GNUnet – An Excess Based Economy

Christian Grothoff

Department of Computer Sciences, Purdue University
grothoff@cs.purdue.edu
http://www.gnu.org/software/GNUnet/

Abstract. This paper describes economic aspects of GNUnet, a peer-to-peer framework for anonymous distributed file-sharing. GNUnet is completely decentralized, all nodes are equal peers. In particular, there are no trusted entities in the network. This paper describes an economic model to perform resource allocation and defend against malicious participants in this situation.
Accountability is generally considered hard for peer-to-peer networks that lack trusted entities [10, 9]. Our approach does not use credentials or payments but is based on trust. The design is close to game theory [5] with peers taking the role of players. Nodes must cooperate to achieve common goals. In such a scenario, it is important to distinguish friendly from malicious nodes, even if the separation may not always be clear.
GNUnet aims to provide anonymity for its users. The design makes it hard to link an action to the node where it originated from. While anonymity requirements make a *global* view of the end-points of a transaction infeasible, the *local* peer-to-peer transactions can be fully authenticated. Our economic model is thus based entirely on this local view of the network and takes only local decisions. Still, it is able to perform resource allocation and to defend against malicious participants that attack the network by using too many resources, making the network resistant to attacks.

1 Introduction

In this work we present a new approach toward resource allocation in an anonymous, decentralized peer-to-peer network. Resource allocation in distributed networks with untrusted hosts is a significant problem for peer-to-peer networks since it is difficult to establish which nodes are worth spending resources on. If sender and receiver are anonymous and therefore not known to intermediaries, it becomes even harder to bound the impact of malicious nodes.

We propose the abandonment of traditional monetary economic models in favor of a trust-based economy. An unrestricted global monetary economy has been criticized since it was first proposed by Smith [16] in the late 18th century. Monetary investments depend on a trusted central authority that guarantees the exchange value of money. Monetary schemes do not work for anonymous peer-to-peer networks. A trusted central authority violates the fundamantal principle that all peers are equal. Furthermore, in anonymous networks the endpoints of

a transaction must stay hidden. The economic model must be able to handle intermediaries that only have a limited, local view of their part of the communication.

In this paper we describe a trust-based economy. While the value of trust cannot be guaranteed, we show that the model ensures that it is in the best interest of all participants to uphold and honor its worth. The model guarantees that no node can gain anything by disobeying the protocol or the rules of the economy.

We have implemented our economic model in GNUnet, an anonymous distributed file-sharing network. GNUnet has the interesting property that all transfers are very small and uniformly sized. GNUnet is also tolerant to packet loss; the resource allocation code can just decide to drop requests if the local node gets too busy. The fine-grained and load-balancing nature of GNUnet also allows for the allocation code to make smooth transitions.

The model we describe is much simpler than schemes for digital cash. Link-to-link authentication is the only cryptographic operation needed by the economic model. Stepping back from monetary schemes to this simpler trust model allows us to fulfill the requirements of a pure peer-to-peer network.

In this paper, we focus on GNUnet's economic model. We assume that the GNUnet framework provides authenticated communications and is able to associate identities with nodes. Every host can create a new identity at any time, but it is impossible for a node to forge the identity of another node. Furthermore, we assume that there are two kinds of interactions of concern: requests and replies. All requests and all replies are of identical size; thus, we can assume all replies require an identical amount of work. Other interactions, such as the propagation of routing information, are assumed to be irrelevant for accounting purposes since the payload should dominate the load in any reasonable network.

All collaborative distributed systems are exposed to the Internet and thus subject to attacks and abuse [1]. Most currently, used systems do not monitor host behavior. This allows nodes to use the distributed network without contributing back to it. Thus, these networks become susceptible to denial-of-service attacks. While this problem does not stop these networks from attracting millions of users, it does degrade performance. It also discourages commercial use of these systems, as there is no incentive to invest.

