An User-centric MIX-net Protocol to Protect Privacy*

Alessandro Acquisti[†]
acquisti@sims.berkeley.edu
UC Berkeley

Workshop on Privacy in Digital Environments: Empowering Users
CSCW 2002, New Orleans, US - November 2002

Abstract

"MIX-net" systems protect the privacy of participants by clouding together their transactions through cascades of third parties. Reliability and trust are therefore open issues in this literature and limit the applicability of these systems. This paper discusses how the MIX approach can be adapted to put the user at the center of the protocol and in control of it, so that each participant can take active steps to protect his or her privacy. The paper also highlights various possible uses of the protocol. Being "in control" comes at a cost, however, and the paper discusses the trade-offs arising from the proposed approach.

1 Introduction

In 1981, David Chaum ([7]) introduced the concept of "MIX" - a third party that combines and forwards messages from several senders to several recipients, so that no relation between any particular sender and any particular recipient can be externally observed.

The MIX approach has been applied to "untraceable" digital pseudonyms (as discussed already in [7]), synchronous and asynchronous communication systems (see [12], [14], [11], and [15]), as well as electronic voting (see e.g. [13]). These applications rely not just on one MIX but on *cascades* of multiple MIXes forming a "MIX-net". The MIX-net clouds the relation between messages (or pseudonyms), senders, and recipients even more: each message may go through several MIXes before reaching its final destination.

Trust and reliability, however, are open issues in the MIX-net literature (see [10] and [9]). In the single MIX version of the protocol, the third party must be trusted not to reveal the correspondence between senders and recipients. In the cascade version of the protocol, collusion among various MIXes can expose the identity of the sender of a certain message. In addition, a certain MIX in a cascade may stop forwarding the messages transiting its server without being immediately detected, thus affecting the reliability of the system.

This paper discusses how the MIX approach can be adapted to put the user at the center of the protocol and in control of it, thereby addressing reliability and trust issues. Unlike in traditional MIX-net systems, in the approach proposed in this paper each user can take active steps to protect her privacy also in environments where other parties cannot be trusted. Furthermore, the protocol satisfies the ACID properties (see [6]) and therefore achieves reliability.

The rest of this paper discusses the general idea behind this MIX-net variation by presenting first an informal description (Section 2) and then a formal, but generic analysis (Section 3). Being in control, however, comes at a cost. The user needs to be more involved in the process. Section 4 therefore discusses the usability and economic trade-offs arising from the proposed approach. The analysis presented in this paper is generic, in that it can be applied to several systems - such as anonymous payment and secure electronic voting. An Appendix at the end of this paper highlights how the protocol can be embodied as an anonymous payment system.

*I thank John Chuang, Doug Tygar, and Hal Varian for comments and suggestions. The usual caveats apply.

[†]Mail: SIMS, 102 South Hall, Berkeley, CA, 94720. Url: http://www.sims.berkeley.edu/˜acquisti.

2 Putting the User in Control

Imagine that Bob wants to complete a certain *transaction* (such as purchasing a good, sending a message, or voting in an election, and so on) with Alice, without letting Alice (a seller, a recipient for a message, an entity organizing an election, etc.) or anybody else know that this particular transaction is being completed by Bob. Let us also imagine that Bob nevertheless has to first identify himself in order to be allowed to start his transaction at all. A third party offers to intermediate between Bob to Alice without being able to associate the two. This party will allow Bob to hide his transactions with other transactions between other "sending" parties and Alice (or other "receiving" parties), in a way that all transactions will be either committed or aborted, and nobody will be able to link any specific parties to any specific transaction. In particular, the third party claims that it will be able to:

1. if needed, verify the *eligibility* of each party to start certain transactions (such as purchasing a good, sending a message, or voting in an election) with other "receiving" parties. This means that the third party will verify, for example, the financial ability to complete a certain transaction, or the eligibility to send certain messages, or the eligibility to vote, and so on;

2. if needed, verify the *validity* of the transaction itself;

3. finally, enable each "sending" party to interact with other untrusted parties in order to complete a transaction with one or more "receiving" parties, in such a way that each receiving party knows that the eligibility of the "sender" of the transaction and the validity of the transaction have been verified, but neither the third party nor any of the receiving parties are able to associate a specific transaction to the sending party that has originated it.

