Using Payments to Promote Cooperation in Anonymity Protocols

Daniel R. Figueiredo Jonathan K. Shapiro Don Towsley *
Department of Computer Science, University of Massachusetts at Amherst

Computer Science Technical Report 03-31 †

September 23, 2003

Abstract

Like many peer-to-peer applications, anonymous communication systems are vulnerable to free-riders, peers that use the system while providing little or no service to others. To complicate matters, the identity of the free-rider is obscured by the very anonymity such systems are designed to provide, imposing challenging design constraints for incentive mechanisms to discourage free-riding. We argue that these constraints are well addressed by requiring currency to be exchanged in return for service. Based on this idea, we propose a novel technique to allow anonymous digital cash payments to be made to those who provide service. We incorporate this technique into a class of peer-peer anonymous protocols that are based on Chaumian mixes while introducing modest message delay overheads and preserving their architectural simplicity. Finally, we formulate an abstract model of self-interested users in such a system and show that a payment based incentive mechanism can significantly improve the degree of anonymity by fostering greater cooperation among peers.

Keywords: anonymous communication, incentive mechanisms, digital cash

1 Introduction

The fundamental goal of anonymous communication is to disguise the identity of one or both parties involved in a bi-directional communication from each other and from any potential eavesdropper. This concept has been discussed in the literature for over two decades and has received attention from numerous researchers, who have targeted different aspects of such systems, ranging from formal definitions of anonymity [14, 20, 25] and practical communication protocols [12, 19, 20] to performance and service degradation under different malicious attacks [16, 27]. Some researchers have recently suggested building incentive mechanisms into such systems as a way to increase their robustness [10].

*This research has been supported in part by the NSF under grant awards ANI-0085848 and EIA-0080119 and by CAPES (Brazil).

†Obsoletes Computer Science Technical Report 03-21

As noted in [9], anonymity is a property of communication that cannot be provided by the sender (or receiver) alone. The sender must rely on one or more nodes that will cooperate to disguise its identity. A number of peer-to-peer anonymous systems have been proposed in the literature [12, 16, 19, 20, 21]. In such systems a group of peers collectively obscure the identity of a message initiator by forwarding the message randomly among themselves an arbitrary number of times before sending it to its intended recipient and returning the recipient's response along the reverse path. In such mechanisms, the message initiator is anonymous within the group of collaborating peers. That is, from the recipient's perspective, all group members are equally likely to have initiated the message. Furthermore, peers that forward messages cannot distinguish between the true message initiator and an intermediate peer along the forwarding path. Most of the anonymous protocols proposed in the literature are based on the mix-network concept initially introduced by Chaum [6], which will be described in Section 4.

An important property of anonymous protocols is the *degree of anonymity* they provide and their resistance to certain types of malicious attacks that attempt to break anonymity [14, 16, 20, 27]. These metrics are usually monotonic in the number of peers in the group; having more peers confers a higher degree of anonymity and higher resistance to malicious attacks.

Like many other peer-to-peer applications, anonymity systems are vulnerable to free-riders, nodes that consume the service without providing service to other nodes in the system. Free-ridership has been observed in many peer-to-peer systems. In file-sharing systems [2, 13], free-riders download files without making them available to other users. In ad hoc networks [4, 5], free-riders decline to forward packets for others while still expecting their own packets to be forwarded. In anonymity systems, free-riders join the system when they need to establish anonymous communications, provide service while they are joined, but then leave the system once their immediate needs are fulfilled. This behavior has the potential to undermine anonymity systems because of two undesirable consequences. First, the presence of free-riders tends to reduce the overall number of peers in the group at any particular point in time, which consequently reduces the degree of anonymity. Second, the frequent turnover in group membership caused by free-riders joining and leaving the system imposes a high group maintenance overhead and can facilitate certain types of malicious attacks [27].

For an anonymous protocol to be scalable and effective, users must behave cooperatively by providing service to each other; yet there certainly are reasons not to do so. There are clear costs and risks associated with committing local resources to an anonymous communication system, such as the costs of dedicating computer resources (e.g., network bandwidth, CPU cycles) and the risk of increased scrutiny due to participation. Notwithstanding numerous examples of dedicated participants in such systems acting altruistically, many (if not most) users are likely to behave selfishly when presented with such costs and risks by becoming free-riders. In systems where the quality of the service provided depends on the number of participating nodes (as in anonymity), providing an explicit incentive to remain joined to the service is of great importance.

In this work, we consider a novel technique that uses digital cash to provide explicit incentives to reduce free-ridership in peer-peer anonymous systems. The main contributions of this paper are:

- We propose two payment based mechanisms that use digital cash and are readily coupled with the operation of the class of anonymous systems that are based on Chaumian mixes (e.g., Tarzan [12], onion routing [19]). The key idea of these mechanisms is to provide the initiator the ability to embed, in each message sent, small anonymous payments destined to those peers who forward the message along its path. Peers that desire service, can either join the system and accumulate cash by providing service to others, or can purchase service with an infusion of cash into the system. We argue that the use of digital cash is well-suited for providing incentives as it preserves the fundamental anonymity properties and the architectural simplicity of anonymity systems.

- We formulate an optimization problem of self-interested peers that are subject to the costs of using the proposed payment based anonymity system and solve it to obtain the optimal fraction of time each peer remains joined to the system and the optimal price each is willing to pay per message sent. Using these results, we show that the incentive mechanisms proposed can significantly improve the degree of anonymity by reducing the amount of free-riding. Our results indicate that peers very sensitive to purchasing service can still use the system free of charge by providing service to others and accumulating revenue for their own demands.

The remainder of this paper is organized as follows. Section 2 provides an overview of the related work on incentives for cooperation. In Section 3 we discuss why a payment mechanism is suited for anonymous systems and provide a short background on digital cash. Section 4 presents our design for an incentive mechanism to be embedded in a mix-type anonymous protocol and discusses security and trust issues along with possible design variants to address them. In Section 5 we analyze the effectiveness of the incentive mechanism using a optimization model to capture peers' cost. Finally, Section 6 concludes the paper.

2 Related Work

A common approach to minimize the impact of free-ridership is to design some incentive mechanism into the functionality of the system. Much recent work in this area has focused on reputation mechanisms, whereby peers—individually or collaboratively—identify free-riders and punish them by declining service to peers with bad reputations [4, 22]. While reputation mechanisms are a promising approach to reducing the incentive to free-ride in many types of systems, they clearly require peers to know each other's identities. In anonymity communication protocols, however, the identity of a free-rider is obscured by the very service the system is designed to provide.

