

# Do You Believe in Tinker Bell? The Social Externalities of Trust

Khaled Baqer and Ross Anderson

Computer Laboratory, University of Cambridge, UK  
forename.lastname@cl.cam.ac.uk

**Abstract.** In the play Peter Pan, the fairy Tinker Bell is about to fade away and die because nobody believes in her any more, but is saved by the belief of the audience. This is a very old meme; the gods in Ancient Greece became less or more powerful depending on how many mortals sacrificed to them. On the face of it, this seems a democratic model of trust; it follows social consensus and crumbles when that is lost. However, the world of trust online is different. People trust CAs because they have to; Verisign and Comodo are dominant not because users trust them, but because merchants do. Two-sided market effects are bolstered by the hope that the large CAs are too big to fail. Proposed remedies from governments are little better; they declare themselves to be trusted and appoint favoured contractors as their bishops. Academics have proposed, for example in SPKI/SDSI, that trust should flow from individual users' decisions; but how can that be aggregated in ways compatible with incentives? The final part of the problem is that current CAs are not just powerful but all-powerful: a compromise can let a hostile actor not just take over your session or impersonate your bank, but 'upgrade' the software on your computer. Omnipotent CAs with invisible failure modes are better seen as demons rather than as gods.

Inspired by Tinker Bell, we propose a new approach: a trust service whose power arises directly from the number of users who decide to rely on it. Its power is limited to the provision of a single service, and failures to deliver this service should fairly rapidly become evident. As a proof of concept, we present a privacy-preserving reputation system to enhance quality of service in Tor, or a similar proxy network, with built-in incentives for correct behaviour. Tokens enable a node to interact directly with other nodes and are regulated by a distributed authority. Reputation is directly proportional to the number of tokens a node accumulates. By using blind signatures, we prevent the authority learning which entity has which tokens, so it cannot compromise privacy. Tokens lose value exponentially over time; this negative interest rate discourages hoarding. We demotivate costly system operations using taxes. We propose this reputation system not just as a concrete mechanism for systems requiring robust and privacy-preserving reputation metrics, but also as a thought experiment in how to fix the security economics of emergent trust.

**Keywords:** Trust · Reputation · Metrics · Unlinkability · Anonymity

## 1 Introduction

Children know the story of Tinker Bell from JM Barrie’s 1904 play ‘Peter Pan; or, the boy who wouldn’t grow up’. She is a fairy who is about to fade away and die, but is revived when the actors get the audience to declare their belief in her. The underlying idea goes back at least to ancient Greek mythology: the Greek gods’ power waxed and waned depending on the number of men who sacrificed to them. More modern references include Jean Ray’s 1943 novel *Malpertuis* [21] puts it as (translation from French): “Men are not born of the whim or will of the gods, on the contrary, gods owe their existence to the belief of men. Should this belief wither, the gods will die.” The same concept was used recently in the 2012 movie *Wrath of the Titans*.

The idea that authority emerges by consensus and evaporates when the consensus does is not restricted to mythology; democratic institutions perform a similar function. In the context of a nation state, or even a professional society, they are developed into a governance framework optimised for a combination of stability, responsiveness and the maintenance of trust.

How are things online? The honest answer is ‘not good’. When talking of trust online the first port of call is the Certification Authority (CA) infrastructure, which has many known failings. A typical machine trusts several hundred CAs, and trusts them for just about everything; if the Iranian secret police manage to hack Comodo, they can not only impersonate your bank, or take over your online banking session, they can also upgrade your software. Since an Iranian compromise caused the browser vendors to close down Diginotar, we have seen corporates moving their certificate business to the two largest players, Verisign and Comodo, in the belief that these firms are ‘too big to fail’ (or perhaps ‘too interconnected to fail’). Firms hope that even if these CAs are hacked (as both have been), the browser vendors would never dare remove their root certs because of the collateral damage this would cause. As for ordinary users, we trust Verisign not because we decided to, but because the merchants who operate websites we use decided to. This is a classic two-sided market failure.

