The following paper was originally published in the
Proceedings of the Sixth USENIX UNIX Security Symposium
San Jose, California, July 1996.

Building Blocks for Atomicity in Electronic Commerce

Jiawen Su and J. D. Tygar
Department of Computer Science
Carnegie Mellon University

Building Blocks for Atomicity in Electronic Commerce

Jiawen Su J. D. Tygar
sjw@cs.cmu.edu tygar@cs.cmu.edu
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Atomicity is clearly a central problem for electronic commerce protocols — we can not tolerate electronic commerce systems where money is arbitrarily created or destroyed. Moreover, these atomicity properties should be retained in the event of component failures in distributed systems. In this paper, we enumerate several classes of atomic protocols. We then give two fundamental building blocks for building atomic electronic commerce protocols: encryption-based atomicity and authority-based atomicity. We then illustrate these building blocks by considering variations of payment-server based protocols that use these different building blocks. The results give a contrast to the class of protocols that we have previously examined in our work with NetBill.

1 Introduction

We study electronic commerce protocols that allow users or agents to transfer funds on a network. In these electronic commerce protocols, a customer *C* makes a purchase by transferring funds to a merchant *M*.

What if communications or some other subsystem fails during the transfer? It is reasonable to require that the resulting electronic commerce system impose an *atomicity* condition: either the funds transfer should complete (*M* has the money; *C* does not) or it should not occur at all (*C* has the money; *M* does not). It is unacceptable if the protocol leaves us in a partial or ambiguous state: for example, if neither *M* nor *C* has the money, then our system has destroyed value. Similarly, if both *M* and *C* have the money, or even if both believe they have the money, chaos or counterfeiting can result.

In [18], the second author argued that this is an example of *money atomicity*: money should not be created nor destroyed by an electronic commerce protocol. The second author then defined two stronger classes of atomicity: *goods atomicity* (transfer of goods occurs if and only if the money transfer happens and certified delivery (all parties can prove the exact content of goods transferred after the fact with a money atomic protocol.) [18] has an extensive discussion of these different types of atomicity, and shows examples of protocols meeting and not meeting these atomicity concerns.

A close examination of published electronic commerce protocols quickly reveals a number that do not satisfy atomicity constraints. These protocols appear to use very strong cryptographic methods, but they fail to consider the possibility of communication failure.

Many of the "electronic currency" protocols (including [3, 8, 13]) exemplify the difficulties of non-atomic systems. Consider the policy for handling communications failures. Consider the policy for handling communications failure as a customer is transferring funds to a merchant: the merchant may or may not have received the funds, but the customer does not know the status of the funds transfer. If the policy is to deny the customer access to the funds that have started to transfer (but may have not been received by the merchant), then this will sometimes result in lost money. On the other hand, if the policy is to allow the customer access to the funds, then the customer and the merchant may both believe that they have access to the same funds. This can result in double-spending. In systems such as these, that will be indicated as a potential fraud, and will result in the customer being charged with counterfeiting — even though the customer may have acted in good faith according to the policy of the system.

Atomicity is not a new concern; it has been considered for years in the transaction processing environment. There

We gratefully acknowledge the support of the National Science Foundation (under cooperative agreement IRI-9411299), the US Postal Service, and Visa International for the support of this research. This paper is solely the opinion of the authors and does not necessarily reflect the opinion of the funding agencies.

are some excellent references on general techniques for achieve atomicity in electronic commerce protocols ([12, 11] are outstanding surveys of the field.) This paper is a first attempt to enumerate some possible building blocks for generating atomic protocols in electronic commerce.

Our concern with these topics is not purely academic. At CMU, we are currently building a system called NetBill. NetBill has a protocol (see [10, 16]) optimized for highly atomic electronic transactions. This paper considers several variations on those protocols which have radically different performance considerations.

We limit our examples in this paper to payment servers; but the second author has recently worked with others to develop a number of atomic versions of electronic cash [4, 5] and the Mastercard/VIS SET Secure Electronic Transaction [15] secure protocol for transmitting credit cards over the network [4, 6].

2 Payment models

Most proposed electronic payment systems fall into three broad categories: electronic currency, debit/credit instrument and secure credit card.

In an electronic currency system, money is denoted as a *token* which is a sequence of digital bits. Most electronic currency systems strive to model traditional currency: they attempt to provide anonymous and unforgeable tokens. However, it is especially easy to copy a byte string, so these protocols use special techniques to prevent double-spending tokens because almost no other things are as easily copyable as digital bits. Chaum pioneered electronic currency protocols [8]; there is a very wide literature on systems derived from Chaum's framework. In Chaum's system, tokens are withdrawn from an issuing bank. To prevent double-spending a token, each token includes hidden account information. If a customer double spends a token, their is sufficient information to uniquely identify the customer.

