# An Atomicity-Generating Layer for Anonymous Currencies

L. Jean Camp, Ph.D.
L-213
Harvard University
Cambridge MA 02138

*Abstract*[*]

Atomicity is a necessary element for reliable transactions (Financial Service Technology Consortium, 1995; Camp, Sirbu and Tygar, 1995; Tygar, 1996). Anonymity is also an issue of great importance not only to designers of commerce systems, (Chaum, 1982; Chaum, 1989; Chaum, Fiat & Naor, 1988; Medvinski, 1993), but also to those concerned with the societal effects of information technologies (Branscomb 1994. Compaine 1985, National Research Council 1996, Neumann 1993, Poole 1983). Yet there has been a trade-off between these two elements in commerce system design. Reliable systems, which provide highly atomic transactions, offer limited anonymity (Visa, 1995; Sirbu and Tygar, 1995; Mastercard, 1995, Low, Maxemchuk and Paul, 1993) . Anonymous systems (Chaum, 1985; Chaum 1989; Medvinski, 1993) do not offer reliable transactions as shown in Yee, 1994; Camp, 1999; and Tygar, 1996. This work illustrates that any electronic token currency can be made reliable with the addition of this atomicity-generating layer.

[*] This work was begun at Carnegie Mellon University with support from the US Postal Service, and an equipment grant from IBM. This work is the opinion of the author and does not necessarily reflect the opinion of any funding agency, employer, or the US Government. I would like to acknowledge Michael Harkavey, Marvin Sirbu, Doug Tygar, and Bennet Yee.

## *Introduction*

Electronic commerce includes sending electronic payments over a public network to obtain electronic goods or promises of the delivery of physical goods. Can customers protect their privacy and their money in such a transaction? Privacy means that the subject of information controls that information. Anonymity means that information has no subject -- that is, identity is not linked to the information. To protect privacy and money, Internet transactions must be atomic and anonymous. Atomicity means that a transaction cannot be broken into its parts (payment , delivery and receipt). An atomic transaction succeeds completely or fails completely.

There are two kinds of money: notational and token. With notational money funds exist as entries in a ledger, such as in a banking record. In the case of notational money the study of database transactions can be directly applied to ensure consistency. Notational money is most common today in point of sale exchanges using ATM cards. Electronic commerce proposals based on checks and credit cards are notational money. However, token moneys are a difficult problem. Token moneys are analogous to cash -- the value is bound to the instrument. Thus sending token moneys over networks can result in the same problems as sending cash through the mail. That is, the recipient may deny its receipt or the cash may be lost, stolen or destroyed in transit. This problem has also been referred to as the "dine and dash" problem (Simon, 1996).

In electronic commerce, the payment message must travel over an insecure network from the customer to the merchant. Without verifiable acknowledgment in the protocol, the customer will not know that the merchant received the payment message. Under the standard transmission control protocol (TCP), a payment may be duplicated when the communications protocol believes the packet containing the payment message was lost on the network. Moreover, a payment message may be destroyed by network failure. If a payment message is lost, delayed, or destroyed, confusion rather than consistency may result.

To better understand how anonymity and atomicity conflict consider the use of Digicash[1] token (Yee, 1994). If a transfer of Digicash tokens from the customer to the merchant is interrupted, then it is possible that both or neither party may believe it has legitimate access to the token. The customer may attempt to resolve this state by canceling the token (by cashing it in), but if the merchant also does this, the result is a race condition. Another possible failure creates an opportunity for theft. The bank cannot trace tokens. Thus if a merchant claims that he was not paid by a particular customer, and the customer otherwise; there is no technique for dispute resolution. There is no link between transaction and token. The customer has been defrauded. In short, what is needed, and what is presented here, is a general technique for anonymous rollback. This problem has been solved in a limited case (Camp, Harkavy, Tygar and Yee, 1996) here I present an extension for the general case.

