stock prices, etc.). In these applications, the customer may be entitled to some refund or compensation when the transaction takes more than the stipulated time to complete. 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 product transacted was delivered on time or not. This is due to the non-existence of global clocks and the ability to tamper with the system clocks of unsecured computers. In this thesis, we propose to design a secure real-time aware protocol that can detect if a transaction took more than the stipulated time to deliver a product or complete a crucial service.

- **Real-Time Methodology**: Since many protocols already exist that are customized to certain applications, it is important to be able to add the real-time awareness property to these protocols. We propose to develop a methodology to convert an existing electronic commerce protocol to a real-time aware electronic commerce protocol while preserving its useful properties. The methodology can then be optimized for various special cases where only a part of the transaction must complete in real-time and for situations where the payment phase is aggregated over multiple transactions (in applications where low overhead is very important).

- **Formal Verification**: We propose to extend the BAN logic [BAN90] to model an intruder who can eavesdrop, intercept messages, decrypt and store the message components encrypted with keys available to the intruder, and introduce fake messages constructed from previously stored message components and plain text components. We will then use this extended logic

to prove that our protocols are not only correct but also secure, atomic, anonymous, and private in the presence of this powerful intruder.

- **Performance Analysis**: We propose to implement our protocols and experimentally analyze the performance of our protocols and compare them with other existing protocols. The cost will be estimated in terms of the time consumed and the resources used for an average run of the protocol.

- **Mobile Payment Scheme**: We will study the problem of designing a viable and inexpensive electronic payment scheme for a mobile customer. There are many account based payment schemes that would need to be verified/validated at the time of payment. This is an expensive option when the customer is a mobile person and the transaction involves only a moderate amount of money. The extra cost is either in the form of accessing the information from a distant location or in maintaining accurate and consistent information at replicated centers. There are a few coin based schemes that are offline, but they are extremely expensive and suitable only for transactions involving large amounts of money. We propose to build a payment scheme where the account information is carried by the customer in a mobile device such as a smartcard or a mobile phone and must be updated only infrequently at the central accounting agency. This eliminates the need to contact a central database or to replicate databases. However, other security issues such as tampering with the record, verifiability of the record, and billing mechanism in case of the loss of the device (and thus the record) by the customer need to be addressed.

4 Proposed Approach, Preliminary Results, and Future Plans

In Section 3, seven problems were identified. In this section, our approach to each of these problems is discussed separately. Preliminary results and future research plans are also described.

- **General Protocol**: We have designed a general protocol for electronic commerce transactions [SS98a] that satisfies all the important five properties (see Section 1). The product and the payment are first exchanged in an encrypted form. The payment key is then revealed before the product key is revealed. We ensure security and privacy by signing all messages with the sender's private key and encrypting it with the receiver's public key. Atomicity is ensured since the customer has the advertisement, the acknowledgement for an encrypted payment, and an encrypted product and can demand the product encryption key in exchange for the payment encryption key any time and any number of times. Since the product encryption key is not revealed until the payment is made, there is no need to know the identity of the customer. Hence, anonymity is ensured. We have also developed a logic based on the BAN logic [BAN90] to verify various properties in the presence of a passive intruder. Using this logic we have verified all the above mentioned five properties for our protocol. In future, we plan to formally verify the properties in the presence of active intruders and evaluate the performance of the algorithm.

- **Auction Protocol**: After studying various auction sites such as Onsale, FirstAuction, UBid, etc., and proposed auction protocols in literature [FR95, BSS96, YWL98, WW98, KHT98,

PPLB98], we have developed a more sophisticated auction protocol with multiple rounds of bidding [Sub98b]. The difficult part of designing auction protocols is ensuring anonymity from the merchant while ensuring that the bidders honor their bid. The payment part of our protocol is similar to the protocol described above. The difference lies in the way many parties are allowed multiple rounds of bidding without revealing each others identity. Each round of bidding has a time bound and at the end of a round, all participants are informed about the highest bid made. If no higher bid is made in a round, the participants are sent a sold notice and the price at which the product was sold. We have also formally verified all the above mentioned five properties using the logic we developed earlier. Future plans include performance evaluation of the protocol and formal verification of the properties in the presence of active intruders.

