Research Notes on Trust:
How to Model and Design Trust in Accountable and Anonymous P2P File Transfer Systems

C. U. I.

April 18, 2019

File Version: 0.1

1 Introduction

1.1 Problem in Brief

Suppose a completely decentralized network where nodes transfer files to/from each other. They use a tor-like system to secure their anonymity. (See Section 3.1) They are responsible for choosing the relay nodes through which the file transfer (both sending and receiving) will be realized[1]. Nodes are encouraged by an accounting system that gives the relay nodes, as well as the sender nodes, some tokens which they can use to 1) buy service (e.g. downloading file) and 2) to protect their anonymity while buying the service (e.g. forming a relay path). Assume also that the accounting history secures anonymity by ignoring the content of the transferred file but only dealing with the amount of transfer. Accounting mechanism is expected to be *tamper-proof*, *append-only* and *easily certifiable*[2]. The system may include malicious nodes which perform some actions resulting in configurations where 1) the malicious nodes are able to spend more than what they actually earned, 2) a correct node cannot spend what it actually earned in response to what she had performed for the others.

Given the system above, find a reputation system / trust function which enables the nodes to select their service providers in such a way that

[1] [Ghosh et al., 2014] (TorPath) has a different approach to the problem so that nodes do not select the relay nodes directly but through an assignment server. Their motivation for this is to simplify and anonymize the payment.

[2] This is where *blockchain* gets on the stage

- secures anonymity
- the total end-to-end (spam-free) throughput increases with respect to the case of no-trust system. (relatively, the time required for building successful relay paths decrease)
- does not increase the vulnerability against correlation attacks [3] [4] (at least as much protection as in current Tor system)
- is based on an accounting system which provides incentives for
  - transaction completion (if you used a service, pay for that! and record it!)
  - transaction validation (Occasionally validate other transactions)
  - collaborate for collective trust calculation
- misbehavior is detected and penalized. Some misbehaviors are:
  - non-validated transactions
  - uncompleted/interrupted/abandoned relays

1.2 On the Projected Contribution

There is not a standard definition of trust yet ([Braga et al., 2018]). That means the existing works on trust evaluate the concept in different ways. It also means that the notion of trust is highly dependent on the formulation of the problem. For example, for a problem in which a node aims at finding peers that can provide more resources (e.g. bandwidth, storage) for it, the trust is about the **performance** (or **effectiveness**[5]). For the problem where a peer has possibility to not obey the rules of the protocol, the trust is about **correctness** (or **validity**?, or **legitimacy**?). For the problem where peers share opinions about each other, the trust is about **informativeness** (or **transparency**?).

Summary: *The meaning of trust is problem-dependent.*

In the problem we stated above, choosing the relay nodes and believing the opinions of other nodes on a specific node require different interpretations of trust. We can roughly call these interpretations as *dimensions of trust.*

[3] *One cell is enough to break Tor's anonymity*, Tor Blog, 2009..

[4] For other types of attacks on Tor: [Johnson et al., 2013]

[5] Ten digital trust challenges, Norbert Schwieters

Some of the dimensions that are detected on our problem until now[6] are as follows:

- Relay-ability: How well a node performs as a relay node. What is the success of the relay paths which involve the node.
- Anonymity-guarantee: Is the relay-candidate node a part of a group of nodes which collaborate to de-anonymize the network/me.
- Accountability?: How correctly a node records its own transactions. Is there any trace of double spending?
- Valid-ability:
- Informativeness?: How correct are the information that a node shares with me. Does it try to give false information to me about a node's performance, reliability, accountability or other dimensions of trust.

Summary: *Our problem has multiple dimensions of trust.*

Question: Can we have a distributed trust algorithm which outputs an n-tuple for every single neighbor of the node where i^{th} value of the tuple corresponds to the trust of the node on that neighbor with respect to the i^{th} trust dimension.

Note: Trust dimensions may be dependent on each other.

1.3 Briefly on the Originality of the Problem

- In Tor Browser, there is not any accounting system: no incentive for being a relay, only expects altruistic behavior from the clients.
- In Tor Browser, the notion of trust is only with respect to the available bandwidth [7].
- In BitTorrent, there is no guarantee on anonymity at all.
- In BitTorrent, there is no relaying: direct connection between the sender and the receiver

[6] Names of the dimensions are subject to change and might not be matched up with the literature

[7] *"For all circuits, we weight node selection according to router bandwidth."* Tor Path Selection and Constraints

- No decentralized accounting system for onion-routing systems:
  - In BitTorrent, accounting system is under construction (BTT project) [8]
  - In TorPath ([Ghosh et al., 2014]), which aims at designing an accounting system for tor-like routing systems, nodes do not select the relay nodes directly but through a *central* assignment server, which breaks the promise of decentralization. Their motivation for this is to simplify and anonymize the payment.

