Proof-of-Work as Anonymous Micropayment: Rewarding a Tor Relay*

Alex Biryukov and Ivan Pustogarov

University of Luxembourg,
{alex.biryukov,ivan.pustogarov}@uni.lu

Abstract. In this paper we propose a new micropayments scheme which can be used to reward Tor relay operators. Tor clients do not pay Tor relays with electronic cash directly but submit proof of work shares which the relays can resubmit to a crypto-currency mining pool. Relays credit users who submit shares with tickets that can later be used to purchase improved service. Both shares and tickets when sent over Tor circuits are anonymous. The analysis of the crypto-currencies market prices shows that the proposed scheme can compensate significant part of Tor relay operator's expenses.

Keywords: Tor; Proof of Work; Crypto-currency; Micropayment; Mining pools

1 Introduction

Many open peer-to-peer systems rely on volunteers donating their resources in order to achieve acceptable level of Quality of Service. E.g. in file-sharing applications, latency and failure rate depends on the number of users sharing their resources. In overlay routing systems packet latency depends on relays donating their bandwidth. Many of these systems suffer from free-riding: users consume resources without donating anything back. Obviously, this rational behavior is motivated by that users don't want to degrade their own performance. While not a P2P network in the traditional sense as there is a clear separation between clients and relays, Tor network suffers from the same free-riding problems: only limited number of relays provide decent bandwidth while the client base is rather large.

A number of incentive techniques were proposed to mitigate selfish behaviour of clients for traditional P2P systems. The bottom line of many of them is that a client is incentivized to donate the same type of resources to the network as he consumes. Unfortunately for Tor such incentives are hardly applicable: the majority of Tor users reside behind ISP NAT's and firewalls and thus cannot be checked by Tor authorities for reachability which prevents them from appearing in the Tor Consensus. In fact, for Tor it might be even undesirable to allow very

* Full version is available at http://eprint.iacr.org/2014/1011.pdf

low bandwidth nodes to become a part of the network [2] (and many clients can provide only limited bandwidth).

Another alternative would be to use a cryptocurrency and make direct payments to Tor relay operators. Many cryptocurrencies are not anonymous however which is in conflict with Tor goals. In this paper we propose a method to reward Tor relays. This method is based on crypto-currencies but does not have to involve direct payments; it rather adopts a mining-pool approach: a Tor relay implements mining pool functionality and provides Tor clients with mining jobs. When a client finds the job which meets requested difficulty, he submits the share to the Tor relay and gets priority tickets in exchange. Tor relays can either join a mining pool and delegate jobs to Tor clients or can do solo-mining and try to solve a block. The proposed approach does not require a central bank or a secure bandwidth measurement mechanism. The proposed approach may also help to solve scalability problem. The more users join the Tor network and use "paid" services, the more profitable it becomes to run a relay, and the more relays are expected to join the network.

The necessity of developing robust and secure incentives to participate in Tor was first mentioned in the Tor design paper [6]. Since then a lot of research has been done in the area [18, 12, 13, 8, 15], however in most cases they involve a central authority or require running a Tor relay. The idea described in [19] is close in spirit to our scheme (though not directly related) and suggests that a client offers a portion of his computation power in exchange for a service.

The rest of the paper is organized as follows. In section 2 we describe the details of our approach. Analysis of the method is given is section 3. Discussion in Section 4 concludes the paper.

2 Proof-of-Work as payment for service

2.1 Design goals

The main objective of the proposed scheme is to compensate Tor relays for providing improved service and to encourage server operator's participation in the Tor network. In addition, we require the following properties. First, the scheme should not degrade the anonymity provided by Tor, i.e. it should not introduce new attack vectors. Second, it should not involve direct payments neither with fiat nor with crypto-currencies. The reason for this is that direct payment even with a digital currency like Bitcoin will reduce user privacy[1] and may become a strong psychological obstacle for adopting a scheme for ordinary users. Third, it should not rely on secure bandwidth measurement mechanisms. Fourth, it should not involve a central bank as in [12]. Sixth, the scheme should not require from users to run a Tor relay in order to get improved service. We analyse these properties in more detail in section 3.

[1] An option of payment via anonymous crypto-currency like ZeroCoin [17] will be discussed in Section 4.