This sounds good, Bob thinks. But how can he trust that this stranger, the third party, will do what it claims? The third party explains that after Bob will have started the transaction with it (and, if necessary, will have given necessary proofs of eligibility according to the transaction he wants to complete),[1] in return it will give Bob a number of unique tokens that can be used to complete that transaction (for example, payment tokens, or tokens representing the ability to cast a future vote, etc.). Bob will then be able to contact other token owners and agree with them to hide his transaction with theirs. In particular, they will all simultaneously return their tokens to the third party, and ask for those tokens to be redeemed into *new* tokens. The new tokens will again be unique, but they will be issued in such a way that the third party no longer will know *which* original creditor is getting *which* new tokens.

This is possible because the tokens owners will send the old tokens and ask for new ones in *two different moments* and under *two different pseudo-identities* (by which I refer to nothing more as two different ways Bob can contact the third party, and the third party can contact both, without anybody else but Bob being able to link the two forms of contact).[2] At the end of this process Bob and each of the other users will receive sets of new tokens that they can "exchange" again or use, for example by spending them to purchase goods from sellers, without revealing their original identity to any seller, or using them to vote anonymously at an electronic election.

This process sounds complicated (and its privacy properties obscure) because the above description is very general. Its generality, however, makes it easy to embody the protocol into several different applications: payment, voting, messaging, etc. The following section presents a formal description of the protocol, and the Appendix at the end of this paper highlights its embodiment as an anonymous payment system.

3 The Generic Approach

This sections offers a generic yet formal description of the approach loosely described above.

[1] In some applications no eligibility at all will be needed. In others - such as voting in an election - it might be necessary.

[2] Imagine, for example, two different Hotmail addresses used by the same individual, or two different postal addresses.

Imagine that there exists a set of sending parties, numbered 1 to N, and a set of receiving parties, numbered 1 to M, and a third party (the MIX). The protocol is composed of the following steps:

1. Each sending party sends a first message, 1, to a third party. Message 1 contains information that the third party needs to know to verify the eligibility of the third party to send a message to one or many of the receiving parties (for example, the real identity or proof of age of the sender). "Eligibility" is a context-dependent concept that will not be discussed here but in specific applications (see [2] and [1]). Message 1 also contains some form of "return information" that the third party needs in order to reply to the sender (for example, a return postal address).

2. The third party verifies the eligibility of each sending party. Then, for each sending party which is deemed eligible, the third party sends a second message, 2, using the return information included in message 1. Each message 2 contains a unique receipt of eligibility. This receipt is associated to the return information included in message 1, in the sense that anybody could recognize that a certain receipt of eligibility has been created for a certain return information (for example, the receipt could be a written statement by the third party address to the sending party, with attached an unique identifier).

3. The N sending parties send each a third message, 3, to the third party. Each message 3 contains the actual data that each sending party wants to be validated and then sent anonymously to the receiving party (or parties), and an unique identifier for that message. Again, "validity" is context-dependent. Each message 3 also contains a "new" return information that allows the third party to contact the sender of each message 3. "New" means that for each sending party the return information contained in message 3 is different from the return information contained in message 1. The two return addresses have no external association or link. Neither the third party or any other party (excluded the sending party itself) should be able to associate message 1 and message 3 sent from the same sending party from the return information alone. This assumes the existence of some form of anonymous channel. While specific electronic implementations are possible, it suffices here to note that message 1 and message 2 could simply be two physical letters posted from different geographical locations and reporting different return addresses.

4. The third party compiles a list of these T of the N messages 3 whose included data can be considered "valid" (where it may be: $T \leq N$), and associates each of them to one single unique identifier created for the list. Thereafter, the third party sends to all N parties who sent messages 1 a fourth message, 4, using the return information contained in message 1. The fourth message is the same for everybody and contains the unique identifiers for T messages containing data that have been deemed valid, as well as the unique identifier that the third party has selected for that list.

5. All N parties receives the same message 4, but only T of them find that the unique identifier associated to their message 3 is contained in the list. Those T sending parties who now know that their message 3 has been validated, reply to the third party by sending each a fifth message, 5. Each message 5 contains for each sender the unique receipt of eligibility that that sender had received from the third party with message 2. Each message 5 also contains the unique identifier for the list sent by the third party with message 4. However, each message 5 does *not* contain any reference to message 3. This means that upon receiving T instances of message 5, the third party will not be able to link the sender of each message 5 to the sender of each message 3. Hence the third party cannot associate the receipt of eligibility to the validated final data contained in message 3 that each sending party wants to send.