Despite the aforementioned issues, there have been efforts in applying reputation mechanisms to anonymous systems. In particular, Dingledine et. al ([9]) gives an overview of two different systems that were enhanced with reputation mechanisms. In the Remailer Networks system, a reputation mechanism based on cluster of nodes is used to provide a more *reliable* service to users but the authors do not address the issue of free-riding. The authors also describe their attempts to couple an anonymous publishing system (the Free-Haven project) with a reputation mechanism to prevent free-riding (publishing content without providing any reliable storage space). In both cases, the complexity and degree of centralization of the systems were significantly increased by the addition of reputation mechanisms. The difficulties encountered suggest that conventional reputation mechanisms might not be suitable to prevent free-ridership in anonymous systems.

It is important to understand the economic forces that drive users to participate in a given peer-peer system. A broad discussion of the economics (costs and benefits) behind anonymity systems has been discussed in [1]. The authors enumerate such economic forces and balance them in a model to understand the implications of user actions. Using a simplified model they show that under some circumstances the system is not feasible as the cost of anonymity exceeds its benefits. The authors also suggest that a payment-based incentive could provide an alternative to reputation mechanisms to cover the excess costs, but do not propose any scheme based on this observation.

The use of payments to promote cooperation has been proposed for peer-to-peer systems other than anonymity systems. MojoNation [3] was a deployed peer-to-peer network for robust file storage and retrieval in which peers traded a form of private currency called *mojo* in exchange for both the storage and retrieval of data. The main intent of requiring such an exchange was to limit any individual peer's ability to cause a denial of service by excessively consuming resources[1]. Buttyán and Hubaux advocate a payment scheme for promoting cooperation in mobile ad hoc networks [5]. The authors propose a system for exchanging a private currency for service and show that this incentive mechanism can push the system to an equilibrium point where peers cooperate. The implementation of their approach requires all participating peers to have a tamper-proof hardware to enforce honest exchange of payments. Zhong, Chen and Yang also propose a payment scheme called *Sprite*, to encourage cooperation in mobile ad hoc networks [28]. However, their system does not require the use of specialized hardware and instead, makes use of a centralized record keeping authority along with a cryptographic scheme for deferred payments. A game-theoretic analysis establishes that when prices are set appropriately, rational peers behave truthfully and cheating or collusion is never a better option. Crowcroft, et al. also consider a payment scheme for ad hoc networks, with a focus on the problem of setting individual prices for service in a distributed fashion [7].

Despite the fact that both ad hoc networks and anonymous communication protocols provide a data forwarding service, the two applications have different requirements and their respective users have different utility functions. In ad hoc networks, local battery power and bandwidth are scarce resources, while in anonymous systems this is less of an issue. In anonymity, one might fear the scrutiny that comes from helping others engage in dubious activity, while this is not an issue in ad hoc networks. Moreover, the primary system requirement of an anonymous system is to preserve the identity of a peer, whereas efficient data transmission is the goal in ad hoc networks. With this in mind, our work differs in many important respects from related work in ad hoc networks. In particular, we must ensure that the exchange of currency itself does not enable an attacker to compromise anonymity. The use of a centralized accounting should not reveal the identity of users involved in a transaction, neither to each other nor to the central authority. We have also chosen to reject any requirements for specialized hardware since this could constrain the scale of deployment and ultimately limit the number of participants (which is not desirable for anonymous protocols).

It is also instructive to contrast an anonymous communication system with an application like as SETI@home [17]. SETI@home performs a massively parallel computation by leveraging the CPU resources contributed by numerous volunteers, who receive little compensation (e.g., being listed as a heavy contributor) and apparently behave altruistically. Such systems are fundamentally different from anonymous peer-peer systems. Most importantly, neither cooperation nor communication among the participating nodes is required in order to contribute to the system as contribution consists solely of local computations. Therefore, the benefits perceived by each volunteer are largely unaffected by the actions of others, who may join and leave frequently or contribute very little of their time.[2] Thus, the very notion of free-rider does not exist, as either

[1] The idea of using currency to limit the power of attackers was also developed by Dailianas and collaborators [8].

[2] Although SETI@home has registered over 10^6 users, only about 10^3 users are active within a given 24 hour period (statistics

a user participates and contributes or not. In such systems, an explicit incentive structure would have little benefit.

3 Background: Digital Cash

One way a peer-to-peer system can encourage cooperative behavior is by requiring the exchange of a token in return for service. One can think of a token presented by the peer requesting service as evidence of its past cooperation. For such a mechanism to be robust against fraud, the token must be irrefutable, which typically requires that it be issued by a trusted third party, and peers must be prevented from reusing previously spent tokens. Furthermore tokens must be transferable in the following sense: a token presented to its issuer can be exchanged for a new token that the presenter may then use to acquire service. To ensure anonymity, it must be impossible for the token issuer to link any individual token use to the peer that initially requested that token. Mechanisms for issuing and exchanging such tokens already exist in the form of several digital cash schemes developed over the last decade. It is therefore reasonable to consider whether these mechanisms can be easily adopted to provide incentives in anonymous communication systems.

We propose an incentive mechanism that makes extensive use of digital cash and some of its fundamental properties. Digital cash is a mature field of research and still a promising practical idea, despite the failure of initial attempts to bring digital cash to the public domain [26]. A reason for this failure was the meager demand for digital cash attributable to the lack of applications that required its use. Although digital cash infrastructures do not currently exist in the public domain, systems that make extensive use of digital cash, such as the one proposed here, could increase the demand and accelerate its deployment.

We will assume the existence of a publicly accessible authority issuing digital cash, which we will call the *Bank*, and a digital cash mechanism with fairly standard properties.[3] Note that the *Bank* need not be a centralized entity nor be under single administration, as long as different entities recognize and value each others' currency. The most important property of the digital cash mechanism is that it renders the relationship between a payer and its purchases untraceable by either the Bank or by the payee. Ideally, a transaction should not even reveal partial information about the payer. Also, the digital cash mechanism should be implementable solely in software and not rely on tamper-proof hardware such as a smart card. Payments using digital cash can be either *off-line* or *on-line*. On-line payments require an interaction with the Bank for each transaction as it occurs. This allows the Bank to prevent malicious users from *double-spending*—repeatedly spending the same unit of currency with different payees. Off-line payments do not involve the Bank at the time of transaction and can only detect double-spending after the fact. Most off-line payment schemes provide a disincentive to double-spend by using cryptographic payment protocols which reveal the identity of the payer if a unit of currency is double-spent, but not if it is spent only once [23].[4] Off-line payment protocols require a challenge-response interaction between payee and payer but do not

in [17]).

[3] An alternative to general-use digital cash is to issue a private form of cash for use only within the anonymity system. We advocate using a public Bank primarily because of the difficulty of preventing side-exchanges of private cash and secondarily to reduce the complexity of the anonymity system by using an external issuer of cash. These issues are discussed in more detail in [11].