2.2 System design

Tor users can get improved service from a Tor relay by producing proof-of-work and sending it to the relay over an anonymous Tor circuit. The relay can then forward this proof-of-work to a crypto-currency mining pool and earn coins. Users are rewarded by relay-specific *priority tickets* which can later be exchanged at the same relay for improved service (higher bandwidth or lower latencies). Tickets are issued by relays using blind signatures [3] and exchanged between users and relays over anonymous Tor circuits. Unlike [12] we do not use any bank entity and tickets are blind-signed by relays themselves.

Setup. In the setup phase a Tor relay first chooses a mining pool, the corresponding crypto-currencies and PoW algorithms (note that the relay can choose a pool which automatically switches to the most profitable currencies). Second, the relay generates a public/private key pair which will be used in generation of priority tickets (this key pair should be different from the relay's onion and identity keys). The relay then includes this information into its descriptor. A client which plans to obtain improved service chooses relays which announce compatible PoW algorithms.

Protocol 1. Ticket Purchase: Client C obtains a priority ticket from relay R

1: $C \rightarrow R$: `SUBSCRIBE` message.
2: $R \rightarrow C$: `JOB` message.
3: C : start mining a share.
4: C : If share w is found, generate random number x and its hash $H(x)$.
5: $C \rightarrow R$: w, $H(x)$.
6: R : check w, if correct pass it to the mining pool.
7: $R \leftrightarrow C$: Generate partially blind signature S over $\{H(x), d\}$, where d is an assigned by the relay timestamp, which specifies the current day.
8: C: Keeps the ticket $T_R = \{S, d, x, H(x)\}$.

Purchasing priority tickets. A relay will provide improved service for clients in exchange for priority tickets. Priority tickets are relay-specific which means that by default they can only be used to purchase service from the relay which issued them (see Protocol 2 if ticket exchange is required). The protocol for client C to obtain a ticket from relay R is described in Protocol 1. Prior to execution of the protocol, the client establishes an anonymous Tor circuit to the relay. All communications are carried over this circuit, including (optionally) the future client traffic. Client C registers for a new mining job with relay R and the relay sends a reply in which it specifies the PoW algorithm, difficulty per share, and data sufficient to construct a share (steps 1–2). At step 3, the client starts solving a new share. At steps 4–5 (given that the client solved the share), the client generates a random value x and its hash $H(x)$ and sends the share to R. The relay verifies the share and produces a partially blind signature S over $H(x)$ with timestamp d as an added factor according to [1]. The tuple $T = \{S, d, x, H(x)\}$ is a priority ticket which the client can later exchange for

the improved service. By reducing the granularity of the timestamp to just the current date makes all clients that got tickets on the same day undistinguishable.
Buying improved service. Every ticket that a client gets can be used to transmit cells with priority access during Δt seconds through the Tor relay which issued the ticket. In order to prevent double-spending, the relay should keep history of spent tickets. To limit the size of this database tickets should expire after e.g. 48 hours.
Priority access. We suggest using Hierarchical Token Bucket Algorithm [14] to provide improved quality of service for users with priority tickets, however other options exist [7]. HTB is a simple algorithm and it is a logical step from the currently employed by Tor Token Bucket algorithm. The priority access scheme should allocate enough resources for "free" users so that people without funds to buy high-speed computers can still have reasonable QoS with Tor.
Ticket exchange. So far in the proposed scheme a client gets tickets from the same relay R_1 for which he is working, and the tickets are valid at this relay only. Such scheme works best if the client provides proof-of-work simultaneously with sending his data over Tor. Assume now that a client pre-mined priority tickets with an intention to spend them later. He might become frustrated if at the time when he decides to spend them relay R_1 is off-line. In such a case relay R_1 may team with a backup relay R_2 and ask it to accept its priority tickets. R_2 can later request payment from R_1 in crypto-coins or by redirecting his clients to mine for R_2. Protocol 2 describes how priority tickets issued to client C by relay R_1 can be spent at relay R_2. When relays R_1 and R_2 are both online they synchronise their databases of spent tickets.

