Proceedings of the 24th International Conference on Very Large Data Bases, August 1998, pp. 1-12

Atomicity versus Anonymity: Distributed Transactions for Electronic Commerce

J. D. Tygar
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3891 USA*

Abstract

Electronic commerce challenges our notions of distributed transactions in several ways. I discuss issues how distributed transactions can apply to electronic transactions, with special emphasis on the role of atomicity. I discuss the application of these ideas to two systems I have helped design and build: NetBill (a system for highly atomic micro-transactions) and Cryptographic Postage Indicia (a system for generating postage on laser printers attached to PCs or other devices.) I discuss the difficulties in integrating atomic, anonymous payment systems and some issues in supporting anonymous auctions. Finally, I conclude with a set of open questions.

*Effective September 1998, the author will hold a joint appointment in between the Electrical Engineering & Computer Science Department and the School of Information Management & Systems, both at the University of California, Berkeley, 94720. The author's e-mail addresses will be tygar@cs.berkeley.edu and tygar@sims.berkeley.edu.

This work was in part supported by DARPA (Contract F19628-96-C-0061), NSF (Cooperative Agreement IRI-9411299), and the US Postal Service. The U.S Government is authorized to reproduce and distribute reprints for government purposes, not withstanding any copyright notations thereon. The views and conclusions contained in this document are those of the author and should not be interpreted as reflecting the official policies, either expressed or implied, of any of the supporting agencies or the U.S. Government.

Proceedings of the 24th VLDB Conference
New York, USA 1998

1. Electronic Commerce

Electronic commerce is clearly among the most exciting developments in Internet based applications today. Here are some measures: Dell reports selling more than three million dollars worth of computers each day from their web site. Ernst & Young reports for that the online stores now offer the best prices for 90% of all consumer goods. 10% of all flower orders received by 1-800-FLOWERS now arrive via the world wide web.

Estimates vary on the amount of electronic commerce now occurring. Here is one measure of the excitement over electronic commerce: the 12 June 1995 issue of *Business Week* includes the following projection of the role of electronic commerce. This projection is probably overly optimistic, but it indicates that electronic commerce is being taken seriously in some quarters.

Year	Traditional Commerce (billion $)	Electronic Commerce (billion $)
1994	5150	245
2000	8500	1650
2005	12000	2950

These figure include apparently include all electronic commerce transactions, including inter-bank and EDI electronic commerce. Suppose we restrict ourselves to consumer sales of goods (not services or financial products) done over the world wide web. IDC projects that $20 billion of such consumer good oriented sales will take place in 1998. And clearly, with the explosion of interest in online stock brokers, banks, and other financial service industry providers, electronic commerce will continue to grow.

How do we realize payment technologies for electronic commerce? How can we support payment,

billing, and delivery of goods? A number of groups have built or are building systems to support this, here is a sample list (by no means comprehensive):

- CMU (NetBill)
- Cybercash
- Digicash
- DEC (Millicent)
- First Virtual
- FSTC (E-check)
- Mastercard and Visa (SET)
- Open Market
- Netscape (SSL)
- US Postal Service

This list is very fluid. Statements that I make about the systems being developed today are likely to become outdated as the systems evolve. Any bibliographic listing of references is will rapidly become dated, but [15], [16], [17], [36], [37], and [38] contain nice summaries of much of this work.

Indeed, another measure of excitement in this work is given by the number of conferences devoted to this area. Besides the proceedings listed above, there are numerous workshops and special meetings in the electronic commerce area. By my count, in the 12 months from July 1 1997 to June 30 1998, there were at least 18 special meetings devoted to electronic commerce, and numerous portions of other conferences with sessions or panels devoted to electronic commerce.

Concepts from the distributed transaction community are used heavily in electronic commerce. In particular, in this paper I explore the extension of the traditional notion of atomicity in electronic commerce, and in particular its interaction with anonymity. After briefly reviewing the properties of atomicity and anonymity, I will consider the atomic properties of several electronic commerce protocols. I will then discuss the development of two highly atomic protocols: the NetBill protocol and cryptographic postage indicia. I will explore emerging protocols including highly-atomic anonymous protocols and anonymous auctions. Finally, I will discuss some open problems in electronic commerce.

This paper is keyed to an expository keynote lecture that I will give in August 1998 at the VLDB conference in New York. Therefore, I have adopted a highly informal tone throughout the paper, and, in particular, most of the cryptographic details are not fully presented. Moreover, as I discuss below, the formulation of formal definitions of types of electronic commerce to be an open problem. For those who crave more technical details, I can refer you to the following references:

- Atomicity in electronic commerce: [33]. The first part of the present VLDB paper contains a high-level overview and informal material derived and updated from [33].
- NetBill protocol: [8] (see also appendix A to [33].) For a high-level overview of NetBill, see [29].
- Anonymous atomic protocols: [4]
- Cryptographic postage indicia: [14]. See also the formal standards [34] and [35].
- Auction protocols: [12]

Note that throughout the text I use male pronouns to refer to merchants and female pronouns to refer to customers.

2. Electronic Commerce Properties

We can characterize a variety of properties for electronic commerce. Particular notable in this paper are the properties of atomicity and anonymity.

