Securing and Scaling Cryptocurrencies

Aviv Zohar
The Hebrew University of Jerusalem, Israel
avivz@cs.huji.ac.il

Abstract

Bitcoin, a protocol for a new permissionless decentralized digital currency hailed the arrival of a new application domain for computer science. Following Bitcoin's arrival, a series of innovations derived from the state of the art in several fields has been applied to cryptocurrencies, and has been slowly reshaping monetary and financial instruments on public distributed ledgers. It was soon clear however that Bitcoin and similar cryptocurrencies still require additional improvements. This challenging domain presents researchers in the field with new and exciting questions. I provide examples from two main research threads, related to the scalability of the protocol and to its underlying incentives.

1 Introduction

The Bitcoin protocol at its core is primarily tasked with synchronizing a ledger of transactions between different participants in the network. Participating nodes collaboratively work to establish an agreed upon record of transactions. If there is consensus on the contents of the ledger, then in fact, there is complete agreement over the ownership of all funds.

Bitcoin bundles together the security of the currency with an incentive system that supports those who participate. Designed as a decentralized P2P system, it gains a higher level of security and stability as more resources are devoted to its operation. Thus, the rewards given to participants in exchange for their work promote the system's long term security.

Consensus mechanisms for synchronizing data in distributed systems have been known for quite some time, but Bitcoin, which was introduced by a pseudonymous creator named Satoshi Nakamoto [2008], had successfully overcome a previously unsolved challenge: How to achieve consensus in an open "permissionless" system, i.e., one in which anyone can freely partake. The ledger maintained within Bitcoin is supported by a data structure known as *the blockchain*. Each individual block is a collection of transactions that was approved by the system. Blocks are organized in a chain structure—as each block contains a cryptographic hash of its predecessor. If the system is to be secure against manipulation, records on the blockchain must become immutable after some time, otherwise transfers of money could be reversed or rerouted.

The main challenge in establishing consensus in the permissionless setting is assuring that attackers cannot launch *Sybil attacks* [Douceur, 2002] in which they freely enter the system under multiple assumed identities and subvert the protocol. Such attacks are feasible since the internet at its core does not provide any verification of identity. Conventional consensus protocols, that only provide guarantees of security as long as adversaries do not control a sufficient number of nodes in the system are susceptible to such manipulations. [1] Subverting the protocol in this way implies that the attacker can double-spend, i.e., use money repeatedly in several transactions by reverting payments after they had already been accepted.

Proof of work and the security of the Nakamoto Consensus Nakamoto's solution to the problem of Sybils and to double-spending was to utilize a new tool within the protocol: cryptographic proof-of-work (PoW) puzzles [Dwork *et al.*, 2003] that are required whenever a new block is added to the blockchain and imply that a significant amount of computation must be expended (Nodes that participate in block generation are called *miners*). While anyone can join the system and and set up a node, the effort required for block creation effectively limits attackers. When faced with two alternative continuations of the blockchain (a fork in the chain) the protocol is set to accept the version that has more proof of work invested in it. Thus, attackers that wish to have their version of events replace that of the rest of the network must out-pace the rest of the network in producing blocks.

Nakamoto shows in his original paper [Nakamoto, 2008] that as long as messages propagate quickly in the Bitcoin sytem, and as long as the attacker controls less than 50% of the computational resources, the probability that an attacker will launch a successful double-spend decreases exponentially fast.[2] Thus providing Bitcoin with some theoretical framework for its basic security.

[1] Byzantine fault tolerance in asynchronous systems can only be guaranteed if fewer than one third of the nodes are adversarial [Lamport *et al.*, 1982].

[2] See also subsequent analysis in several works [Rosenfeld, 2014; Sompolinsky and Zohar, 2015; Garay *et al.*, 2015; Pass *et al.*, 2016].

The security guarantees provided by Nakamoto fail if one of the aforementioned assumptions does not hold. If the attacker is not relatively weak, or if messages do not propagate quickly (relative to the block creation time). Below I elaborate on Bitcoin's underlying incentive scheme and on how it may break down and threaten the first assumption, and how Bitcoin's operation at larger scale threatens the second one. The two assumptions thus form the basis for discussion of the weaknesses of the current protocol, and direct us towards solutions to address these faults.