The main requirement for the GNUnet economy is to prevent the abuse of the network. The economic system is supposed to detect nodes that use the network without contribution and limit their impact while giving preference to nodes that do contribute. While trust in the GNUnet economy may potentially be mapped back to real-world currencies, the primary purpose is geared more towards resource allocation than actual payment.

GNUnet is designed as an anonymous network. While each node has an identity, the economic model must not require knowledge about the end-to-end identities of a transaction. Each transfer in GNUnet may involve a large number of peer-to-peer intermediaries. In order to guarantee anonymity, no intermediary

may have any knowledge about the ultimate sender or receiver of a transaction [3].

Thus, the economic model must be based entirely on the link-to-link transactions that occur as a part of the actual end-to-end transfer. All economic decisions must be based on this local view of the world. While intermediaries may not know the identities of the actual entities that initiated the transaction, they must be protected by the economy against abuse by any constellation of adversaries.

In GNUnet it must be possible for nodes to join or leave the network at any time. Furthermore, new nodes cannot be expected to perform any kind of key exchange, authentication or registration with a central authority. Joining GNUnet should only require the establishment of a link-to-link encrypted connection with an *arbitrary* node on the network.

No node in the network should ever be trusted. In particular, there cannot be any kind of "bank" or other form of credit authority. For the economic model, nodes can only trust their own records. These restrictions are necessary, as GNUnet is an open system where anyone can join no one is supposed to trust any kind of central authority. As far as we know, this restriction makes it impossible to use any of the known forms of digital cash.

The design of the Internet does not allow any protocol to guarantee that malicious hosts will not bombard a host with traffic. This is usually not a problem, however, as an attack of this kind requires the adversary to have more bandwidth than the victim can sustain.

Some distributed networks cannot cope with simple flooding because the network allows the adversary to cause significantly more traffic than the adversary could generate on its own. This is possible whenever some node in a network forwards a query. Sending a request from a malicious node to even one more node is already sufficient to double the amount of traffic on the network; if each node forwards a given request to n other nodes until the tree formed by the propagation of transactions reaches some depth d (a typical behavior for gnutella), an adversary can use this to multiply the effect of his attack by a factor of n^d, even if no replies are sent.

Thus, one of the main goals of GNUnet's economic model is to make it impossible for adversaries to use the multiplier effect (which is inherent in the search process) to attack the network.

2 Related Work

Anonymity have been part of the research on digital cash since the very beginning. The problem with all exisiting digital cash systems is that they require the existance of a bank, which is a trusted authority that has a secret key that is used to create cash [6, 14]. The requirements of an anonymous peer-to-peer system make all of these schemes impossible to use in our setting.

Many designs concerned with anonymity have either no provisions for accountability [7, 13, 15] or use ridiculous stop-gap measures like the 100k per file

limit (but no limit on the number of files) as in [17]. Our scheme for anonymity is described and compared to these designs in [3].

The best-known example for a system that uses a variant of the digital cash systems to protect a peer-to-peer network is probably Mojo Nation [12]. Mojo Nation is a distributed file sharing system in which hosts need to provide bandwidth and drive space to earn *Mojo*. *Mojo* can then be used to request services from other hosts. This credit system protects the network against *freeloaders*, nodes that use the network but do not contribute to it. While some of the design goals are similar to GNUnet's, Mojo Nation relies on a central authority to govern the use of *Mojo*.

The only other system with design goals that are similar to GNUnet (anonymity, accounting and decentralization) is Free Haven [8]. Free Haven is a censorship-resistant storage that uses a "buddy system" to detect if servers fail to keep their promise to store a certain document. While the buddy system can fail if hosts conspire, this system charges for storage, whereas GNUnet charges for requests. In our opinion, it is much better to reward hosts that make content available than to make them pay. In [10] the authors of Free Haven discuss various payment schemes, but they fail to produce "any clear solution for an environment in which we want to maintain strong anonymity".