2.1. Atomicity

Atomicity allows us to logically link multiple operations so that either all of them are executed or none of them are. For example, in transaction processing, one may execute a sequence of code as follows:

<begin-transaction>
state-changing operation 1;
state-changing operation 2;
. . .
state-changing operation *n*;
<end-transaction>

When this block of operations is executed, all of the state-changing operations from 1 to *n*, inclusive will be executed or the state of the system will be as if none of them had been executed.

Why would atomicity every fail to occur? Well, if the transactions are being executed in a distributed environment on multiple processors, then one of the processors executing a state-changing operation or communication between two processors executing state-changing operations may fail. In that case, it may be impossible to complete the entire block of state-changing operations. In these cases, it is necessary to roll-back the processors to a state consistent with the transaction have never been begun in the first place.

Atomic transactions form the cornerstone of modern transaction processing theory. (Nancy Lynch and her

fellow researchers have written an encyclopedic book about atomic transactions [19]; a tremendous resource for those implementing atomic transaction processing systems is the standard textbook [11]; for a thorough review of powerful roll-back methods in the context of computer security and electronic commerce, see [30], [31], and [32].) The "A" in *ACID Transactions* stands for "atomic", no non-atomic distributed transaction system would ever be tolerated by customers of data processing.

However, as we shall see below, the story is quite different in the world of electronic commerce protocols. Most of the proposed protocols are not atomic. For example, if I interrupt a communication between a merchant and a customer, I can often throw an electronic commerce protocol, into an ambiguous state. Money or electronic cash tokens may be copied (with different parties each believing that it has the true, valid copy) or destroyed.

I define three levels of atomicity to protect electronic commerce protocols.

2.1.1. Money Atomicity

Money atomic protocols effect the transfer of funds from one party to another without the possibility of the creation or destruction of money.

For example, a cash transaction is usually money atomic (unless the possibility exists of counterfeiting or destruction of money).

This is a basic level of atomicity that each electronic commerce protocol should satisfy.

2.1.2. Goods Atomicity

Goods-atomic protocols are money atomic, and also effect an exact transfer of goods for money. That is, if I buy a good using a goods-atomic protocol, I will receive the good if and only if the money is transferred. For network protocols, goods atomicity is especially important for information goods. There must be no possibility that I can pay without getting the goods, or get the goods without paying. (Anyone who has had an interrupted file transfer while downloading information on the Internet is aware of the importance of goods atomicity.)

For example, a cash-on-delivery parcel delivery is a good real-world approximation to an electronic commerce protocol. I get the parcel exactly when I have paid the delivery agent.

Goods atomicity is an important property that each electronic commerce protocol intended for information goods should satisfy.

(We originally defined the notion of goods-atomicity in 1995, see [5] or [29] for example. However, in 1997, Franklin and Reiter redefined this property as *fair-exchange,* see [9]. Note that fair-exchange does not include the properties of certified delivery mentioned below.)

2.1.3. Certified Delivery

Certified delivery protocols are money atomic and goods atomic protocols that also allow both a merchant and a customer to prove exactly which goods were delivered. If I buy a document entitled "How to make a million dollars fast on the Internet" and receive an electronic copy of some unrelated or garbage material, I will want to complain to an authority. To rapidly resolve the question, both the merchant and the customer will want to be able to prove the exact contents of what was delivered.

For example, a certified delivery protocol corresponds to a cash-on-delivery parcel delivery where the contents of the parcel is opened in front of a trusted third-party who immediately records in an indestructible form the exact contents of the parcel.

Certified delivery protocols are helpful for scenarios where merchants and customers may be untrusted. Today, there is no effective way to distinguish a large trusted WWW merchant from a fly-by-night impressive electronic storefront that actually connects to a shop that contains a fraudulent operation.

2.2. Anonymity

Some people want to keep their purchases private. They do not want to have third-parties (or even merchants) know their identity. This concern may arise because the customer is buying a good of questionable social value (e.g., pornography); or because the customer does not want to have his name added to a marketing or mailing list; or for illegal reasons (e.g., to evade taxes); or simply because the customer personally values privacy.

Although most paper money contains serial numbers, cash transactions can often have anonymous properties. Serial numbers are rarely traced and recorded, and if I buy something from a merchant who does not know me or from a vending machine, my purchase is often effectively anonymous.

David Chaum has been the most influential advocate of anonymous electronic commerce protocols. He has written a number of highly influential papers on topics such as "anonymous digital cash", (in particular [6]) these in turn have inspired all electronic commerce researchers. Modern researchers have improved his protocols; a representative sophisticated example of the current version of his protocols can be found in [3].

Here is the way these protocols work:

a) a customer withdraws money from the bank, receiving a cryptographic token which can be used as money;

b) the customer applies a cryptographic transformation to the money that still allows a merchant to checks its

validity, but make it impossible to trace the customer's identity;

c) the customer spends the money with the merchant. (in doing so, she applies a further cryptographic transformation so that the merchant's identity is used in the data);

d) the merchant checks to make sure that he has not received the token previously;

e) the merchant sends the goods to the customer;

f) at a later point, the merchant deposits his electronic tokens at the bank; and

g) the bank checks the tokens for uniqueness; the identities of the customers remain anonymous except in the case when a customer had double-spent a token—if a token was double-spent, the identity of the customer is revealed and the network police are notified of attempted counterfeiting.