Protocol 2. Ticket Exchange: C gets improved service at R_2 by providing a ticket issued by R_1

Client C obtained ticket $T_{R_1} = \{S_1, d, x, H(x)\}$ from relay R_1. R_2 is a backup relay for R_1

1: $C \rightarrow R_2 : T_{R_1}$
2: R_2 : verify signature S_1 and timestamp d.
3: R_2: If correct, register T_{R_1} as spent (sync this with R_1).
4: R_2 : If T_{R_1} is correct, provide priority access.
5: $R_2 \rightarrow R_1$: `PAYMENT_REQUEST` (Once every N served tickets).

Assume that client C has ticket $T_{R_1} = \{S_1, d, x, H(x)\}$ issued by relay R_1. The objective of the Protocol 2 is for the client to be able to get improved service from relay R_2 while preserving the following properties: (1) A colluding client and relay should not produce "free" tickets which can later be used at other relays; (2) Double spending of the same ticket at two different relays should be prevented.

"Free" tickets created by colluding client C and relay R_1 are avoided by that R_2 requests payment for each batch of N served tickets (either in crypto-coins or by delegating new mining work). We can envision that in practice relays R_1, R_2

might be run by the same operator or by two operators, who trust each other. In the second case the amount of trust can be regulated by the size of N. In case R_1 stops paying, relay R_2 will stop accepting its tickets. In order to prevent double-spending of the same ticket at relays R_1 and R_2 they should regularly synchronise their databases of spent tickets.

Mining strategies. The operator of a Tor relay which accepts PoW shares has two possibilities. First, he can decide to do solo-mining, by making his crypto-currency address a part of `JOB` messages sent to clients in the hope that one of the submitted shares will also solve a block. This strategy requires significant computational power at a large number of Tor clients. Second, the Tor relay operator may decide to ask for work from a large mining pool and then delegate this work to clients. The operator then resubmits the shares found by the clients to the mining pool. Note that the mining pool requests the relay to generate a share of difficulty lower than the current block's difficulty in the hope that one share will also solve the block. The Tor relay may use the same strategy towards Tor clients: it may request to generate PoW with difficulty lower than that indicated by the mining pool in the hope that a client's PoW will also solve the mining pool's share. With this approach the Tor relay may regulate how many tickets are issued to different clients, proportional to their mining power.

Donations. Clients that just want to support Tor relays without requesting any bandwidth can submit shares without requesting anything back.

3 Analysis

3.1 Profitability

Motivation. According to the performance statistics maintained by the Tor project[2] [21], it takes roughly between 10 and 15 seconds to download a 5MB file over the Tor network on average (which results in 333 KB/s). While such speeds are likely to be enough for general Web-surfing they might be frustrating for bulk file downloads, watching videos, or having a video conference [11]. The later types of traffic could be the reason why Tor clients may decide to get improved service from Tor relays. This might be especially true for Bittorrent users. Bittorrent over Tor has been problematic for both Bittorrent users and Tor relay operators: users did not get enough speed, and Tor operators are concerned that bulk file downloads consume a lot of bandwidth and thus decrease Quality of Service (QoS) for Web-surfing users.

Another reason why a Tor client would want to have higher capacity/lower delays is to improve QoS for his hidden services. The current version of Tor Hidden Services suffers from high delays and low speeds [10] which significantly reduces the number of users.

Choosing crypto-currencies There are more than 400 different crypto-currencies nowadays [5] (however only few of them achieved noticeable market capitalisation and are less susceptible to huge fluctuations in market value towards fiat

[2] For June – September 2014.

currencies). According to [4] and [22] the following PoW algorithms are used in existing crypto-currencies: Blake-256, Groestl, HEFTY1, JHA, Keccak, NeoScrypt, Quark, Scrypt, Scrypt-Adaptive-Nfactor, Scrypt-Jane, SHA-256, X11, X13.

Profitability of mining a digital currency obviously depends on the miner's hash-rate, price of electricity, the currency's difficulty, and its current market price. The miner's hash-rate can vary significantly depending on hardware. Table 1 shows hash-rates achievable for different algorithms on Intel Core i7-2760QM (4 cores at 2.40GHz). The table also includes maximum revenue[3] for each algorithm for the 1st of September 2014 according to [4] (averaged over multiple observations). Electricity costs are estimated to be 11 cents per day given that max power of the CPU is 45W. During the day we also observed short periods of time when the revenue jumped to 11 cents per day. Also note that hash rates achievable on GPU's can be an order of magnitude higher. We assume that an average user of our protocol does not use ASICs.