An interesting solution for certain attacks that involve an adversary sending large numbers of requests is Hash Cash [2]. In this design, a host that wants to initiate a transaction (e.g. send E-mail) must perform an expensive computation for the receiver in order to have the request processed. The original goal of Hash Cash was to fight spam. The approach has problems with transitivity and applications which create legitimate large amounts of traffic (e.g. mailinglists). The high computational demands of this approach are another potential issue, particularly since these demands must be kept high with respect to the current technology in order to keep the Hash Cash meaningful.

3 Trust Economy

The economic model in GNUnet departs from the traditional schemes involving money. In the capitalistic world, each entity *owns* certain assets. These assets can be converted, exchanged or donated, but they are always under the *control* of the entity that owns them. Any concept that involves money requires that money always belongs to one entity, can be transferred between entities and cannot be created by anyone except for a trusted authority.

While this works to some extend in the capitalistic world, the concept of money cannot be applied to an anonymous, distributed network without any authorities. The main problem is that without trusted authorities, money cannot be created. The problems of a monetary economy date back to the mercantile system and were analyzed and brought to wide attention by Engels [11].

The automated exchange of digital cash in a peer-to-peer network also suffers from the problem that the receiver or the sender of money might be malicious and not deliver (or at least not deliver what was negotiated). This problem

arises because the remote operation of exchanging money for any other good is not atomic.

For these reasons, GNUnet does not use money. While some of the properties of money would be desirable for GNUnet, we will see that most are not required to meet the goal of allocating resources for entities that have contributed to the network and limiting the impact of attackers.

Trust

GNUnet's economy is based on trust. In this respect, GNUnet is very similar to human relationships – or players in a game with rules that make it non-obvious which players have compatible goals. A node in GNUnet trusts certain other nodes. The level of trust is measured as a non-negative integer. Each node keeps track of how much it trusts each of the other nodes it has had contact with in the network.

The first thing to notice with this setup is that no node *owns* trust. The amount of trust a node has earned is stored at the other nodes. The cultural difference in comparison to the money-based economy is even greater as this implies that the *other* nodes are in control of the trust that a node has earned, not the node itself.

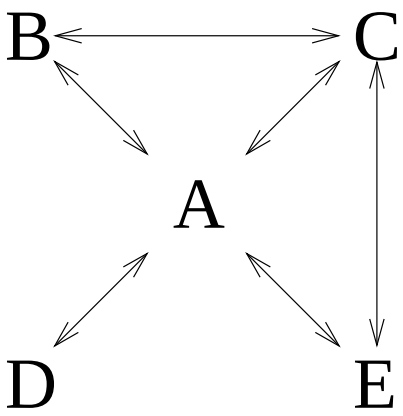

Fig. 1. Trust in a small GNUnet.

Node	A	B	C	D	E
A	∞	100	3	0	42
B	0	∞	0	-	-
C	5	200	∞	-	2
D	1	-	7	∞	-
E	140	-	3	-	∞

Trust is not symmetric and nodes that have never communicated have no opinion about one another. We prefer the word *trust* over *reputation* because reputations are usually transitive. Trust does not have this property. If node A trusts node B this does not imply that if node E trusts node A, E will also trust B. Trust is also different from *faith* in that it is based on actual, concrete experiences. Note that this does not exclude the possibility of nodes gambling, that is making a choice without clear evidence that the choice is going to be correct. Gambling can still be a reasonable strategy if the available information based on trust is insufficient to make a decision.

3.1 Building Trust

In GNUnet, each node evaluates the behavior of the other nodes that it communicates with and forms its own opinion. Nodes form their opinions about other nodes based upon the two basic protocol primitives that exist in GNUnet: queries, and replies.

A query is considered network usage, and nodes that send huge amounts of queries will lose trust. A reply is considered to be a contribution to the network. Nodes that send replies earn trust.

Each query in GNUnet comes with a *priority*. This priority defines the amount of trust that the sender node S is willing to risk for this query. The receiver R of the query can reduce the amount of trust that R had in S by that amount. If the receiver can answer the query, the sender S of the query will increase its trust in the receiver R by the priority of the query.