Can we expect salvation from governments? Probably not any time soon. Governments have tried to assume divine powers of their own, first during the crypto wars by attempting to mandate that they have master keys for all trust services operating in their jurisdiction, and second by trying to control authentication services. Such initiatives tend to come from the more secret parts of states rather than the most accountable parts.

Can we users ourselves do better? The SPKI/SDSI proposal from Ellison, Rivest and Lampson attracted some research effort in the late 1990s and showed how every individual user could act as their own trust anchor, but the question is how to deploy such a system and scale it up; the one user-based system actually deployed in the 1990s, PGP, remains widely used in niche applications such as CERTs and anti-virus researchers, but never scaled up to mass use. The application of encrypting email suffers from strong network externalities in that I need my counterparties to encrypt their email too. This has become the norm in specific communities but did not happen for the general population.

Can we scale up deployment in other applications from a club that provides a small initial user base? One case where this happens is in the Internet inter-connection ecosystem, where trust among some 50,000 ASes is founded on the relationships between about a dozen Tier-1 providers, who form in effect a club; their chief engineers meet regularly at Nanog conferences and know each other well. But how could a service scale up from a few dozen users?

## 2 Motivation

In this paper we present another example for discussion. We propose an anonymous online reputation system whose goal is to let people get better quality of service from a distributed proxy service such as Tor. Our proposed new trust service has limited scope; if it works, it can provide lower latency, while if it fails its failure should be evident. The more people trust it, the more effective it becomes; if people observe that it's not working and lose faith in it, then it will fade away and die. What's more, multiple such trust services can compete as overlays on the same network.

The Tor network [9] consists of volunteer relays mixing users' traffic to provide anonymity. The list of relays is disseminated through a consensus file which includes the IP addresses of all relays. IP addresses are required to allow a user's client (Tor software) to locally decide which relays to use to route traffic. However, an attacker (say a *sensor*) can also download the consensus file, extract relay addresses, and block traffic by cooperating with local ISPs or using a nation-wide firewall. Thus, *victims* of a technically competent censor need private relays to connect to one of the publicly known relays. The private relay must not be known to the censor, or it too will be blocked. These private relays, called *bridges*, act as transient proxies helping victims to connect to the Tor network. Bridges are a scarce resource, yet play a critical role in connecting censorship victims to the Tor network. Therefore we want to incentivise Tor users to run more bridges.

The system proposed in this paper was originally designed to motivate non-malicious node interaction in anonymous remailer networks. We then realised that the design fits into the literary theme for the Twenty-third International Security Protocols Workshop, "Information Security in Fiction and in Fact".

## 3 System design

The system consists of competing *clubs*; each is managed by a club *secretary*. This is a major design difference from using a quorum of Directory Authorities (DAs) organised as a failover cluster, as currently implemented in Tor. The club secretaries, acting as Bridge Authorities (BAs), are responsible for disseminating information regarding club *members*. Each secretary is supported by a community of members who use its tokens as a currency to prioritise service. Members

help censored users (*victims*) to circumvent censorship by volunteering their resources to act as bridges, and can claim token rewards for their help. To the secretaries, the performance of members is visible and measurable.

Secretaries clear each others' tokens, just as banks clear each others' notes. Through private token payments, we can analyse the behaviour of nodes and determine which are actively and correctly participating in the network. Tokens are blindly-signed objects used to request services from other nodes. Tokens lose value over time, to demotivate hoarding. We discuss now the details of operation.

### 3.1 Member registration

Members can join any club they choose; loyalty to a club is determined by the incentives and performance it offers. Members can participate in one club or many, volunteering for whoever provides reliable services. Members offer service to a club by broadcasting their services: "this is my key, address, etc and I'll be available for contact between 11:00 and 11:30; send victims my way". This process can be automated using an uncensored and trusted means of communication. We assume that most members are outside the censor's jurisdiction, though some will have ties of family or friendship to the censor's victims. Thus some members may be motivated by the wish to help loved ones while others are altruistic and others are revenue maximisers. Some members will be within the censored jurisdiction.