Now consider when a communication failure happens around step (c). The customer has no way of knowing if a merchant has received her token or not. The customer has two options:

- The customer can return her electronic token to the bank (or spend it on a different merchant.) If she does this, and the merchant actually received her token, then when the merchant cashes in the token, the customer's anonymity will be revealed. Even worse, the customer will be likely to be accused of fraud.
- The customer can take no action, failing to return her token. If she does this, and the merchant never received her token, then she is in danger of losing her money. She will have never received the good she attempted to purchase, and she will be unable to use her money.

In either case, money atomicity breaks down. (It is worth noting that the commercial version of Digicash does not use this off-line approach, perhaps because of the weaknesses described above. Instead, the commercial version of Digicash apparently uses a fully on-line protocol.)

In many countries, most anonymous transactions are illegal. For example, in the United States, the Money Laundering Act (12 USC §1829) requires that electronic commerce systems should both

- promptly report any transaction valued over $10000.
- record any transaction valued over $100.

These requirements have not been tested in court for digital cash systems. However, it is not clear that digital cash systems will be upheld as satisfying the requirements

I also note that it is often possible to achieve a limited form of anonymity by having a proxy agent complete purchases for the customer. In this case, the transaction may be easily traced to the proxy agent, which keeps private the identity of the true customer (see [7] for details).

2.3. Security

Can we trust anyone in cyberspace? Communications can be easily intercepted, messages can be inserted, and the absolute identity of other parties may be uncertain. Clearly, security will be important for any electronic commerce protocol.

By contemporary standards, the current form of credit cards, which reveal a customer's identity and charge numbers to a merchant and to anyone who can obtain a copy of the receipt, would be unlikely to be accepted if they were introduced newly today.

Many electronic commerce systems depend on some ultimate, trusted authority. For example, NetBill depends on the trustworthiness of a central server. However, even in the case where one uses a trusted server, one can minimize the effects of the security failures of that server. For example, in NetBill, detailed cryptographically-unforgeable records are kept so that if the central server was ever corrupted, it would be possible to unwind all corrupted actions and restore any lost money.

2.4. Transaction size

The average credit card transaction has typically been estimated to be on the order of $50. Depending on the arrangements made with a bank, a merchant pays approximately 30¢ plus 2% of the purchase price for each and every transaction. For many telephone or mail order businesses, the actual rate is closer to 50¢ plus 2.25%.

If one is engaging in a transaction that is only worth 10¢ or even 1¢, the standard credit card rates would dominate the cost of the item. Thus, a number of parties have proposed support for *microtransactions* or transactions less than $1. (By no means is 1¢ the minimum transaction value of interest; Mark Manasse at Digital Equipment Corporation's System Research Center has developed an electronic commerce system named Millicent [20].)

Both NetBill and cryptographic postage indicia are motivated by the idea of supporting microtransactions. Some of the design decisions made for those systems can only be understood by the microtransaction requirement. However, a detailed discussion of microtransactions is beyond the scope of this paper.

(For those who are curious: the key idea behind most microtransaction protocols is to aggregate many small transactions charged using specially optimized protocols;

then charge the aggregated total as a large value transaction. This idea is a beautiful application of protocol nesting. For a discussion of microtransactions in NetBill, see [29]; for a completely different approach, see [20].)

3. Non-atomic Protocols

3.1. Digicash

Digicash (as described in [6]) uses an anonymous digital cash protocol. As discussed in Section 2.2, digital cash protocols are not money atomic; indeed, in the event of communication failure, they can fail to be anonymous as well. (It is worth noting that the commercial version of Digicash has apparently abandoned the approach of using a purely off-line version of their protocol.)

Finally, digital cash protocols use several rather computationally intensive cryptographic operations, so the question of their applicability to general micro-transactions is not clear.

3.2. First Virtual

First Virtual allows users to freely buy goods. First Virtual then uses e-mail to confirm each and every transaction with the customer. Setting aside the acceptability of flooding a user with e-mail for purchases in this way, this model clearly preserves money atomicity, although it clearly fails goods atomicity (since the customer can buy an item without paying for it.) First Virtual apparently considers goods atomicity to be relatively unimportant. (Indeed, First Virtual takes a dim view of communications security and encryption in any form; in [1], they argue that communications security is "irrelevant" and they dismiss electronic commerce designers who postpone deployment of their systems to perfect strong security guarantees.)

First Virtual's system can easily be a target of fraud and atomicity failures. It is somewhat better than digital cash, but inferior to other electronic commerce systems.

Ultimately, First Virtual translates each electronic commerce transaction into a credit card transaction, making First Virtual in its current form of limited value for microtransactions. (First Virtual suggests using aggregation, but they can not aggregate across different merchants in a single credit card transaction.)

3.3. SSL

The Secure Socket Layer (SSL) approach sets up a secure communication channel (using cryptography) to transfer a customer's credit card number to the merchant. This approach is equivalent to reading your credit card number over the phone to a merchant using a secure telephone connection.

This approach offers money atomicity to the extent that credit card transactions are money atomic. However, its security properties are less clear; for example, since a (potentially unscrupulous) merchant has the customer's credit card number, he can use it to commit fraud. (Merchant fraud is one of the most serious problems facing the credit card industry [41]. Lyndon LaRouche is a well-known example of a person who committed merchant credit card fraud.)

Goods atomicity is not addressed by SSL.

In its current form SSL is clearly of limited value for microtransactions.