In debit/credit instrument systems, customers and merchants register accounts with payment servers. When a customer buys goods from a merchant, he or she signs a payment instrument directed to his payment server. Some examples of debit/credit instrument systems are NetBill [16] and NetCheque [14].

Secure credit card systems base the payment on traditional credit cards [2, 15]. A customer's credit card number is securely transferred to a merchant in case of payment. Then the merchant processes the payment through a traditional credit card transaction. These protocols often add a twist: they attempt to protect the merchant from obtaining the customer's credit card number. Instead, the transaction is processed only by the acquiring bank of the merchant; preventing a wide class of credit card fraud: merchant fraud where the merchant misuses the customer's credit card number.

3 Security considerations

There are many cryptographic and security considerations that must be satisfied to make a safe, correct, and secure electronic commerce protocol. To make our discussion more focused, we do not consider these issues in the protocols discussed below. In particular **the protocols below in sections 4 and 5 should not be considered to be secure for actual use.** In fact, integrating security concerns with atomicity is a non-trivial problem and is illustrated in detail in [10, 18].

In particular, here is a (not exhaustive) list of some considerations that apply (some excellent references on further essential elements for secure protocols is contained in [1, 17]):

- **Privacy:** Communications must be protected from eavesdropping from an unintended party. A private communications channel, or a virtual private communications channel, should generally be used in the protocol. There are several ways that this could be achieved: a physically secure line might be used, a shared symmetric key cryptosystem could be exploited, messages could be encrypted in the public key of the intended recipient, etc. These various techniques lead to radically different performance considerations: for example, the costs of key exchange and encryption can change the cost equation making some protocols feasible and other infeasible.

- **Nonrepudiation:** For showing some aspects of certified delivery, it may be necessary to prove to a third party that a certain communication was really received from another party. For example, it is certainly useful for a receipt to be nonrepudiable, so that the holder of a receipt can use it as proof in case of a dispute over a transaction. One way to achieve this is through the use of digital signatures; however these can add substantially to the running time of the protocol.

- **Protection against replay:** One of the most common attacks on protocols involves replaying certain messages. Anyone who has been double-billed on his credit card for a single purchase is familiar with the consequences of this attack. To address this, we use the techniques of *idempotence* (discussed in section 4 and *nonces*. Nonces are unique identifiers that associate a set of messages uniquely with a single transaction. Unfortunately, the generation of nonces and the protection against their malicious use is nontrivial. For example, if an opponent can guess a valid nonce value, he may be able to insert the message in a way that confuses the analysis.

By putting the issues of privacy, nonrepudiation, and protection against to the side for a moment, we can focus on the vital issues of atomicity. We do not consider here the full integration of these elements.

4 Atomic building blocks

We consider protocols that preserve atomicity. These protocols must maintain an all-or-nothing property for transfer of funds (and digital goods) — either the transaction completes or the effect is restored as if the transaction did not occur. We give two building blocks: encryption-based atomicity and authority-based atomicity.

Now, in the building blocks below, we have several parties defined: *S*: A sender who attempts to send some data *m* (such as digital goods, a value token, a credit card number, etc) to a recipient; *R*: A receiver who receives the data; and *A*: A trusted third party recipient of the data. We can assume that *S* and *R* may try to deviate from the protocol, but we assume that *A* will always comply with the protocol requirements. (In fact, in more sophisticated analyses, such as [10, 18], we can weaken the assumption and consider the possibility of a corrupt third party; but here we do not consider that case.)

We use the notation $\{m\}_k$ to denote that message *m* is encrypted under key *k* (for example, under a symmetric key cryptosystem such as DES.) We use TID to denote a unique and unguessable (see section 3) transaction ID. And we use the notation sig *x* to denote *x* in a digitally signed format.

Finally, we note that all of our messages are designed to be *idempotent*, they can be repeated more than once with no additional effect. For example, a message that says, "decrease my account by one unit", is not idempotent since its repetition will result in the account being decremented by more than just one unit. However, a message that is uniquely identified by a transaction ID can be made idempotent since the receiver of the message can ignore repeated copies of the same message tagged with the same transaction ID. (See [11] for more on the subject of idempotence.)

4.1 Encryption-based atomicity

Our first building block prevents data (such as a token or digital goods) from being read by the recipient before a key can be sent to a trusted third party. This third party can then arbitrate to ensure atomic delivery.[1]

1.	S $\rightarrow$ R:	$\{m\}_k$, TID
2.	R $\rightarrow$ S:	sig $\{m\}_k$, TID
3.	S $\rightarrow$ A:	*k*, TID
4.	S $\rightarrow$ R:	*k*, TID

Note that if communications fail before or during step 3, the transaction will not take place. On the other hand, if communications fail after step 3, *R* may fail to receive the decryption key *k*. However, if we trust the authority *A* to freely give *k* on request, requests it, then *S* and *R* can always complete the transaction even if they (or their communication links) fail after step 3.