6. The third party, upon receiving exactly T instances of message 5, each containing a unique receipt of eligibility and the unique identifier for the list, creates T receipts of validation for the T selected messages. A receipt of validation is a message that shows that the data contained in each message 3 has been deemed "valid" by the third party. Each receipt of validation is unique and is not repeated for any other sending party, and is of course

associated to the third message in a way that every other party could recognize that a certain receipt of validation has been created for a certain third message or the data it includes (for example, in an application based on cryptography, the receipt of validation might contain an hash of message 3). The third party then sends the unique receipts of validation to the T parties in T sixth messages, 6. Importantly, the third party uses the return information contained in each of the T selected third messages, rather than the return information associated to messages 1 or 5.

7. Upon reception of the receipts of validation, each sending party can now create the seventh and final message which has to be sent "anonymously" to the final recipient. Each seventh message contains the data originally contained in the third message, and the associated receipts of validation. "Anonymously" means that the recipient of the message will know that the sender is an eligible sender and that the message is a validated message. However, the recipient will be unable to link a specific receipt of validation to a specific receipt of eligibility. So, for example, in a payment application the merchant might receive a message containing payment tokens (the "validated" data), but will be unable to link them to the credit card information that the sender passed to the third party in order to receive the tokens (the "eligibility" receipt). In a voting application, the electoral authority receiving a certain valid voting token (the validated "data") will be unable to link it to the legal identity of the voter (whose eligibility to vote will have been verified in message 1). In an anonymous messaging application, each sender might receive a list of recipients in message 1, and an encrypted message in message 3, so that it would forward the encrypted messages to entire sets of selected recipients with only the designated recipient being able to decrypt the message, but nobody else being able to link a specific message to a specific recipient.

Two simple variations of the above protocol let users control the level of privacy they can achieve by participating to the system.

A first variation is the following. Rather than using their receipts of validation in message 7 to complete the desired transaction, senders can start again the protocol from step 3, creating new data and asking for new receipts of validation to be sent back in a repeated step 6, every time sending back, in a repeated step 5, the receipt of validation received in the previous round of the protocol. Through these repetitions of the protocol, old receipts of validation are "redeemed" into new ones together with those of other senders, so that the relation between each original receipt of eligibility and each receipt of validation is clouded even further.

A second variation involves letting the senders independently forms groups to receive further messages from the third party. After receiving message 2, one or more of the N sending parties broadcasts a new message, 3, so that all other N parties and the third party may receive it (for example, it publishes it on a public board). Message 3 contains the return information to contact the sending party and an invitation to other parties to send to that return address a fourth message, 4. Each message 4 should copy or refer to message 3 and contain return information to contact each of the senders of message 4. Let us imagine now that some or all of the parties reading message 3 decide to send each one message 4 to the sender of 3. Then the sender of 3, upon reception of N or less than N messages number 4, will select R of them (with $0 \leq R \leq N$), and will send to the R selected parties (as well as to the third party) a fifth message, 5, using the return information that the R parties have included in their messages number 4. This message 5 contains an unique identifier to represent the R senders as a group. The R senders receive this identifier and paste it in a sixth message, 6, to the third party, which contains for the rest all the other information of message 3 in the main protocol above. The main difference is that now the third party will send a message to N about which of the R parties have been validated.

These variations give the users more control on the system and on their own privacy, as discussed in the next Section.

4 Why the Protocol Empower Users, and Its Trade-offs

In the generic protocol described in Sections 2 and 3 a third party acts as a MIX, intermediating between parties who do not know or trust each other. The protocol presented here therefore is related to several strands in the cryptographic literature: MIX-nets ([7]), but also onion routing ([11]), and crowds ([16]); ACID properties and privacy in ecommerce (e.g. [6]); as well as ANDOS protocols (e.g., [5]), group signatures [8], and the cocaine auction protocol ([17]).

While detailed properties, current known vulnerabilities, and extensions of the protocol in specific embodiments are discussed at length in [2] and [1], here I offer an overview of why the protocol addresses trust and robustness issues in the MIX-net literature and how it empowers the users with respect to the level of privacy they want to achieve.