As miners increase the computational power invested in solving PoW puzzles, the rate of block creation may increase. The Bitcoin protocol reacts to such changes by adjusting the difficulty of the cryptographic puzzle, aiming to maintain an expected period of 10 minutes between consecutive blocks.

2 Bitcoin's Scalability Challenge

As Bitcoin adoption increases, additional users wish to transact with the currency. The network is then required to process more transactions per second. Unfortunately, the protocol itself comes with a built-in limit on the number of transactions: Blocks are currently limited to 1MB. As a typical transaction takes up around 0.5 KB, the parameters mentioned above along with a block inter-arrival time of 10 minutes imply a rate of approximately 3-4 transactions per second.

If more transactions are to be processed, either the size of blocks or the block creation rate must increase. In each case, the propagation time of blocks in Bitcoin's P2P system increases relative to the block creation rates, a fact that leads to decreased security of the protocol [Sompolinsky and Zohar, 2015; Garay *et al.*, 2015; Pass *et al.*, 2016]. Indeed, as block propagation slows, one node may create a block while completely unaware that its current view of the blockchain is incomplete – another block has been created to extend the blockchain, and is currently being relayed around the network. As the Bitcoin protocol only adopts blocks on the longest chain, one of the two blocks currently extending the chain will be discarded. Such discarded blocks do not contribute to the length of the main chain, and in fact represent computational effort that was wasted by honest participants. At extreme transaction rates, this waste implies that attackers no longer need to hold the majority of computational power to successfully double-spend.

2.1 GHOST and Inclusive Blockchains

One of the approaches to address the scalability challenge has been to revise the underlying protocol that governs the blockchain. In an early work [Sompolinsky and Zohar, 2015] we had noticed that blocks that were created in parallel and form forks in the network do not add to its security. In addition to deriving careful bounds on such security, we have also suggested taking off-chain blocks into account while deciding on the chain that nodes view as valid. The GHOST protocol (Greedy Heaviest Observed Sub-Tree) replaces the longest-chain rule used by the Nakamoto consensus. As each block references a single parent predecessor (via its cryptographic hash), blocks form a tree with the first block (the genersis block) as its root. The chain is then selected as follows:

- Begin at the root, and proceed towards the leaves.
- At each fork, proceed to the subtree with the greatest collective proof-of-work.
- Stop once an unextended block is reached. The selected chain is then the path taken from the root to this leaf.

Indeed, it can be shown that once the honest nodes agree about the existence of a block in the chain, it gains weight extremely fast. As GHOST considers the weight of blocks in the subtree and not just the chain, two blocks that are built in parallel to one another may disagree about the suffix of the chain, but both contribute weight the chain prefix that they share.

Unfortunately, the GHOST protocol in its original form is still susceptible to attacks at high transaction rates, specifically those that delay the processing of transactions. Modified versions that further restrict which blocks count towards the weight of a subtree perform better.

A companion paper on Inclusive Blockchain protocols [Lewenberg *et al.*, 2015b] appeared at the same time as GHOST and suggested a way to include not only the weight of blocks that were off-chain, but also their contents. The key insight in inclusive protocols was that in order to produce a consistent ledger, it is sufficient to provide a total ordering over all known blocks including blocks that are not on some selected chain. Under the inclusive protocol, each block lists all known predecessor blocks that had not been extended by any other known block. Blocks in this case establish a direct-acyclic graph structure. Once a main chain is selected (using the longes-chain rule or GHOST), producing a total order is simple: each block in the main chain adds its off chain predecessors just before it in the total order. Given a total order, a consistent ledger is then produced by reading all transactions in order (from the first block to the last one), and accepting all transactions other than those that are inconsistent with earlier accepted ones. That is, a transaction that represents the transfer of funds that had already moved is considered invalid and can be rejected.

Extensive game theoretic analysis from the paper shows that since miners can now collect a reward even if their blocks are not on the main chain they should adjust their behavior. Several nice benefits ensue from the change: First, miners are now incentivized to ensure that the transactions they select are less likely to be included in blocks that are created by others which greatly increases the transaction throughput (in the classic protocols conflicting blocks often contain the same transactions). Second, unlike in the classic protocol, users get a differentiated level of service depending on the amount of fees they offer to the miners. Finally, the protocol could better accommodate miners whose communication is slower. These would lose less money.