3.4. SET

Visa and Mastercard have developed a combined protocol called Secure Electronic Transaction (SET) that has strong security properties [21]. This was formed from a variety of previously published protocols: STT (Visa/Microsoft), SEPP (MasterCard) and the iKP family of protocols (IBM). SET, and the protocols from which it is adapted, is an example of a secure credit card based protocol. In SET, the customer digitally signs a purchase request and a price and then encrypts payment information (in the form of a credit card number, for example) with a bank's public key. The merchant acknowledges the purchase, and forwards the request to the bank. The bank processes the request, and if the prices match, the bank charges the customer's account and instructs the merchant to complete the sale.

Like SSL, this approach offers money atomicity to the extent that credit card transactions are money atomic. However, the security properties of SET are superior since they prevent merchant fraud. Goods atomicity is not addressed by SET.

I should mention that the SET protocol is unusually complex (the description of the protocol exceeds several hundred pages) and it is questionable whether the protocol is actually secure in practice or not. Even basic properties are questionable. For example, SET went to considerable lengths to prevent a merchant from obtaining a consumer's credit card number. However, it turns out that some merchants organize customer records based on the purchase account used by a consumer. For this reason, SET explicitly allows a "back-door" by which a merchant may receive a consumer's credit card number. Is this a security problem? It is not clear. Even popular media such as the *New York Times* has weighed in with doubts about SET.

In its current form, SET is of limited value for microtransactions.

4. NetBill

My co-researchers and I developed NetBill to provide all three levels of atomic transactions. Here, I give a broad sketch of the NetBill format and some rough arguments of why it satisfies all three atomicity conditions: money

atomicity, goods atomicity, and certified delivery. However, to keep my explanation simple, I do not cover the details of the protocol, leaving that for more detailed presentations (see [8] or [33]). For example, I do not discuss here how NetBill protects against message replay, communication security, or various timing attacks.

The NetBill protocol is between three parties: a customer, a merchant, and the NetBill server. Think of a NetBill account held by a customer as equivalent to a virtual electronic credit card account.

Here is the outline of the NetBill protocol

a) The customer requests a price from the merchant for some goods. (This step is necessary because the price of a good may depend on the identity of the customer; for example, a student ACM member may qualify for a discount on some items)

b) The merchant makes an offer to the customer

c) The customer tells the merchant that she accepts the offer.

d) The merchant sends the information goods requested encrypted by key K.

e) The customer prepares an electronic purchase order (EPO) containing a digitally signed value for: <price, cryptographic-checksum of encrypted goods, time-out>. The customer sends the signed EPO to the merchant

f) The merchant countersigns the EPO. The merchant also signs the value of K. The merchant sends both values to the NetBill server.

g) The NetBill server checks the signature and countersignature on the EPO. It then checks the customer's account to ensure that sufficient funds exist to approve the transaction, and also checks that the time-out value in the EPO has not expired. Assuming that all is OK, the NetBill server transfers price funds from the customer's account to the merchant's account. It stores K, and the cryptographic-checksum of the encrypted goods. It then prepares a signed receipt that includes the value K. It sends this receipt to the merchant.

h) The merchant records the receipt, and forwards it to the customer (who can then decrypt her encrypted goods.)

This protocol thus transfers an encrypted copy of the information goods, and records the decryption key in escrow at the NetBill server. Now let us see how this protocol provides various types of atomicity protection.

Money atomicity: all funds transfers occur at the NetBill server, and since the NetBill server uses a local atomic database to store fund values, no money can be created or destroyed.

Goods atomicity: if the protocol fails as a result of communications failure or processor failure before the NetBill server atomically processes the transaction in step (g), then no money changes hands, and the customer never receives the decryption key — he gains no access to the encrypted information goods. On the other hand, if step (g) succeeds, then both the merchant and NetBill server will record the value of K. Normally, these values would be forwarded back to the customer as a result of step (h), but if something goes wrong, the customer can obtain K from either the merchant or NetBill server at any time.

Certified delivery: since we have goods atomicity, we know that the customer received something in exchange for money. Now, suppose that the customer claims that he receives goods different from what she ordered. Then, since NetBill server has a cryptographic checksum of the encrypted goods that is countersigned by both the customer and the merchant, the customer can present her encrypted goods to a judge and verify that she has not tampered with the goods. Now, since a merchant-signed value of K is stored at both the customer and the merchant, the judge can decrypt the goods and determine whether the goods were as advertised as not.

Thus NetBill presents an example of a highly-atomic electronic commerce protocol. We have currently built an alpha version of NetBill at Carnegie Mellon (in conjunction with our development and operations partners, Cybercash, Mellon Bank and Visa International), and we hope to prove that NetBill is not only highly-atomic but that it has the performance, scalability, and efficiency to handle a large number of microtransactions. The protocol has been licensed for commercial use to Cybercash, and in large part, the Cybercash *CyberCoin* protocol uses a NetBill style approach.

4.1. Anonymous Atomic Protocols

For a long time, the possibility of an atomic, anonymous protocol was open. As discussed above, conventional approaches to anonymity have been at odds with money atomicity, not to mention goods atomicity and certified delivery.

In [4], we showed that anonymous, atomic protocols were fully possible.

The basic idea came straight out of distributed transaction design. We postulate a publicly readable transaction log as a separate entity (from the consumer, merchant, and bank). Individuals log entries in the transaction log; encrypting them and signing them using public-key primitives as necessary to protect the message. Using this approach, we can easily implement a version of two-phase commitment. As necessary users can write messages but they can write them to anonymous identities associated only with a public key. The full details are worked out in [4].