1.4 Current State of Implementation (Tribler)

Features which are already -partially- supported by Tribler:

- Tor-like file transfer
  - *Important:* Tribler does not use the state-of-the-art Tor system at all and does not guarantee the same level of quality, suitability and anonymity.[9]
  - Seemingly, the basic caveat in securing anonymity in Tribler is that a node chooses its middle and exit relays with the help of the chosen entry relay node. In other words, a client who wants to build a relay circuit with 3 nodes does the following:
    * selects the entry relay node from available nodes
    * receives the list of available nodes (for being middle relay) from the entry node
    * selects the middle node from the list of available nodes
    * receives the list of available nodes (for exit) from the middle node by means of the entry node
    * selects the exit node from the list of available nodes

    Question: Is it secure to trust the list provided by the entry node?
- optional anonymity (nodes have chance not to use relaying)
- basic accounting mechanism (Trustchain) which
  - uses the idea of PeerReview ([Haeberlen et al., 2007])

[8] BitTorrent White Paper, v0.8.7
[9] see Specification of Tribler Anonymity, Tribler Wiki

  - endangers anonymity (due to the correlation of block creation with the accounting ???)
  - is not scalable enough (goal: less block creation for the same amount of transactions)
  - lacks protection against lost transactions (which are realized but not successfully accounted)

Features which are to be designed:

- **Trust** system (see Section3.3)
- More **scalable** distributed ledger (Either a more **scalable** version of Trustchain ([Otte et al., 2017]) or a new blockchain-based solution)
- Improved payment system (**accounting**) (Tribler issue #4255) which
  - secures anonymity
  - [design choice] is fair (exit nodes receive at least as much as other relays, since it sacrifies its anonymity)
  - [design choice] the sender of a file should get at least positive incentive even in the case of choosing anonymous file sharing. (In the current implementation, a seeder who has chosen to share anonymously loses tokens as long as it continues to share)

2 Formulation of the Problem

2.1 Offline Version

In the offline version of the problem, the requests for files are assumed to be known in advance, and do not change during the execution of the algorithm.

[*... to do...*]

2.2 Online Version

In the online version of the problem, the requests for files are dynamic and revealed over time.

[*... to do...*]

3 Proposed Model (Informal)

3.1 Anonymous File Transfer

- Tor-like file transfer network
- Nodes use the network for sending/receiving files to/from others
- Each node can have three main roles in a single end-to-end file transfer:
  - Sender: The node who provides the service (in our case, service is a file)
  - Receiver: The node who consumes the service
  - Relay node: Transfer the encrypted data from one node to another node for the purpose of increasing anonymity of sender and receiver nodes
- Sender relay path: The path consisting of
  - the sender
  - the relay nodes selected by the sender (number of relay nodes on the path can be between 0 and 3)
- Receiver relay path: The path consisting of
  - the receiver
  - the relay nodes selected by the receiver (number of relay nodes on the path can be between 0 and 3)
- A relay node may have the following sub-roles depending on its position on the path:
  - Entry (guard) node
  - Middle node
  - Exit node
- The most extreme cases of a file transfer:
  - Both sender and receiver demand **full**-anonymity[10]. (Figure 1)
  - Both sender and receiver demand for **no**-anonymity. (Figure 2)