- **Real-time Aware Protocol and Methodology**: The focus of this work is two-fold : a) to design protocols to detect violation of real-time constraints b) to design a methodology to convert a given electronic commerce protocol to one capable of detecting real-time violations. We have developed real-time aware protocols for general transactions and for auctions [SS97] and have formally verified that these protocols ensure all the five useful properties mentioned earlier. We have also developed a general methodology [SS98b] to extend any protocol to a real-time aware protocol. The real-time detection is achieved using a secure, trusted clock in proximity to the customer. Furthermore, we have formally verified that this methodology preserves all the useful properties of the original protocol. In future, we plan to design a methodology to detect delays in specific parts of the transaction and to do a performance evaluation of the real-time aware protocols.

- **Formal Verification**: The use of formal logic for protocol verification became prevalent with the development of the BAN logic in 1989 [BAN90]. Since then several papers have been published reporting problems with the BAN logic [Mea96, Sne91, Syv93] and several others proposing BAN-style logics that overcome many of these limitations [GNY90, SvO94]. We have extended a semantics for BAN-style logic [BM97] to model a passive intruder capable of intercepting messages, decrypting and storing message components that can be obtained with keys known to the intruder. Using this logic we have proved the five above mentioned properties for our protocols. In future, we plan to extend this logic to model active intruders capable of introducing fake messages constructed from stored message components, known keys, and plain text.

- **Performance Analysis**: We have modeled the electronic commerce system as a queue with the merchant as the server and the jobs or requests coming from the customers. The jobs are processed in a FIFO manner. The server and the queue are both memoryless. We plan to use this model to analyze the system resources and the time consumed by an transaction in each protocol.

- **Billing for mobile customers**: The main problem here is to design a workable payment scheme that is suited to a highly mobile customer. The basic approach is to have a 'record' with verifiable account information that is not alterable by the customer. This record may be created or modified by participating merchants. The record specifies the spending limit and the amount already spent. After every transaction, the merchant changes the amount spent and returns the record to the customer. Some of the issues we plan to address are

impersonation, replay attacks, lack of privacy, and correct (or acceptable) billing when the record is stolen or lost.

5 Expected Significance and Contribution

With heightened interest in electronic commerce applications and concern for security and privacy, the last few years have seen intense research in security aspects of electronic commerce. Many research groups have worked in this area providing support for varied applications. However, there is still a critical need for general transaction, auction, and real-time protocols that are secure, atomic, anonymous, private, and incur nominal cost. The main objective of this thesis is to design and develop electronic commerce protocols that ensure all the five properties mentioned above. Several notable contributions have already been made:

We have proposed a general transaction protocol [SS98a] that is attractive to customers intending to make only a few transactions, reluctant to store money in a special account, or not willing to pay high overhead charges for an electronic transaction. Our protocol involves no third party except to arbitrate disputes. The protocol is also simple and easy to implement.

We have also proposed an auction protocol [Sub98b] that is fast, efficient, allows multiple rounds of bidding, can be used to implement English or Dutch auctions, is fully automated, and ensures all five properties mentioned above.

The real-time protocols and methodology proposed in this thesis have significant potential for bringing important real-time commerce applications to the electronic front and for improving the real-time characteristics of those applications that have already become electronic [SS98b]. The real-time aware general transaction protocol and real-time aware auction protocols have direct applications in the stock market which constitutes the largest and most receptive electronic commerce market. Furthermore, our real-time aware methodology extends real-time properties to existing protocols without stripping off any other useful property from the original protocol.

In conclusion, this thesis proposes secure protocols general enough to be directly applied to most electronic commerce situations. On completion, this thesis promises to bring more applications, customers, and merchants to the electronic front by satisfying their conflicting needs.

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.

[BAN90] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. In *ACM Transactions on Computer Systems*, 1990.

[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.