Thus, if S sends a query with priority 10 and receives an answer from R, the trust that S has in R will be increased by 10 whereas the trust that R has in S will be decreased by 10. The total difference in trust between the two nodes would thus be 20.

New nodes that join the network start as *untrusted*. The reason for this is that the creation of a new identity only requires the creation of a new public key pair. If a host S sends a query with a priority that is higher than amount of trust t that the receiver R has in S, the priority is automatically bounded by t. This ensures that no host ever has anything to gain by creating a new identity.

3.2 Resource Allocation and Excess

Our main motivation for the introduction of an economic model is resource allocation. If a GNUnet node is too busy to process all requests, it must decide which requests to drop. A request with a lower priority is going to be less valuable for the node, thus a busy node will drop the requests with the lowest priority first.

This scheme has the desirable property that nodes that contributed to the network will receive better service than nodes that did not contribute. Furthermore, client applications will try to keep priorities as low as possible in order to avoid being charged unnecessarily. This makes it possible for GNUnet to allocate resources to the important requests.

The scheme described so far suffers from the intrinsic problem of infusing the network with trust. In order to solve this problem, we make use of a fundamental property that computers have had for some time: excess resources. Most of the time, computers and networks have excess resources, meaning that their capacity is not fully used. In this situation, resource allocation is trivial since all requests can be fulfilled. A node that uses the network at a time when there is an abundance of resources available cannot reduce the performance of the network. It does not matter whether or not the node contributes back to the network. As long as the network has excess resources, freeloading is *harmless*.

GNUnet nodes detect the current load at the local node and stop charging for service if the load is under a certain threshold. At this point, requests to that node will be answered without the node reducing the trust it has in the sender. However, the economic model is not entirely disabled. If a node receives replies, it still credits the sender for the reply. In this manner, this process infuses the network with trust.

It is essential in order for the economic model to work that it be possible to increase the trust of well-behaved nodes without decreasing the trust of other nodes by the same amount (i.e., no zero-sum in the trust economy). Otherwise, no node would ever trust any other node because no one has trust to start with.

3.3 Knowledge

An important question in this economic model arises from the previously mentioned fact that the trust that a node A has earned is stored on other nodes. For the sake for the argument, we assume the trust is stored at node B. The potential problem here is that node B could decrease its trust in A arbitrarily. Similarly, B could decide not to honor the trust it has in A and give its preference to a request from the new node C instead.

Resolving this dilemma requires a fundamental realization: in any peer-to-peer system, a node can never be guaranteed that a service that it has provided in the past will pay off in the future. The reason is that there simply are no guarantees. For example, the creditor could always just disappear. While universal credit schemes may allow the node to *cash in* elsewhere, these schemes can still need to answer the question of who pays for the node that joins, uses the network without contribution and then disappears. In an open network that allows this kind of harmless freeloading, there is always the chance that a node does not get paid for providing resources.

Thus, A can never be sure that the trust it has earned will *buy* it anything in the future (again, trust is not money). This still does not quite answer the question of why B would not reduce its trust in A. The real reason that keeps B from tampering with A's record is self-interest.

It is in B's best interest to have *knowledge* about A. Knowledge about A's performance in the past helps B to make good decisions. If B were to lose that knowledge, B's decisions would be less informed and thus potentially harmful for B. Note that it does not matter if B forgets, ignores or disregards its knowledge about A. All that matters is what B bases its decision on. What kind of decisions

does B have to make? The only important decision that B makes that depends upon its trust in A is whether it should drop a request from A or a request from C if B is so busy that it can only answer one of them.

Suppose A is a good host and has answered thousands of B's queries in the past, and B decides to purge its trust in A. Then, a new node[1] M starts to flood B with requests. B does its best to answer the queries from M, but can't really answer all of them. At this point, A decides to send an important query to B. B has no proper records of A's past, thus B cannot decide if it should keep answering queries from M or if it should answer A's query. B might drop A's query, missing a great opportunity to increase its own standing with A. When B later asks A about something, A may now have decided that B is a malicious node and prefer answering requests from C.