The atomicity-generating layer presented here provides the highest level of atomicity, and removes the risk of loss or stolen tokens. Previous work has defined degrees of atomicity (Camp, Sirbu and Tygar, 1995; Tygar 1996). In *money atomic* transactions, money is conserved. In transactions with *goods atomicity*, there is money atomicity and the delivery of information goods is linked to payment. The merchant is paid if and only if some goods

[1] I refer to the family of Chaum's blinded tokens with no embedded identity information as "Digicash tokens" for reasons of brevity.

are delivered. *Certified delivery* subsumes money atomicity, goods atomicity, and in addition requires that the merchant is paid if and only if the *promised* goods are delivered. In certified delivery the content of the goods can be proven to a third party. Note that goods atomicity and certified delivery are only applicable for information goods.

Currently published token currencies do not provide money atomicity. Simple anonymous rollback is not feasible. Without proper record-keeping every anonymous individual could claim to have sent money to another. Token currencies illustrate the possible trade-off between atomicity and anonymity suggested in the discussion of rollback (for examples see Rivest and Shamir, 1996; Yee, 1994; Camp, 1999; and Tygar, 1996).

Digicash (Chaum, 1985) is the canonical anonymous token currency. Yet Digicash has no atomicity as described above. In the later version of Digicash, (Chaum, Fiat & Naor, 1988), Chaum attempted to provide money atomicity, through encoding identity into each token to be spent. Encoding identity allows double-senders to be identified, thereby resolving the conflict between anonymity and accountability in the case of double spending. The addition of integrity provides sufficient information for dispute resolution in issues of payment, but not enough information to resolve disputes over goods delivery. This alteration again illustrates the perceived trade-off between atomicity and anonymity.

MicroMint (Rivest and Shamir, 1996), which uses hashing rather than public key operations to affordably generate large quantities of electronic cash, offers no money atomicity in its most simple form. In order to provide money atomicity there is an extended version of MicroMint where customer identity is included in every token. Thus the extension of MicroMint to provide money atomicity depends on the requirement that every consumer identify herself to the merchant to verify her right to spend a token. This is another illustration of the conflict between atomicity and anonymity: to make rollback possible the customer provides identity information.

An alternative approach to ensuring atomicity without limiting anonymity requires that every user have tamper proof, trusted hardware usually in the form of an active smart card (Brands, 1993; Brickell, Gemmell, and Kravitz, 1995; Burk & Pfitzmann, 1989, Boly et al. 1994). This requires that for commerce every user have a smart card or tamper-resistant hardware. It would also require that every commerce access location (such as consumers' home machines) have a smart card reader. The layer described here has four fundamental differences with the tamper-proof trusted party approach.

First, there is a fundamental philosophical difference between the distributed trust approach in this layer and the smart card or wallet approach. The trusted party for the layer is trusted only with the minimal information necessary to resolve disputes. There is no external observer who is completely trusted, and who could commit theft if it alone were subverted. The trust is not place completely in the customer's wallet. The approach in trusted hardware systems is essentially the trusted computing base approach where the tamper proof smart card is the completely trusted base. Trust and security are then built in an axiomatic manner on that base. The approach here is fundamentally a survivability approach. Assume that any single party can be dishonest or subversive. Assure that theft can not happen without subversion of multiple parties. This layer distributes trust for survivability rather than concentrating it in a trusted base.

Second, this layer differs from the tamper-proof hardware-based designs in implementation requirements. This layer does not require that every consumer or every merchant own completely trustworthy hardware. This system could be implemented on the Internet today

without the addition of the card and card-reading infrastructure. This hardware is required by trusted parties implemented in specialized, consumer-specific hardware assumed in the trusted computing base approach.

Third, this system could be added to many token currencies. The creation of the token or script may require multiple exchanges. Spending the money may require multiple exchanges; however the steps shown here can be wrapped before and after the money is exchanged. The token need not be a single token and the steps for obtaining a token need not be as described below. If there is a breakpoint, after which time the merchant has the token, the messages which lead to this point can be treated as a token. Furthermore, messages 1-4 in Figure 1 could consist of any number of messages as long as there is some digital information at the end of the exchange which represents and is bound to monetary value. Given that there are over one hundred proposed electronic commerce mechanisms (many of which are, of course, notational; see Mackie-Mason, 1998) the value of a general solution to the problem of anonymous token atomicity is clear.