[BM97] A. Bleeker and L. Meertens. A Semantics for BAN Logic. In *DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[BSS96] C. Beam, A. Segev, and J. G. Shanthikumar. Electronic Negotiation through Internet-based Auctions. In *CITM working paper 96-WP-1016, Haas School, Berkeley*, 1996.

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

[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.

[GL97] 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.

[GNY90] 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.

[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.

[KHT98] H. Kikuchi, M. Harkavy, and J. D. Tygar. Multiround Anonymous Auction Protocols. In *IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, 1998.

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

[Mea96] C. Meadows. Analyzing the Needham-Schroeder Public Key Protocol: A Comparison of Two Approaches. In *Proceedings of ESORICS*, 1996.

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

[PPLB98] C. Peng, J. M. Pulidod, K. J. Lin, and D. Blough. The Design of an Internet-based Real Time Auction System. In *IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, 1998.

[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. Applied Cryptography. 1994.

[Sne91] E. Snekkenes. Exploring the BAN Approach to Protocol Analysis. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1991.

[SS97] S. Subramanian and M. Singhal. Protocols for Secure, Atomic Transaction Execution in Electronic Commerce. In *OSU-CISRC-10/97-TR49*, 1997.

[SS98a] S. Subramanian and M. Singhal. Design and Verification of a Secure, Atomic Transaction Execution Protocol for Electronic Commerce. In *OSU-CISRC-10/98-TR12*, 1998.

[SS98b] S. Subramanian and M. Singhal. Detecting Violation of Real-Time Constraints in Secure Electronic Commerce Transactions. In *14th International Information Security Conference IFIP SEC'98*, 1998. To appear.

[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.

[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.

[Sub98a] S. Subramanian. Candidacy Exam: Written Part. 1998.

[Sub98b] S. Subramanian. Design and Verification of a Secure Electronic Auction Protocol. In *17th IEEE Symposium on Reliable Distributed Systems*, 1998. To appear.

[SvO94] P. Syverson and P. van Oorschot. On Unifying some Cryptographic Protocol Logics. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1994.

[Syv93] P. Syverson. Adding Time to a Logic of Authentication. In *Proceedings of First ACM Conference on Computer and Communications Security*, 1993.

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

[WW98] M. P. Wellman and P. R. Wurman. Real Time Issues For Internet Auctions. In *IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, 1998.

[YWL98] X. Yi, X. Wang, and K. Lam. A Secure Auction-like Negotiation Protocol for Agent-based Internet Trading. In *17th IEEE Symposium on Reliable Distributed Systems*, 1998. To appear.

A Secure, Atomic, Electronic Commerce Protocol for General Transaction Execution

An electronic commerce transaction can be broken into three stages. In the first stage, the merchant, say Bob, advertises that he has a product he wishes to sell. In the second stage, the interested buyer, say 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.

A protocol for atomic, secure, private electronic transactions between an anonymous customer, Alice, and a merchant, Bob, is illustrated in Figure 2.

The 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 non-repudiation) and to decrypt secure messages. Furthermore, any person can generate a public and private key pair (that does not coincide with the set of universal identities) and use it as a pseudonym.

(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 2: 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 signed copy of Bob's advertisement (for authentication), her public key (for further communication), and 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 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. Furthermore, 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 sends her a signed copy of the product decryption key. He also secures this message with Alice's public key so that only she can read the message.

B Formal Proof of Security, Atomicity, Anonymity, and Privacy

We first develop the logic below and then use it to analyze the protocol and prove that all the above mentioned properties are ensured

B.1 The BAN-Style Logic 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 [BAN90] in 1989. Since then several papers have been published reporting problems with the BAN logic [Mea96, Sne91, Syv93] and several others proposing the BAN-style logics that overcome many of these limitations [GNY90, SvO94].

In analyses using the 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 enhanced the logic by introducing useful concepts such as correctness of the beliefs of the participants, order of the receipt of messages etc. Some tools such as model checkers and theorem provers are also used to automate the proof process.

Traditionally, formal logic has been used to validate a protocol's correctness - but not to verify its security in the presence of intruders and other outside factors. Some theorem provers model an intruder [GL97]. However, many theorem provers are not based on formally verified models and their assumptions need to be checked for correctness.