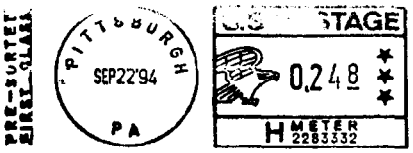

Figure 1: Traditional indicia are easy to copy.

Moreover, [4] also draws a distinction between one-sided certified delivery (where only the consumer can prove the contents of an item delivered) and two-sided certified delivery (where both parties can prove the contents.) If the burden of proof is always on the consumer to prove the results of the transaction, then one-sided certified delivery is sufficient to provide full protection and permits a substantially simplified protocol.

5. Cryptographic Postage Indicia

Is it possible to achieve money atomicity without using a central server? Yes, one way to do this is to use secure hardware. For example, FIPS 140-1 [23] specifies support for tamper-proof and tamper-resistant devices that can store information and perform processing tasks. These devices are secure in the sense that any attempt to penetrate them will result in erasure of all information stored inside them. We could use this to store an electronic wallet; when a charge is made, the electronic wallet withdraws funds.

We call these tamper-proof devices *secure-coprocessors.*

Now the design of such a system is not easy [42], and there are quite a few risks associated with customer approval of transactions [10]. However, with careful design it can be made to work.

My research group has been working with the US Postal Service to develop standards for PC-generated laser printed indicia for postage meters. These are designed to meet the needs of the Postal Service Information-Based Indicia Program [35].

As Figure 2 shows, it is trivial to copy traditional indicia using a scanner and a computer. It is equally easy to forge dates and postage values on counterfeited indicia. (Note: if you ever decide to take up the life a criminal and forge indicia, make sure to add smudges to the indicia — indicia that are reproduced too clearly can easily be recognized as forged.)

Using a secure coprocessor, it is easy to store an account balance for postal customers. This account balance is decremented whenever postage is printed. Now, the secure coprocessor prepares a cryptographically signed message that contains envelope data (sender address, receiver address, date sent, and sequence number). This information is then printed on the envelope using an efficient data representation such as PDF-417 [18]. Figure 2 shows Lincoln's Gettysburg address encoded in PDF 417. PDF 417 normally encodes 400 bytes per square inch.

When mail is received at a postal sorting facility, the data block is checked to see if they match the address used for sorting, and to verify the uniqueness of the sequence number. (Note that all mail to given address will be processed by a single sorting station.) Indicia remain valid for six months. (The US Postal Service claims to deliver more than 90% of all first class mail within three days of being sent and more than 99% in seven days. Thus, six months would appear to be a generous bound for mail delivery.) The database stored at a local sorting station can regularly be purged of entries with a date older than six months

If an adversary attempts to break money atomicity by forging indicia, he must do one of two things:

- copy existing indicia, which then will only be valid for the encrypted delivery address, and will be caught at the sorting station; or
- attempt to find the value used to digitally sign the cryptographic indicia, which will require opening the secure coprocessor, erasing all the vulnerable data within.

For a more technical exposition on secure coprocessors, see [14], [42], and [43].

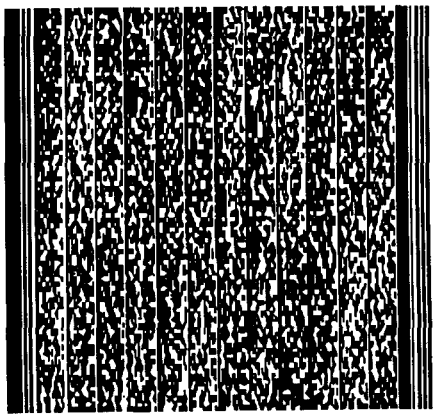

Figure 2: PDF 417 encoding of Abraham Lincoln's Gettysburg Address.

6. Anonymous Auctions

A challenging extension of the ideas I discuss in this paper is their application to auctions. Of course, auctions are vital, and auction markets (such as stock markets) are central to our modern financial systems.

In [12], we present a fully anonymous auction technique. To put this in context, consider the following desiderata for auctions:

Economic design: we want the auction to be designed on solid economic principles and for participants to have incentives to bid as they truly value the item. This is known as the participant's *valuation*, and is also called the *indifference price*. If bidders bid less than their true valuations, it is possible that the final winning bid may be artificially low. I illustrate this below in the discussion of sealed-bid auctions.

Fast execution: we want to have the auction run quickly.

Privacy: we want the auction to be private, for others to not know our actual bids. We don't even want the auctioneerto know the bids. The only exception to this rule is that we will reveal the final price the item is sold at. This may at first this may seem like a paradoxical condition, but it is commonly achieved in Dutch auctions, discussed below. Note that this is a quite useful requirement; otherwise we give away detailed information on our preferences that may be used in the future to inform "shills" who work for the seller to attempt to artificially drive up the price an item is sold at (creating a disincentive to bid the true valuation.)

Anonymity: we don't want our identities to be revealed. One way to achieve this is to use an intermediary to anonymously forward our bids. Note that privacy is different from anonymity; privacy protects the values of the bids while anonymity protects the identities of the bidders. Even if our bids are anonymously forwarded, participants (such as the auctioneer) may learn the distribution of our bids.

[12] discusses how to hold a true auction that combines the first three features. If we add anonymizing intermediaries [7] to the mix, we can achieve an auction with all four properties.