Consider the solution of atomic exchanges without tamper-proof hardware, meaning that it is possible for any single player to be corrupt. An analysis of protocols for Internet commerce based on notational currency illustrates reliability is often simplified by creating a single ledger (Mackie-Mason, 1996). This single ledger in these cases is a networked entity controlled by the creator of currency who also serves to moderate disputes. The creation of a single ledger means that there is a concentration of information -- thus implying a threat to privacy and survivability.

Counter-intuitively, anonymity is provided through the creation of a publicly readable transaction log. The transaction log provides forced serialization. The transaction log receives and records messages and then publicized the recorded messages. The log also acts as a time-keeper, aborting transactions that expire

I first describe the necessary assumptions and parties to the protocol. In the following section I describe the protocol itself. I then show that the use of this layer provides anonymity. I discuss the trust assumptions, and possible results of worst-case failures.

### *1. Assumptions*

I assume all parties can perform basic cryptographic operations. All parties have well-known or verifiable public keys. Other assumptions are that the private elements of public keys are not disclosed; that tokens are verifiable; and that the bank can distinguish those tokens to be used with certified delivery. Distinguishing tokens to be used with certified delivery enables the bank to refuse their deposit without appropriate transactional records. Communication channels are assumed to be secure.

As is common in electronic commerce protocols (Sirbu, 1995; Mastercard, 1996; Mastercard, 1995; Visa, 1995) message signatures are computed on hash values of the plaintext and then appended to the plaintext to form a signed message.

There are four parties to each transaction: a bank, a customer, a merchant and a transaction log. In the following discussions the customer is assumed female; the merchant male; and the bank and log neuter. This allows me to use she, he and it without worrying that the reader may confuse the noun referenced by the pronoun.

The transaction log and bank are separated for clarity of exposition. There is no reason that the entities referred to here as transaction log and bank are not merely separate machines in the same facility or possibly separate severs on the same machine, analogous to the ledgers and vault of a depository institution. It is possible that separation between the log and the bank increases survivability.

Each transaction has an expiration time. This implies that the banks and the logs have synchronized clocks. The customers and merchants are then responsible for synchronizing with their log or bank of choice. Although this is a nontrivial problem, there are established approaches to the problem of synchronizing distributed secure transactions (Smith, 1992).

## *2. A Transaction: Preparation & Purchase*

Figure 1, below, illustrates the preparation steps required in an anonymous atomic purchase.

In the first step the customer obtains a token from her bank. In order to assure anonymity, the bank must be unable to match the token received in step three with the token sent in step two. There are multiple techniques for providing anonymous tokens, and many such techniques would be useful here. Notice I make no claims about introducing a new technique for generating anonymous tokens -- only removing the previously necessary trade-off between the use of anonymous currencies and atomicity. The flexibility of this atomicity-generating layer allows steps one and two to be as many steps as necessary -- as long as an anonymous token is the result. The only specification necessary is that the resulting token be anonymous and, as noted above, identifiable by the bank as for use only with certified delivery.