We assume that keys can be exchanged successfully between members and secretaries: either the secretary publishes public keys somehow, or passes them on to new members as part of the recruitment process (about which we are agnostic). Within the censored jurisdiction, one or more designated *scouts* communicate with victims: we assume these are existing members. We assume the existence of innocuous store-and-forward communications channels such as email or chat; only a handful of censored jurisdictions ban Gmail and encrypted chat completely.

### 3.2 A simple threat model

Suppose that Alice and Bob belong to a club organised by Samantha to provide bridge services. Alice volunteers her IP address at time  $t$  to Samantha; a victim, Victor, contacts Sam to ask for a bridge; Sam gives him Alice's IP address and a short one-time password  $N_A$ ; Victor contacts Alice and presents  $N_A$ ; Alice shows  $N_A$  to Sam, who checks it, and Alice connects Victor to the Tor network. The protocol runs

$$\begin{aligned} A &\rightarrow S : IP \\ S &\rightarrow V : IP, N_A \\ V &\rightarrow A : N_A \\ A &\rightarrow S : N_A \\ S &\rightarrow A : OK \end{aligned}$$

The first problem with this simple protocol is that Samantha has to be online all the time; as she's a bottleneck, the censor can take down the system by running a distributed denial of service attack on her, and even without that we have two messages more in the protocol than we probably need. Our first attempt at improvement is to make the nonce  $N_A$  one that Alice can check; Alice shares a key  $K_{AS}$  with Samantha and we construct the nonce  $N_A$  by encrypting a counter  $k$  with it. The protocol is now

$$\begin{aligned} A &\rightarrow S : IP \\ S &\rightarrow V : IP, \{k\}_{K_{AS}} \\ V &\rightarrow A : \{k\}_{K_{AS}} \end{aligned}$$

Alice can now check the nonce directly, so Samantha doesn't have to be online.

The next problem is harder; it's that the censor's shill, Vlad, can also ask for an IP address, and if Samantha gives him one, the censor will block it. This is the real attack right now on Tor bridges; the censors pretend to be victims, find the bridges and block them. Various mechanisms are used from restricting the number of IP addresses given to any inquirer, and trying to detect Sybil inquirers using analytics; but if you have a repressed population where one percent have been coopted into working for the secret police, then telling Vlad apart from Victor is hard (at least for Samantha who is sitting safely in New York).

### 3.3 A more realistic threat model

In what follows we assume that of the two representative club members, Alice is in the repressed country, while Bob is sitting safely in exile. Alice, if honest and competent, is better than average at telling Victor from Vlad, perhaps because of family ties, friends, or ethnic or religious affiliations. Alice might be undercover, or might have some form of immunity; she might be a diplomat, or religious official, or sports star. She might hand over bridge contact details to victims written on pieces of paper, or on private Twitter messages to fans. The full gamut of human communications, both online and offline, are available for members who act as scouts to get in touch with victims.

We now introduce another layer of indirection into the protocol. After Bob volunteers to be a bridge, Samantha gives the scout Alice a token for her to give to a victim Victor, constructed as  $\{k, N_{AS}\}_{K_{BS}}$ . When this is presented to Bob, he can decrypt it and recognise the counter, so he knows Samantha generated it for him, and grants bridge service to the victim. He sends it to Samantha, who can recognise it as having been generated for Alice, and can thus note that Alice managed to recruit Victor (or alternatively, if Bob's IP address then ended up on the blacklist, that Alice recruited Vlad by mistake). Formally

$$\begin{aligned} B &\rightarrow S : IP \\ S &\rightarrow A : IP, \{k, N_{AS}\}_{K_{BS}} \\ V &\rightarrow B : \{k, N_{AS}\}_{K_{BS}} \\ B &\rightarrow S : N_{AS} \end{aligned}$$

$N_{AS}$  can of course be constructed in turn by encrypting a counter; but once we start encrypting a block cipher output and a counter under a wider block cipher, we are starting to get to the usability limit of what can be done with groups of digits written on a piece of paper. As AES ciphertext plus an IP address is about 50 decimal digits. In some applications, this may be all that's possible. In others, we might assume that both scouts and victims can cut and paste short strings, so that digital coins and other public-key mechanisms can be used.