[4] It is reasonable to assume that among users of an anonymous communication protocol, such a disincentive would be particularly effective in preventing double-spending.

involve the Bank at the time of payment. We do not require other generally desirable properties of digital cash, like efficient transferability and divisibility, although such properties could potentially be exploited to make our scheme more efficient. Both the on-line and off-line digital cash systems that will be used in the incentive mechanism that follows have been fully designed and their description can be found in [23] and references therein.

4 System Design

Before describing the proposed incentive mechanism, we first review the basic operation of the mix network which was initially proposed by Chaum [6] and forms the basis of some anonymous protocols proposed (e.g., [12, 19]). The recursive encryption characteristic of Chaumian mixes is a fundamental requirement of our mechanism.

In a simplified mix network, a peer wishing to send an anonymous message (known as the *initiator*) to some destination D, constructs a path through a set of collaborating peers in the system. The last peer on this path is responsible for forwarding the message to its ultimate destination D. The initiator selects the intermediary peers of the path, possibly at random, and using their respective public keys constructs a message that is recursively encrypted and has the form:

$$O = S_1, \{S_2, \{S_3, \{\ldots, \{S_L, \{R, D\}_{K_L^+}\}_{K_{L-1}^+}\}\cdots\}_{K_2^+}\}_{K_1^+} \tag{1}$$

where S_i is the address of the i-th peer in the path, L is the path length, and $\{X\}_{K_i^+}$ denotes message X encrypted with public key K_i^+.[5] In onion routing ([19]), this recursively encrypted message is known as an *onion* and we adopt this nomenclature in the subsequent text.

After constructing, the onion, the initiator then forwards it to the first peer of the path, S_1. Each intermediary peer i in the path will have access to a payload after decrypting the message with its private key. The payload contains the address of the next hop S_{i+1} and an encrypted payload to be passed to this next hop. Eventually, the message reaches the last node in the path, which then forwards message R to destination D. The peers forward any response from D along the reverse path.

The essence of our proposed scheme is to embed a small digital cash payment in each hop of the anonymous path. The initiator includes a payment C_i for the i-th hop of the path within the encrypted payload destined for that hop. The inherent source-routing mechanism (initiator determining the anonymous path) provided by the above protocol is particularly well suited for integration with a payment mechanism, as the initiator can safely embed a payment for each hop in the path. We consider two possible extensions of anonymous protocols to support digital cash payments—one using on-line payment and the other using off-line payment (see Section 3).

[5] For clarity of presentation, we present simplified encryption for the onion. Real protocols rely on public keys only initially to distribute symmetric keys, which are used thereafter.

4.1 Protocol With On-Line Payments

In the on-line protocol, each intermediate node must contact the Bank to verify the validity of its payment. This determines whether the received cash payment has been previously spent. To discourage double spending, each peer should validate its payment before returning any response messages along the reverse path, buffering data if necessary.

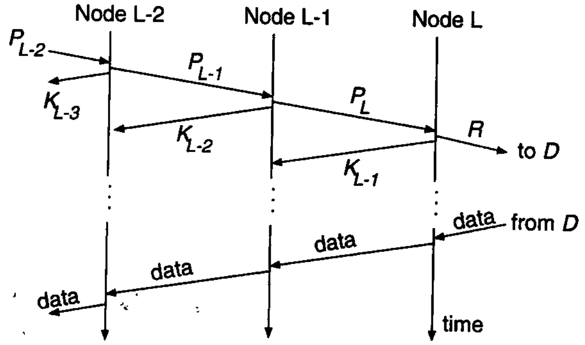

Figure 1: Message exchange for the last three hops of a path in the on-line protocol. Not shown in the diagram is an interaction with the Bank for each node to claim its payment.

To prevent intermediate nodes from receiving payment without providing service, an acknowledgment mechanism is used to finalize the payment only after the payload has been properly forwarded. The payment for node i is encrypted with a symmetric key generated by the initiator and accessible only to node $i+1$. On receiving a message, node i sends the encrypted symmetric key to its predecessor in an acknowledgment message enabling the previous hop to obtain its payment. On receiving an acknowledgment from its successor, node i decrypts the key and its own payment C_i and contacts the Bank to validate it. Figure 1 illustrates the operation of the on-line protocol.

The payload received by node i is

$$P_i = \{S_{i+1}, P_{i+1}, \{C_i\}_{K_i}, \{K_{i-1}\}_{K^+_{i-1}}\}_{K^+_i} \tag{2}$$

K_{i-1} is a symmetric key and the notation $\{X\}_K$ denotes a message X encrypted with symmetric key K.

The last peer in the path forwards the unencrypted request R to destination D. Since D is not assumed to participate in the anonymous protocol, the key to its payment is provided in the message payload to this final peer. Thus, the last payload is given by:

$$P_L = \{D, R, \{C_L\}_{K_L}, K_L, \{K_{L-1}\}_{K^+_{L-1}}\}_{K^+_L} \tag{3}$$

Clearly, in this protocol the initiator must trust the last peer in the path to correctly forward the message to the final destination. Moreover, if a message generates responses from the destination, the last node and all other nodes in the path should forward the reply along the reverse path to the initiator. We discuss in more detail both issues of trust and response messages in Sections 4.4 and 4.5, respectively.

4.2 Protocol With Off-Line Payments

The off-line protocol differs from the on-line version mainly in how a node interacts with the Bank. Instead of interacting with the Bank every time a payment is received, a node can accumulate payments and redeem them in batch at a later point in time, for example, when the node is idle or about to leave the system. However, recall that the detection of double-spending using off-line digital cash requires the payee to issue a challenge to the payer. Since the payer must remain anonymous we provide a mechanism to forward the challenge from intermediate nodes back to the initiator along the reverse path. Challenges are suitably encrypted so that only the initiator can read them.

After forwarding the message to destination D, the last hop creates a message containing an encrypted challenge $\{q_L\}_{K_L}$ and sends it along the reverse path. Note that the key used to encrypt the challenge is the symmetric key that was generated by the initiator to encrypt the payment for this node. Each node i on the reverse path appends its encrypted challenge $\{q_i\}_{K_i}$ as this message travels toward the initiator, who thus receives the following message containing encrypted challenges from all intermediate nodes:

$$\{q_L\}_{K_L}\{q_{L-1}\}_{K_{L-1}}\cdots\{q_1\}_{K_1}$$

After receiving the message with the challenges, the initiator constructs an onion containing a response r_i for each hop. When the final hop on the path receives its response r_L, it can then forward any response that might have originated at the destination back towards the sender. Note that the final hop on the path will immediately send the message to the destination but should buffer any reply until r_L is received from the initiator.

