Anonymous Credit Cards

Steven Low
Room 2C-219
AT&T Bell Laboratories,
600 Mountain Avenue,
Murray Hill, NJ 07974
Phone: 908-582-5313
E-mail: slow@allegra.att.com

Nicholas F. Maxemchuk
Room 2C-225
AT&T Bell Laboratories,
600 Mountain Avenue,
Murray Hill, NJ 07974
Phone: 908-582-6240
E-mail: nfm@allegra.att.com

Sanjoy Paul
Room 4G-532
AT&T Bell Laboratories,
101 Crawfords Corner Road
Holmdel, NJ 07733
Phone: 908-949-1637
E-mail: sanjoy@boole.att.com

Submitted to 1994 IEEE Symposium on Research in Security and Privacy, Oakland, California

Anonymous Credit Cards

Abstract

This paper describes a novel technique for generating an *anonymous credit card* which combines the privacy of cash transactions with the security, record-keeping and charging mechanisms of credit cards. Using this scheme, an individual can make purchases without revealing his identity while a shop can sell items to an *unknown* individual without the fear of being cheated.

1 Introduction

Before the introduction of credit cards, transactions were made using cash. However, people are reluctant to carry a large amount of cash for fear of having it stolen. Lost cash cannot be reclaimed. More importantly, the buying power of a customer is limited by the amount of cash he carries with him. Banks realized the potential of introducing credit cards which virtually eliminate all these problems. Although the credit cards eliminated several problems, they introduced some new problems which did not exist before. For example, a store knows the identity of an individual and the purchases he makes. The credit card company knows the stores a customer shops at. If the store sends a list of purchases made by an individual to the credit card company, the latter can easily compile a profile of the individual's spending habits and invade his privacy. This information can be sold to advertisers, to help them focus their advertising. At some point, the ability to collect information compromises an individuals' right and willingness to deviate from the norm. Cash solves the privacy problem. The dollar bill that you spend doesn't have your name on it, and it is extremely difficult, although not impossible, to track the present owner of a dollar bill. However, cash has a security problem as mentioned earlier. In this paper, a mechanism is described that has the desirable characteristics of cash and credit cards.

One simple solution to the privacy problem is to pass legislation to make it illegal to track spending habits. Another simple solution is to create a *trusted* entity that promises to keep the credit card holders identity secret. However, if we do not want to impose these restrictions, the problem becomes interesting and non-trivial.

David Chaum in [DC85] proposed an interesting cryptographic protocol which could be used by a customer in an automatic transaction processing without letting the bank relate to what merchandise he buys. His protocol uses public key encryption mechanism in which the bank has a pair of keys p and s (where $p.s$=1 in appropriate modular arithmetic) for each type of dollar bill. The key p is publicly known and the secret key s is known only to the bank. If the customer wants a $1 bill, he chooses a random note number n and a random number r and sends $[1]=n.r^p$ to the bank, where p is the public key of the bank corresponding to $1. The bank, after receiving [1], sends $[2]=[1]^s=n^s.r$ to the customer, where s is the secret key corresponding to $1. The customer then removes r and sends $[3]=n^s$ to the store he shops at. The store verifies the signed $1 bill, gives the merchandise to the customer and claims $1 from the bank later on. Note that the bank and store cannot collude to link a customer with the merchandise because the customer removes the random number r before handing over the signed bill to the store. However, there are several disadvantages of this scheme: (1) the customer handles *electronic cash* and hence it may be possible for him to duplicate the bits representing cash, (2) it may be possible to generate *new* electronic cash by combining *old* ones [LMP93]. This is a major shortcoming of the above technique.

This paper describes an alternative approach which avoids the drawbacks of [DC85]. In contrast to [DC85], all fund transfers in our system take place electronically between banks. The bits representing money are never in the possession of individuals. Thus we reduce the problem of *duplication*. We simulate *electronic credit* as opposed to *electronic cash*.

The rest of the paper is organized as follows. Section 2 describes the basic idea of the protocol, the cryptographic tools we use, the information stored at each site and the formats of messages exchanged between the players. Section 3 contains the protocol specification in terms of communicating extended finite state machines. Section 4 analyses our protocol with respect to collusion and the *leakage* of information, followed by a conclusion and appendix.

2 Our Protocol

2.1 The basic idea

The operation of the protocol requires a customer C to have two accounts in different banks. The first bank which we call B_c knows the person's identity, and is willing to extend him credit. In the second bank which we call B_p (sometimes referred to as a Swiss bank), the account is under a pseudonym P - a numbered Swiss account. C's account in B_c will be referred to as A_c=(B_c,C) and his account in B_p will be referred to as A_p=(B_p,P).

A customer C can withdraw funds from B_p by proving his identity; however, B_p does not know the person, and is not willing to extend credit. B_c knows C and it puts credits in the account of P in B_p without knowing P. C spends these credits. At the end of the month, B_p gives B_c a bill, which it trusts B_c to pay. B_c presents the bill to C which it trusts the customer to pay. When the customer pays his bill, B_c places additional credits in P's account in B_p.