[10] *full* means *maximum allowed* in this context

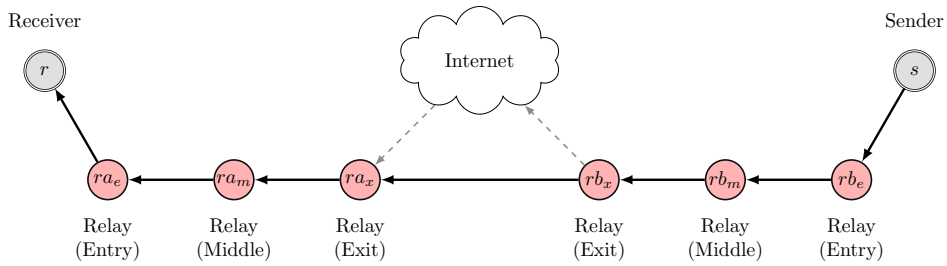

Figure 1: Full anonymity

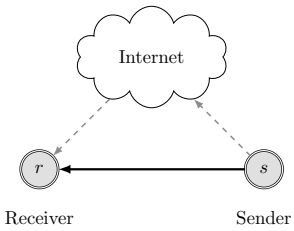

Figure 2: No anonymity

3.2 Incentives for relaying

The higher number of (trustable) relay nodes, the higher anonymity. Thus the nodes must be encouraged to participate relaying/sharing. Solution: A fully **Distributed Accounting Mechanism** where nodes can gain a value (which they should have in order to consume something from the system) by performing work for others.

- During a file transfer from a sender to the receiver, all of the relay nodes as well as the sender node perform work for the receiver. The receiver has to pay a value (e.g. *tokens*) to all for it.
- A node may have the option to pay less in exchange with decreased/zeroized level of anonymity.
- For a node to hold (or earn/mine) a value (e.g. *tokens*), it has to perform work for others, either by sharing a file or by serving as a relay node for a transfer.
- Accounting model is given in Section 6

3.3 Trust/Reputation System

Trust is *problem-dependent* and *has multiple dimensions*. (see Section 1.2).

We may need a trust system in the following aspects:

- **Trust on *relay-ability*** (under question!):
  - How does a node choose the ones from whom it will get the relaying service.
  - Possible criteria: number of involvement on a successfully completed relay (this may increase vulnerability against DoS attack)
  - [Idea for preventing DoS attack] Mastermind-like[11] problem solving for the detection of relay nodes who deny the service only when it is the middle node, but perform well when they control both the entry and exit.
  - [Design question] Different criteria for evaluation of suitability for different roles (entry, middle, exit)
  - The reason why this feature is under question is that creating a bias toward some nodes may make the system more vulnerable to attacks, by making the trusted nodes a "good" target for an attacker. In Tor system, this vulnerability is reduced by *ensuring that all nodes in the system are used to some extend, but nodes with more bandwidth and higher stability are used more often* [Bauer et al., 2007]. We can keep it in mind and give chance to nodes with low trust to become relay nodes to some extent. Another option would be to
- **Trust on accountability/reliability**:
  - Number and amount of validated Trustchain records (volume).
  - *Requirement:* Continuous check of validity.
- **Trust on performance**:
  - Served bandwidth (used in Tor [12])
  - Ongoing works on random walk studies in Tribler[13] actually use the transaction history of a node to determine its trust on performance:

[11] Mastermind (board game), Wikipedia Article

[12] *"For all circuits, we weight node selection according to router bandwidth."* Tor Path Selection and Constraints

[13] Real time random walk, C.U. Ileri; Incremental page-rank, A. Stannat

* A node learns about the transactions happened in the network.
* It draws a local vision of the network from its own perspective. The vision is actually a directed graph where the nodes are peers, the edges are the transactions, the weights of the edges are the amount of transactions and the direction of the edges are the directions of transferred files.
* A node executes a random walk on the directed local vision.
* Number of walks that passed through a specific node determines the trustability of the node.
* *Note:* Intuitively, the current approach is sybil-resistant in the sense that the total number of trust value that a node can have is not dependent on the number of sybils it created, but on the total amount of work performed by all the sybils of the node.
* Possible formulation:

$$TP_i(j) = \alpha \times RW_i(j) + (1 - \alpha) \times RWG_i(j) \tag{1}$$

$$RWG_i(j) = \frac{\sum_{k \in \Gamma(i)} TI_i(k) \times TP_k(j)}{\sum_{k \in \Gamma(i)} TI_i(k)} \tag{2}$$

  - $TP_i(j)$: i's trust on j's **performance**
  - $RW_i(j)$: Number of visit of j from i's local random walks
  - $RWG_i(j)$: RW results that i learns from its neighbors about j.
  - $TI_i(j)$: i's trust on j's **informativeness**
* Other possible measures of trust calculation on performance:
  - Centrality
  - Gauss ??
  - Hitting time ??