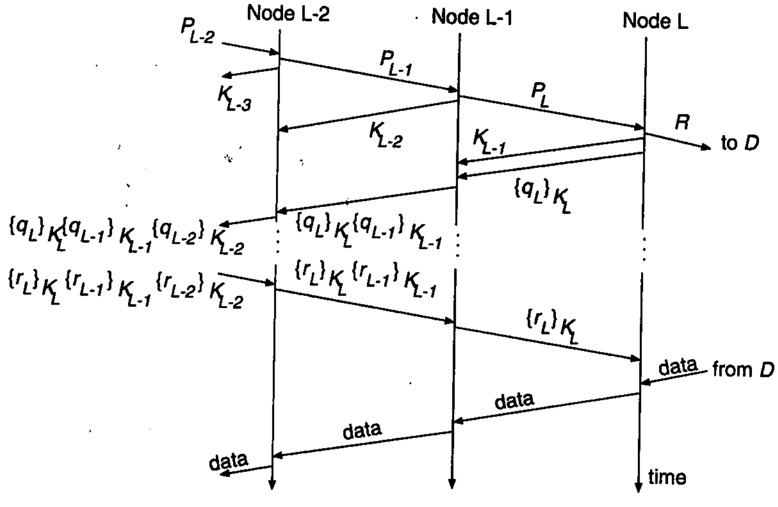
Node L-2
Node L-1
Node L
P_{L-2}
P_{L-1}
P_L
K_{L-3}
K_{L-2}
K_{L-1}
R
to D
$\{q_L\}K_L$
$\{q_L\}K_L\{q_{L-1}\}K_{L-1}\{q_{L-2}\}K_{L-2}$
$\{q_L\}K_L\{q_{L-1}\}K_{L-1}$
$\{r_L\}K_L\{r_{L-1}\}K_{L-1}\{r_{L-2}\}K_{L-2}$
$\{r_L\}K_L\{r_{L-1}\}K_{L-1}$
$\{r_L\}K_L$
data
from D
data
data
data
time

Figure 2: Message exchange for the last three hops of a path in the off-line protocol.

Like the on-line version, the off-line protocol also uses encrypted cash with encrypted keys that are returned through acknowledgments so that nodes must forward requests in order to receive their payment. Figure 2 illustrates the operation of the off-line protocol. The off-line protocol has a few advantages, as it can potentially reduce the overhead associated with Bank interactions and also reduce the ability of the Bank to break anonymity. These issues will be discussed in the following sections.

4.3 Attacks on Anonymity

It is important to understand what new attacks on anonymity and on the system itself can be introduced by a payment mechanism such as the one proposed above. Clearly, the Bank is a new potential attacker and collusions between peers and the Bank becomes possible.

We argue that the Bank alone will not trivially be able to reveal the identity of the initiator. The anonymous property of digital cash will prevent the Bank from linking a given cash payment to the peer to which that cash was issued. However, a more sophisticated Bank might use traffic analysis to correlate the issue of digital cash to a given peer with its later redemption by other peers. The on-line version of the protocol is particularly vulnerable to such an attack, as peers receiving payment must immediately contact the Bank. To counteract traffic analysis, an initiator might buy large sums of digital cash (in small cash units) infrequently, as opposed to requesting the Bank to issue digital cash on a per message time-scale. The off-line protocol is less subject to such an attack as peers can exchange the payments received at any point in time, possibly even randomly and infrequently.

Furthermore, even if the Bank colludes with one or more nodes the initiator is not trivially revealed. Since the Bank cannot add any definite information to what colluding nodes might already know, at best they can join efforts in performing traffic analysis. In this case, traffic analysis can be stronger as more events might be correlated, but again, we could counteract using similar ideas as above. We also envision the Bank to be a public accessible authority providing digital cash services also to various other entities, making traffic analysis of payments much harder, as transactions from the anonymous protocol will be interleaved with uncorrelated transactions.

The off-line protocol is subject to the following attack if the initiator is not careful. In order to prevent double-spending of digital cash, the identity of the initiator will be revealed if the initiator provides a response to two different challenges for the same digital cash payment. However, a careful initiator cannot be tricked into providing two responses to two different challenges for the same digital cash and can safely issue only one response per digital cash payment.

If our mechanism indeed provides the necessary incentives to reduce the number of free-riders (see Section 5), then known attacks on anonymity will become harder. Increasing the average number of peers in the system and reducing the average turnover rate will increase the degree of anonymity and provide for a more resistant system. For example, consider both the predecessor and intersection attacks on anonymous systems [27]. Both attacks take advantage of the time varying membership of the system and become more effective with a smaller average number of peers and with a higher turnover rate [27]. By providing an incentive to participate in the system, both attacks become less effective as the rate of path establishment decreases (e.g., fewer broken paths) and the group membership changes more slowly.

One could ask if a free-rider is more subject to such attacks than a peer that always cooperates. In this case, there would already be an implicit incentive for a free-rider to cooperate, since it would be protecting its own identity. However, it is not clear that this is the case; a peer is not more or less subject to such attacks if it unilaterally decides to free-ride. The increased resistance to these attacks requires the peers to collectively make such a decision, and an explicit incentive must be present in order to drive the system in this direction.

Finally, by introducing a payment-based system we allow malicious users to attempt to exploit the system in order to gain money. New attacks designed to obtain digital cash from the system can emerge. Although we are not aware of any trivial and effective attack, it is possible that by colluding with each other, a set of malicious users could pose a threat to the system. An observation that inherently limits the effectiveness of such attacks is the fact the anonymous path is defined solely by the initiator, reducing the chances of colluding nodes appearing in the same path.

4.4 Whom to Trust and Malicious Behavior

The problem of exchanging digital products (e.g., digital cash for message forwarding) between two parties that do not necessarily trust each other has been widely studied over the years. Protocols that guarantee the success of such transactions are known as fair exchange protocols and a number of them have been proposed in the literature [15, 18]. However, efficient protocols that avoid after-the-fact disputes rely on a common trusted third party. Protocols that do not use a trusted third party can have a high communication overhead or unreasonable assumptions (such as requiring the parties to have identical computing power), limiting their practicality [18]. Moreover, fair exchange protocols are even more complicated if anonymity is required among the parties.

Although the system proposed above could potentially make use of an existing fair exchange protocol that provides anonymity among the parties, we choose instead, for the sake of system simplicity, to accept a certain amount of inherent distrust among the parties involved in a transaction. Distrust can be tolerated because a peer that has participated in an unfair transaction can retaliate against the misbehaving party, as we explain below. The possibility of retaliation will motivate peers to behave fairly when engaging in a transaction. However, it could be possible that stronger punishment mechanisms for misbehaving nodes are needed; we leave this question for future investigations. A possibility would be to use a reputation mechanism, such as the one suggested in [4], to discriminate against nodes with bad behavior.