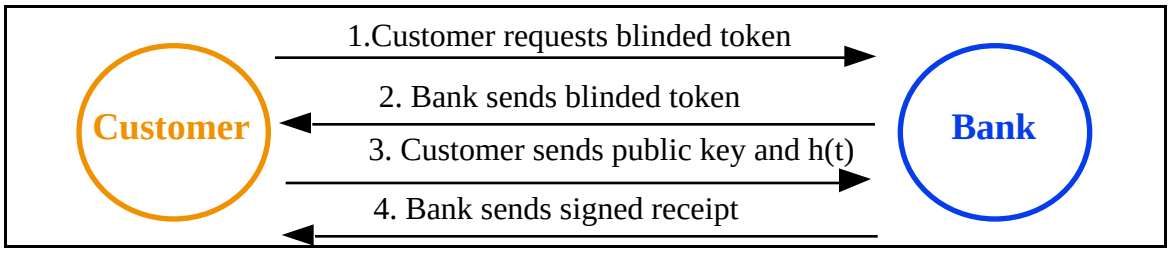


Figure 1: Preparation for an Anonymous Atomic Transaction

I assume the customer generates a public/private key pair to be used uniquely for the purpose of assuring certified delivery for this particular transaction. Call this pair q and Q to distinguish them from the pair c and C normally used by the customer and linked to her identity. The key pair (q, Q) is linked to the ownership of the token, not to the identity of the customer. Since the key pair is linked only to the token, then the key pair is as anonymous or pseudonymous as the token to which it is linked.

Thus the steps are:

| | |
|---|---|
| 3: C -> B | h(anonymous token),Q |
| 4: B -> C | $(h(\text{anonymous token}),Q)_b$ |

Key pairs q and Q can be generated off-line in advance of purchases. Notice that the key needs only to be strong enough to make obtaining the token cost prohibitive.

In the fourth step the bank returns a signed receipt for the token and the public key. This receipt enables the customer to demand that the subject token not be accepted for deposit unless accompanied by a receipt signed by the token specific key, q, indicating customer commitment to the transaction. The receipt is, in effect, a public key certificate, issued by the bank, binding the public key Q to the rightful holder of the token t.

The customer may now begin the transaction, as shown below in Figure 2. The steps are numbered consecutively, with the assumption that they follow the preparation steps above. Each step also has an associated label to simplify the following discussion of atomicity. The customer is assumed to have the merchant's certificate and key; and the merchant is assumed to have the certificate that binds the public key Q to the rightful holder of the token. As is common in electronic commerce protocols, the exchange of certificates is not shown.

P1: M -> C (contract, $E_k$(goods), TID, L, Q)$_m$

P2: C -> B (t, expiration, M, L, price, TID, Q)$_q$

P3: B -> M (expiration, M, L, price, TID)$_b$

P4: M -> L (expiration, k, TID)$_m$

P5: L ((expiration, k, TID)$_m$)$_l$

P6: M -> C (k)

Table 1 provides the definitions of the terms used above.

| **Field** | **Description** |
|---|---|
| contract | The contract includes a human-readable description of the goods being ordered, the price, and the hash of the encrypted goods |
| TID | Transaction identifier. Globally unique identifier. |
| L | The log chosen as the point of serialization for this transaction. |
| t | The token. |
| expiration | A time after which the transaction, if not completed, is to be aborted. |
| M | The merchant’s identity. Includes account number and merchant bank identifiers as necessary. |
| k | A secret key used only to encrypt the goods. |

Table 1: Fields in the Anonymous Certified Delivery Protocol

The messages in a transaction using anonymous certified delivery are shown in Figure 2.

To initiate the transaction the customer must contact the merchant. Such contact may allow the customer's identity to be inferred. However, I assume here that this contact allows the customer to remain anonymous, implying that the customer is using an anonymizer or anonymous remailer.

During purchase step one, limits on the purchase or special discounts can be negotiated, if the customer wants to include such an offer. For example, the repeated use of a token-specific key would create a pseudonym that would allow for subscriptions or repeat visit discounts. This would allow merchants to link sequential transactions; however, it would not allow merchants to link these purchases to a unique individual.

Before purchase step two, three elements of the transaction must be agreed upon: description of the item, price, and transaction identifier (TID). The transaction identifier must be globally unique. Some combination of a serial number and the merchant's identity can provide a unique TID.

Purchase step two is the provision of the contract. In preparation for purchase step two the merchant generates a secret key, k, and encrypts the requested merchandise with this key. Notice that k is a secret key used for symmetric encryption rather than the private part of a private/public key pair. The merchant then sends the encrypted goods. Because the merchant encrypts the information, there has been no exchange of valuables.