In a more general design, we have to think not just about running scouts to contact victims and tell victims apart from censors, but also about scouts who are eventually turned, and about clubs that fail because the club organiser is turned, or has their computer hacked by the censor, or is just incompetent. We also have to think about dishonesty: about a bridge operator or scout who cheats by inflating his score by helping nonexistent or Sybil victims. How far can we get with reasonably simple mechanisms?

### 3.4 Payment system

We avoid using external payments, as offering cash payments as incentives to volunteers risks trashing the volunteer spirit (this is why the Tor project has always been reluctant to adopt any form of digital cash mechanism for service provision; volunteering is crucial for Tor's operation). Furthermore, we avoid using complicated zero-knowledge protocols or creating huge log files to protect against double-spending; large audit trails cannot scale very well. We prefer a lightweight mechanism that uses blind signatures made with regularly changing keys and member pseudonyms to provide privacy and unlinkability, as well as symmetric cryptography to create data blobs verifiable by the secretary or bridge. The token reward can then be used to pay for other services in Tor. For example, club members who run successful bridge services might enjoy better quality of service by using service tokens to get priority.

We now sketch a design using blind signatures rather than just shared-key mechanisms.

**3.4.1 Member identifier** After registering a member, the secretary creates a series of data blobs as the member's *identifiers*. An identifier can be the result of encrypting the member's name plus a counter or random salt with a symmetric encryption algorithm and a key known to the secretary; identifiers change constantly, and the secretary alone can link them to members other than by context. As well as knowing which members correspond to which identifiers, the secretary also notes which victims are introduced to which identifiers as possible bridges. This is to make it hard for a member to generate fake victims and claim rewards without providing them with service. It does mean the secretary is completely trusted, but secretaries compete with each other to provide effective service. We discuss all this in more detail later.

**3.4.2 Victim accounts** Secretaries maintain reputation scores not just for members but also for victims. Upon being introduced by a scout, a victim gets a default score of 1 allowing them to make a single bridge request. After a successful initial request, the victim’s account is reduced to zero for the current period. This is one of a number of rate-limiting mechanisms to prevent Vlad from draining the IP pool.

Secretaries use victim accounts to mitigate a few possible attacks, which we explore in more detail in the discussion section. One purpose is Vlad detection: if the members a victim learns about are not censored after a period of time – the IP addresses are not blacklisted and the scout is not turned – the victim’s account is increased; the opposite is true if a scout is arrested. Eventually, a diligent secretary should be able to identify fake victims by intersecting groups of victims and groups of censored members. It follows that the censor would have to refrain from blocking members if he wants to learn about new members. If he blocks members immediately, he betrays his skills. The conventional law-enforcement approach would be to block immediately, while the intelligence approach would be to merely observe quietly until all or most of the members are identified. Forcing the censor to make a strategic choice opens up all sorts of possibilities.

Secretaries have a clear incentive to protect their members’ identities; if the censor can spot and arrest the scouts and block the bridge IP addresses, the club will be ineffective and volunteers will help other clubs instead. Some volunteers will be picky, as if they join a badly-run club their IP address will be quickly blocked in that club’s country of interest. Other volunteers may be happy to help victims in fifty countries and consider it a badge of honour to be blocked in a dozen of them. And if participation is rewarded not just with honour but with improved quality of service, then this will mostly be forthcoming from the jurisdictions in which you are not blocked.