Obviously, if B arbitrarily increases its trust in another node, this will also have a bad impact on B's performance as B may increase its trust into an attacker and give preference to a node that is very unlikely to ever be beneficial for B.

Thus, if B is malicious (or better, stupid) and arbitrarily changes its trust in A, it will also have a bad effect on the performance of B. This is again very similar to human relationships in which knowing who is trustworthy is beneficial for both sides. The major difference is that it is much harder for humans to obtain a new identity and thus "negative" trust is also useful in human relationships.

3.4 Transitivity

A fundamental requirement for an economic model is transitivity. Without transitivity, the problem would be much easier but could not provide any anonymity.

The problem with transitivity is that a group of collaborating malicious hosts must not be able to trick a set of intermediaries into providing resources for free. Suppose the nodes have a constellation as shown in figure 2.

In this situation, node B receives a query from node A with priority 10. Suppose B decides to forward this query to C and D. Intuitively, B would reduce its trust in A by 10 and forward the query with priority 5 to C and D. The problem with this approach is, that if A, C and D conspire against B, they could now have C send a query with priority 10 to B (which may get forwarded to A and D), and then have D send a query with priority 10 which is forwarded to A and C. If each of the hosts A, C and D answers each of the queries, B will have the same trust in A, C and D than it had before A's initial request after this third round. Thus, A, C and D have used B's resources but not really contributed anything back.

While this behavior is acceptable as long as B is idle, B must be able to defend against this kind of scheme if A, C and D use this scheme to deplete B's

[1] Note that for the model it does never matter if actions are carried out by a single node or a large group of nodes. For the economy, there is no difference between one *large* node with huge bandwidth and many smaller nodes that have the same total bandwidth.

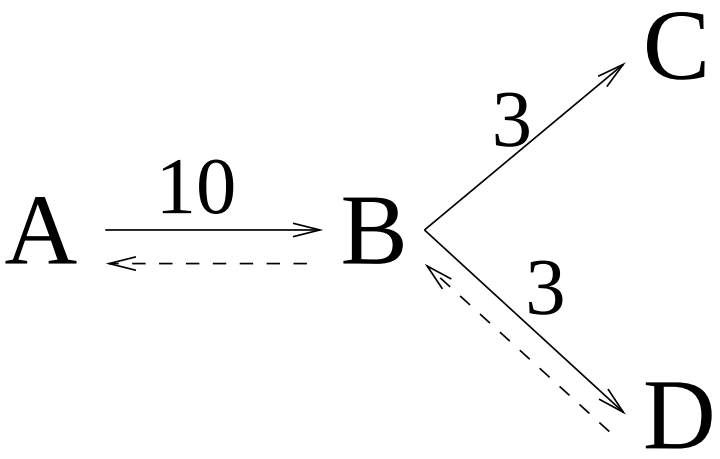

Fig. 2. Transitivity in the GNUnet economy.

resources. Our solution to this problem is to have B charge for its service by making the sum of the priorities of the forwarded queries lower (and not equal) to the priority of the query that was received. In this way, the trust that B has in A, C and D would degrade over time given the scenario above.

3.5 Under Attack!

GNUnet's economic model was launched with the goal of fending off attacks by malicious hosts that abuse the network, flood it with queries, or just make excessive use of it without contributing in return. We now evaluate to what extent GNUnet's trust-economy can live up to this expectation.

Suppose the attacker A has a bandwidth of c bytes available for the attack. In the internet, nothing can prevent A from throwing the entire c bytes at a network like GNUnet. Thus, if GNUnet's economy works optimally, the damage will be bounded by the processing of c bytes.