Since the properties (such as security, atomicity, anonymity, privacy) cannot be verified for an exhaustive set of existing protocols, we use formal verification methods to prove that no useful property is violated by the application of our methodology to any existing transaction protocol. The logic we use here is based on a semantics of the BAN logic [BM97, BAN90]. We extend the semantics of the BAN logic to model a powerful intruder. The intruder can:

- see all messages sent on the network,
- recursively decipher and see components of messages encrypted with known keys,
- construct messages using seen messages, message components, plain text components and known keys, and
- introduce these composed messages into the network.

In this section, we state the language (or syntax), the axioms, the model, and the semantics of this logic. We also develop some theorems that are later used to analyze the protocol.

B.1.1 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 Z contains two message components X and Y, $Z = (X, Y)$.)
- 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
- shared_key_of : Key $\times$ Principal $\times$ Principal $\longrightarrow$ 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, $[X]^k$ denotes X encrypted by k)
- rightly_believes : Principal $\times$ Formula $\longrightarrow$ Formula
  Here, P rightly_believes φ := (P believes φ) $\wedge\varphi$

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

B.1.2 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:

   $\vdash (X, (Y, Z)) = ((X, Y), Z)$
   $\vdash (X, Y) = (Y, X)$

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

   $\vdash \varphi \rightarrow P$ believes φ

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

   $\vdash P$ believes $(\varphi \rightarrow \psi) \rightarrow (P$ believes $\varphi \rightarrow P$ believes $\psi)$

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

   $\vdash P$ once_said$(X, Y) \rightarrow P$ once_said 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:

   $\vdash P$ once_said $[X]^k \wedge \exists R \ni k$ private_key_of $P \rightarrow P$ once_said X
   $\vdash P$ once_said $[X]^k \wedge k$ shared_key_of $P, R \rightarrow P$ once_said X

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

   $\vdash P$ sees $(X,Y) \rightarrow P$ sees X

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

   $\vdash P$ sees $k \rightarrow P$ possesses k

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

   $\vdash P$ believes k public_key_of $Q \rightarrow P$ possesses k
   $\vdash P$ believes k private_key_of $Q \rightarrow P$ possesses k
   $\vdash P$ believes $\exists R, Q \ni k$ shared_key_of $R, Q \rightarrow P$ possesses k

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

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

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

    $\vdash P$ possesses $k \rightarrow P$ sees 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:

    $\vdash k$ private_key_of $P \wedge R$ sees $[X]^k \rightarrow P$ once_said X
    $\vdash k$ shared_key_of $P, Q \wedge R$ sees $[X]^k \rightarrow P$ once_said $X \vee Q$ once_said X

B.1.3 The model

The environment consists of a finite collection of *principals*. The local state of a principal P is defined as the tuple (B_P, O_P, S_P, 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 possesses.

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.

B.1.4 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 $X \in \mathcal{S}_P$;
$s \models k$ shared_key_of P, Q	if for all $R : [X]^k \in \mathcal{S}_R$ implies $X \in \mathcal{S}_P \vee X \in \mathcal{S}_Q$;

It can be proved that the above logic is sound with respect to the given semantics [BM97].

In Section B.1.5 below, some theorems are developed that are later used to analyze the properties of protocols modified by our methodology.

B.1.5 Proving properties of a protocol

In this section, some terms are formally defined to express the properties of and the relationships between the system state and the protocol actions. These definitions are then used to develop and express theorems that are useful in analyzing protocols. k and $\frac{1}{k}$ are used to represent encryption and decryption keys. in the case of shared keys, $k = \frac{1}{k}$.

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 A to a new state B on execution of a. The new state is given by the set of predicates: $\{B : \{\mathcal{A} \cup \mathcal{T}(a) \vdash B\}$.