The protocol described above assumes that either the payer or the payee are to be trusted in a particular transaction. For example, the initiator is trusted to provide valid digital cash to intermediate peers, as a peer first forwards the message to only then receive a key for the payment. If a payment is invalid a peer can retaliate by refusing to forward subsequent requests for this path or refusing to forward response messages back to the initiator, causing the path to be broken and forcing the initiator to create a new path. An initiator that continuously misbehaves will not be able to receive responses from its destination and will have to create a new anonymous path for each request. However, the creation of a new path is known to reveal information about the initiator's identity (predecessor and intersection attacks [27]). Initiators, who presumably value anonymity, therefore have an incentive to behave honestly with respect to issuing payments.

The trust relationship is inverted in the transaction between the initiator and the last peer in the path. In this case, the last peer is trusted to provide service, as it receives its payment prior to forwarding the message to its final destination. If the last hop misbehaves, the initiator can retaliate by declaring a broken path and creating a new one which might not include the last hop. A peer that continuously misbehaves will be less likely to appear in the paths that are formed, reducing the amount of money it receives from the system.

There is also a trust relationship between a peer and its next hop neighbor. A peer trusts its successor to return an acknowledgment with the key for its encrypted payment. A misbehaving neighbor might neglect to do so. However, since not much can be gained through this behavior (certainly no additional cash), a self-interested peer is unlikely to engage in this action. As a form of retaliation the peer that did not receive its key might stop forwarding requests and/or responses creating a broken path, which would force the initiator to create a new path. Again, the hope is that the misbehaving peer will eventually appear in fewer anonymous paths.

4.5 Paying for Responses

In our system design thus far, we have focused on providing a payment from the initiator to all intermediate nodes along the forward path—that is, the path taken by the message as it travels to the destination. We have implicitly assumed that while joined to the system, nodes behave cooperatively by forwarding responses from the destination along the reverse path. We must take care, however, that the payments on the forward path do not create an undesirable incentive to remain joined but to behave uncooperatively by collecting payments and not properly forwarding response messages. An initiator whose response message is dropped by some intermediate node would be unable to identify the individual node at fault but could declare the path to be broken and create a new path, perhaps entirely disjoint from the original one. However, it might be desirable to have further incentives to prevent nodes from dropping responses.

A reasonable idea is to consider splitting the payment between the forward and reverse paths. The initiator would then embed two encrypted payments for an intermediate node of the path giving the keys to the forward and reverse payments to the node's successor and predecessor, respectively. The successor returns the forward key in an acknowledgment to the request message, as described above, and the predecessor returns the reverse key in a similar way when the response message is passed back toward the initiator. Some additional complexity is needed to prevent a colluding neighbor from providing the reverse key before the response comes back.[6] The main drawbacks of this approach are: (i) the initiator must trust the last hop on the path to behave honestly by withholding this additional key required in the reverse path until the response is returned (besides forwarding the original request to the proper destination); (ii) the initiator cannot provide a payment that is proportional to the size of the response (at best, it might be able to estimate this value).

An alternative approach is for the initiator to embed only forward payments in the request and to provide reverse payments in a separate recursively encrypted messages after it has received the response, which perhaps can be piggybacked in the next request it sends. This obviously would allow the initiator to pay in proportion to the size of the response, but would also require all intermediate nodes to place even more trust the anonymous initiator, since now they must wait for payments after service has been provided. Of course, intermediate nodes can punish an initiator for non-payment by dropping subsequent requests and forcing the initiator to create a new path (which, as described above might not be in its best interest).

At present, we feel that the second approach outlined above (response payments in a separate message) is more appropriate for the following reasons: First, it provides the flexibility of having payments proportional to the size of the responses. Second, retaliation is potentially more effective against the initiator, as it jeopardizes its anonymity by having to create multiple paths. Third, to the extent that paths are long lasting, the opportunity to piggyback the payments is realistic and the overheard will be small. We intend to explore

[6] We will not describe this mechanism in detail here due to lack of space.

this topic of reverse path payments more fully in future work.

4.6 Performance and Delays

The payment mechanism proposed above will clearly introduce additional communication and computation overheads to an existing anonymous system that will materialize in the form of delays. This additional overhead is present in every message generated and forwarded through the system. In the on-line protocol nodes along a path are now required to forward an acknowledgment to its predecessor, decrypt the payment and contact the Bank to deposit the digital cash. While in the off-line protocol, a node needs to generate and encrypt a challenge, forward it back to the initiator, receive and verify the corresponding response and, at a later point in time, contact the Bank.

If we assume a scenario where computation resources are plentiful, additional delays will be dominated by the transactions. In particular, in the on-line and off-line protocols the transactions between a node and the Bank and between a node and the initiator will, respectively, dominate the additional delays. However, note that both types of transactions can occur in parallel with waiting for a response message from the destination, masking the actual delay imposed by the transaction. Thus, we expect the perceived additional delay imposed by our protocols to be small.

The use of digital cash will also impose overheads as the Bank must be contacted in order to issue digital cash for the initiator, and issuing anonymous digital cash requires processing and the exchange of messages. To mitigate this delay nodes should buy digital cash in large batches well prior to establishing their own anonymous communication.

5 Incentive Model

In this section we present and evaluate a model for the incentive mechanism introduced in the previous section. The goal of this analysis is to provide a qualitative confirmation of our intuition that the proposed mechanism increases the level of cooperation in the system when users are sensitive to paying money or waiting to receive service. The approach taken here is to model each user of the system as a selfish agent interested in optimizing a function of a small number of local decision variables. Since the optimal values of these variables will depend on the values chosen by other users, the system equilibrium will be defined by a fixed-point at which all users have simultaneously optimized their local objectives. In this section, we establish the existence of a system equilibrium and evaluate the conditions under which the level of cooperation at equilibrium will be higher than what would be achieved with no incentive mechanism

Consider the anonymity system described in the previous section, where a user must embed a payment of amount q for each hop of the forwarding path. For simplicity, we will assume that all paths through the system are of a fixed length L, thus the total cost to send a message is Lq. Let N denote the total number of users that are willing to use the system. We assume that user $i, 1 \leq i \leq N$, generates messages at a fixed rate λ_r^i that must be delivered anonymously through the system. In order to send its messages anonymously, a peer must join the system for a minimum amount of time s, during which we assume it cooperates by forwarding traffic for others.

Each individual user i optimizes over two decision variables—its level of cooperation c_i and the amount p_i that it pays from external funds for each message sent through the system. Let s denote the time required for the system to deliver an anonymous message. Hence, we have that $s\lambda_r^i$ is the minimum fraction of time a user must be joined to the system in order to satisfy its own needs. Note that s is much smaller than $1/\lambda_r^i$, in particular, we assume $s\lambda_r^i$ to be much less than one. Thus, the level of cooperation c_i, which can be interpreted as the fraction of time the user is joined to the system, takes a value in the range $[s\lambda_r^i, 1]$. Note that c_i is bounded away from zero since we assume a user cooperates while it receives service. Although no assumptions are made with respect to the amount paid to send a message, there is no rational reason for a user to pay more than Lq—the total cost to send a message through the system.