All the interactions in the system are carried out through an intermediary which we call the communications exchange cx. For example, B_c interacts with B_p through cx. The customer stores his Swiss account number A_p in an encrypted form in B_c such that B_c can use it to transfer funds to A_p without knowing either the bank B_p or the specific account A_p. Similarly, C stores his account number A_c in an encrypted form in B_p such that B_p can communicate with B_c and send a bill for A_c without knowing either B_c or A_c.

During transfer of funds, B_c communicates with cx which can determine the destination bank B_p, but not the customer P in that bank. cx informs B_p which is able to determine the customer C. In effect, cx trusts the source bank B_c to provide the funds it is transferring, and the destination bank B_p trusts cx. When funds are transferred, separate cryptographic procedures are used to sign the transfers and to prevent duplication attacks on the system. cx knows the source and destination bank for all fund transfers, and is responsible for settling the accounts. The signed messages provide an audit trail if there are discrepancies.

At the point of sale, a customer uses a card for his account in B_p. He contacts B_p via the communications exchange and provides it with an encrypted account number A_s of the store at possibly a different bank B_s and the amount of sale. B_p verifies the customer's identity and if he has sufficient credit, it transfers funds to A_s via cx without knowing either the bank B_s or the customer S. B_s

then notifies the store S that it has the funds, and the customer is given the merchandise.

We refer to C, B_c, P, B_p, S, B_s and cx as the players involved in the protocol.

2.2 Cryptographic Tools and Message Formats

Public key cryptography [DH76][R78] has been used as the basis of encryption and digital signatures [DH76][RSA78] in our protocol. Each player in our system has a public key and a secret key. The public key is known to all the other players but the secret key is known only to its owner. We use the following notation in our paper. $EP_A(m)$ is a message m encrypted with the public key of player A. $sgn_A(m)$ is a message m signed by player A. $sgn_A(m)$ denotes a message m sent in the clear, followed by a function of the message encrypted with the secret key of the player A. $EX_c(m)$ is a message m encrypted with a secret, unshared key (which is different from the secret key used by C to sign a message) of C.

The protocol messages are sent from a source to a destination. A source, or a destination, is one of C, P, S, B_c, B_p, B_s. The message formats and some important notations are summarized in table 1.

The communication from a source src to a destination dst is accomplished by a message pair $m_1(src, dst)$ and $m_2(src, dst)$. $m_1(src, dst)$ is sent by src to cx and $m_2(src, dst)$ is sent by cx to dst. The specific formats of the messages is described in table 1. Message $m_1(src, dst)$ uses format $F(msg, dst)$ or $F_{src}(msg, dst)$ and message $m_2(src, dst)$ uses format $X(msg, dst)$ or $X_{src}(msg, dst)$.

When a message arrives at a destination, the destination needs to verify if the message is authentic. Sometimes the communication exchange needs to check the authenticity of a message. We use the predicate $C(src, X)$, defined in table 1, to indicate the results of this test.

2.3 Information Stored at Sites

In order that the transactions occur anonymously, the customer stores encrypted information at each site. The information is encrypted using the public keys of cx and the destination in the format $D^e(A_x)$ defined in table 1.
C stores $D^e(A_p)$ in A_c, to allow B_c to transfer information to A_p without knowing B_p or P.
P stores $D^e(A_c)$ in A_p, to allow B_p to transfer information to A_c without knowing B_c or C.
In addition, P stores a set of questions Q_i and answers A_i at B_p as $CR(C) = EX_c(Q_i), EX_c(A_i)$. The questions are personal, little known, information that the customer would like asked of himself to prove who he is. This information provides additional security over the conventional *PIN* checking mechanism of ATM transactions. During a transaction at a store, B_p picks an encrypted question at random from the set, and sends it to the point of sale. The customer's smart card decrypts the question and presents it to the customer. The customer answers the question, the card encrypts the answer, and sends it to the bank for comparison with the encrypted, stored answer.

3 Protocol Specification

In this section, we give an abstract specification of our protocol in terms of communicating extended finite state machines. For notations, see the appendix. We assume that messages are transmitted between sites with a reliable transport mechanism, such as the TCP/IP [C91]. Therefore, we do not need to explicitly recover lost messages as part of our protocol.

Notation	*Definition*	*Explanation*
$D^e(A_x)$	$EP_{cx}[B_x, rn_1, EP_{B_x}(X, rn_2)]$	Encrypted destination for A_x (X=C, P or S).By sending $D^e(A_x)$ to cx, a source can send a message to account A_x, without knowing B_x or X. cx decrypts $D^e(A_x)$ to determine B_x. cx forwards the message to B_x without learning X. B_x decrypts $EP_{B_x}(X, rn_2)$, to determine X. rn_1 and rn_2 are random numbers.
$F(msg, dst)$	$EP_{cx}(msg, dst)$	Message msg to be forwarded to destination dst by cx, encrypted with the public key of cx. dst may itself be encrypted with the public key of cx.
$F_{src}(msg, dst)$	$EP_{cx}[src, sgn_{src}(msg, t(src, dst)), dst]$	Message msg timed-stamped and signed by the source src to be forwarded to dst by cx, encrypted with the public key of cx. The time-stamp $t(src, dst)$ increases for each successive message between src and dst and prevents replay attacks.
$X(msg, dst)$	$EP_{dst}(msg)$	This is a message msg sent by cx to dst, encrypted with the public key of dst.
$X_{src}(msg, dst)$	$EP_{dst}[src, sgn_{src}(msg, t(src, dst))]$	Message msg time-stamped and signed by the source src sent by cx to dst, encrypted with the public key of dst.
$C(src, X)$	$C(src, X)$ is true if sgn_{src} is correct, $timestamp$ is ok and X is true. $X = T_{src}$, pin_{ok} or ac_{ok}. T_{src} is true if src is trusted. pin_{ok} is true if pin is correct. ac_{ok} is true if $answer$ is correct and $credit$ is available. If only sgn_{src} and $timestamp$ need to be checked, $C(src)$ is used.	