The GHOST and Inclusive protocols when applied together suggest that mechanisms relying on direct acyclic graphs can do better than regular chains. A direct acyclic graph provides a causal order over blocks, and a more comprehensive picture of the state of each miner at the time blocks are created. Given that block x lists block y among its predecessors, then x was provably created after y. Hence, it is possible to generalize chain based mechanisms altogether: given

a causal order, find a way to produce a consistent ledger of transactions using the contents of all blocks.

2.2 The SPECTRE Protocol

Most blockchain protocols effectively include some hard-coded upper bound on the propagation time of messages in the system. For the system to be secure, this upper bound must indeed hold. As a result, wide margins must be taken when selecting which parameter to use. Such is the case with Bitcoin's block creation rate. For the system to be secure, the expected 10 minute interval between blocks should surpass the propagation time of blocks. The end result is slow confirmation times for transactions that must wait for inclusion in blocks. In fact, the same problem of a bound on propagation time that directly affects processing of transactions occurs in other protocols as well.

The question then arises: can we produce a workable protocol that would not include an explicit bound on message delays? The SPECTRE protocol [Sompolinsky *et al.*, 2016] provides one such example. In SPECTRE, blocks form a DAG structure, but the protocol does not yield a robust total order. Instead, SPECTRE provides weaker properties that are still sufficient to obtain a consistent ledger. The benefit of using weaker guarantees, is that the protocol can now work without an explicit bound on message delays. Transactions become irreversible at a rate that depends on the true delay in the network, without explicitly encoding anything in the protocol. The result is a highly scalable DAG-based protocol that adjusts to changing network conditions.

The miner protocol in SPECTRE is straightforward: Each miner creates blocks that reference several predecessor blocks. Let $Past(x)$ denote all blocks in the DAG that precede x in the causal order represented by the DAG (that are reachable from x by following the hash references), and by $Future(x)$ all blocks that are preceded by x.

SPECTRE nodes hold an independent pairwise vote between pairs of blocks. The vote indicates if $x < y$ (i.e., that x *defeats* y), or if $y < x$ (Y *defeats* X). The voters are all blocks in the DAG. The vote of each block B with regards to a pair x, y is not encoded in its contents, but is rather inferred from the causal structure:

- If $x, y \in Past(B)$, then block B votes according to the majority decision of the DAG that precedes it: $Past(B)$, i.e., votes are recursively computed on the smaller DAG, and then adopted.
- If $x \in Past(B)$ but $y \notin Past(B)$, then block B votes $x < y$
- If $y \in Past(B)$ but $x \notin Past(B)$, then block B votes $y < x$
- Otherwise, $x, y \notin Past(B)$, and B adopts as his vote the majority decision of all blocks in $Future(B)$.

The protocol above terminates as the recursion operates over smaller and smaller DAGs each time. The result of the election is a relation $<$ that is obtained by the majority vote of all blocks. It is important to note that as we know from social choice theory, the pairwise relation $<$ may be non-transitive (this is known as the Condorcet paradox in social choice [Gehrlein and Fishburn, 1976]). Thus, SPECTRE does not obtain a full order over blocks. Instead, a different property is gained:

Property 1 (informal): If the attacker holds under 50% of the computational power, and if block X is published to honest nodes immediately, only blocks that are published soon after block X will ever defeat it (per the relation $<$, with high probability).

We show that this property can be utilized to construct a set of accepted transactions that is resilient to attacks. The caveat is that a pair of conflicting transactions may sometimes both be rejected by the protocol (this is a consequence of the existence of Condorcet cycles which imply the $<$ relation is non-transitive). This is not harmful to the operation of ledger used for payments, as conflicting transactions represent dishonest attempts to double-spend. Transfers by different individuals are in fact never in conflict and can be quickly accepted by the protocol.

2.3 Additional Scalalbility Approaches

In addition to protocols like SPECTRE, other PoW based protocols that had been suggested to improve scalability include Bitcoin-NG [Eyal *et al.*, 2016] that cleverly sets apart key blocks that carry PoW from micro blocks that carry transaction data, and the hares and rabbits protocol which is also DAG based [Bentov *et al.*, 2017].