As with the system defined in Section 4, users accumulate revenue while joined by forwarding messages generated by other peers. Since we assume that a user must necessarily send all anonymous messages it generates, the rate of cash injected into the system by user i is $Lq\lambda_r^i$. Assuming that peers on an anonymous path are chosen uniformly at random by the initiator, the rate at which a user accumulates cash by forwarding messages is given by the aggregate rate of cash injected by all users divided by the average number of users that are joined to the system. Let λ_c^i be the rate at which user i that is joined obtains revenue from the system. Thus,

$$\lambda_c^i = \frac{Lq\sum_{j=1,j\neq i}^{N}\lambda_r^j}{\sum_{j=1,j\neq i}^{N} c_j}. \tag{4}$$

Because a user only accumulates revenue while it is joined to the system, the long term rate at which user i accumulates revenue is $\lambda_c^i c_i$.

Of the total amount Lq of revenue required to send a message, user i will contribute p_i from external funds. The remaining balance $Lq - p_i$ must be collected by serving other users in the system. If the user has not yet collected this balance, then he/she will have to serve additional requests from other users in order to accumulate the necessary funds, thereby incurring a waiting time before his own message can be dispatched. The average time spent waiting per message sent will be denoted by $w(p_i, c_i)$. Observe that the waiting time is clearly a function of the local decision variables and rate parameter. For example, if a user is willing to pay the full price to send each of its anonymous messages ($p_i = Lq$), then his waiting time is zero. Similarly, if a peer is permanently joined to the system ($c_i = 1$) and has a relatively low message request rate (λ_r^i small), one might expect his waiting time to be near zero even if it is unwilling to pay anything ($p_i = 0$).

Each user then performs a local optimization to minimize the weighted sum of the costs involved in participating in the system. In particular, we consider the following three costs:

Level of cooperation: We assume that users suffer some cost for being cooperative for a variety of reasons ranging from the commitment of local resources for forwarding other users' traffic, to the increased risk of scrutiny incurred by participating in an anonymity preserving system. Because this cost is difficult to quantify, we model it simply as a linear function of the decision variable c_i.

Net cash flow rate: Users naturally value money, so there is a cost associated with paying money in order for a user to send messages anonymously through the system. Because users are both paying out and receiving money, each user will see a net cash flow rate, $\lambda_r^i p_i - \lambda_c^i c_i$, which we will treat as a cost.[7]

[7] Note that if a user receives money at a higher rate than it spends, the net cash flow will take a negative value.

Average waiting time: Since users are also sensitive to delays, there is a cost associated with the average waiting time $w(p_i, c_i)$ before a message can be sent. An analytical expression for the average waiting time can be determined in several ways; we consider a queueing model that captures the user's behavior within our system.

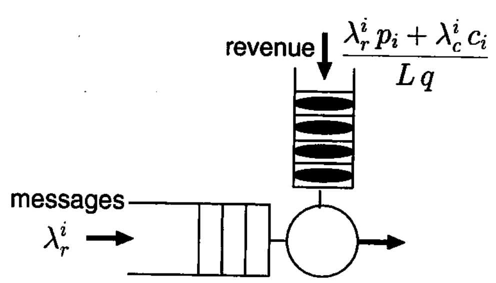

Figure 3: Leaky bucket—a model for the waiting time of a given user before it can send out an anonymous message.

The interaction between anonymous messages generated by a user and the amount of revenue accumulated is well captured by what is known as a *leaky bucket* model [24], illustrated in Figure 3. A leaky bucket is composed of two separate queues that are used to store messages and tokens, respectively. When a message arrives to an empty message queue and a token is available in the token queue, the message is immediately dispatched and one token is consumed. If no tokens are present when a message arrives, the message has to wait until a token is generated and only then is it dispatched. Both messages and tokens are generated according to fixed rates. In our case, messages represent anonymous messages generated by users at rate λ_r^i. Tokens map directly to revenue accumulated by the user, which is generated from his own payment and from being joined to the system. The overall rate at which a user accumulates revenue is given by $(\lambda_r^i p_i + \lambda_c^i c_i)/(Lq)$, which is normalized to match the cost of sending a single anonymous message. Assuming that both queues have infinite storage capacity and that both messages and tokens arrive according to Poisson processes, the waiting time of a message in the leaky bucket before it is dispatched, can be approximated by the waiting time of an M/M/1 queueing system [24]. Using a well-known result for the M/M/1 queue, the average waiting time for user i, is:

$$w(p_i, c_i) = \begin{cases} 0 & \text{if } p_i \geq Lq \\ 1/\left(\frac{\lambda_r^i p_i + \lambda_c^i c_i}{Lq} - \lambda_r^i\right) & \text{otherwise} \end{cases} \qquad (5)$$

Note that we explicitly model the fact the waiting time is zero when $p_i \geq Lq$.

A necessary condition for stability in the leaky bucket model above is that the token arrival rate must be greater than the message arrival rate. This condition makes intuitive sense, since failure to satisfy it would mean that the user is unable to send all of his messages.[8]

In a diverse group of users, we expect each user to have different sensitivities to each of the costs enumerated above. For example, one user might be very sensitive to paying to send its anonymous messages while another might be more concerned with the waiting time. To capture the heterogeneity among users, we assign user-dependent weights α_i, β_i and γ_i to each of the costs.

[8] An alternate modeling option here would be to allow the user to reduce its anonymous message rate to match available revenue. We do not consider this option here.

Having laid the necessary groundwork, we may now write the local optimization for user i

$$\min_{p_i, c_i} \ \alpha_i w(p_i, c_i) + \beta_i(\lambda_r^i p_i - \lambda_c^i c_i) + \gamma_i c_i \tag{6}$$

subject to

$$\lambda_r^i p_i \textstyle\sum_{j=1}^N c_j - \lambda_r^i L q \sum_{j=1}^N c_j + c_i L q \sum_{j=1}^N \lambda_r^j > 0 \tag{7}$$

$$s\lambda_r^i \leq c_i \leq 1, \ \ 0 \leq p_i. \tag{8}$$

The feasible region for this optimization is defined by the leaky bucket stability constraint, which is expanded in relation (7), and by the bounds on the decision variables (8). It can be shown that the objective function (6) is convex within this feasible region and that the feasible region is itself a convex set. By these convexity properties, we can conclude that there is a unique feasible optimal solution for each user. However, due to the nonlinearity of constraint (7) and the objective (6) (note the product $p_i \, c_i$), solving this optimization problem analytically is challenging and we therefore rely on numerical solution methods. However, a simple approximation obtained by ignoring the nonlinear term could be considered in targeting an analytical solution, although we leave this for future investigations.