Table 1: Message Formats

3.1 Protocol Component Machines

As described earlier, our protocol involves the entities: cx, B_c, B_p and B_s in addition to the customer C, and the shop S. P specifies the customer at the point of sale.

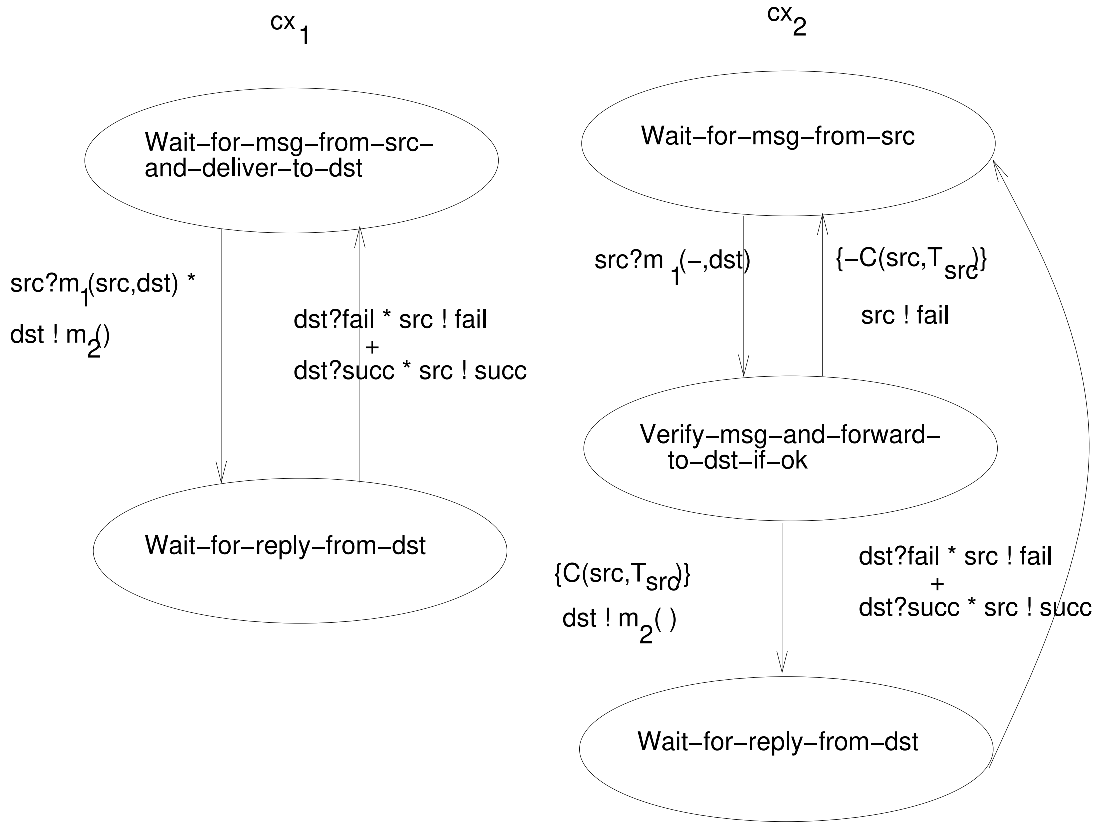

Figure 1: The Communications Exchange cx: (a) cx_1 transfers a message from src to dst (b) cx_2 transfers a message from src to dst after verification

cx has two component machines (Fig.1):

1. cx_1 is used for forwarding a message from a source to a destination.

   cx_1 receives a message $m_1(src, dst)$ from a source src and forwards it to the destination dst as message $m_2()$.

   Note that the terms src and dst are used as parameters which assume different values at different times. For example, if $m_1(A_c, A_p)$ is sent to cx_1 by B_c, then $src = B_c$ and $dst = B_p$.

2. cx_2 is used for verifying a message from a source before sending it to a destination.

   cx_2 receives a message $m_1(src, dst)$ from a source src, checks if $C(src, T_{src})$ is TRUE and if so, it sends out $m_2()$ to dst.

B_c has two component machines:

1. $B_{c,cr}$ is used to extend credit to a customer C.
2. $B_{c,bill}$ is used for billing a customer.