Note that without a secure channel any observer could steal the encrypted merchandise if the transaction is successful -- implying that this channel must be secured using the readily available public keys. The merchant may choose a common method, such as SSL, or more expensive proprietary options. The mechanism for establishing the secure channel does not affect the overall protocol -- as long as this technique does not expose customer identity information. The establishment of a secure channel is a source of cost, thus merchants should be able to choose their encryption methods based on their sensitivity to theft. After the end of this step the token has not been exposed, nor has the customer received a useful item.

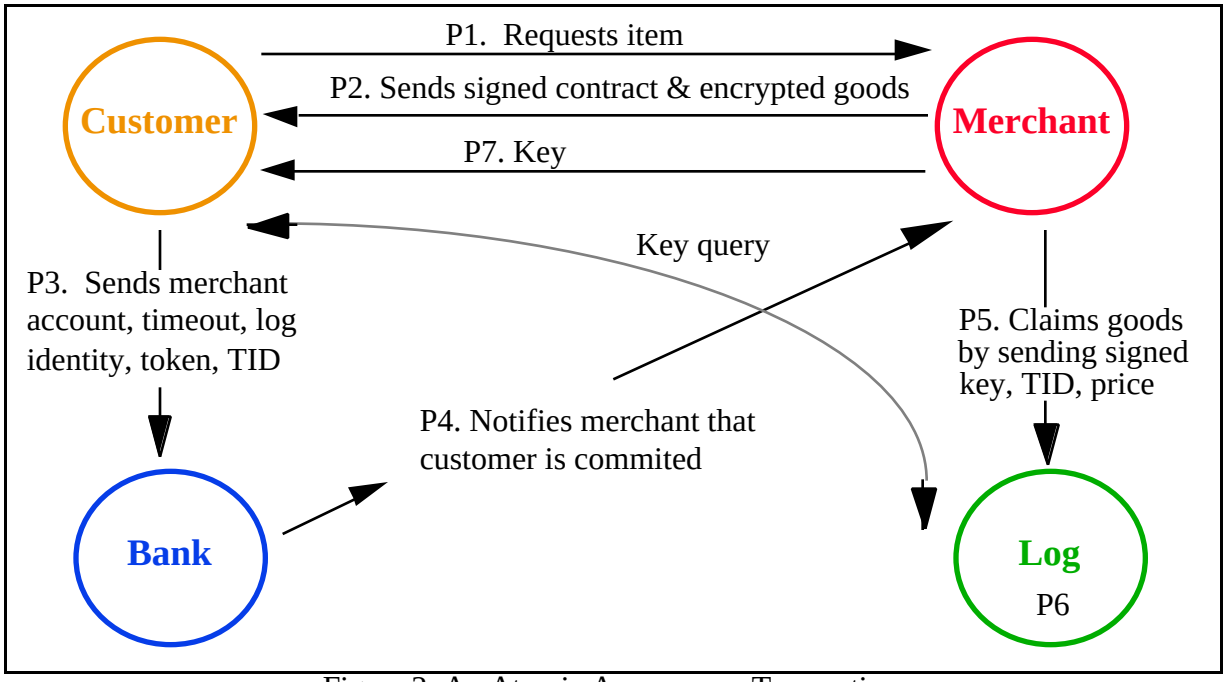


Figure 2: An Atomic Anonymous Transaction

In purchase step three the customer commits to the transaction by exposing the token. The customer sends the merchant's identifier, expiration, the amount of the transaction, the

transaction identifier, the log chosen for this transaction, and the token. The merchant's identifier may be any unique identifier. All this is signed with the token-specific key, q. This signature, together with the receipt ("certificate") from the bank proves that the customer is authorized to spend the token.

In purchase step four the bank notifies the merchant that the customer has committed. If the material signed by the bank is not what is claimed by the customer, the merchant can refuse to commit.

In purchase step five the merchant commits to the transaction by providing the key to the log. If the information is not correct, the customer may claim a refund. Thus it is not in the interest of the merchant to provide false information.

In purchase step six the log generates the global commitment by making the key publicly available. The customer may demand this receipt in order to obtain the key, if it is not promptly delivered.