5.1 Solving the model

To make our model tractable, we aggregate users into a fixed number of classes, M, where users within each class have identical behavior. (Note that subscript i will now denote a class and not a user.) We will also assume that each class contains an identical number of users, although this assumption can readily be relaxed by associating weights with the different classes. Moreover, we assume that the number of users in each class is large enough such that we can approximate λ_c^i by ignoring its dependence on a particular user i. Thus, all users accumulates revenue from the system at the same rate, λ_c. Using a numerical solution technique, we can solve the resulting fixed point problem by solving the local optimization problem for each class sequentially and iterating until the solutions for all classes stabilize. This stable set of decision variables defines the system equilibrium.

We consider two classes of users, $M = 2$, where users in class 1 are very sensitive to paying but are relatively insensitive to waiting, meaning that payments from external funds have more value than waiting for service or remaining joined to the system. This behavior is modeled accordingly by adjusting the weights that balance these different costs; a larger weight value indicates higher sensitivity. Users in class 2 have the opposite behavior and are very sensitive to waiting for service while being more willing to pay for service. In the case study that follows, both classes are equally sensitive to remaining joined to the system.[9]. Using these two classes, we investigate how the system equilibrium given by the optimal choices for the price paid per message $p^* = \{p_1^*, p_2^*\}$ and for the level of cooperation $c^* = \{c_1^*, c_2^*\}$, will differ between each class under different demands for anonymous message (λ_r^1 and λ_r^2).

We start by inspecting the system equilibrium as a function of the fraction of demand generated by the two classes. Figure 4 illustrates both p^* and c^* when the total message demand is kept constant ($\lambda_r^1 + \lambda_r^2 = 2$), but the fraction of the demand generated by each class varies (x-axis denotes the fraction generated by class

[9] Parameters for the model are: $\alpha_1 = 5$, $\beta_1 = 50$, $\gamma_1 = 4$, $\alpha_2 = 50$, $\beta_2 = 5$, $\gamma_2 = 5$, $L = 10$, $q = 0.1$, $s\max\{\lambda_r^1, \lambda_r^2\} \leq 0.01$

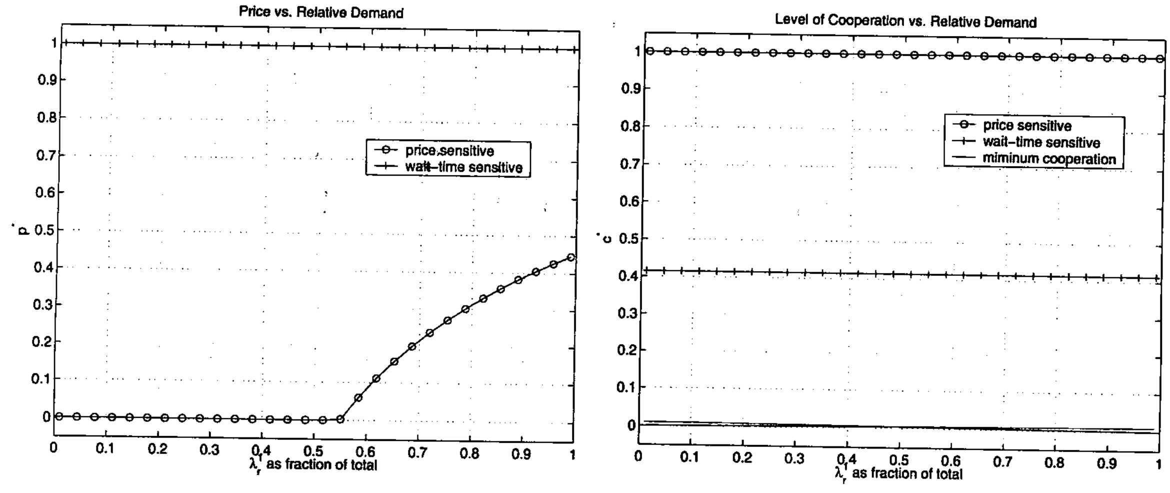

Figure 4: Optimal prices and levels of cooperation for classes 1 and 2 as a function of the fraction of total demand for the two classes.

1). Note that users from class 2, who are sensitive to delays and less sensitive to payments, will always pay the full price to send their messages ($p^* = Lq = 1$), independent of the fraction of demand they generate. In contrast, users from class 1 will not pay if the fraction of messages they generate is below 0.55, and will only start paying if this fraction is larger.

Contrary to users from class 1 who remain joined to the system 100% of the time independent of the fraction of their demand, users from class 2 remain joined to the system for 40% of the time, as illustrated by Figure 4. Since class 2 peers have a non-negligible sensitivity in paying for service, a user can recover part of its cost by serving message requests from others by remaining joined to the system. An interesting observation is that the revenue provided by users that are more likely to free-ride (being insensitive to payments), can under certain conditions, enable price sensitive peers to use the service free of charge and even generate profit by remaining joined to the system.

Figure 5 illustrates the system equilibrium as a function of the total demand on the system, when the ratio λ_r^1/λ_r^2 is kept constant (equal to one). Note that both classes pay full price when the demand is low, as each class cannot collect enough revenue from the system to satisfy its own needs. As demand increases, eventually class 1 reduces its payments taking advantage of the willingness of class 2 to pay for immediate service. As demand increases even higher, class 2 also reduces its payment.

At the same time class 1 and 2 reduce their respective payments, they increase th fraction of time joined to the system, as illustrated by Figure 5. Note that both classes eventually remain joined to the system 100% of the time, collecting revenue for their own messages. However, in the current model, the optimal price paid by at least one of the classes when both class are fully cooperative is bounded away from zero (although it can be very small). A system where optimal prices can be exactly zero when nodes are willing to cooperate 100% of the time seems interesting and is the subject of future study.