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* $P \to Q : X^k$, $k \in \mathcal{K}_P$	$:=$ *for some* S: $\mathcal{A}$ *allows* $P \to S : X$
$\mathcal{A}$ *allows* $P \to Q : X^k$, $k \notin \mathcal{K}_P$	$:= \mathcal{A} \vdash P$ *sees* X^k
$\mathcal{A}$ *allows* $P \to Q : (X, Y)$	$:=$ $\mathcal{A}$ *allows* $P \to Q : X$ *and* $\mathcal{A}$ *allows* $P \to Q : Y$
$\mathcal{A}$ *allows* $P \to Q : \phi$	$:= \mathcal{A} \vdash P$ *believes* $\phi \vee P$ *is an intruder*
$\mathcal{A}$ *allows* $P \to Q : X$	$:= True$ *(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
$\mathcal{R}[\phi \wedge \psi]$	$:= \mathcal{R}[\phi] \wedge \mathcal{R}[\psi]$
$\mathcal{R}[\phi \vee \psi]$	$:= \mathcal{R}[\phi] \vee \mathcal{R}[\psi]$
$\mathcal{R}[\phi \to \psi]$	$:= \mathcal{R}[\phi] \to \mathcal{R}[\psi]$
$\mathcal{R}[\forall x : \phi]$	$:= (\forall x : \mathcal{R}[\phi])$
$\mathcal{R}[\phi]$	$:= \phi$*, 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 [BM97].*

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 [Sub98a].*

Theorem 2 *If $\mathcal{A} \vdash Y \notin \mathcal{S}_R$, $a = P \rightarrow Q : X$, $Y \notin cts[X]$, $k \notin cts[X]$, where k is any key, and $\{\mathcal{A}\}a\{\mathcal{C}\}$, then $\mathcal{C} \vdash Y \notin \mathcal{S}_R$. For a proof, see [Sub98a].*

Intuitively, R cannot see a previously unseen message if it is not part of the key-less 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 [Sub98a].*

Intuitively, R cannot acquire 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 [Sub98a].

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

Theorem 4 *If $\{\mathcal{A}\}a\{\mathcal{C}_1\}$, $a = P \rightarrow Q : X_1$, $b = P \rightarrow Q : (X_1, X_2)^k$, $\{\mathcal{A}\}b\{\mathcal{C}_2\}$, and $(\mathcal{A} \vdash k \in S_R) \wedge (Y \neq (X_1, X_2)^k) \wedge (Y \notin cts[X_2]) \wedge (K \notin cts[X_2])$, where K is any key, then $\mathcal{C}_1 \vdash Y \in \mathcal{S}_R \Leftrightarrow \mathcal{C}_2 \vdash Y \in \mathcal{S}_R$.*
For a proof, see [Sub98a].

Intuitively, R only sees components of the key-less attachment and whatever R would have seen if the message was sent without the attachment.

Theorem 5 *If $a = P \rightarrow Q : (X_1, Z_1)^(k_1), (X_2, Z_2)^(k_2), \ldots, (X_n, Z_n)^(k_n)$, $\{\mathcal{A}\}a\{\mathcal{C}\}$, $\mathcal{A} \vdash X_1 \in S_R$, $\mathcal{A} \vdash X_2 \in S_R$, $\cdots$, $\mathcal{A} \vdash X_n \in S_R$, $K \notin cts[Z_1] \cup cts[Z_2] \cup \ldots \cup cts[Z_n]$, where K is any key, then $\forall Y \notin cts[Z_1] \cup cts[Z_2] \cup \ldots cts[Z_n]$, $\mathcal{A} \vdash Y \in S_R \Leftrightarrow \mathcal{C} \vdash Y \in S_R$*
For a proof, see [Sub98a].

Intuitively, R only sees components of the attachments and whatever R has seen before the message was sent, if the message sent is a concatenation of previously seen messages with key-less attachments.

In the next section, we show that the useful properties of a protocol are preserved after application of our methodology to any protocol.