6.1. Auction Types

How do existing auction types stack up against our desiderata?

Consider these three broad categories of auctions that have been proposed:

Increasing-price auction (English auction): In this type of auction, a good or commodity is offered at increasing prices. It may initially be offered for K tokens, at successive points of time i it is bid at $K + i * delta$ tokens (*delta* may be a function of previous bids and other factors). At each unit of time, one or more parties can bid for the item. At the end of the auction, the highest bidder takes the item; he pays the price he bid. This is the sort of auction found at Sotheby's and Christie's.

This type of auction has many disadvantages: the time necessary to conduct the auction is potentially proportional to the price at which the item is sold; the communication costs may grow super-linearly in the ultimate price at which the item is sold (since at lower prices, multiple bidders may simultaneously bid for an item); moreover, this type of auction leaks an enormous amount of information (a careful observer will be able to deduce information about the price that each party is willing to pay for the auctioned good.)

However, the auction does have a very desirable feature: in economic terms, it allocates the good to the bidder with the highest valuation, since the bidder with the highest valuation will be willing to outbid all other bidders. (It is good for consumers too; using economic terminology, it maximizes the *consumer surplus*.)

Sealed-bid auctions: In this type of auction, each party sends a sealed bid to an auctioneer who opens all bids. The auctioneer determines the highest bid and sells the item to that bidder for the bidding price. This type of auction can execute in a single round of communication between the bidders and the auctioneers. However, it has disadvantages. First, the auctioneer will know the exact price that each party is willing to pay. Second, it does not support optimal distribution of goods.

In a sealed bid auction, participants will have beliefs about what others will bid. If a participant believes that she will have the highest bid, and the second highest bid will be substantially beneath that, then she has an incentive to lower her bid. For example, if she values an item at $1,000, but believes that the second highest bidder values the item at $500, then she is likely to place a bid slightly higher than $500. If she is wrong about the distribution of other bids, then the final item will not go to the party that values it most, and the seller will have given up the item a price lower than he would have achieved with an English auction.

Decreasing-price auction (Dutch auction): This type of auction is similar to the English auction in that the bidding price varies over time; however in this case, the price decreases and at time i is $K - i * delta$. The first bidder will take the item. This type of auction has the advantage of preserving maximum privacy; no information is revealed except the winning bid and bidder.

However like the increasing-price auction, it may be time consuming, and like the sealed-bid auction, it is not economically efficient.

In Nobel-prize winning work [40], the economist William Vickrey designed a type of auction that combined

the best features of an increasing-price auction and a sealed-bid auction. Vickrey's technique, called a second-price auction, works like a sealed-bid auction, in that all bids are sealed and sent to an auctioneer. Like a sealed bid auction, the highest bidder wins. But the price the winner pays is the price that the second highest bidder has bid. For example, suppose that we bid 100 tokens and the second highest bid is 10 tokens. Then we will win the bid, but we will only have to pay 10 tokens to secure the good. This auction runs in constant time, and maximizes consumer surplus, but it is still highly centralized and does not protect the privacy of the bids.

[12] gives a private-bid version of a second-price auction. This auction

- runs in a single round of bid submissions (like a sealed-bid auction),
- is efficient enough for practical implementation,
- will maximize consumer surplus and will give incentives for participants to submit bids at their true valuations (like an English auction), and
- will preserve bid privacy (like a Dutch auction).

This is quite an unusual result. In the end, only the second highest bid is revealed. The auctioneers and participants (except for the winner) will be completely unaware of the numerical value of the highest bid (or an other bid besides the second highest).

6.2. Auctions with private bids

The length of this article does not permit a full presentation of how auctions can be handled with private bids. Instead, let me present a simpler result.

First, consider the following mental experiment. Suppose that we want to calculate the average salary of all the people in your department, without revealing any single salary. Here is how we can do it: Pick a large modulus M, much greater than the dollar value of everyone's salary (for example, one trillion). We specify a set of three "accumulators". Each person picks two random values a and b modulo M, and then picks a third value c such that $a+b+c$ mod M is equal to his/her salary. Now, each employee privately sends his/her a value to accumulator 1, his/her b value privately to accumulator 2, and his/her c value privately to accumulator 3. The accumulator privately sum each of their values modulo M (so accumulator 1 sums all the a values, etc.) and then they report them. If we sum the three accumulators' values modulo M, we'll have the sum of everyone's salary, and then it is trivial to find the average salary.

Note that with only the a value (for example) or even with two of the three values, one can not determine the salary of the employee. All three values are needed to gain any information whatsoever about the value of a particular employees salary.

Now, we can see how to extend this to compute max of a number of values. Suppose we all pick a value between 1 and 100, how can we privately find the maximum value of all these entries. Suppose we generate an array vector where every value less than our guess is a random value modulo M, and every value greater than our guess is 0. If we take the private element-wise sum of all these entries, then with high probability, the max value will correspond to the highest non-zero entry in the sum vector. This is essentially how one can hold a private sealed-bid auction. (To determine the winner, all bidders would need to also cryptographically commit to their bid, and then they could prove that they bid highest. Note that we ignore the issue of ties for highest bids.)

With a little bit of thought, the reader will realize that if M is a large prime, then finding the min of a set of numbers corresponds to computing the element-wise product of a set of vectors. Using this, the reader may begin to see how to compute the second price bid. Proving the full privacy properties is tricky, and making it efficient is very tricky, so I refer the reader to [12] for all the details.