**3.4.3 Token creation** Account payment mechanisms can be replaced by blind tokens at one or more stages in the process. In the simplest implementation, we can reward Bob for providing bridge services with tokens that offer better quality of service from Tor nodes. Given the way Tor works, this requires some form of anonymous payment or certification. So when Bob services a request from Victor, Bob can now use Alice’s nonce  $N_{AS}$  to request an anonymous token from Samantha rather than just banking the credit. He does this by generating a well-formed token  $C_B$ , blinds it with a multiplier, and sends it to Samantha, who generates a blind signature [4] and returns it. With simple RSA blinding, where  $e$  is the public signature verification exponent,  $d$  the private signing exponent and  $n$  the public modulus, we have

$$\begin{aligned} B \rightarrow S &: N_{AS}, r^e C_B \pmod{n} \\ S \rightarrow B &: r C_B^d \pmod{n} \end{aligned}$$

Bob now unblinds  $C_B$  by dividing out the blinding factor  $r$ . This token is unlinkable and can be used for interacting with Tor nodes. The token  $C_B$  includes a random number, generated by Bob, to detect double-spending. Unlinkable

tokens can now be used to request other services by embedding them in the request; for example, if victims can handle public-key mechanisms, Victor might use such a token to request bridge service from Bob. However, it's in prioritising anonymous service requests that blind tokens really come into their own.

**3.4.4 Defining time using key rotation** As noted earlier, we assume that new public signature verification keys are announced for each future epoch. (It is possible to use an identity-based signature scheme by setting the public key for epoch  $i$  to be the value of  $i$ ). We can simplify matters if we define a time interval as an epoch; this enables us to avoid using timestamps in token protocols (we'd prefer to avoid the complexity of using partially blinded signatures that contain timestamps). Changing public keys frequently also greatly reduces the amount of state that must be retained to detect double spending, a known problem with blind payment systems. If tokens are used only by members such as Tor nodes with high-quality network service, this is probably a reasonable simplifying measure. We note in passing that nodes have an incentive to pay attention to the stream of signature traffic, to ensure they are not cheated by being passed a stale token.

**3.4.5 Validation and value of tokens** Tokens expire if they were issued using a signing key that was retired or revoked; for example, a signing key may be deemed retired if it was first used a certain number of epochs ago. We propose that the number of epochs since the token's signing key was first used is a deflator, which will decrease the value of a token. The formula used might be exponential, to represent a negative rate of interest; it is not clear that it matters all that much whether deflation is exponential or linear. Club secretaries can refresh tokens with new ones of an appropriately lower value, according to rules we will discuss later, and will reject double spend attempts. Expired tokens will also be recognised and rejected by all nodes, limiting the volume of data needed to detect double-spending.

### 3.5 Generating trust and reputation metrics

Each secretary acts as a bank and keeps members' accounts: each member's balance is the amount of correctly performed identifiable services plus the number of tokens claimed or refreshed in each epoch. These balances act as a proxy for reputation. The balances also deflate over time, but significantly less quickly than tokens in circulation. For example, if a token loses 20% of its value at each epoch when in circulation, it might lose only 5% when banked. Thus a member wanting to maximise its reputation has to do useful work and bank tokens promptly. The history of members' bank balances may also be made available to other members; there are second-order issues here about the potential identifiability of members involved in particular campaigns, so whether the secretary publishes member account history or smoothed metrics derived from it would depend on the application.



The overall effect is that the network as a whole can take a view on how much it trusts particular members. The more tokens Bob has in the bank, the higher his reputation. Note that there is little incentive for a group of colluding nodes to manipulate the reputation system by trading among themselves; they achieve nothing except to decrease their original endowment of tokens by the amount of time these tokens are not in a bank.

Another advantage of using a rapidly depreciating currency is that we can use the reputation system itself as an indicator of member liveness. We want to avoid sending requests to offline nodes; yet if we contact each node directly to request proof of liveness (for example with an ICMP packet), we open up a denial-of-service attack where a malicious node saturates the system with liveness checks. A real-time reputation system may help us avoid this.