B.2 Formal Analysis of the Proposed Protocol

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

B.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 $I\}$;
$\mathcal{K}_A = \{\frac{1}{a}, a, b, i\}$;
$\mathcal{K}_B = \{\frac{1}{b}, b, i\}$;
$\mathcal{K}_I = \{\frac{1}{i}, b, i\}$;

B.2.2 Properties

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

Security:	$\frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I$ (the intruder must not find out the private keys of the parties involved or get the payment or the product being transacted)
Anonymity:	$id \notin \mathcal{S}_I \wedge id \notin \mathcal{S}_B$ (the merchant or intruder must not find out the universal public key (id) assigned to the customer for identification)
Privacy:	$a \notin \mathcal{S}_I$ (the intruder must also not find out the public key (i.e., pseudonym) that the customer is using for the transaction)
Atomicity:	$(product \in \mathcal{S}_A \wedge payment \in \mathcal{S}_B) \vee$ $(product \notin \mathcal{S}_A \wedge payment \notin \mathcal{S}_B) \vee$ $(product^E \in \mathcal{S}_A \wedge [payment^e]^{\frac{1}{b}} \in \mathcal{S}_A)$ (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)

B.2.3 Verification

In this section, we first analyze the protocol in the presence of passive attacks to formally show that the above mentioned properties are ensured. Next, active attacks are included in the analysis. Protocol cost is discussed separately in Section B.2.4.

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. Assuming only passive attacks, the protocol may still be interrupted by either the customer or the merchant failing to execute the next step. The protocol steps that are executed, however, may not be tampered or forged. We must show that $\mathcal{C}$ holds after *every* properly executed step of the protocol. That is, if P_n is the nth of the protocol, then we must show:
$\forall_{i=1}^{5}\ \{\mathcal{R}[\mathcal{A}]\}\ P_1;P_2;\ldots;P_n\ \{\mathcal{R}[\mathcal{C}]\}$.
We formally prove this below.

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

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

(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,
$\mathcal{A} \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I \wedge a \notin \mathcal{S}_I \wedge$
$product \notin \mathcal{S}_A \wedge payment \notin \mathcal{S}_B$ (4)
$\frac{1}{a} \notin [product\ description,\ price]^{\frac{1}{b}} \wedge \frac{1}{b} \notin [product\ description,\ price]^{\frac{1}{b}} \wedge$
$payment \notin [product\ description,\ price]^{\frac{1}{b}} \wedge product \notin [product\ description,\ price]^{\frac{1}{b}} \wedge$
$a \notin [product\ description,\ price]^{\frac{1}{b}}$ (5)
From Theorem 2, (4) and (5) above,
$\mathcal{A} \cup \mathcal{T}(P_1) \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I$ (security) (6)
$\mathcal{A} \cup \mathcal{T}(P_1) \vdash a \notin \mathcal{S}_I \wedge id \notin S_I \wedge id \notin S_B$ (anonymity, privacy) (7)
$\mathcal{A} \cup \mathcal{T}(P_1) \vdash product \notin \mathcal{S}_A \wedge payment \notin \mathcal{S}_B$ (atomicity) (8)
From (6), (7), and (8),
$\mathcal{A} \cup \mathcal{T}(P_1) \vdash \mathcal{C}$ (9)
From Theorem 1, (1), (2), (3), (9), and the fact that all initial beliefs of all principals are valid,
$\{\mathcal{R}[\mathcal{A}]\}\ P_1\ \{\mathcal{R}[\mathcal{C}]\}$ (10)

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

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

From (1) and (2) and Definition 6,
$\mathcal{A} \cup \mathcal{T}(P_1)$ is *positive* (11)

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

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

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

From (11), (13) and Definition 6,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2)$ *positive* (25)
From (12) and (16) and Definition 5,
$\mathcal{A}$ *allows* $(P_1;P_2)$ (26)
From Definition 6,
$\mathcal{T}(P_3)$ is *positive* (27)
From $\mathcal{T}(P_2)$ above and Axiom 6 and Axiom 9, A sees $[payment^e]$ (28)
From Definition 5 and (28),
$\mathcal{A} \cup \mathcal{T}(P_1;P_2)$ *allows* $B \rightarrow A\ :\ [[payment^e]^{\frac{1}{b}},\ [product^E]^{\frac{1}{b}}]^a$
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), and (22) ,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3) \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I \wedge$
$a \notin \mathcal{S}_I \wedge id \notin S_I \wedge id \notin S_B$ (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: $A \rightarrow B\ :\ [[product^E]^{\frac{1}{a}},\ [\frac{1}{e}]^{\frac{1}{a}}]^b$