Furthermore, suppose the attacker A uses t bytes (where $t \geq 0$) to earn trust in GNUnet at some point, reducing the capacity available for the attack to $c - t$. Finally, suppose that GNUnet has ϵ bytes excess capacity in the network. In other words, ϵ bytes are available to build trust. This means that as long as A's impact is smaller than ϵ, the network's performance would not be affected at all.

The scoring system for trust described above bounds the amount of damage d that A's attack can cause as follows:

$$d \leq t + (c - t) + \epsilon = c + \epsilon \tag{1}$$

Thus the damage that A can do to the network is the capacity that A has on its own plus the excess bandwidth ϵ on the network. Since *epsilon* is by definition the amount of traffic that does not degrade GNUnet's performance, the effective

damage an attacker can do is bounded by the network capacity c available to the attacker. The economic model performs within ϵ of the optimum since no model can prevent A from sending c bytes of noise to the network.

4 Discussion

The economic model described so far is not able to solve all the resource allocation problems in GNUnet. In this section, we discuss potential problems and describe some open issues with our current model.

4.1 Implications for Anonymity

The trust scheme has only minimal implications for anonymity. Since the knowledge required by the scheme is purely local, the only trust-related information that is added to the peer-to-peer protocol is how much trust the sender node is offering for an answer. There are two cases in which an adversary can use that number to break anonymity.

In the first case, a node that offers an extraordinary amount of trust, the adversary might be able to tell that the node is the originator of the request since it might be unlikely that the node trusts anybody else enough to indirect a request with a priority this high. An implementation can thward this attack by limiting the trust offers to values commonly encountered. This should be done anyway since unreasonably high offers do not make any sense in the economic picture either. Why would a node want to offer significantly more for an item than what it is worth?

The second case is more subtle. If a node uses the importance of the requests to decide whether to process it at all, but does not apply this test to its own requests, the requests originating from the node may have a distinctively low priority compared to indirected requests. The solution is obvious, the node must apply the same rules (with respect to filtering low priority requests under load) to its own requests.

We conclude that a correct implementation of the trust based economy does not have any impact on the anonymity properties of the system since trust is purely local knowledge, does not leak discriminating information into the protocol and thus does not make it easier for an adversary to break anonymity.

The Zero Priority Problem

If content is migrated between nodes in GNUnet, nodes try to capture, store and serve content that is as valuable as possible. The most significant indicator for valuable content is a matching query with a high priority (from a host that had an adequate trust-level). The problem here is that if a node is idle, it will not charge for queries. In order not to be charged for the indirection itself, it will reduce the priority to zero when forwarding (the node cannot tell if the node that it forwards the query to is also idle).

The receiver of this zero-priority query now has two problems. First, the receiver node cannot earn any trust when answering that query. We assume that this is not an issue because if the node is also idle, it does not matter to the node if it answers the query or not. The situation in which one node is idle and another is very busy is unlikely if the network exhibits reasonable load-balancing characteristics. Furthermore, it is generally desireable to have nodes directly connected to many other nodes. In that case, there is a high chance that when an unsuccessful query is repeated, it will not take the same path with first an idle and then a busy node next time.

Content migration with zero-priority queries is another problem. As the priority of the query serves as an indicator for the value of the content, only nodes that have excess space will copy zero-priority content that floats by. All other nodes will decide that the content that they are storing has a higher priority and will merely indirect the reply without replicating the content. We currently do not have a good solution to this problem because we have to prevent a malicious node from using zero-priority queries which request useless data to manipulate the network into discarding more valuable data.

A trivial solution to the zero priority problem would be to charge a little bit for forwarding even when idle. The problem with this solution is that it then becomes harder for the system to infuse the network with trust.

Reply Verification

Forwarding queries and replies for other anonymous participants poses another problem. Suppose B forwards a query that was issued by A to C. B is generally not supposed to learn anything about either the query or the answer (see also the portions on deniability in [3]).