The protocol presented in this paper lets each participant hide his transaction with those of other participants. The protocol also satisfies "ACID"[3] properties, because transactions/messages are verifiably completed or aborted. Each step in the above protocol represents a transaction that is either committed or aborted for all participants. This means that the inactivity of the MIX or its inability to mix messages and transactions can be immediately exposed without compromising privacy and anonymity (see Appendix and [2] and [1] for proofs). Trust requirements are reduced with respect to traditional MIX-net systems, because each participant sends messages 1 and 3 in *two different moments* and under *two different pseudo-identities* that cannot be linked by the MIX. In fact, not only the third party cannot compromise anonymity, but the parties with a *potential* ability to compromise anonymity are those with the least incentive to do so: the users themselves. Even so, assuming that just a small share of parties will *not* collude with the third party is enough to achieve a "probabilistic" level of privacy, in the sense that each participant can adopt measures to preserve his anonymity against collusion among other parties and bring the probability of remaining anonymous asymptotically close to one, given the other conditions. In this sense the protocol empowers the user: specifically, each user can call for other users to "exchange" their receipts of validation (or tokens) through the third party, and can decide how many times to repeat this process. In this approach, therefore, the users are actually responsible for and in control of their own level of anonymity, and the risk of collusion is minimized because the parties with the potential ability to collude are those with the least incentive to do so: the users themselves.[4] One property of this protocol is, in fact, that customers who do not trust each other still have a shared interest in performing as many exchanges or rounds as they can, and therefore in coordinating with each other.

All of this comes at a cost, however.

Every privacy concerned individual, when trying to protect their privacy, faces trade-offs between the advantages from using anonymous technologies (for example, the avoidance of future risks coming from privacy intrusions) and their costs (attention, time spent using the system, resources involved, etc.). These trade-offs can be formalized - see also [3] - and then applied to specific systems, as in [4]. Here I just highlight some economic implications of the protocol presented in this paper.

Firstly, the user has to spend more attention in the process, following its steps, and taking, for highest security, the lead. This might be expensive: in order to achieve higher and higher privacy confidence, the user might use the system repeatedly. Hence, compared to other MIX-net protocols which do not empower the users, the one proposed here trades higher protection (because the risk of collusion and of interference by "bad" MIXes or other participants is lower) and higher reliability (because the protocol satisfies the ACID requirements) with a more direct intervention (and therefore effort) by the customer and a possibly larger amount of messages.

Secondly, while this variation on the MIX approach can be used in various applications (included anonymous messaging), it is most suited to areas where ACID properties and robustness

[3] ACID stands for atomicity, consistency, isolation, and durability. See [6].

[4] After the first round, the MIX third party knows that the customer must have received one of the sets of newly created receipts. The user can therefore repeat steps 3 to 7 n times, clouding further and further any initial relation between his receipt(s) of validation and receipt(s) of eligibility, until he is satisfied with the probabilistic level of privacy he has achieved. In an untrusted environment where the share of users who collude with the MIX is π (with $0 < \pi < 1$), the probability p_a of remaining anonymous will be function of: $p_a = f(\pi, n, s_i)$ where n is the number of rounds each user goes through and s_i is the size of the group the user is exchanging receipts with at round i. p_a is positively correlated with n and s_i and is negatively correlated to π.

are critical, such as payments and elections, because receipts of validations (or tokens) can be dissociated from the nature of the transaction. Not so with anonymous messaging.

The above considerations suggest that users with different privacy sensitivity might tend to use this system differently (some only "exchanging" receipts once, some others exchanging them repeatedly), or using different, less secure, and more automated systems altogether. Similar conclusions are reached in [4] for traditional MIX-net systems. When the privacy concerns of individuals are distributed with enough variation, systems like the one proposed here might generate equilibria where users with the highest evaluations of privacy will act as MIXes (in MIX-net systems) or as active promoters (calling for exchanges, in the variation proposed here) while the others will just participate as passive users thus providing enough "noise" to ensure privacy.

5 Conclusions

This paper presented a protocol that allows users to protect their own privacy through the intermediation of an untrusted third party. The protocol is comparable to the MIX-net approach, but satisfies ACID properties and lightens the trust and reliability issues associated to traditional MIX-net systems. In this protocol privacy is preserved because no party is able to associate specific participants with the information regarding their transactions, and participants can adopt measures to preserve their anonymity against collusion among other parties. We discussed with more detail the version of the protocol that protects the privacy of participants to financial transactions, and we proposed some preliminary analysis of the trade-offs arising from putting the user in control of her own privacy protection.