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

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,
$\mathcal{A} \cup \mathcal{T}(P_1;P_2;P_3;P_4) \vdash \frac{1}{a} \notin \mathcal{S}_I \wedge \frac{1}{b} \notin \mathcal{S}_I \wedge payment \notin \mathcal{S}_I \wedge product \notin \mathcal{S}_I \wedge$
$a \notin \mathcal{S}_I \wedge id \notin S_I \wedge id \notin S_B \wedge product^E \in \mathcal{S}_A \wedge [payment^e]^{\frac{1}{b}} \in \mathcal{S}_A$ (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: $B \rightarrow A\ :\ [[\frac{1}{e}]^{\frac{1}{b}},\ [\frac{1}{E}]^{\frac{1}{b}}]^a$

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

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

We now add active intruder to our analysis. An active intruder as defined here is one who introduces a fake message into the network, that may be mistaken for a message from one of the participants in the protocol. Formally, if step P_n constitutes $P \rightarrow Q : X^k$, we must show:
$\mathcal{A} \cup \mathcal{T}(P_1,\ ...,\ P_{n-1})$ **not** allows $I \rightarrow Q : x^k$ (or i_n), where x is distinct from X, but of the same form for it to be recorded as the protocol message content instead of X.

It is reasonable to assume that all encryption keys and pseudonyms are regenerated for each transaction. In other words, a, $\frac{1}{a}$, e, $\frac{1}{e}$, E, and $\frac{1}{E}$ have never been used in previous transactions. Therefore, the intruder has no strings signed, secured, or encrypted with these keys from any previous transaction.

$i_1 = I \rightarrow everybody\ :\ [product\ description,\ price\,]^{\frac{1}{b}}$
$\frac{1}{b} \notin \mathcal{K}_I$
Hence, from Definition 7 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1)$ **not** *allows* i_1 (59)

(Here, we are also considering the situation where the intruder may interrupt P_1 and use the message contents to forge a new message)
$i_2 = I \rightarrow B \ : \ [[[product\ description,\ price]^{\frac{1}{b}}]^{\frac{1}{a}},\ a,\ payment^e]^b$
$payment \notin \mathcal{S}_I$
(The payment here has to be a coin owned by A or a withdrawal from an account owned by A. Otherwise, the transaction would be a simple transaction between B and I, where I has taken the pseudonym a.)
From Definition 7 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1,\ P_2)$ **not** *allows* i_2 (60)
$i_3 = I \rightarrow A \ : \ [[payment^e]^{\frac{1}{b}},\ [product^E]^{\frac{1}{b}}]^a$
$\frac{1}{b} \notin \mathcal{K}_I$
From Definition 7 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1,\ ...,\ P_3)$ **not** *allows* i_3 (61)
$i_4 = I \rightarrow B \ : \ [[product^E]^{\frac{1}{a}},\ [\frac{1}{e}]^{\frac{1}{a}}]^b$
$\frac{1}{a} \notin \mathcal{K}_I$
From Definition 7 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1,\ ...,\ P_4)$ **not** *allows* i_4 (62)
$i_5 = I \rightarrow A \ : \ [[\frac{1}{e}]^{\frac{1}{b}},\ [\frac{1}{E}]^{\frac{1}{b}}]^a$
$\frac{1}{b} \notin \mathcal{K}_I$
From Definition 7 we know that,
$\mathcal{A} \cup \mathcal{T}(P_1,\ ...,\ P_5)$ **not** *allows* i_5 (63)
From (59), (60), (61), (62), and (63) we conclude that no active intrusion is possible.

B.2.4 Low Cost

The protocol presented above does not require any complex, computationally intensive computation (encryption and decryption are only moderately intensive and are not very expensive). Furthermore, there are only five messages exchanged between the customer and the merchant. Therefore, the cost incurred is nominal [GS96].