An interesting line of work utilizes reductions from PoW based consensus to classic consensus mechanisms in which participants are known and Sybil attacks do not occur [Decker *et al.*, 2016; Kogias *et al.*, 2016; Pass and Shi, 2016; Abraham *et al.*, 2016]. This line of research usually produces mechanisms with different properties, most notably, current reductions result in protocols that are not self stabilizing and might not recover from catastrophic events such as wide-range network splits. On the other hand they build upon many advances in classic consensus and often obtain highly scalable protocols.

Off-chain transaction channels Another direction that has been suggested to scale cryptocurrencies is to utilize off-chain transaction channels. These make use of Bitcoin's scripting system to set up 'joint accounts' involving two participants on the blockchain. The balance within the channel can then be shifted from one user to the other using direct communication between the two. Channels only affect the blockchain when they are opened or closed and thus allow for multiple transfers going back and forth between the two individuals. Transacitons channels are designed to work in a trustless manner. If one of the participants stops the interaction, or worse, decides to withdraw more funds than they are entitled to, the other can stop them and often punish any wrongdoing by claiming all the funds in the channel. Several techniques to create such channels exist, including bi-directional channels [Decker and Wattenhofer, 2015], and one that leverage trusted executed environments [Lind *et al.*, 2016]. Channels can be composed to longer paths effectively establishing networks through which payments can be routed [Poon and Dryja, 2015]. Routing in such credit network requires additional privacy as well [Moreno-Sanchez *et al.*, 2015; Heilman *et al.*, 2017].

3 Incentives as a Foundation for Security

The need to ensure that a potential attacker's computational resources will be dwarfed by those of the rest of the network drove Nakamoto to design an incentive scheme into Bitcoin. Miners who successfully solve proof-of-work puzzles and create blocks are rewarded in bitcoins. Such payments come from two sources: newly minted money, and transaction fees that are paid by users. Rewards thus inherently support the security of Bitcoin.

Many economic and game theoretic questions naturally arise. Is the protocol in equilibrium, or can users manipulate it to gain more? Does the fee market behave well and will enough revenue be produced? Unfortunately, early research quickly recognized that several problems exist with the protocol. While miners get paid for PoW and block creation, they do not receive proper rewards for other actions such as the long term storage of transaction data, the propagation of messages, and more. In fact, it can be shown that miners have disincentive to propagate transactions to their competitors [Babaioff *et al.*, 2012].

Other economic issues arose with establishment of mining pools that allow miners to band together and split the rewards for blocks between them. Payment distribution mechanisms within pools lower the risks of participants, but are also subject to manipulation and attacks [Rosenfeld, 2011; Eyal, 2015; Lewenberg *et al.*, 2015a].

Another line of work beginning with [Kroll *et al.*, 2013] examines the incentives to mine as stated by the protocol. A key result in this direction of research is the discovery of a selfish mining strategy by Eyal and Sirer [2014] which reveals that miners that are sufficiently large and can communicate quickly can earn more than their fair share in the protocol by selectively delaying the publication of blocks they create. Such miners can then push others out of the market and eventually grow to own a majority of the computational resources. The latter outcome is particularly troubling since such miners can launch double-spending and censorship attacks on the protocol.

While Eyal and Sirer provided a particular strategy that earns more than honest behavior, they did not find the optimal deviation. In a follow-up paper [Sapirshtein *et al.*, 2015] we provided an algorithm that utilizes MDPs to produce the optimal deviation from the protocol.[3] This then allowed us to see how resilient Bitcoin is exactly to deviation from the protocol and also allowed us to examine the result of optimal deviations for some protocol variants that had been proposed to alleviate the problem. While Eyal and Sirer result required attackers to own some amount of computational resources in order to profit from a deviation, our work has shown that if propagation delays of blocks are accounted for in the model, attackers can in fact profit from deviating regardless of their size. The same type of analysis used for optimal deviations has subsequently been applied to find optimal double spending attacks and to better quantify the security of the protocol in the worst case [Sompolinsky and Zohar, 2016; Gervais *et al.*, 2016].

[3] Additional improvements over the startegy of Eyal and Sirer appear in [Nayak *et al.*, 2016].