However, the economic model requires B to ensure that the reply from C is a valid answer to the query. Otherwise, C would be able to profit by replying to a query with extraneous data in order to earn credit. If B can not verify the validity of the reply, C could answer all queries with garbage and earn credibility while disrupting the system. Since only the initiator is supposed to be able to decrypt the reply, this situation would be very hard to handle with the scheme described so far.

In [4] we describe a scheme in which an intermediary is able to verify that a reply matches a query without being able to decrypt the reply. The scheme requires the responder to either have the content (or invert a one-way-function) in order to be able to prove that the responder actually knows the answer to the query. This makes it impossible for a malicious node to send back invalid data as an answer to arbitrary queries.

Beyond Query-Reply

The current model is geared towards the simple query-reply service that is used for the file-sharing application of GNUnet. For other applications, like E-mail, other types of requests will be required. Some of these applications may have

very different requirements for the economic model (e.g. the sender of the E-mail should probably be charged, and not the receiver).

We currently believe that the trust-economy can be extended toward this type of system if we can ensure that the fundamental rule of contribution increasing the odds of better service is preserved. Still, introducing a system like mail with much stronger reliability requirements is going to be an interesting challenge for the future.

5 Future Work

User feedback is the only real way to determine with any degree of certainty that content is actually valuable. This feedback is very hard to implement in an anonymous network, particularly because malicious hosts could lie. In the future we plan to propagate a user's evaluation of content back along the path the data originated from. Of course, back-propagation should be decided based upon the available bandwidth, the ranking of the nodes involved, and the evaluation of the pseudonym of the user. Since it is possible that the user who ranked the content is malicious, only content rankings from trusted nodes should be considered.

Feedback is a particulary tricky problem for the economic model as it can be either a valuable contribution or a malicous deception. Thus, it is unclear how to account for feedback.

Another possibility is to have users sign content that they insert into the network using pseudonyms and to use rankings for these pseudonyms to evaluate content rather than node-rankings. The primary problem with pseudonyms is that they may open the network to intersection attacks against anonymity (which nodes were online when content under this pseudonym was made available).

6 Conclusion

We have presented an accounting system that allows accountability without a central server in an anonymous network based on trust. The economic model prevents any host from harming the performance of the network by forcing hosts to contribute first. The excess capacity of the network is used to infuse trust into the system. Since trust is a purely local property with addition and subtraction as the only operations, the performance overhead is minimal.

References

1. E. Adar and B. Huberman. Free riding on gnutella. Technical report, Xerox Parc, Aug. 2000.
2. Adam Back. Hash cash - a denial of service counter-measure.
3. K. Bennett and C. Grothoff. gap - practical anonymous networking. 2002.
4. K. Bennett, C. Grothoff, T. Horozov, and I. Patrascu. Efficient sharing of encrypted data. In *Proceedings of ASCIP 2002*, 2002.

5. H. Bierman and L. Fernandez. *Game Theory with Economic Applications.* Addison-Wesley, 1993.
6. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash,.
7. I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability.* International Computer Science Institute, 2000.
8. R. Dingledine. The free haven project. Master's thesis, Massachusetts Institute of Technology, 2000.
9. R. Dingledine and P. Syverson. Open issues in the economics of anonymity, 2002.
10. Roger Dingledine, Michael J. Freedman, and David Molnar. *Accountability.* 2001.
11. Friedrich Engels. *Umrisse zu einer Kritik der Nationalökonomie.* 1844.
12. Mojo Nation. Technology overview, Feb. 2000.
13. Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
14. Ronald L. Rivest and Adi Shamir. Payword and micromint: Two simple micropayment schemes. In *Security Protocols Workshop*, pages 69–87, 1996.
15. R. Sherwood and B. Bhattacharjee. P5: A protocol for scalable anonymous communication. In *IEEE Symposium on Security and Privacy*, 2002.
16. Adam Smith. *Wealth of Nations.* 1776.
17. Marc Waldman, Aviel D. Rubin, and Lorrie Faith Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *Proc. 9th USENIX Security Symposium*, pages 59–72, August 2000.