In purchase step seven, the merchant completes the transaction by sending the key to the customer. Requiring that the merchant send the key to the customer reduces the load on the log, since most transactions will be without conflict or failure.

Note that if the transaction is aborted rather than completed the log would generate the following entry and send it to the bank:

P7: L -> B (M, TID, failed)]

This global abort message would replace the global commit message which entailed the publication of the key, k.

If the merchant commits with the log but fails to provide the key, the customer can query the log for the key. The query, shown as a dotted line in Figure 2, consists of two messages:

Q1: C -> L (TID)
Q2: L -> C (TID, k)]

This protocol provides goods atomicity and certified delivery. Consider the possible failures at each step. If purchase steps one or two, request and contract respectively, fail, then there is no transaction. If purchase steps three through six fail, then the merchant is never paid and the customer has only encrypted goods, which are of no use. If purchase steps three or four, customer commitment and notification respectively, fail then the merchant is never paid and the customer never obtains useful goods. If the merchant replaces the valid key with a bogus key or changes the expiration, the customer would have the necessary proof to obtain a refund or prevent the merchant from claiming payment (depending upon the time when the merchant attempted the replacement). If purchase step five, merchant commitment, fails, then the key is never delivered, the goods remain encrypted, and the merchant's deposit is never completed. The same is true if purchase step six fails; there is no key delivery and no deposit.

Purchase step seven is the global commitment. When purchase step seven is completed both key delivery and deposit are assured. Deposit is assured because the bank will credit the merchant in the absence of an abort message. Key delivery is assured because the customer can query the log directly and obtain the key if the merchant fails to deliver it directly. This assures goods atomicity.

Note that there is no need to encrypt purchase message seven: it is simply the key. The key need not be signed for there is already a verifiable copy of the key in the transaction log.

The protocol provides certified delivery since the description of merchandise and the item delivered can be verified from purchase step two. The contract includes a description of the item requested and a checksum of the item delivered, and is accompanied in purchase step two by a unique transaction identifier. All three fields are digitally signed by the merchant. If upon decryption the item delivered does not match the description, then the customer can obtain a refund.

The merchant could send a bogus key and claim the deposit. In this case the holder of the token-specific public key could present the actual merchandise as delivered to the merchant's or customer's bank, and show that the decrypted merchandise does not match the description. The bank would refund the customer's money by generating a token of equal value and transmitting it to the customer using the anonymous public key for secure delivery[2]. The value of this token would be debited from the merchant's account.

There is one trust assumption in this protocol: the merchant must trust the log to record received messages. If the log fails to record the merchant's commitment, but instead passes the key to the consumer, then the consumer will gain access to the goods while the merchant will not be able to demand payment. This trust requirement is the reason that the merchant selects the log in purchase step two.

The existence of a requirement for trust between the merchant and the transaction log is one argument for combining the log with the bank, since banks already play the role of trusted financial intermediary. In this case, purchase steps two, three and four (the contract, customer's commitment and notification, respectively) would not include the log selected, since the selection of the bank would imply a transaction log. Thus, while the merchant would not select the log, the merchant would instead extend trust to the customer's bank.

Conversely, the requirement for trust is also an argument for separating the log and the bank. Banks could then work as filters with a function analogous to the current role of acquiring banks in the credit card system, and refuse to work with logs which were the subject of many complaints or suspected of fraud.

## *3 Security*

There are four public key sets in this system: the transaction log's, the token-specific, the bank's and the merchant's.

[2]Recall the assumption of secure, anonymous channels. This assumes the bank does not get significant identity information during the transmission, for example, though knowledge of IP address.

The use of variable public keys limits the amount a customer can lose if a private key is lost. If a key is lost, both the customer and the thief could 'prove' ownership of a token. Thus refunds are not possible in the case of lost keys. Customer loss could be limited by requiring that keys be changed at a given monetary increment, or limiting the denomination of a single token if no keys are reused.