6 Appendix: An Application to Anonymous Payments[5]

This appendix highlights a particular application of the above approach to anonymous payments. A more detailed description is available in [2]. This application is based on the following assumptions:

- All parties can perform basic cryptographic operations.
- Parties might have known or verifiable public keys.
- All signatures can be verified by the receiving parties.
- Encrypted messages cannot be decrypted without the proper keys.
- Each party can have several pseudo-identities that represent different ways other parties can contact that party (e.g., different email addresses, or postal addresses, or on-line accounts). For each pseudo-identity, the party might have an associated set of public and private keys. For each party, one of these pseudo-identities will be the actual legal identity of that party.[6] External parties cannot link together the different pseudo-identities of one same party from the pseudo-identities alone. Similarly, if the parties are communicating on-line, their identities cannot be revealed by their IP addresses alone.
- There are several selling parties and several buying parties. There is one MIX party, that will be called the facilitator F.

The notation is the following:

- $E_X\{.\}$ means that message (.) has been encrypted with key X.
- $A \rightarrow B : t$ represents the communication of t from A to B.
- $A_{A\_1} \rightarrow B_{B\_1}$: represents a communication from the pseudo-identity 1 of A to the pseudo-identity 1 of B.
- $[1, 2, ..., X] \rightarrow B$: represents x distinct communications from $1, 2, ...$, and X to B.
- $A \rightarrow * : t$ represents a broadcast communication of t from A to all other parties.
- $A \rightarrow *_{BB} : t$ represents a publication of a message t by A on some "bulletin board" BB, where other parties can access the message.

The protocol begins with a customer C, who wants to purchase anonymously from a merchant M a good whose price is advertized as M_amount. C starts the transaction with the facilitator MIX F, generating a long number ($C\_transaction\_id$) and sending:

1. $C_{C\_t} \rightarrow F : E_{C\_tPR}$ $\{E_{F\_PB}\{C\_transaction\_id, C\_amount, C\_tPB, \textit{account information associated to } C\_t\}\}$ where C_t is the pseudo-identity t for the customer C. At the beginning of the protocol we set $t = 1$,[7] and therefore we consider the pseudo-identity "$C\_t = 1$" to be the real, publicly verifiable identity of the customer, associated to some payment instrument he owns (for example, the customer's credit card information, with his legal name, billing address, etc.); C_tPB is a public key that C wants F to use to encrypt the tokens received in exchange for the funds he is transferring; C_amount is the amount that C wants to transfer to F; and E_{F_PB} $\{.\}$means that the message has been encrypted with F's public key; *account information associated to* C_t is the information that F needs in order to charge the customer's account. This message is signed with C_tPR, a private key belonging to C and associated to the pseudo(real)-identity $C\_t = 1$.

[5] Parts of this section are derived from an earlier paper: [2].

[6] For simplicity, also this real identity will be called a "pseudo-identity" for that party.

[7] The notation C_t is preferred to C_1 for expositional reasons, because the protocol can be used recursively (see the rest of this Section).

F charges C_amount to the customer's account and creates n tokens T, where $n = \frac{C\_amount}{unit\ value\ of\ token}$ and T is defined as: $T = E_{F\_PRT}\,\{E_{C\_tPB}\{\text{random number}\}\}$. That is, each token is a random number encrypted with C_tPB - the public key of the pseudo-identity - and is signed by F with a private key F uses for its tokens, F_PRT.

2. $F \rightarrow C_{C\_t} : E_{C\_tPB}\{C\_transaction\_id, T^{C\_t}_{1,...,n}\}$ where $T^{C\_t}_{1,...,n}$ represents the n tokens T that F is sending to customer C_t.

3. $C_{C\_t} \rightarrow *_{BB} : E_{C\_tPR}\,\{C\_tPB, n_{C\_t}\}$ where C_t publishes a signed message on some form of public board accessed by other customers of F, requesting other customers (whom C_t does not know) to exchange their tokens Ts with him. In other words, C_t is offering n_{C_t} of its tokens (where n_{C_t} can be equal to n) to be exchanged with other customers' tokens, in order to redeem his current tokens into new ones.