Hashing algorithm	Rate on Intel Core i7-2760QM	Currency	Revenue per day
Blake-256	9,6 Mh/s	Blakecoin	n/a
Groestl	1 Mh/s	Diamond	2.1
HEFTY1	128 Kh/s	Heavycoin	n/a
JHA	308 Kh/s	Jackpotcoin	2.2 cents
Keccak	5.2 Mh/s	Maxcoin	0.7 cents
Quark	300 Kh/s	CNotes	3.8 cents
Scrypt	40 Kh/s	42	0.8 cents
		Litecoin	0.65 cents
		Dogecoin	0.26 cents
Scrypt-N	20 Kh/s	Vertcoin	2.3 cents
Scrypt-Jane	360 h/s	Yacoin	n/a
SHA-256d	9.6 Mh/s	Peercoin	0.01 cents
		Bitcoin	0.008 cents
X11	360 Kh/s	Smartcoin	3.8 cents
		Darkcoin	2.5 cents
X13	104 Kh/s	Marucoin	n/a

Table 1. Hash rates of the proof-of-work algorithms on Intel Core i7-2760QM

Profit estimation. In order to estimate[4] how much a Tor relay can earn using the proposed scheme we first make the following assumptions:

– Among 2,000,000 daily Tor clients (according to the Tor statistics), only 500,000 are real users and the rest belong to botnets [16]. I.e. only 500,000 users can mine.

[3] Revenue can be smaller when trying to exchange due to small market size.

[4] These are of course very rough estimates: it's not possible to learn the current hardware of Tor users, estimate the fraction of non-botnet Tor users, the number of Tor users which would be willing to mine, and the number of new (Bittorrent over Tor) users.

– Moreover we assume that each user's session takes about 1 hour and every user is willing to mine with a hash-rate similar to that from Table 1. The later implies that clients will spend 100% of CPU on mining during 1 hour period. If clients decide to use less fraction of their CPU, the revenue of a Tor relay will decrease proportionally.

Income of a Tor relay obviously depends on the number of users which establish their circuits through this relay. This in turn depends on the relay's consensus bandwidth. We consider the case in which the scheme motivates running a Tor Exit node (currently there are only about 1,000 Exits out of 6,000 Tor relays). The green line in Figure 1 shows the income of an Exit relay under the assumption that each client can mine an equivalent of 3.8 cents per 24 hours of which a fraction of 1/24 is received by the relay during a 1 hour session. For such a case top Tor relays (with consensus bandwidth 200,000 KB/s) can earn about 500 USD per month. A middle-tier relay with consensus bandwidth 10,000 KB/s can earn about 25 USD. The green line in Figure 2 shows monthly incomes assuming 11 cents per client per day (in which case a top Tor relay can earn up to 1,600 USD).

Running a high-bandwidth Tor relay obviously means high costs. In order to estimate the incurred costs we assume that the rental price is: 25 EUR per month for a relay with consensus weight less than 15,000; 40 EUR for weight between 15,000 and 50,000; 70 EUR for consensus weight larger than 50,000. In addition we assume that 10 TB of traffic is included into the server's price and one has to pay 2 EUR per additional 1 TB [9]. It is important to note that we consider costs which Tor relays already have regardless whether they use the proposed rewarding scheme or not. Note also that in order to compute traffic costs of a relay we take its consensus bandwidth (which represents the relay's speed in KB/s), and assume that the relay constantly transmits with such speed which results in upper bound of traffic costs.

Costs to run an Exit relay of specific bandwidth and corresponding profitability of running such a relay (given the income produced by mining clients) are shown in Figures 1 and 2 with blue and red lines. A Tor relay partially compensates its costs in case of 3.8 cents per day per client; when clients mine an equivalent of 11 cents per day, the relay's costs are lower than its income. Additional income can be used for the server upgrade or to provide better free services.