The results above show that the incentive mechanism induces a much higher level of cooperation than the system would have otherwise (see the minimum level of cooperation in each figure). In particular, if

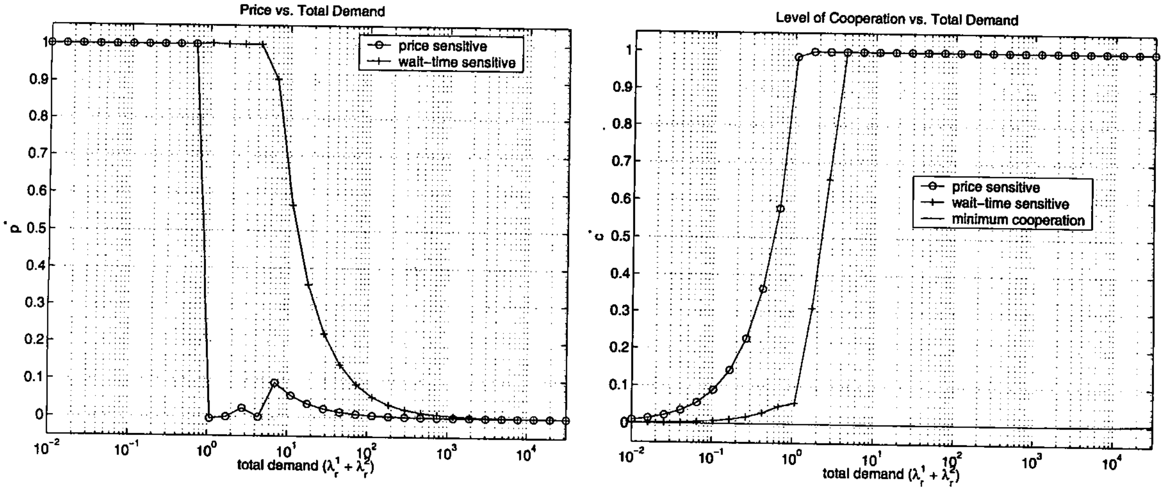

Figure 5: Optimal prices and levels of cooperation for classes 1 and 2 as a function of the total demand on the system. For these plots, the ratio for of demand for the two classes is fixed at $\lambda_r^1/\lambda_r^2 = 1$.

no incentive mechanism is adopted, a user would be joined solely for the time required to satisfy its own demand. This fraction of time is the ratio between s, the time to service a single request, and $1/\lambda_r^i$, the message interarrival time, hence $s\lambda_r^i$. In any system, $s\lambda_r^i$ must be less than one to ensure stability. However, in practical systems and under most user behavior we expect $s\lambda_r^i$ to be much smaller than one as s should be very small in an efficient system (in the above analysis, we set s such that $s\lambda_r^i$ is smaller than 0.01). Although these assumptions may vary, the results shown above illustrate that incentive mechanisms such as the one proposed, can provide a significant improvement on the level of cooperation by reducing the amount of free-riding, and hence, improving the degree of anonymity offered by the system.

6 Conclusion

This paper presented a novel payment based technique that makes use of digital cash to provide incentives for cooperation in peer-peer anonymous communication protocols. We propose two mechanisms that rely on on-line and off-line infrastructures for digital cash, respectively, that can be coupled with a class of existent anonymous protocols that are based on a mix-network ([12, 19]). The key idea of our mechanisms is to provide the initiator the ability to embed anonymous payments to those peers who perform forwarding services. We argue that our mechanisms preserve anonymity and the architectural simplicity the system. We also believe that the additional delay overhead introduced by our mechanism should be modest.

We formulate an abstract model of self-interested users that are subject to the costs of using a payment based anonymity system, such as the one proposed here. Using this model we demonstrated that the incentives provided can significantly improve the degree of anonymity by fostering greater cooperation among peers, that is, reducing the amount of free riding.

Although we show the existence of a system equilibrium using a centralized solution technique, it is not yet clear how users would achieve this equilibrium in a decentralized setting. In addition, we have assumed that the price paid and received for forwarding a message a single hop, q, is determined a priori, whereas in practice this price would likely be set by market mechanisms. An interesting question is to understand the impact that a fluctuating price hop price will have on the system equilibrium and how this can be introduced into the actual incentive mechanism.

References

[1] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the economics of anonymity. In *Proc. Seventh International Financial Cryptography Conference - FC03*, Jan 2003.

[2] Eytan Adar and Bernardo A. Huberman. Free riding on gnutella. *First Monday*, 5(10), 2000.

[3] B. Wilcox-O'Hearn. Experiences deploying a large-scale emergent network. In *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, MA, March 2002.

[4] S. Buchegger and J.-Y. Le Boudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. In *Proc. WiOpt'03 (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks)*, 2003.

[5] L. Buttyan and J. P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5), October 2003.

[6] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[7] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modelling incentives for collaboration in mobile ad hoc networks. In *Proc. of WiOpt'03*, 2003.

[8] Apostolos Dailianas. *Use of Currency for Access Control in Large-scale Information Systems*. PhD thesis, Columbia University, Dept. of Computer Science, 2000.

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Reputation in P2P anonymity systems. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, 2003.

[10] Roger Dingledine and Paul Syverson. Reliable mix cascade networks through reputation. In *Proc. Sixth International Financial Cryptography Conference - FC02*, Mar 2002.

[11] Daniel R. Figueiredo, Jonathan K. Shapiro, and Don Towsley. Incentives for cooperation in anonymity systems. Technical Report UM-CS-2003-021, University of Massachusets at Amherst, Dept. of Computer Science, 2003.

[12] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proc. of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[13] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for sharing in peer-to-peer networks. *Lecture Notes in Computer Science*, 2232, 2001.

[14] Yong Guan, Xinwen Fu, Riccardo Bettati, and Wei Zhao. An optimal strategy for anonymous communication protocols. In *Proc. 22nd IEEE International Conference on Distributed Computing Systems (ICDCS 2002)*, Jul 2002.

[15] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An intensive survey of non-repudiation protocols. *Computer Communications Journal*, 25(17):1606–1621, 2002.

[16] Brian N. Levine and Clay Shields. Hordes: A protocol for anonymous communication over the internet. *ACM Journal of Computer Security*, 10(3), 2002.

[17] SETI@home Project. http://setiathome.ssl.berkeley.edu, 1999–2003. Based at University of California at Berkeley.

[18] Indrajit Ray and Indrakshi Ray. Fair-exchange in e-commerce. *SIGecom Exchanges*, 3.2:9–17, 2002.

[19] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection*, 1998.

[20] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[21] Marc Rennhard and Bernhard Plattner. Introducing morphmix: Peer-to-peer based anonymous internet usage with collusion detection. In *Proc. of the Workshop on Privacy in the Electronic Society (WPES)*, Washington, DC, November 2002.

[22] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[23] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, 2nd edition, 1996.

[24] Mischa Schwartz. *Broadband Integrated Networks*. Prentice-Hall Inc., 1996.

[25] Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In *Formal Methods' 99*, Lecture Notes in Computer Science 1708, pages 814 – 833. Springer-Verlag, 1999.

[26] Peter Wayner. Electronic cash for the net fails to catch on. *The New York Times*, November 28 1998.

[27] Matt Wright, Micah Adler, Brian N. Levine, and Clay Shields. Defending anonymous communication against passive logging attacks. In *Proc. of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.

[28] S. Zhong, J. Chen, and Y. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *Proc. of Infocom 2003*, 2003.