4. $[1,2,...,X]_{[1,2,...,X]\_(t+1)} \rightarrow C_{C\_t}$:
$E_{C\_tPB}\left\{E_{[1,2,...,X]\_(t+1)PR}\,\{[1,2,...,X]\,\_\,(t+1)\,PB, E_{C\_tPR}\,\{C\_tPB, n_{C\_t},\}\}\right\}$ where $[1,2,...,X]$ are x customers that reply to C_t with their pseudo-identities $t+1$, accepting C_t's proposal to "exchange" n_{C_t} and sending some *new* public keys, $[1,2,...,X]\,\_\,(t+1)\,PB$, which they want F to sign the "new" tokens with. Note that $[1,2,...,X]_{[1,2,...,X]\_(t+1)} \rightarrow C\_t :$ represents x communications from $1,2,...,$ and X to C_t. These communications are separate (customers do not know each others and coordinate only through the board), but here they are represented together for concision. Note also that at this moment there is yet no guarantee that all the $[1,2,...,X]$ customers actually have valid tokens to redeem. Finally, note that the other customers contact the original sender C not through their original identities $[1,2,...,X]\,\_\,(t=1)$ (which were the publicly known identities used to pay for the original tokens); instead, they contact C through *new* identities $[1,2,...,X]\,\_\,(t+1)$ and provide new public keys $[1,2,...,X]\,\_\,(t+1)\,PB$ that have no publicly known link with the previously used identities and public keys.

5. $C_{C\_t} \rightarrow F, [1,2,...,X]_{[1,2,...,X]\_(t+1)}$:
$E_{C\_tPR}\,\{n_{C\_t}, [C,1,2,...,X]\,\_\,(t+1)\,PB, [C,1,2,...,X]\,\_\,(t+1)\,PB\}$
where $[C,1,2,...,X]\,\_\,(t+1)\,PB$ and $[C,1,2,...,X]\,\_\,(t+1)\,PB$ have the usual meaning except that now also C_t is sending the public key corresponding to its new pseudo-identity C_t+1, that he wants F to encrypt the new tokens with. With this message, C is telling F that x customers want to exchange n_{C_t} tokens with him. Implicitly, C asks F to create $n_{C\_t} * x$ new tokens if and only if F will have received $n_{C\_t}* \, x$ original tokens-key pairs from x users of the protocol. In this protocol F agrees to create $n_{C\_t}* \, x$ new tokens and encrypt each n_{C_t} of them with the public keys $[C,1,2,...,X]\,\_\,(t+1)\,PB$, respectively, after it receives the private keys to decrypt $n_{C\_t}* \, x$ tokens which have not been redeemed or spent before. Note, however, that F itself cannot link old tokens and new tokens - in fact, *nobody* (except the individual who owns both) can: for example, even if C is sending inside his message from his original pseudo-identity C_t also his own new public key corresponding to its new pseudo-identity C_t+1, F cannot link any specific $[C,1,2,...,X]\,\_\,(t+1)\,PB$ to any specific $[C,1,2,...,X]\,\_\,(t+1)\,PB$, because the new pseudo-identities are all presented together.

6. $[C,1,2,...,X]_{[C,1,2,...,X]\_t} \rightarrow F, [C,1,2,...,X]_{[C,1,2,...,X]\_(t+1)}$:
$E_{[C,1,2,...,X]\_tPR}\left\{S, E_{F\_PB}\left\{T^{[C,1,2,...,X]\_t}_{1,...,n_{c\_t}}, [C,1,2,...,X]\,\_tPR\right\}\right\}$ where S is message 5 above that each customer $1,2,...,X$ is now accepting and signing. Each customer, after seeing that his new public key $(t+1)\,PB$ is indeed listed in message S, agrees to actually send to F the tokens he received from it with the key to decrypt them ($[C,1,2,...,X]\,\_tPR$), under *his original pseudo-identity* $[C,1,2,...,X]\,\_t$. So, for example, 2_1 will send her keys because she knows that 2_2 will then receive new tokens - but 2 is the only customer who knows that the keys used by 2_1 and those used by 2_2 are actually used by the same customer. F can verify the validity of the tokens (i.e., that they have not been spent or redeemed before) and the fact that they come from the legitimate owner (because the private key must decrypt valid tokens), but it is not able to link any specific member of $[C,1,2,...,X]\,\_t$ with any specific member of $[C,1,2,...,X]\,\_t+1$. Again, it only knows

that the same customer must be part of both groups. On the other side, if the customer has found his new public key in the message S, then he can safely send out the private key that decrypts the tokens he has originally received, because *if* new tokens will be generated, he will receive them. Tokens *will* be generated *if* all customers listed in S do send in their old tokens.
F verifies that it has now $x * n_{C\_t}$ valid tokens-key pairs, where valid means that by decrypting the tokens with the keys it has received it obtains valid random numbers, and that those random numbers have not been successfully redeemed or spent before. If that is not the case, the transaction aborts and C starts again from step 3. Otherwise, the transaction is committed:

7. $F \rightarrow * : E_{F\_PR}\{E_{C\_(t+1)PB}\left\{T^{C\_(t+1)}_{1,...,n_{C\_t}}\right\}, E_{1\_(t+1)PB}\left\{T^{1\_(t+1)}_{1,...,n_{C\_t}}\right\}, ..., E_{X\_(t+1)PB}\left\{T^{X\_(t+1)}_{1,...,n_{C\_t}}\right\}\}$
where the notation represents a broadcast of a message by F where $x * n_{C\_t}$ new tokens are created by F, and each n_{c_t} of those are encrypted with (respectively) one of the new $[1, 2, ..., X]\_(t+1)\,PB$ keys it has received in message 5. Note again that F receives the messages described in 5 from the pseudo-identities $t+1$ and the messages described in 6 from the pseudo-identities t, but is unable to link them. Therefore, F is sending the new tokens without being able to know which original customer is getting which set of tokens. It only knows that the original customer must have received *one among* the new sets of tokens.

References

[1] Alessandro Acquisti. An anonymous, fair evoting/recommendation system. Technical report, School of Information Management and Systems, UC Berkeley, 2002.

[2] Alessandro Acquisti. Anonymous transactions in untrusted environments. Technical report, School of Information Management and Systems, UC Berkeley, 2002.

[3] Alessandro Acquisti. Protecting privacy with economics: Economic incentives for preventive technologies in ubiquitous computing environments. In *Workshop on Socially-informed Design of Privacy-enhancing Solutions, 4th International Conference on Ubiquitous Computing - UBICOMP '02*, 2002.

[4] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. Open issues in the economics of anonymity. Technical report, University of California, Berkeley, MIT, and Naval Research Lab, 2002.

[5] Gilles Brassard, Claude Crepeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - Crypto '86*, pages 234–238. Springer Verlag, LNCS 263, 1987.

[6] Jean Camp, Micheal Harkavy, J. Doug Tygar, and Bennet Yee. Anonymous atomic transactions. In *USENIX Workshop on Electronic Commerce*, pages 123–133, 1996. citeseer.nj.nec.com/camp96anonymous.html.

[7] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[8] David Chaum and Eugene van Hejist. Group signatures. In *Advances in Cryptology - EUROCRYPT '91*, pages 257–265. Springer Verlag, LNCS 547, 1991.

[9] Roger Dingledine, Michael J. Freedman, David Hopwood, and David Molnar. A reputation system to increase MIX-net reliability. In Ira Moskowitz, editor, *Information Hiding - IH '01*, pages 126–141. Springer-Verlag, LNCS 2137, 2001. http://www.freehaven.net/papers.html.

[10] Roger Dingledine and Paul Syverson. Reliable MIX cascade networks through reputation. In Matt Blaze, editor, *Financial Cryptography - FC '02*. Springer Verlag, LNCS (forthcoming), 2002. http://www.freehaven.net/papers.html.

[11] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):39–41, 1999.

[12] B. Gulcu and G. Tsudik. Mixing email with BABEL. In *Symposium on Networked and Distributed System Security*, 1996.

[13] M. Hirt and K. Sako. Receipt-free voting based on homomorphic encryption. In *Eurocrypt*, 2000.

[14] David Mazières and M. Frans Kaashoek. The design, implementation and operation of an email pseudonym server. In *Computer and Communications Security - CCS '98*. ACM Press, 1998.

[15] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDN-mixes: Anonympous communication with very small bandwidth overhead. In *GI-ITG Conference: Communicartion in Distributed Systems*, pages 451–462, 1991.

[16] Michael K. Reiter and Aviel D. Rubin. Anonymous web transactions with Crowds. *Communications of the ACM*, 42(2):32–38, 1999.

[17] Frank Stajano and Ross J. Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding Workshop*, pages 434–447. Springer Verlag, LNCS 1768, 1999.