3.2 Anonymity

In this section we discuss anonymity of the proposed scheme. In Protocol 1, after client C mined a share he sends it to the corresponding Tor relay along with the hash of a random number (to be blindly signed). All communications are done over anonymous circuits, so that the Tor relay does not learn the originator of the messages (unless it is a Guard node). In addition blind signatures prevent the Tor relay from distinguishing client C from other clients. Finally shares generated by client C contain a Bitcoin address of either the Tor relay or a mining pool

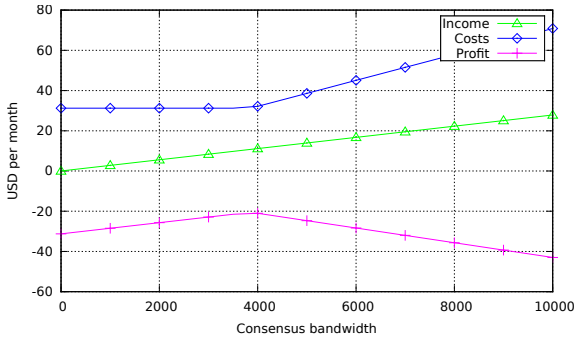

Fig. 1. Income, costs, and profit of an Exit relay in case of 3.8 cents per day per miner.

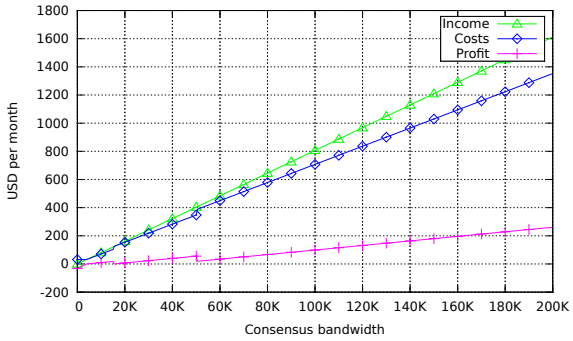

Fig. 2. Income, costs, and profit of an Exit relay in case of 11 cents per day per miner.

(the client is even not required to have a crypto-currency account), thus they don't reveal the identity of the client in spite of known attacks against Bitcoin (and hence Altcoins) anonymity [20].

A curious relay can however learn the hash rate of a client, thus it may recognize repeated connections from the same client. In order to mitigate such an attack a client is advised to randomize its hash rate. The same holds if a client decides to pre-mine bandwidth tickets from a relay.

We also note that a powerful miner can try to DoS the paid traffic of a relay, by taking all the paid traffic of a relay for itself. However such behavior is not rational, since it is economically more reasonable for such miner to just earn shares in the mining pool.

4 Conclusion and discussion

Mining Bitcoins or Altcoins on consumer-grade hardware, GPUs or even first generation ASICs (for Bitcoin) is not profitable nowadays. This is due to the fact that the difference between the price of mined coins and the electricity

costs is negative. Delegating mining (and thus electricity costs) to others while keeping the earned coins obviously makes it positive[5]. In this paper we propose a scheme to reward a Tor relay in which it subscribes for mining jobs at a crypto-currency mining pool and delegates these jobs to Tor clients (thus clients indirectly pay for electricity). The Tor relay then keeps all earned coins and in turn issues priority tickets and sends them to the clients. Priority tickets can be exchanged for the improved service at the same relay. The proposed scheme has four desirable properties: (1) it does not rely on a central bank; (2) it preserves user anonymity; (3) it removes a psychological barrier since clients do not pay directly (and thus the risk of their money being stolen is removed); (4) Tor relays are rewarded with crypto-currency coins which can be exchanged for fiat currencies and partially cover their operational expenses. A relay's income can vary significantly depending on crypto-currency exchange fluctuations, number of Tor clients willing to mine, hardware, etc. In a concrete example, assuming that clients mine for Exit relays only and if each client is able to mine an equivalent of 11 cents per day and mines 1 hour per day, an Exit relay with Consensus bandwidth 100,000 KB/s can earn 800 USD per month; such revenue should completely cover the relay's traffic costs and may allow the operator to upgrade to a more powerful server.

The proposed scheme does not decrease anonymity provided by the Tor network. All shares submitted by clients are anonymous and contain a Bitcoin address of either a Tor relay or a mining pool, thus attacks against Bitcoin anonymity become inapplicable. A curious relay can however learn a client's hash rate. Also in the case of pre-mining for later usage the relay will learn that the same user tries to go through it later on the same day.