7. Open Problems

The field of electronic commerce has many open problems. Here are some of my favorites:

- What other atomicity models exist in electronic commerce (besides money atomicity, goods atomicity, and certified delivery)? Is there a general schema?
- What is the minimum number of message exchanges necessary in an atomic purchase? (For example, with some thought, we can reduce the 6 core steps in the NetBill protocol to 5 steps; can we reduce it further?)
- What atomic electronic commerce mechanisms can be built for multiple banks or billing servers?
- Can atomicity be used for continuously delivered information (such as continual stock market updates) or very large objects (such as video programs)?
- Can we give a formal definition for atomicity (in the sense of electronic commerce)?
- How can we prove that a protocol is atomic (in the sense of electronic commerce)?
- Is it possible to express atomic properties in terms of model checking? (See [13] for details.)
- Can we extend electronic commerce auctions to full auction markets, such as stock markets?

- Can we protect redistributed information or reselling of information? (This is the so-called superdistribution of Mori and Kawahara [22].)
- Can we devise effective *digital watermarks* that clearly indicate the purchaser of illegally pirated or redistributed information?
- How can we represent and enforce electronic contracts governing the use, distribution, and payment conditions for information goods and software?
- Can we make a fault-tolerant version of electronic commerce protocols that remain stable even when banks fail? (The results of T. Rabin and Ben-Or [26] seem to be appropriate here.)
- Can we build systems to allow anonymous charitable contributions? Can we extend them to allow documentation so that one can take a tax credit?
- What is the minimum level microtransaction that can be supported in electronic commerce? The minimum level atomic microtransaction?
- We can express money as tokens or as entries in a server (see [5]) — is there anyway to express a formal equivalence between these two methods?

8. More Information

- More information on NetBill can be found at

  `http://www.ini.cmu.edu/netbill/`.
- For a consumer interface to NetBill, see

  `http://www.netbill.com/`
- More information on cryptographic postage indicia and secure coprocessors can be found at

  `http://www.cs.cmu.edu/afs/cs/project/dyad/www/`.
- Many of the papers I cite can be found at my web site

  `http://www.cs.cmu.edu/~tygar`.
- In September 1998, I will move to UC Berkeley, so look for my home pages at

  `http://www.sims.berkeley.edu/~tygar`
  or
  `http://www.cs.berkeley.edu/~tygar`

8.1. Acknowledgments

- Bennet Yee, as all of the work described here regarding secure coprocessors is joint work I've done with Bennet. We jointly observed that Chaum-like digital cash protocols fail to work properly if communications are interrupted, thus inspiring our work. Portions of our work previously appeared in [42] and [43].
- Nevin Heintze, who contributed to the later development of cryptographic postage indicia as represented in [14]; and Ali Bahreman, who started me thinking about certified delivery in [1].
- Ben Cox, Tom Wagner, and especially Marvin Sirbu, my collaborators in the NetBill protocol; see [8] and [29].
- Jean Camp, who made an initial division between money atomicity and goods atomicity, see [5].
- Mike Harkavy, who perfected anonymous anonymity in [4] and anonymous auctions in [12].
- Thomas Alexandre, Brad Chen, Howard Gobioff, Maurice Herlihy, David Johnson, Hiroaki Kikuchi, Mahadev Satyanarayanan, Sean Smith, Alfred Spector, Jiawen Su, Mark Tuttle, Jeannette Wing, and Hao-Chi Wong, for their useful comments.

References

[1] A. Bahreman and J. D. Tygar. "Certified Electronic Mail." In *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pages 3–19, San Diego, CA, February 1994.

[2] N. Borenstein. "Perils and Pitfalls of Practical Cyber Commerce: the Lessons of First Virtual's First Year." Presented at *Frontiers in Electronic Commerce*, Austin, TX, October 1994.

[3] E. Brickell, P. Gemmell, and D. Kravitz. "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change." In *Proceedings of the Sixth ACM-SIAM Symposium on Discrete Algorithms*, pages 457–466, 1995.

[4] L. Camp, M. Harkavy, J. D. Tygar, and B. Yee, "Anonymous Atomic Transactions." In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 123–133, November 1996.

[5] L. Camp, M. Sirbu, and J. D. Tygar. "Token and Notational Money in Electronic Commerce." In *Proceedings of the First USENIX Workshop on Electronic Commerce*, pages 1–12, July 1995.

[6] D. Chaum, A. Fiat, and M. Naor. "Untraceable electronic cash." In Advances in Cryptology:

Crypto '88 Proceedings, Springer Verlag, pages 200–212, 1990.

[7] B. Cox. *Maintaining Privacy in Electronic Transactions.* Information Networking Institute Technical Report TR 1994–8, Fall 1994.

[8] B. Cox, J. D. Tygar, and M. Sirbu. "NetBill Security and Transaction Protocol." In *Proceedings of the First USENIX Workshop on Electronic Commerce,* pages 77–88, July 1995.

[9] M. Franklin and M. Reiter. "Fair Exchange with a Semi-Trusted Third Party." In *Proceedings of the 4th ACM Conference on Computer and Communications Security,* pages 1-5, April 1997.

[10] H. Gobioff, S. Smith, and J. D. Tygar. *Smart Cards in Hostile Environment.* CMU-CS Technical Report CMU-CS-95-188, September 1995.