B_p has four component machines:

1. $B_{p,cr}$ is used to handle credit extensions for a customer P.
2. $B_{p,auth}$ is used for authenticating a customer during transaction.
3. $B_{p,xfer}$ is used for transferring funds from P's account in B_p to S's account in B_s.
4. $B_{p,bill}$ is used for billing B_c, the bank of the customer C.

P describes the actions of the customer at the point of sale.

3.2 Phases of the Protocol

The protocol has three distinct phases:

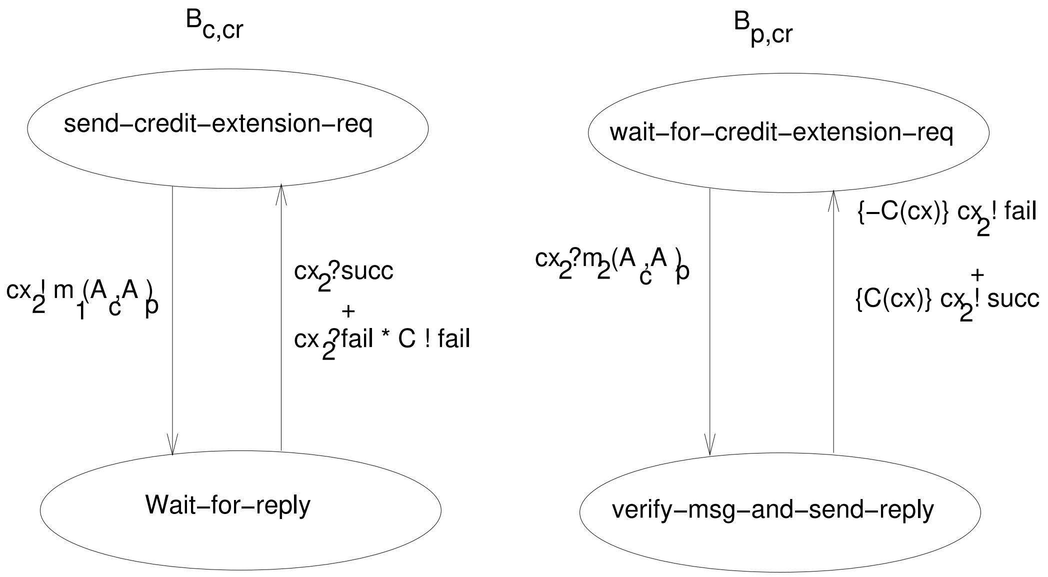

Figure 2: Extension of credit by B_c to A_p

(I) Extension of credit by B_c to A_p.
This involves component machines cx_2, $B_{c,cr}$ and $B_{p,cr}$ (Fig.2).

1. $B_{c,cr}$ sends the following message authorizing an extension of credit.
   $m_1(A_c, A_p) \;=\; F_{B_c}(\delta_{CR}, D^e(A_p))$ where δ_{CR} is the credit limit for C selected by B_c.
2. $B_{p,cr}$ awaits the message
   $m_2(A_c, A_p) \;=\; X_{cx}[(\delta_{CR}, EP_{B_p}(P, rn_2)), B_p]$
   It sends a $succ$ ($fail$) signal to cx_2 depending on whether $C(cx, true)$ is true or false.

(II) Transaction between P and B_p.

This phase has two sub-phases:

(a) Authentication of P by B_p

(b) Transfer of fund from P's account in B_p to S's account in B_s.

(II (a)) Authentication of P by B_p involves P and $B_{p,auth}$ (Fig.3).

1. P sends the following message requesting a transaction with B_p.

   $m_1(P, A_p) \;=\; F[X_P(M_P, B_p), D^e(A_P)]$ where

   $M_P = \delta_M, D^e(A_s), EP_{cx}(S, rn_5), pin, EX_c(\delta_M,\ S,\ M,\ comments)$

   δ_M is the amount of credit that P is requesting.

   $D^e(A_s)$ is the account that the money should be transferred to — S provides C with A_s, and C rather than S constructs $D^e(A_s)$.

   $EP_{cx}(S, rn_5)$ is the address at the store to send requests for further information to.

   pin is P's personal identification number, that he types in when using the card.

   $EP_{cx}[rn_6, EX_c(\delta_M,\ S,\ M,\ comments)]$ is information that the customer associates with the purchase. It can only be read by C, and can be used to generate an itemized bill, or to trace questionable purchases.

   rn_3 and rn_4 in $D^e(A_P)$ are different from rn_1 and rn_2 used in $D^e(A_p)$ at B_c.

2. $B_{p,auth}$ receives

   $m_2(P, A_p) \;=\; X_P(M_P, B_p)$

   If $C(P, pin_{ok})$ is true, then

3. $B_{p,auth}$ sends the following message containing a question Q_i for P.

   $m_1(A_p, P) \;=\; F(X_{B_p}(EX_c(Q_i), P), EP_{cx}(S, rn_5))$

4. P receives

   $m_2(A_p, P) \;=\; X[X_{B_p}(EX_c(Q_i), P), S]$.