Message 2 provides *S* with a receipt of the fact that *R* received message 1 intact. Later, if their is a dispute over the contents of the message, this signature, together with a signed copy of *k* obtained from *A*, can be used to prove the contents of *m* to any third party (such as a digital judge!)

4.2 Authority-based atomicity

A different approach to atomicity can be to have the trusted authority *A* not only hold cryptographic keys but also the messages themselves in escrow. This way, *A* becomes a trusted communication agent. This can result in fairly expensive storage costs for *A*. Here is a simple authority-based atomic transfer of message:

1.	S $\rightarrow$ A:	*m*, R, TID
2.	A $\rightarrow$ R:	"message available", TID
3.	R $\rightarrow$ A:	"send me the message", TID
4.	A $\rightarrow$ R:	*m*, TID

Here, *A* will store message *m* and continue to transmit message 2 until *R* finally picks up his message.

5 Atomic protocols

In this section, we analyze various electronic commerce models and apply our building blocks to achieve atomicity for debit/credit instruments.

[1]The careful reader will note that this protocol and the ones that follow do not provide privacy, nonrepudiation, or protection against replay attacks. We have distilled these elements out of this set of building blocks. In particular, because this protocol and the ones that follow are not designed to be secure, we do not recommend that they be used without modification. See section 3 for more discussion of these issues.

(We discuss one of the three major electronic commerce models below. What about the other two models electronic currency and secure credit card information? Electronic currency presents special challenges for atomicity because it required us to provide anonymous and atomic transactions. The integration of these two elements is extremely complicated; recently Jean Camp, Mike Harkavy, Bennet Yee, and the second author have developed a family of protocols to achieve this, and a description of that protocol is available in [4, 5]. For secure credit card transactions, Jean Camp, Marvin Sirbu, and the second author [4, 6] have recently shown how to integrate certified delivery with the new SET protocol [15].)

5.1 Debit/Credit Instrument

In a debit/credit model, both customers C and merchants M register accounts with a payment server P. (P is assumed to act as a trusted third-party authority, and to abide by the conditions discussed above.) The payment server is assumed to use a locally atomic database to register transfer of funds among accounts, so money atomicity follows trivially. However, the problem of goods atomicity (money exchanged for goods) and certified delivery (proof of the contents of items delivered) is non-trivial.

We assume the case that the items being purchased are digital goods to illustrate the atomicity concerns.

In [10, 18] we consider this problem at length. Here, we merely summarize a few alternatives.

Case 1: Without atomicity

1.	C $\rightarrow$ M:	price-inquiry, TID
2.	M $\rightarrow$ C:	price, TID
3.	C $\rightarrow$ M:	instrument, TID
4.	M $\rightarrow$ P:	instrument, TID
5.	P $\rightarrow$ M:	status of payment, TID
6.	M $\rightarrow$ C:	item, TID

Comments. Here the item is the digital goods to be sold. The example above is clearly not goods atomic, since the merchant could decline to send message 6. Note that the instrument in this case could be an invoice, and that in steps 3 and 4, both M and C have an opportunity to agree on the price of the item. In most cases, the instrument should be signed by C and include the TID and an item description to ensure that the instrument originated in a nonrepudiable fashion from C and is not being replayed. (Note that this is not a secure protocol; see section 3 or [10, 18] for more details.) They payment is handled online by the payment server, which then must report the success or failure of the transfer to the customer.

Case 2: Encryption-based atomicity

1.	C $\rightarrow$ M:	price-inquiry, TID
2.	M $\rightarrow$ C:	price, $\{item\}_k$, TID
3.	C $\rightarrow$ M:	instrument, sig $\{item\}_k$, TID
4.	M $\rightarrow$ P:	instrument, k, sig $\{item\}_k$, TID
5.	P $\rightarrow$ M:	status of payment, TID
6.	M $\rightarrow$ C:	(if successful) k, TID

Comments. Here, we have the item send encrypted under k in step 2. C then prepares a signed copy of the encrypted item, which in step 4 should be counter-signed by the merchant, so both parties agree on the contents of the item. The instrument should contain information such as the TID and a description of the item for maximum protection. When the counter-signed item, together with the decryption key k and the instrument, are registered at P, it enables P to give proof of certified delivery — contents are registered at P. If steps 5 or 6 fail, then M and C can directly query P to discover the status of the transaction.

Note that instead of a full signature in steps 3 and 4, it is desirable to take signatures of cryptographic checksums (such as MD5) of the data. This not only reduces storage costs at P, but it provides additional privacy against P reading the contents of the item sent from M to C.

Finally, note that this protocol is only an example of the application of two-phase commitment [11, 12] to the problem of electronic commerce. Two-phase commitment is a well known technique for achieving atomicity.