4 Conclusions

Cryptocurrencies are a relatively new application domain in computer science, one that is quickly growing and changing. In spite of their recently growing adoption, the fate of cryptocurrencies is still unknown, as they have yet to enter widespread main-stream use. Disagreements over the future path of the protocol mixed with internal politics are now affecting the deployment of protocol changes to Bitcoin. Meanwhile, other cryptocurrencies that are less conservative appear on a regular basis and often contain more risky innovation. The era of decentralized cryptocurrencies thus brings with it a plethora of research questions and directions for further exploration spanning many sub-fields within computer science.

The establishment of public permissionless cryptocurrencies has also piqued the interest of the financial industry that now actively seeks to improve its infrastructure by applying modern practices. Distributed ledgers and blockchains along with their power to remove the third party from interactions have attracted attention in this regard. Many additional advances in public blockchains, such as the adoption of advanced cryptographic tools, smart contracts, and other techniques are thus making their way from the wild-west of open source permissionless cryptocurrencies into the more structured domain of closed financial systems.

Acknowledgments

The author is thankful for support by the Israel Science Foundation (grant 616/13) and by the Hebrew University's Cybersecurity Center (grant 039-9230).

References

[Abraham *et al.*, 2016] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *arXiv preprint arXiv:1612.02916*, 2016.

[Babaioff *et al.*, 2012] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 56–73. ACM, 2012.

[Bentov *et al.*, 2017] Iddo Bentov, Pavel Hubáček, Tal Moran, and Asaf Nadler. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. 2017.

[Decker and Wattenhofer, 2015] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18. Springer, 2015.

[Decker *et al.*, 2016] Christian Decker, Jochen Seidel, and Roger Wattenhofer. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, page 13. ACM, 2016.

[Douceur, 2002] John R Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[Dwork *et al.*, 2003] Cynthia Dwork, Andrew Goldberg, and Moni Naor. On memory-bound functions for fighting spam. In *Annual International Cryptology Conference*, pages 426–444. Springer, 2003.

[Eyal and Sirer, 2014] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[Eyal *et al.*, 2016] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59. USENIX Association, 2016.

[Eyal, 2015] Ittay Eyal. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE, 2015.

[Garay *et al.*, 2015] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[Gehrlein and Fishburn, 1976] William V Gehrlein and Peter C Fishburn. Condorcet's paradox and anonymous preference profiles. *Public Choice*, 26(1):1–18, 1976.

[Gervais *et al.*, 2016] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.

[Heilman *et al.*, 2017] Ethan Heilman, Foteini Baldimtsi, Leen Alshenibr, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted tumbler for bitcoin-compatible anonymous payments. 2017.

[Kogias *et al.*, 2016] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 279–296. USENIX Association, 2016.

[Kroll *et al.*, 2013] Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013. Citeseer, 2013.

[Lamport *et al.*, 1982] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[Lewenberg *et al.*, 2015a] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 919–927. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[Lewenberg *et al.*, 2015b] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 528–547. Springer, 2015.

[Lind *et al.*, 2016] Joshua Lind, Ittay Eyal, Peter Pietzuch, and Emin Gün Sirer. Teechan: Payment channels using trusted execution environments. *arXiv preprint arXiv:1612.07766*, 2016.

[Moreno-Sanchez *et al.*, 2015] Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Kim Pecina. Privacy preserving payments in credit networks. 2015.

[Nakamoto, 2008] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[Nayak *et al.*, 2016] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 305–320. IEEE, 2016.

[Pass and Shi, 2016] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model, 2016.

[Pass *et al.*, 2016] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. *IACR Cryptology ePrint Archive*, 2016:454, 2016.

[Poon and Dryja, 2015] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2015.

[Rosenfeld, 2011] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.

[Rosenfeld, 2014] Meni Rosenfeld. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009*, 2014.

[Sapirshtein *et al.*, 2015] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. *arXiv preprint arXiv:1507.06183*, 2015.

[Sompolinsky and Zohar, 2015] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

[Sompolinsky and Zohar, 2016] Yonatan Sompolinsky and Aviv Zohar. Bitcoin's security model revisited. *arXiv preprint arXiv:1605.09193*, 2016.

[Sompolinsky *et al.*, 2016] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. Spectre: A fast and scalable cryptocurrency protocol. Technical report, Cryptology ePrint Archive, Report 2016/1159, 2016. http://eprint.iacr. org, 2016.