   If $C(B_p, true)$ is true, then

5. P sends the following message containing the answer A_i

   $m'_1(P, A_p) \;=\; F(X_P(EX_c(A_i), B_p), D^e(A_P))$

6. $B_{p,auth}$ receives

   $m'_2(P, A_p) \;=\; X_P(EX_c(A_i), B_p)$

   If $C(P, ac_{ok})$ is true, then

7. $B_{p,auth}$ sends a *start* signal to $B_{p,xfer}$, the component machine responsible for transferring funds to B_s from P's account in B_p.

(II (b)) Transfer of fund from B_p to B_s involves $B_{p,xfer}$ and B_s (Fig.4).

1. $B_{p,xfer}$ awaits the *start* signal from $B_{p,auth}$ and then
2. $B_{p,xfer}$ sends the following message containing an amount to be transferred to S's account in B_s.

   $m_1(A_p, A_s) \; = \; F_{B_p}((1-\alpha)\delta_M, D^e(A_s))$ where

   $\alpha\delta_M$ is the service charge to the store for the credit card transaction, and is split by the parties involved by any negotiated rule.
3. B_s receives

   $m_2(A_p, A_s) \; = \; X_{cx}(((1-\alpha)\delta_M, EP_{B_s}(A_s, rn)), B_s)$

   If $C(cx, T_{cx})$ is true, then
4. B_s sends *succ* signal which is received by $B_{p,xfer}$ via cx_2 and A_p is updated.
5. B_s also sends the following message containing a notification of receipt of money to S.

   $m_1(A_s, S) \; = \; F(X_{B_s}(rcvd \; \delta_M, S), S)$
6. B_s updates account A_s when it receives a *succ* signal from S via cx_1.

(III) Transfer of billing information from B_p to B_c involves $B_{p,bill}$ and $B_{c,bill}$ (Fig.5).

1. $B_{p,bill}$ sends the following message containing the amount spent by B_p on behalf of B_c during the billing period.

   $m_1(A_p, A_c) \; = \; F_{B_p}[(Rqst, \; EP_{cx}(rn_6, EX_c())), D^e(A_c)]$ where

   $Rqst$ is the amount B_p is requesting from B_c and $EP_{cx}(rn_6, EX_c())$ is the encrypted itemized bill for C
2. $B_{c,bill}$ receives

   $m_2(A_p, A_c) \; = \; X_{cx}[(Rqst, \; EX_c(), EP_{B_c}(A_c, rn_2)), B_c]$

   If $C(B_p, T_{B_p})$ is true, it sends Bill = $Rqst, EX_c()$ to C.

4 Collusion

The objective of the protocol is to make certain that no single party has enough information to link C to either S or M, and to make it necessary for as many of the parties to collude as possible before the customer is compromised. To analyze the protocol, we form an information-player table. The table contains the players involved in the protocol and the information available to each one of

INFO	PLAYER									
	B_c	B_p	B_s	S	cx and $m()$					
					A_c, A_p	A_p, A_c	P, A_p	A_p, P	A_p, A_s	A_s, S
B_c	O				O	O				
C	I_c									
B_p		O			O	O	O		O	
P		X								
B_s			O	O					O	
S			I_s	I_s				I_s		I_s
M				I_m						
$m_1(A_c, A_p)$	X				X					
$m_2(A_c, A_p)$		X			X					
$m_1(A_p, A_c)$		X				X				
$m_2(A_p, A_c)$	X					X				
$m_1(P, A_p)$				e			X			
$m_2(P, A_p)$		X					X			
$m_1(A_p, P)$		X						X		
$m_2(A_p, P)$				e				X		
$m_1(A_p, A_s)$		X							X	
$m_2(A_p, A_s)$			X						X	
$m_1(A_s, S)$			X							X
$m_2(A_s, S)$				X						X

Table 2: Information-Player Table

them. If a piece of information is uniquely known to a player, it cannot be used for collusion. Such information is removed from the table. Furthermore, assuming that there are a large number of money transfers in the system, the quantities δ_{CR} and δ_M cannot be used for collusion. They are also removed. In addition, if two or more pieces of information are available to exactly the same players, there is no advantage to keeping track of them all. In such cases, we keep only one piece of information and discard the rest. With this simplification, the information-player table is shown in table 1.[1].

There are two types of information: (a) information common to all the messages at a particular location and (b) information unique to a single message at a particular location. An entry corresponding to a player and an information of type (a) is marked with an O while an entry corresponding to a player and an information of type (b) is marked with an X. A few entries are marked with an e to denote that the information is available to the corresponding player if he eavesdrops.

I_c, I_s and I_m refer to the identities of the customer, the store he shops at and the merchandise he buys respectively. When messages pass through cx, cx learns information about the parties who exchange that information. cx can associate the information with the message.

[1]The information in $m_1(P, A_p)$ and $m_1'(P, A_p)$ have been combined and represented by the information in $m_1(P, A_p)$ in the following tables. Similarly, the information in $m_2(A_p, P)$ and $m_2'(A_p, P)$ is represented by the information in $m_2(A_p, P)$.