## 4 Discussion

The system proposed in this paper is mainly concerned with enabling a group of users to maintain situational awareness in a censorship avoidance system. Each club of users can set up their own bridge authority maintained by a club secretary; an authority trusted by more users will become larger. In equilibrium we hope that clubs would settle each others' tokens, just as banks clear each others' notes. It can happen of course the community supporting a particular club fails. Our system is designed to enable social externalities to determine the level of trust in the system. If nodes believe in Tinker Bell, then they can bank their tokens with her bank; but if trust and belief fade, then tokens for Tinker Bell deplete, her reputation metrics decrease, and her bank eventually goes bust. The mechanisms to deal with this are a matter for the implementation.

### 4.1 Mitigating collusions and malicious members

There are a number of possible ways in which members might behave improperly. Alice and Bob might collude, so that Bob pretends to service Sybil victims invented by Alice; Samantha can mitigate the risks of this to some extent by randomising the allocation of IP addresses to scouts, and running appropriate analytics. The worst case is if the target country's intelligence service manages to hack Samantha's computer; then it's game over. (For that reason we posit multiple competing clubs.) The next most severe attack might be if the target country's intelligence service manages to subvert Alice and most of her fellow in-country members, and uses their bridge resources to make innocuous network connections, rather than blacklisting them, thereby denying the resources to censorship victims and denying both Bob and Samantha knowledge that Alice has been subverted<sup>1</sup>. While a normal censor will act as a policeman and block bridge IP addresses, a more strategic adversary might prefer to leave them be

<sup>1</sup> Such subversion might involve a national-scale malware implementation programme; see for example Gamma's 'Project Turkmenistan' disclosed on wikileaks.

and exhaust the resource. Detecting and defeating such attacks requires further channels of information. Ultimately we rely once more on the facts that there are multiple clubs, and multiple channels of communication between censorship victims and their family, friends or co-religionists in exile.

#### 4.2 Mitigating Sybil attacks

Members can claim tokens by fabricating victims, but those members end up ‘burning’ their victims’ account balances (and their email identities) if they don’t use the identifiers. Recall that victims are assigned to members randomly, and Samantha can run analytics to determine which scouts and which bridges have outlying patterns of victims. Diagnosis is not always going to be straightforward; Samantha might suspect that some victims are bogus, or that some members refuse to service victims, but it may in fact be the case that some victim group cannot contact members due to censorship or DoS attacks.

It does little good if multiple members collude to exchange each others’ identifiers; they end up burning their fabricated victims’ identities, as long as there are honest members acting correctly serving genuine victims. Members are better off acting correctly.

#### 4.3 Security economics

In this proposal, we attempt to incentivise correct behaviour, to generate metrics to identify which nodes are most interconnected, and empower nodes to shift trust to other nodes. In other words, we facilitate the mechanisms required to democratise trust and power, by empowering participating nodes to vote (transact using tokens) for the node that deserves their trust. Moreover, by creating a system to generate useful metrics, this design can be used to facilitate research on the security economics of users’ interactions, inspired by [1, 8].

## 5 Related work

In their paper *On the economics of anonymity* [1], the authors argue that the actions of interacting nodes in the network must be visible for informative decision-making about malicious behaviour. Through the trust and reputation metrics introduced in this paper, we can now understand node interactions better while preserving privacy. A thorough and insightful discussion based on practical experience is provided in [8], whose authors state that in order to provide the mechanisms for verifiable transactions and reputation ratings in anonymity systems, they needed to retrofit appropriate metrics. In fact, those authors already wondered whether it might be possible to create a reputation currency that might “expire, or slowly lose value over time”. The redesigns that the authors discuss in [8] were originally introduced in [6] and [10]. In [6], the authors integrate into Mix networks (anonymous remailers such as Mixmaster [17]) the role of witnesses: semi-trusted nodes that act as referees to service rejection or abnormal