- **Trust on informativeness**:
  - Assume node A wants to evaluate node B according to a dimension of trust, say accountability. If A tries to validate all transactions of B by itself, it has to perform a lot of computations. Another node C which wants to evaluate node B should also validate all transactions of B. It is more convenient for A

and C to share their *knowledge* about the actions of B, than doing everything on their own. Assume A makes computations and informs C about the result:

  - Should C trust A's opinions about B? At what rate?
  - C could evaluate A's opinions on B by checking the correctness of opinions. By this, C starts to feel confidence in A's opinions and may decrease the percentage of correctness-checks. So, by means of trust, it gains information without consuming computation.
  - A's credibility on its report about B from the eyes of C is the trust-of-informativeness that C gives to B.

- Trust on informativeness is dependent on
  - the volume of the information shared.
  - correctness of the information shared.
  - [design choice] the trust (w.r.t. informativeness) of the others to you. (transition of trust)
- [Intuition:] Trust on informativeness may show recursive behavior: My current trust on you is the recursive evaluation of your past reports. Also, if transition of trust is taken into account, then the random walk approach can be used here: Build directed graph according to informativeness trust of peers against each other, and perform random walks to see who is more trusted.

- **Trust on seniority?**:
  - The length of the maximal time frame that includes any two transactions of the node.
- **Trust on purity?**:
  - The maximal time frame that ends at the current time and covers the moments where a node has not done a suspicious transaction.

Transition of Trust: Opinions of trusteds peers (transition of trust?), in other words, effect of reputation. (See model of [Mui et al., 2002]. Possible extension to a reputation system where nodes share their opinions on the trustability of a specific node and thus determine its *reputation*.

Evaluation of Trust/Reputation System: Any designed trust system is expected to secure the following:

- does not increase the vulnerability to correlation attacks (It has to guarantee at least as much protection as in-use Tor system).
- increase total throughput (the number of successfully completed data transfer over all relays)
- incentives for transaction validation
- enables a scalable transaction recording system (e.g. more scalable Trustchain: more transactions with less chains).

4 Possible Attacks

The designed trust system should guarantee a protection against all kinds of attacks on trust and reputation systems ([Hoffman et al., 2009]), anonymous P2P systems ([Cambiaso et al., 2019]) and accounting systems.

4.1 Sybil Attacks

Attacker creates multiple identities by which it gains an advantage over the case of single identity.

4.2 Spams

4.3 Traffic Analysis Attack

An attacker who controls the entry and exit relay nodes, it can achieve whole information about the sender, receiver and content of a specific transfer.

Note: It is mentioned in [Bauer et al., 2007] that as the network size increases, the likelihood of this kind of attack becomes negligible. The article refers the readers to [Dingledine et al., 2004, Reiter and Rubin, 1998, Syverson et al., 2001]. It should be checked if the analyses are still valid.

4.4 Selective Denial of Service Attack

The relay node which understands that it is the middle relay node refuses to transfer the file and causes the relay not to succeed, for the purpose of negatively affecting the trust scores of entry and exit nodes. As the scores of others are decreased, its probability to become entry and exit node, which accordingly increases its chance for a "successful" traffic analysis attack, gets bigger too.

4.5 Slander Attacks

Malicious nodes may collude to lie about the reputation of a particular neighbor and cause serious damage to the overall trust evaluation systems. [Velloso et al., 2008]

4.6 Double Spend Attacks

5 Related Works

5.1 Trust and Reputation Systems

- Survey on Computational Trust and Reputation Models [Braga et al., 2018].
- [... *to be filled* ...]

5.2 Accounting on Anonymous Networks

- [... *to be filled* ...]

5.3 Scalable Distributed Ledgers

- [... *to be filled* ...]

6 Accounting Mechanism

Note: Mechanisms discussed below mainly targets the token accounting system in tribler. However any trust model can be regarded as an accounting mechanism where peers exchange/use/collect trust values.

We first have to decide on our accounting model with respect to the general specifications of accounting mechanisms, some of which are:

- **Bilateral Interactions**: When A performs work for B, and thus increases its own score (e.g. tokens), does it mean that B consumes its own score by having a job performed by A.

  It other words, when A gains t_a tokens from the trade with B, assume that B loses t_b tokens? Which one is true?

  * $t_a = t_b$
  * $t_a = r \times t_b \; r > 1$
  * $t_a = r \times t_b \; 0 < r < 1$

For example, if we assume $t_a = t_b$, after a file transfer, the sum of balances of all peers involved on the whole path must remain same.

Observation: Our model is bilateral.