INFO	Collusion cx and				
	B_c	B_p	B_s	S	S_e
B_c	O	O			
C	I_c				
B_p	O	O	O		O
P		X			
B_s		O	O	O	O
S		I_s	I_s	I_s	I_s
M				I_m	I_m
$m_1(A_c, A_p)$	X	X			
$m_1(P, A_p)$		X			X
$m_1(A_p, A_s)$		X	X		
$m_1(A_s, S)$			X	X	X

Table 3: Two Party Collusion

4.1 Two Party Collusion

If two parties have two messages that have the same unique piece of information, they can combine those messages.

Players B_c, B_p, B_s, and S have no unique information in common, pertaining to C. Therefore, if any pair of them collude they learn nothing more about the customer. Each of these entities has messages in common with cx and can collude with cx to increase their information. For instance, B_c has $m_1(A_c, A_p)$ and $m_2(A_p, A_c)$ in common with records in cx. After collusion with cx, B_c learns $m_2(A_c, A_p)$ and $m_1(A_p, A_c)$. Similarly, B_p has $m_2(A_c, A_p)$, $m_1(A_p, A_c)$, $m_2(P, A_p)$, $m_1(A_p, P)$ and $m_1(A_p, A_s)$ in common with records in cx. After collusion with cx, B_p learns $m_1(A_c, A_p)$, $m_2(A_p, A_c)$, $m_1(P, A_p)$, $m_2(A_p, P)$ and $m_2(A_p, A_s)$.

The store has two different amounts of information, depending upon whether or not he eavesdrops on the customer's conversation. These are listed as S and S_e.

Note that if cx is trusted, no amount of collusion among the banks and stores can compromise the customer. Moreover, since none of the colluding entries have I_c and either I_s, or I_m, if cx is dishonest, and at most one of the other players is dishonest, then I_c cannot be linked to I_m.

After one of the parties colludes with cx, the next party that needs to be compromised is determined from the non-unique information, that is marked as "O" in the table. For instance, if B_c and cx collude, then B_c learns that C's Swiss account is in bank B_p.

Considering all possible two party collusions, we get a new information-player table. After removing identical rows, it reduces to the following table.

4.2 Three Party Collusion

When cx, B_c, and B_p collude on $m_1(A_c, A_p)$, the customer can be linked with the stores he shops at. There are no other combinations of three parties that can collude to link this information, and there are no combinations of three parties that can collude to link the customer and the merchandise.

INFO	Collusion cx and				
	B_c	B_p		B_s	
	B_p	B_s	S_e	S	S_e
B_c	O	O	O		
C	I_c				
B_p	O	O	O	O	O
P	X	X	X		
B_s	O	O	O	O	O
S	I_s	I_s	I_s	I_s	I_s
M			I_m	I_m	I_m
$m_1(A_c, A_p)$	X	X	X		
$m_1(P, A_p)$	X	X	X		X
$m_1(A_p, A_s)$	X	X	X	X	X
$m_1(A_s, S)$		X	X	X	X

Table 4: Three Party Collusion

INFO	Collusion cx			
	B_c and B_p		B_p and B_s	
	B_s	S_e	S	S_e
C	I_c	I_c		
P	X	X	X	X
S	I_s	I_s	I_s	I_s
M		I_m	I_m	I_m

Table 5: Four Party Collusion

4.3 Four Party Collusion

Out of the nine ways to pick 4 colluders, 4 of the ways can be accomplished by mutual unique information, and are listed in the table.

With four parties, it is possible to link the customer with both the store and the merchandise, if the store eavesdrops on the customer's conversation. Otherwise, it is only possible to link the customer with the store, as with three parties.

From this observation, we note that, since a bank is expected to collude with itself, the number of colluders necessary to link a store and customer is not reduced if the store and customer or pseudonym have account in the same place, but is reduced if the customer and pseudonym have accounts in the same place.

If the store cannot eavesdrop, the customer and merchandise cannot be linked unless all five parties collude. Therefore, it is worthwhile making any effort to prevent the store from seeing the messages that the customer sends to the communication exchange.

5 Conclusion

We have presented a cryptographic protocol which can be used to implement a credit card with the anonymity of cash. This protocol allows people to preserve their privacy without losing the advantages associated with credit cards. We firmly believe that this is an important issue for our society in the near future. Currently, we are working on an expanded version of this paper which does a more formal analysis of our protocol and shows how our protocol can be further simplified.

6 Appendix

The following notations are used in CEFSM:

1. *src?msg* means that a message *msg* is expected from the source *src*.

2. *dst!msg* means that a message *msg* is sent to the destination *dst*.

3. { *cond* } means that the transition with this label is *enabled* if *cond* is true.

   { - *cond* } means that the transition with this label is *enabled* if *cond* is false.

4. *[block]* means that when the transition with this label is exercised, the actions within the *block* are executed.

5. As an abbreviation, an edge will be labeled by $a * b$ to denote an edge labeled by an input/output operation a followed by an edge labeled by an input/output operation b. Similarly, an edge labeled by $a + b$ between two states will denote two edges connecting the same two states, one labeled by input/output operation a and the other labeled by input/output operation b.

References

[C91] Douglas E. Comer, "Internetworking with TCP/IP, Volume I, Second Edition," Prentice Hall, Inc., 1991.