behaviour (but this introduces multiple trust bottlenecks and can be abused if the witnesses are compromised). In [10], the authors decided to drop the witness construction. Network paths are constructed in cascades; if one node in the path fails, every node in the path is rated negatively. Without proof of service failure to pinpoint the node responsible, this design is vulnerable to an adversary joining multiple correctly-operating cascades, perhaps in order to route traffic to other adversary-owned cascades (which would de-anonymise users). Similarly, Free Haven [7] uses reputation to reward correctly-operating servers that store data and fulfil their contracts. The reward is a higher reputation that allows a server to store its own data with other servers (so long as no issues occur when validating service contracts). In fact, Free Haven is one of the first designs to use reputation as a form of currency. Moreover, the stamp-trading system discussed in [18] suggested that reputation built through proofs of providing services can be used as a currency to facilitate node interaction.

In the context of anonymity networks, there have been many proposals to rate and incentivise service-providing nodes (Tor relays). Most of these designs rely on bandwidth verification. For example, in [11], the authors suggest granting priority to the traffic of high-bandwidth nodes using gold stars. However, this can profile relays that also run clients by isolating the gold star traffic from the ordinary variety. More traditional approaches include XPay [5] and PAR [2] which use digital cash to reward relays. A more novel approach is discussed in BRAIDS<sup>2</sup> [13] which employs a similar architecture to the one proposed in this paper; but BRAIDS uses partially blind signatures to embed timestamps in tickets. This allows the Directory Authority (DA) in a BRAIDS-enabled system to expire tickets based on timestamps. In contrast, we expire tokens by key rotation, which enables tokens to be unlinkable. BRAIDS nodes create relay-bound tickets that can be verified by the relay; this increases efficiency, but limits the freedom to transact, as a relay has to contact a DA to issue new tickets for other relays. If a Tor path includes a relay that refuses service, the node must create new relay-bound tickets (generating another request to DA), since the original tickets will not be accepted by another relay. In LIRA [14], the authors attempt to increase efficiency by introducing a probabilistic micropayment protocol into their design (in fact, the authors use a similar construction to MicroMint [22]). In rBridge [23], the authors aimed to solve a different problem: rewarding users using a credit system based on how long information about a Tor bridge remains secret from an adversary (measured by its reachability and non-censorship). This credit system can then be used to obtain information about another bridge if the original bridge is censored.

Other authors were inspired by cryptocurrencies: In [15], high-bandwidth relays are awarded with *Shallots* which are redeemable for *PriorityPasses* that can be used to classify and prioritise Tor traffic. In TorCoin [12], TorPaths are used to verify paths constructed in Tor, and can be viewed as an enhancement for Tor’s bandwidth verification. Another approach that aims to preserve users’

---

<sup>2</sup> We initially designed our system without knowledge of BRAIDS then amended this paper to refer to it, but did not set out to design an improvement to BRAIDS.

privacy was proposed in [3] that uses Proof-of-Work shares as micropayments for relays. Relays can then submit those shares to a mining pool and claim rewards in Bitcoin [19]. This provides anonymity for users but not for relays if they use the pure Bitcoin protocol.

Trust and reputation metrics are used for various reasons. For example, Ad-vogato [16] was used to create attack-resistant metrics to correctly rate user-generated content. Another prominent example is Google’s PageRank [20], which rates web pages based on how many other pages point to it and creates a reputation rating for how reliable a page is (essentially, the pointers to a page are votes for that particular page but are weighted recursively by their own reputation).

A common problem with most proposals for incentivising Tor relays by using bandwidth verification schemes, or user-generated feedback, is that the authors do not discuss Sybil attacks which involve the adversary creating many circuits to her own relays to game the system. We attempt to solve this through key rotation which provides token expiry and decay.

A further issue is that an overlay on Tor that enables some club of users to enjoy priority service would risk a substantial decrease in the size of the anonymity set. In our proposal, a large number of clubs can each have a secretary acting as their own DA, and the secretaries can clear each others’ tokens.