Case 3: Authority-based atomicity

1.	C $\rightarrow$ M:	price-inquiry, TID
2.	M $\rightarrow$ C:	price, TID
3.	C $\rightarrow$ M:	instrument, TID
4.	M $\rightarrow$ P:	instrument, item, TID
5.	P $\rightarrow$ M:	status of payment, TID
6.	P $\rightarrow$ C:	(if successful) item, TID

Comments. Authority-based atomicity requires the payment server to retain much more information than it would under encryption-based atomicity. In particular, it must retain the full contents of the item indefinitely. Certified delivery is satisfied by the escrowed copy of the item stored at the payment server. Since messages 5 and 6 may not be received, the payment server must respond to queries from either M or C on the status of the payment.

In addition to the storage required in this protocol, there is also a very serious difficulty in that the payment server can easily read the contents of the item. Now, it may be possible to solve this problem through the use of public key cryptography: M could encrypt the item in C's public key before transmitting it to P. However, to verify the contents of the message, it would be necessary for C to disclose his private key to a third party (such as a digital judge.) This would have the unfortunately drawback of disclosing all of C's items received encrypted under his public key to the third party. A better solution in this case would use a one-time only public-private key pair. Safely handling the key management information in this case would add substantial complexity to the protocol.

6 Open Problems

This work leaves a number of important issues open including:

- lower performance bounds on atomic protocols;
- the performance impact of integrating privacy, non-repudiation, and protection against replay attack into atomic protocols;
- techniques to prove atomicity (recent unpublished work of the second author with Nevin Heintze, Jeannette Wing, and H. C. Wong gives some preliminary indications that model checking may be an especially fruitful technique to attack the problem of checking atomicity of protocols); and
- removing blocking (our protocols are all derived from two-phase commitment, and thus share the blocking characteristics of that protocol — there are a variety of so-called "non-blocking" commitment protocols that have been studied and some of these may yield important results in the electronic commmerce sphere [11, 12].)

For a more comprehensive list of open problems, see [18].

7 Acknowledgements

Discussions with the following people have illuminated our understanding of electronic commerce atomicity concerns: Jean Camp, Ben Cox, Nevin Heintze, Mike Harkavy, Cliff Neuman, Marvin Sirbu, and Bennet Yee.

References

[1] M. Abadi and R.M. Needham. Prudent Engineering Practice for Cryptographic Protocols. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, May 1994. pages 122-136.

[2] M. Bellare *et al*. iKP — A Family of Secure Electronic Payment Protocols. In *Proceedings of the First USENIX Workshop on Electronic Commerce*, July 1995.

[3] S. Brands. Untraceable Off-line Cash in Wallet with Observers. *Advances in Cryptology-Crypto'93*, 1994.

[4] J. Camp. *Privacy and Electronic Commerce.* PhD thesis, Engineering and Public Policy Department, Carnegie Mellon University. To appear.

[5] J. Camp, M. Harkavy, J. D. Tygar, and B. Yee. *Atomic Electronic Cash.* To appear.

[6] J. Camp, M. Sirbu, J. D. Tygar. *Certified Delivery with SET.* To appear.

[7] D. Chaum. Security without Identification: Transaction Systems to Make Big Brothers Obsolete. *Communications of ACM*, 28(10), 1985. pages 1030-1044.

[8] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Advances in Cryptology-Crypto'88*, 1989. pages 644-654

[9] D. Chaum and T.P. Pedersen. Wallet Databases with Observers. *Advances in Cryptology-Crypto'92*, 1993.

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

[11] J. Gray and A. Reuter. *Transaction Processing: Techniques and Concepts.* Morgan Kaufmann, San Mateo, CA, 1994.

[12] N. Lynch, M. Merrit, W. Weihl, A. Fekete. *Atomic Transactions.* Morgan Kaufmann, San Mateo, CA 1994.

[13] T. Okamoto and K. Ohta. Universal Electronic Cash. *Advances in Cryptology-Crypto'91*, 1992.

[14] B. Neuman, G. Medvinsky. Requirements for Network Payment: The NetCheque Perspective. *Proceedings of IEEE Compcom'95*, March 1995.

[15] Secure Electronic Transactions Specification. This information is available from the WWW pages of both Mastercard (www.mastercard.com) and Visa (www.visa.com) home pages; for example, see http://www.mastercard.com/set/set.htm.

[16] M. Sirbu and J. D. Tygar. NetBill: An Internet Commerce System Optimized for Network Delivered Services, *IEEE Personal Communications*, August 1995. pages 6-11.

[17] P. Syverson. Limitations on Design Principles for Public Key Protocols. In *Proceedings of 1996 IEEE Symposium on Security and Privacy*, pages 62-72.

[18] J. D. Tygar. Atomicity in Electronic Commerce, *ACM/IEEE Conference on Principles of Distributed Computation*, 1995, pages 8-26.