[D84] D. W. Davies, "Smart cards, digital signatures, and negotiable documents," Proceedings of the International Conference on Secure Communication Systems, London, UK, February 22-23, 1984, pp. 1-4.

[DC85] David Chaum, "Security without Identification: Transaction Systems to make Big Brother obsolete," Communications of the ACM, vol. 28, no. 10, pp. 1030-1044, Oct. 85.

[DC86] David Chaum, "A secure and privacy-protecting protocol for transmitting personal information between organizations", Advances in Cryptology - CRYPTO '86. Springer Verlag, 1987.

[DH76] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. IT-22, pp. 644-654, Nov. 1976.

[EMT89] D. Estrin, J. C. Mogul and G. Tsudik, "Visa protocls for controlling interorganizational datagram flow", IEEE Journal on Selected Areas in Communications, 7(4), May 1989.

[GU86] L. C. Guillou and M. Ugon, "Smart card: A highly reliable and portable security device," Lecture Notes in Computer Science 263; Advances in Cryptology: Proc. Crypto '86, A. M. Odlyzko, Ed., Santa Barbara, CA, Aug. 11-15, 1986, pp. 464-479. Berlin: Springer-Verlag, 1987.

[LMP93] Steven Low, Nicholas F. Maxemchuk and Sanjoy Paul, "Analysis of Anonymous Credit Cards", under preparation.

[M88] Judy H. Moore, "Protocol failures in cryptosystems", Proceedings of the IEEE, 76(5), pp. 594-602, May 1988.

[R78] M. O. Rabin, "Digitalized Signatures," Foundations of Secure Computation, R. A. DeMillo, D. P. Dobkin, A. K. Jones, and R. J. Lipton, Editors, pp. 155-168. New York: Academic Press, 1978.

[RSA78] R. L. Rivest, A. Shamir, and L. Adelman, "A method for obtaining digital structures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-127, Feb. 1978.

[S92] Gustavus J. Simmons, editor, "Contemporary Cryptology: The Science of Information Integrity", IEEE Press, 1992.