Finally we would like to mention that if altcoins with strong anonymity (ex. Zerocoin [17]) become widely adopted it would be easy to integrate such payments into our scheme. A client will need to send together with the payment the blinded value for signing. The relay will need to broadcast a transaction with this value signed, from which the client will be able to derive the signature and thus the priority ticket.

Usages other than Tor The proposed scheme can be used not only to reward Tor relays. The same approach can be adopted by entities which accept payments. We note, that for this scheme to be successful it may be useful to go for memory-hard proofs of work, which would have no advantage in GPU or ASICs. Scrypt function used in some alt-coins (ex. Litecoin) comes close to be adequate for this purpose, though more energy-optimal tradeoff-resistant proof-of-work functions can be designed for this task. We envisage that widespread use of such CPU mining in exchange for services may become a basis for a widely used micropayment system, which in turn becomes a strong alt-currency used by consumers (what is currently lacking in the Bitcoin universe, where the main activities are mining and hoarding of coins).

[5] Our scheme thus also gives an interesting use case for the old mining gear which is otherwise obsolete. This might be the only way to buy lots of priority traffic on Tor relays.

References

1. Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Bellare, M. (ed.) Advances in Cryptology CRYPTO 2000, Lecture Notes in Computer Science, vol. 1880, pp. 271–286. Springer Berlin Heidelberg (2000)
2. Alsabah, M., Bauer, K., Elahi, T., Goldberg, I.: The path less travelled: Overcoming tor's bottlenecks with traffic splitting. In: Proceedings of the 13th Privacy Enhancing Technologies Symposium (PETS 2013) (July 2013)
3. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R., Sherman, A. (eds.) Advances in Cryptology, pp. 199–203. Springer US (1983)
4. CoinWars: Crypto Currencies: http://www.coinwarz.com (2014)
5. Crypto-Currency Market Capitalizations: http://coinmarketcap.com (2014)
6. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (August 2004)
7. Evans, J.W., Filsfils, C.: Deploying IP and MPLS QoS for Multiservice Networks: Theory & Practice. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007)
8. From Onions to Shallots: Rewarding Tor Relays with TEARS: http://dedis.cs.yale.edu/dissent/papers/hotpets14-tears.pdf (2014)
9. Hetzner Online Server Auction: https://robot.your-server.de/order/market (2014)
10. Hidden Services need some love: https://blog.torproject.org/blog/hidden-services-need-some-love (2014)
11. How much bandwidth does Skype need?: https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need (2014)
12. Jansen, R., Hopper, N., Kim, Y.: Recruiting new Tor relays with BRAIDS. In: Keromytis, A.D., Shmatikov, V. (eds.) Proceedings of the 2010 ACM Conference on Computer and Communications Security (CCS 2010). ACM (October 2010)
13. Jansen, R., Johnson, A., Syverson, P.: LIRA: Lightweight Incentivized Routing for Anonymity. In: Proceedings of the Network and Distributed System Security Symposium - NDSS'13. Internet Society (February 2013)
14. Linux HTB Home Page.: http://luxik.cdi.cz/ devik/qos/htb/ (2014)
15. M., G., Richardson, M., Ford, B., Jansen, R.: A TorPath to TorCoin: Proof-of-Bandwidth Altcoins for Compensating Relays. In: 7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs) (July 2014)
16. Massive spike of Tor users caused by Mevade botnet: http://www.net-security.org/secworld.php?id=15530 (2014)
17. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: IEEE Symposium on Security and Privacy (2013)
18. Ngan, T.W.J., Dingledine, R., Wallach, D.S.: Building Incentives into Tor. In: Sion, R. (ed.) Proceedings of Financial Cryptography (FC '10) (January 2010)
19. Ostrovsky, R.: A proposal for internet computation commerce: How to tap the power of the web. In: Presentation at CRYPTO '98 rump session (1998)
20. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Financial Cryptography and Data Security (FC'13). Springer (2013)
21. Tor Metrics: Performance: https://metrics.torproject.org/performance.html (2014)
22. Windows GPU Miners for the More Commonly Used Crypto Algorithms: http://cryptomining-blog.com/2595-windows-gpu-miners-for-the-more-commonly-used-crypto-algorithms/ (2014)