## 6 Conclusion

In this paper, we sketched a design for an anonymous reputation system that can provide a quality-of-service overlay for an anonymity system like Tor or Mixminion. Unlike most electronic trust services today, it has the right incentives for local and democratic trust management. Groups of users can each establish their own token currency to pay for forwarding services, and the nodes that work hardest can acquire the highest reputation, enabling them to get still more work. Groups can clear each others’ tokens. And finally, any group that fails to compete will find that its failure becomes evident; its users can desert it for other groups and it can just fade away.

**Acknowledgements.** The first author thanks colleagues Laurent Simon and Stephan Kollmann for discussions regarding anonymity networks.

## References

1. A. Acquisti, R. Dingledine, and P. Syverson. On the economics of anonymity. In *Financial Cryptography*, pages 84–102. Springer, 2003.
2. E. Androulaki, M. Raykova, S. Srivatsan, A. Stavrou, and S. M. Bellovin. PAR: Payment for anonymous routing. In *Privacy Enhancing Technologies*, pages 219–236. Springer, 2008.
3. A. Biryukov and I. Pustogarov. Proof-of-Work as anonymous micropayment: Rewarding a Tor relay. In *Financial Cryptography and Data Security*, page 10. Springer International Publishing, 2015.

4. D. Chaum. Blind signatures for untraceable payments. In *Advances in cryptography*, pages 199–203. Springer, 1983.
5. Y. Chen, R. Sion, and B. Carbunar. XPay: Practical anonymous payments for Tor routing and other networked services. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 41–50. ACM, 2009.
6. R. Dingledine, M. J. Freedman, D. Hopwood, and D. Molnar. A reputation system to increase mix-net reliability. In *Information Hiding*, pages 126–141. Springer, 2001.
7. R. Dingledine, M. J. Freedman, and D. Molnar. The free haven project: Distributed anonymous storage service. In *Designing Privacy Enhancing Technologies*, pages 67–95. Springer, 2001.
8. R. Dingledine, N. Mathewson, and P. Syverson. Reputation in P2P anonymity systems. In *Workshop on economics of peer-to-peer systems*, volume 92, 2003.
9. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.
10. R. Dingledine and P. Syverson. Reliable mix cascade networks through reputation. In *Financial Cryptography*, pages 253–268. Springer, 2003.
11. R. Dingledine, D. S. Wallach, et al. Building incentives into Tor. In *Financial Cryptography and Data Security*, pages 238–256. Springer, 2010.
12. M. Ghosh, M. Richardson, B. Ford, and R. Jansen. A TorPath to TorCoin: Proof-of-bandwidth altcoins for compensating relays. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2014.
13. R. Jansen, N. Hopper, and Y. Kim. Recruiting new Tor relays with BRAIDS. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 319–328. ACM, 2010.
14. R. Jansen, A. Johnson, and P. Syverson. LIRA: Lightweight incentivized routing for anonymity. Technical report, DTIC Document, 2013.
15. R. Jansen, A. Miller, P. Syverson, and B. Ford. From onions to shallots: Rewarding Tor relays with TEARS. *HotPETs.(July 2014)*.
16. R. Levien. Attack-resistant trust metrics. In *Computing with Social Trust*, pages 121–132. Springer, 2009.
17. U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster protocol–version 2. *Draft, July*, 2003.
18. T. Moreton and A. Twigg. Trading in trust, tokens, and stamps. In *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, 2003.
19. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.
20. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. 1999.
21. J. Ray. *Malpertuis*, volume 142. Marabout, 1943.
22. R. L. Rivest and A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. In *Security Protocols*, pages 69–87. Springer, 1997.
23. Q. Wang, Z. Lin, N. Borisov, and N. Hopper. rBridge: User reputation based Tor bridge distribution with privacy preservation. In *NDSS*, 2013.