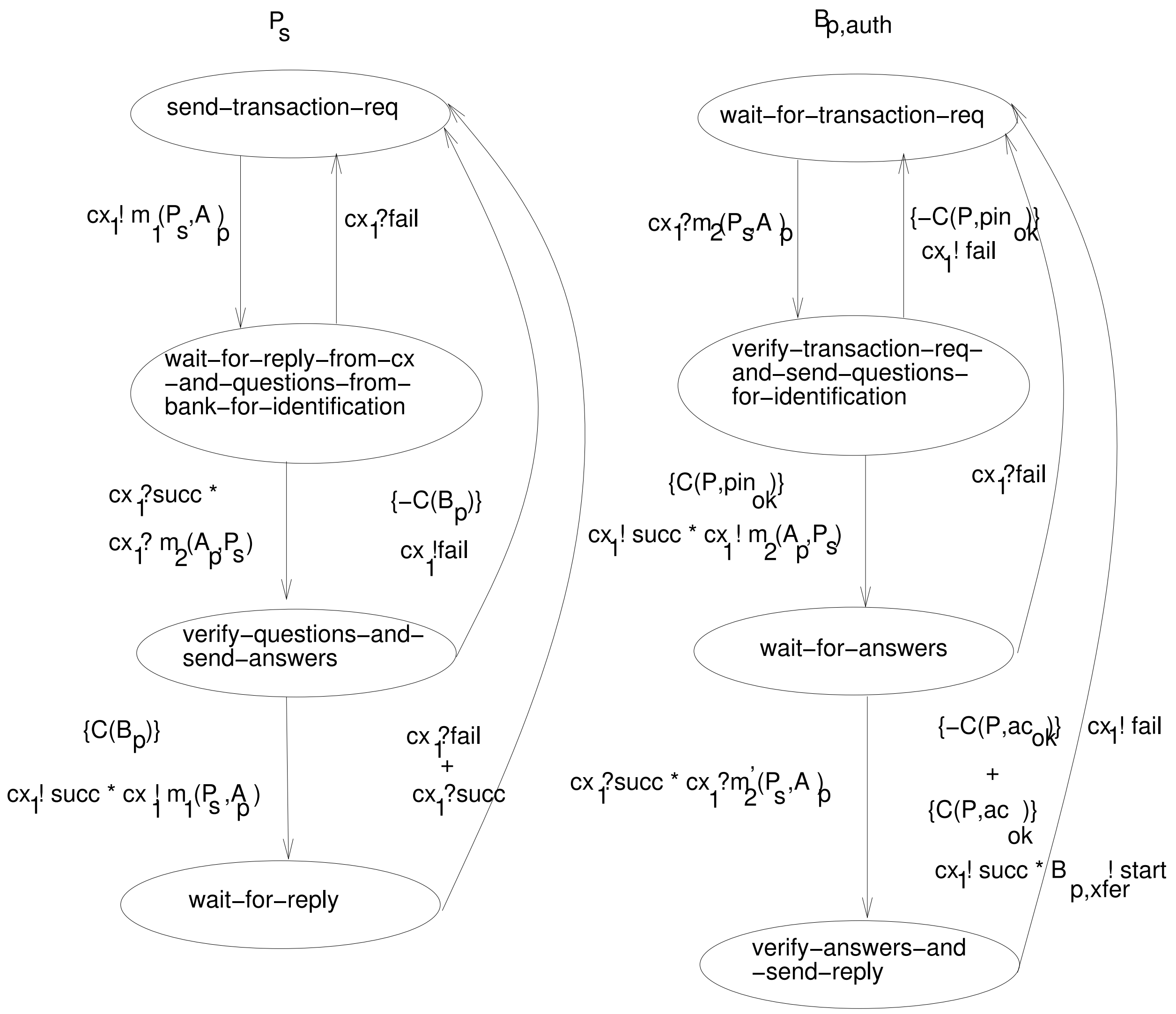

Figure 3: Authentication of P by B_p

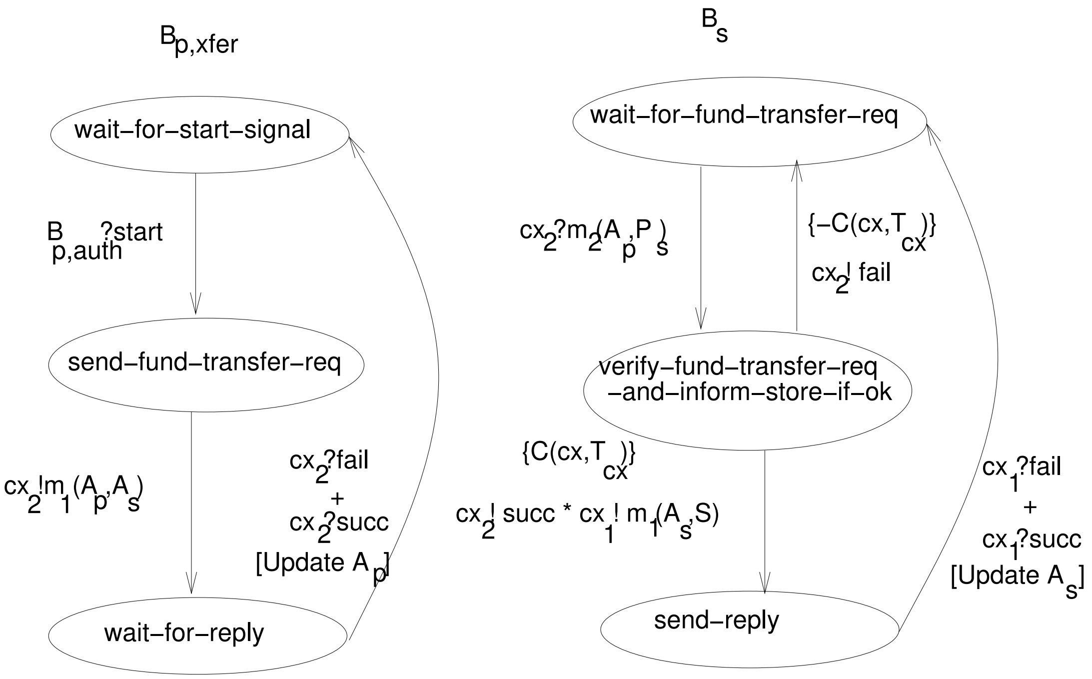

Figure 4: Transfer of fund from B_p to B_s

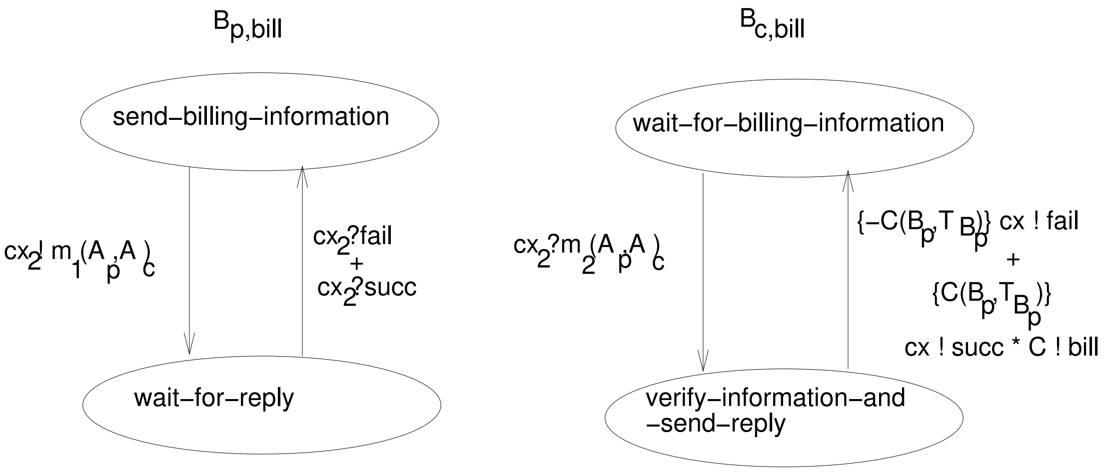

Figure 5: Billing Information from B_p to B_c