If an attacker obtains the token-specific private key then the attacker has obtained deposit authorization. However, without the token as well, the attacker cannot complete a transaction.

If an attacker obtains a merchant's key then the attacker can cause the merchant much difficulty in terms of resolving conflicts but cannot force refunds or obtain free merchandise. The attacker can construct a purchase order with a product description, price and faked record of a delivery. (This would require only that the attacker has a token and the corresponding keys. Presumably these are easy to obtain legitimately.) Then the attacker can create a false key, endorse the faked purchase order and send it as a commit. It would appear that the merchant unsuccessfully attempted to defraud the customer. The attacker can then request that the merchant prove the merchandise. The merchant could then detect the attack, decline the transaction, and refuse the transaction amount. There would be no deposit in the merchant's account, so the merchant would owe neither money nor the key.

If an attacker could obtain the keys of the transaction log, the attacker could masquerade as the log. In that case the attacker could steal merchandise as previously described.

## *4. Privacy*

The information available to each party in an anonymous certified delivery transaction is shown below in Table 2.

Note that information transmitted through communications channels, such as IP address or email from the browser, is not included. Note that to assure anonymity the customer must use an anonymizer or remailer.

| Information Party | Merchant | Customer | Date | Amount | Item |
|---|---|---|---|---|---|
| Merchant | Full | None | Full | Full | Full |
| Customer | Full | Full | Full | Full | Full |
| Transaction Log | Full | None | Full | None | None |
| Bank | Full | None | Full | Full | None |
| Electronic Observer | Full | None | Full | None | None |

Table 2: Information Available with Anonymous Certified Delivery

Note that a public key is not needed for each new token. In that case the customer's public key resembles a pseudonym. The probability of linking any pseudonym, in this case the public key, to real identity increases as the pseudonym is used in many locations, used frequently or over a long time period. The use of a merchant-specific pseudonym would

provide credentials whereby a repeat customer could obtain the appropriate discounts or a subscriber could obtain periodicals.

## *5. Conclusions*

Issues of reliability and consumer protection are addressed in this protocol. Before the publication of this work there was no protocol for generalizable anonymous rollback. The anonymous certified delivery layer fulfills the most stringent technical requirements for atomicity. Previously published anonymous currencies could not provide the highest level of atomicity without the addition of this layer or the use of specialized, trusted hardware.

Implementation issues which must be considered with this protocol are the storage of purchase-specific keys, and managing time-outs over long distances with uncertain transmission times. Simple issues, such as time zone differences, can be solved by using a standard time zone. However , tightly coordinated times may be critical for some information, such as stock information. Such products would require a well-recorded standard (e.g. Eastern Standard Time). Time sensitivity could be negotiated as part of the transaction and recorded in the product description.

Since tokens can be bundled, with a single key for many tokens, long-term deliveries or streaming deliveries can be accomplished whereby the keys are released at pre-set intervals. The value and timing of each partial delivery would necessarily be part of the negotiation and product description.

A significant issue in this layer is the addition of a large database and the requirement that it store keys and records for significant time periods. This memory storage is the largest drawback to this protocol. However, multiple logs could compete so that there is not a single point of storage. The selection of a log would again be subject to negotiation .

Notice that this protocol is optimized for on-line delivery. In the case of physical goods, the receipt and confirmation of order can be secured in this method but the delivery of the physical good cannot be verified.

## *References*


Brands, S., 1993, "Untraceable off-line cash in wallet with observers," *Advances in Cryptology - CRYPTO '93,* Springer-Verlag; Berlin, 302-318.

Branscomb, A.,1994, *Who Owns Information?*, Harper Collins Publishers Inc., New York, NY.

Boly, J. P., et al, 1994, The ESPRIT Project CAFE - High Security Digital Payment Systems, ESORICS, Spring, 217-230.

Brickell, E., Gemmell, P., and Kravitz D., 1995, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change," *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, California, 22-24 January, 457-466.

Burk, H. & Pfitzmann, A., 1989, Digital Payment Systems Enabling Security and Unobservability, *Computers & Security*, August 5, 399-416.