[11] J. Gray and A. Reuter. *Transactions Processing: Techniques and Concepts.* Morgan Kaufmann, 1994.

[12] M. Harkavy, H. Kikuchi, and J. D. Tygar. "Auctions with Private Bids." To appear in the *Proceedings of the Third USENIX Workshop on Electronic Commerce,* October 1998.

[13] N. Heintze, J. D. Tygar, J. Wing, and H. Wong. "Model Checking Electronic Commerce Protocols." In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 147–164, November 1996.

[14] N. Heintze, J. D. Tygar, and B. Yee. "Cryptographic Postage Indicia." In J. Jaffar and R Yap, editors, *Concurrency, Parallelism, Programming, Networking, and Security,* Springer-Verlag, Lecture Notes in Computer Science 1179, December 1996.

[15] R. Hirschfeld, editor, P*roceedings of Financial Cryptography International Conference, FC'97.* Springer-Verlag, Lecture Notes in Computer Science 1318, 1997.

[16] R. Hirschfeld, editor, P*roceedings of Financial Cryptography International Conference, FC'98.* To appear.

[17] IEEE Spectrum. *Special Issue on Electronic Money.* February 1997.

[18] S. Itkin and J. Martell. *A PDF417 Primer: A Guide to Understanding Second Generation Bar Codes and Portable Data Files.* Technical Report Monograph 8, Symbol Technologies. April 1988

[19] N. Lynch, M. Merritt, W. Weihl, A. Fekete. *Atomic Transactions.* Morgan Kaufmann, San Mateo, 1994.

[20] M. Manasse. "The Millicent Protocols for Electronic Commerce." In *Proceedings of the First USENIX Workshop on Electronic Commerce,* pages 117–123, July 1995.

[21] MasterCard, Inc. and Visa, Inc. SET Draft Specification.

[22] R. Mori and M. Kawahara. "Superdistribution: the Concept and the Architecture." In *Transactions of the Institute of Electronics, Information, and Communication Engineers (Japan),* E73(7) pages 1133–1146.

[23] National Institute of Standards and Technology. *FIPS 140-1: Security Requirements for Cryptographic Modules.* January 1994

[24] National Institute of Standards and Technology. *FIPS 180: Federal Information Processing Standard: Secure Hash Standard (SHS).* April 1993.

[25] National Institute of Standards and Technology. *FIPS 186: Federal Information Processing Standard: Digital Signature Standard (DSS).* May 1994.

[26] T. Rabin and M. Ben-Or. "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority." In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 73–85, May 1989.

[27] R. Rivest, A. Shamir, L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." In *Communications of the ACM,* 21(2), February 1978.

[28] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* New York: John Wiley & Sons, 1994.

[29] M. Sirbu and J. D. Tygar. "NetBill: An Internet Commerce System Optimized for Network Delivered Services." In *IEEE Personal Communications*, 2(4) pages 3439, August 1995.

[30] S. Smith. *Secure Distributed Time for Secure Distributed Protocols.* Ph.D. Thesis, CMU-CS

Technical Report CMU-CS-94-177, September 1994.

[31] S. Smith, D. Johnson, and J. D. Tygar. "Completely Asynchronous Optimistic Recovery with Minimal Rollbacks." In *Proceedings of the 25th International IEEE Symposium on Fault-Tolerant Computing*, pages 362–372, June 1995.

[32] S. Smith and J. D. Tygar. "Security and Privacy for Partial Order Time." In *Proceedings of the ISCA International Conference on Parallel and Distributed Computing Systems*, pages 70–79, October 1994.

[33] J. D. Tygar. "Atomicity in Electronic Commerce." In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 8–26, May 1996. This work, without the technical appendices, but with a few technical updates in D. Denning and P. Denning, editors, *Internet Besieged,* Addison-Wesley and ACM Press, pages 389–406, 1998. The latter work will further be reprinted in the May/June 1998 issue of *ACM Net Worker* magazine, together with an updating sidebar.

[34] US Postal Service. *Information Based Indicia Program: Indicia Specification*, July 1997.

[35] US Postal Service. *Information Based Indicia Program: Postal Secure Device Specification*, July 1997.

[36] USENIX Association. *Proceedings of the First USENIX Workshop on Electronic Commerce*, July 1995.

[37] USENIX Association. *Proceedings of the Second USENIX Workshop on Electronic Commerce*, November 1996.

[38] USENIX Association. *Proceedings of the Third USENIX Workshop on Electronic Commerce*, October 1998 (to appear).

[39] H. Varian. "Economic Mechanism Design for Computerized Agents." In *Proceedings of the First USENIX Workshop on Electronic Commerce,* pages 13–22, July 1995.

[40] W. Vickrey. "Counterspeculation, Auctions, and Competitive Sealed Tenders." Journal of Finance (volume 16), pages 8-37, 1961.

[41] Visa USA and Anderson Consulting. *1992 Credit Card Functional Cost Study.* September 1992.

[42] B. Yee. *Using Secure Coprocessors.* Ph.D. Thesis, CMU-CS Technical Report CMU-CS-94-149, May 1994.

[43] B. Yee and J. D. Tygar. "Secure Coprocessors in Electronic Commerce Applications." In *Proceedings of the First USENIX Workshop on Electronic Commerce,* pages 155–170, July 1995.