- **Transition of Information:** Can others learn about an interaction between two parties?
  - When A and B make a transacton, can C know it?
  - If yes, how?
    - Asking directly to A and B?
    - Asking the others about A and B? (This one is used in [Seuken and Parkes, 2011])
- **Duration of interaction**
  - Time interval of a single interaction?
  - Does an active transaction lock the parties? (I suspect this is the case in TrustChain but aimed to be solved by XChange)
  - How many simultaneous transactions can a party make?
- **Durability of *value***:
  - Does the score (e.g. token, coin) a party get from a transaction decay over time. (When it comes to trust o performance, decay of score over time may make sense)
- **Dependency on identity**: Assume same amount of work is performed by two parties A and B. Is the score they get differs with respect to their identities?
- **Aggregation vs. Average**: Assume peers B's and C's trust scores on A are $t_b(a)$ and $t_c(a)$, respectively. Assume further that peer D learns about and relies on $t_b(a)$ and $t_c(a)$. Does D aggregate or average $t_b(a)$ and $t_c(a)$ to determine $t_d(a)$.
- **Risk of transaction**: What is the risk of a single transaction.
  - What is the or cost of making a transaction with a -mistakenly trusted- agent.

7 Discussion and Other Notes

7.1 Adding a Dimension to Trust

Assume we have a distributed trust algorithm which outputs an n-tuple for every single neighbor of the node where i^{th} value of the tuple corresponds to the trust of the node on that neighbor with respect to the i^{th} dimension of trust.

Assume now that the system that implements the trust algorithm had a new feature for which the current trust algorithm is not sufficient, so that we discovered a new dimension of trust. Can we easily tune the function to make it support the new trust dimension?

Summary: *Our trust algorithm may guarantee the flexibility of understanding of trust.*

8 On the Next Sprint

- Study and categorize all existing trust and reputation systems according to the idea of multidimensionality of trust.
- Design dimension-specific trust functions: A trust function for each of the dimensions fo trust. (We may eventually find out that there is no need to have a trust function, so that random trust (i.e. no trust) provides better guarantees.)

References

[Bauer et al., 2007] Bauer, K., McCoy, D., Grunwald, D., Kohno, T., and Sicker, D. (2007). Low-resource routing attacks against tor. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20. ACM.

[Braga et al., 2018] Braga, D. D. S., Niemann, M., Hellingrath, B., and Neto, F. B. D. L. (2018). Survey on computational trust and reputation models. *ACM Computing Surveys (CSUR)*, 51(5):101.

[Cambiaso et al., 2019] Cambiaso, E., Vaccari, I., Patti, L., and Aiello, M. (2019). Darknet security: A categorization of attacks to the tor network.

[Dingledine et al., 2004] Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC.

[Ghosh et al., 2014] Ghosh, M., Richardson, M., Ford, B., and Jansen, R. (2014). A torpath to torcoin: Proof-of-bandwidth altcoins for compensating relays. Technical report, NAVAL RESEARCH LAB WASHINGTON DC.

[Haeberlen et al., 2007] Haeberlen, A., Kouznetsov, P., and Druschel, P. (2007). Peerreview: Practical accountability for distributed systems. *ACM SIGOPS operating systems review*, 41(6):175–188.

[Hoffman et al., 2009] Hoffman, K., Zage, D., and Nita-Rotaru, C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1.

[Johnson et al., 2013] Johnson, A., Wacek, C., Jansen, R., Sherr, M., and Syverson, P. (2013). Users get routed: Traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 337–348. ACM.

[Mui et al., 2002] Mui, L., Mohtashemi, M., and Halberstadt, A. (2002). A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 2431–2439. IEEE.

[Otte et al., 2017] Otte, P., de Vos, M., and Pouwelse, J. (2017). Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems*.

[Reiter and Rubin, 1998] Reiter, M. K. and Rubin, A. D. (1998). Crowds: Anonymity for web transactions. *ACM transactions on information and system security (TISSEC)*, 1(1):66–92.

[Seuken and Parkes, 2011] Seuken, S. and Parkes, D. C. (2011). On the sybil-proofness of accounting mechanisms.

[Syverson et al., 2001] Syverson, P., Tsudik, G., Reed, M., and Landwehr, C. (2001). Towards an analysis of onion routing security. In *Designing Privacy Enhancing Technologies*, pages 96–114. Springer.

[Velloso et al., 2008] Velloso, P. B. B., Laufer, R. P. P., Duarte, O. C. M., and Pujolle, G. (2008). A trust model robust to slander attacks in ad hoc networks. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, pages 1–6. IEEE.