Camp, L. J. ,Sirbu M. and Tygar, J. D., 1995, "Token and notational money in electronic commerce," *Usenix Workshop on Electronic Commerce*, July, New York, NY.

Camp, L.J., Harkavy, M., Tygar, J.D. and Yee, B., 1996, "Anonymous atomic transactions," *2nd Usenix Workshop on Electronic Commerce Proceedings*, Usenix, Berkeley, CA, 123-134.

Camp, L.J., 1999 *Internet Commerce: Fundamentals and Systems*, MIT press, Cambridge MA, expected January 1999.

Chaum, D., 1985, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, Vol. 28, 1030-1044, October.

Chaum, D., 1988, "Untraceable Electronic Cash", *Crypto 88*, Springer-Verlag, Berlin, 319-3127.

Chaum, D., 1989, "On-line cash checks," *Advances in Cryptology - EUROCRYPT '89*, 288-293.

Chaum, D., 1992, "Achieving electronic privacy," *Scientific American*, Vol. 267, 76-81.

Compaine B. J., 1988, *Issues in New Information Technology*, Ablex Publishing; Norwood, NJ

Financial Service Technology Consortium, 1995, *Electronic Payments Infrastructure: Design Considerations*, http://www.llnl.gov/fstc/projects/commerce/public/epaydes.htm, November.

Gray, J. and Reuter, A., 1993, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann Publishers; San Francisco, CA.

Low, S., Maxemchuk, N.F. and Paul, S., 1993, "Anonymous credit cards," *First ACM Conference on Computer and Communications Security*, November.

Mastercard, 1995, *Secure Electronic Payment Protocol Specification Draft Version 1.1*, http://www.mastercard.com/Sepp/sepptoc.htm, November, Part 2.

Mastercard, 1996, *Secure Electronic Transaction Technology, Draft.*, http://www.mastercard.com/SETT.

Mackie-Mason & White, K., 1996, "An Axiomatic Approach to Evaluating and Selecting Digital Payment Mechanisms" *Telecommunications Policy Research Conference*, Solomons Island, MA, Sept.

Mackie-Mason, J., 1998, "Commercial Protocols, Digital Currencies and Providers", *Telecommunications Information Resources on the Internet*, http://www.spp.umich.edu/telecom/net-commerce.html, August.

Medvinski, G. and Neuman, B.C., 1993, "NetCash: A design for practical electronic currency on the Internet," *Proceedings of the First ACM Conference on Computing and Communications Security*, November.

National Research Council, 1996, *Cryptography's Role in Securing the Information Society*, National Academy Press, Washington, DC.

Neumann, P. G., 1993, "Risks of surveillance", *Communications of the ACM*, v36, n8, p122(1) August.

Okamoto, T. and Ohta, K., 1991, "Universal electronic cash," *Advances in Cryptology-CRYPTO '91*, Springer-Verlag; Berlin, 324-336.

Pool, I., 1983, *Technologies of Freedom*, Harvard University Press, Cambridge MA.

Sirbu, M., and Tygar, J. D., 1995, "NetBill: an Internet commerce system optimized for network delivered services," *IEEE ComCon*, San Francisco, CA. March 6

Simon, 1996, "Anonymous Communication and Anonymous Cash", *Crypto '96*, Springer -Verlag, Berlin, 61-73.

Smith, S., 1992, *A Theory of Distributed Time*, Ph.D. dissertation, Carnegie Mellon University. Available as Carnegie Mellon University technical report CMU-CS-92-231.

Tygar, J. D., 1996, "Atomicity and electronic commerce," *Proceedings of 1996 Symposium of Principles of Distributed Computing*, ACM Press, Philadelphia, PA.

Visa, 1995, *Secure Transaction Technology Specifications Version 1.1*, http://www.visa.com/visa-stt/index.html, November.

Yee, B., 1994, *Using Secure Co-processors*, Ph.D. dissertation, Carnegie Mellon University. Available as Carnegie Mellon University technical report CMU-CS-94-149