Blockchain System Foundations

Mohammad Javad Amiri Sujaya Maiyya Victor Zakhary
Divyakant Agrawal Amr El Abbadi
Department of Computer Science, University of California, Santa Barbara
Santa Barbara, California
{amiri,sujaya-maiyya,victorzakhary,agrawal,amr}@cs.ucsb.edu

1 INTRODUCTION

Bitcoin [21] is considered the first successful global scale peer-to-peer cryptocurrency. The Bitcoin protocol explained by the *mysterious Nakamoto* allows financial transactions to be transacted among participants without the need for a trusted third party, e.g., bank, credit card company, or PayPal. Bitcoin eliminates the need for such a trusted third party by replacing it with a distributed ledger that is fully replicated among all participants in the cryptocurrency system. This distributed ledger is referred to as *blockchain*.

Blockchain is a secure linked list of blocks containing financial transactions that occur in the system and linked by hash pointers. The main challenge that Bitcoin addresses is to maintain a consistent view of this replicated blockchain in a secure and fault-tolerant manner in a *permissionless* setting and in the presence of malicious participants. Unlike *permissioned* settings where all the participants in the system are known *a priori*, a permissionless setting allows participants to freely join and leave the system without maintaining any global knowledge of the number of participants. To address these challenges, Bitcoin builds on foundations developed over the last few decades from diverse fields [22], but primarily from the fields of **cryptography** [8, 24], **distributed systems** [10, 16, 17] and **data management** [9, 19, 27].

Bitcoin uses a notion of *miners* who need to perform a computationally challenging *Proof of Work (PoW)* puzzle before they can add any block of transactions to the replicated blockchain. Since the PoW puzzle is computationally hard, very few miners can successfully solve the puzzle, and hence a successful miner can add a block to the blockchain and be guaranteed, with very high probability, to be unique. Many concerns have been raised about the wasted massive energy requirements to *mine* one Bitcoin block. This mining approach to determine the process eligible to add a new block to the block chain is in contrast to the distributed systems approach, that has been promoting the use of Byzantine Agreement or consensus, which is efficient and more egalitarian. In fact, consensus protocols such as Paxos have been quite successful in recent years in laying the foundations of large global scale data management system. Unfortunately, Paxos has many limitations, especially from a global cryptocurrency point of view, including the requirement of a permissioned setting, and that participants can only fail by crashing. An alternative to Paxos that tolerates malicious failures is Practical Byzantine Fault-Tolerance (PBFT) [10]. Although it tolerates malicious failures, PBFT still requires a permissioned setting, and requires a large number of message exchanges, hence does not scale to the large number of participants expected in permissionless cryptocurrencies.

In this tutorial, our goal is to present to the database community an in-depth understanding of state-of-the-art solutions for efficient scalable blockchains. We progress towards this goal by starting from a detailed description of the protocols and techniques underlying the design of Bitcoin. Since most recent innovations in blockchain design depend critically on consensus protocols in malicious settings, we outline the basic foundations of distributed fault-tolerant consensus protocols. This is followed by a discussion of recent state-of-the-art permissioned blockchains. Since the participants are known and identified, permissioned blockchains can benefit from many techniques developed in the area of distributed computing over decades for reaching consensus, replicating state, and broadcasting transactions. We discuss various aspects of permissioned blockchains in the context of confidentiality [2, 7], verifiability [6, 20], performance [4, 7, 12, 15, 25, 26], and scalability [3, 5, 11, 13].

The wide adoption of permissionless open blockchain networks by both industry and academia suggests the importance of developing protocols and infrastructures that support peer-to-peer atomic cross-chain transactions. A two-party atomic cross-chain swap protocol was originally proposed by Nolen [1, 23] and generalized by Herlihy [14] to process multi-party atomic cross-chain swaps. Both Nolan's protocol and its generalization by Herlihy use smart contracts, hashlocks and timelocks to achieve atomic cross-chain swaps. These protocols require synchronous network assumptions and are not fault-tolerant. We therefore present a recently proposed atomic fault-tolerant cross chain protocol [28]. Finally, we give an overview of Fides [18], a database system that can detect malicious behaviour using blockchain.

2 ACKNOWLEDGEMENT

Partially funded by NSF grants CNS-1703560 and CNS-1815733.

REFERENCES

[1] 2018. Atomic cross-chain trading. https://en.bitcoin.it/wiki/Atomic_cross-chain_trading.

[2] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. CAPER: a cross-application permissioned blockchain. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1385–1398.

[3] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. On Sharding Permissioned Blockchains. In *Second International Conference on Blockchain*. IEEE, 282–285.

[4] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. ParBlockchain: Leveraging Transaction Parallelism in Permissioned Blockchain Systems. In *39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1337–1347.

[5] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. SharPer: Sharding Permissioned Blockchains Over Network Clusters. *arXiv preprint arXiv:1910.00765* (2019).

[6] Mohammad Javad Amiri, Joris Duguépéroux, Tristan Allard, Divyakant Agrawal, and Amr El Abbadi. 2020. SEPAR: A Privacy-Preserving Blockchain-based System for Regulating Multi-Platform Crowdworking Environments. *arXiv preprint arXiv:2005.01038* (2020).

[7] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *arXiv preprint arXiv:1801.10228* (2018).

[8] Adam Back. 2002. Hashcash-a denial of service counter-measure. (2002).

[9] Philip A Bernstein, Vassos Hadzilacos, and Nathan Goodman. 1987. Concurrency control and recovery in database systems. (1987).

[10] Miguel Castro, Barbara Liskov, et al. 1999. Practical Byzantine fault tolerance. In *OSDI*, Vol. 99. 173–186.

[11] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. 2019. Towards Scaling Blockchain Systems via Sharding. In *SIGMOD Int. Conf. on Management of Data*. ACM.

[12] Christian Gorenflo, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. 2019. Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. In *Int. Conf. on Blockchain and Cryptocurrency (ICBC)*. IEEE, 455–463.

[13] Suyash Gupta, Sajjad Rahnama, Jelle Hellings, and Mohammad Sadoghi. 2020. ResilientDB: Global Scale Resilient Blockchain Fabric. *Proceedings of the VLDB Endowment* 13, 6 (2020), 868–883.

[14] Maurice Herlihy. 2018. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 245–254.

[15] Jae Kwon. 2014. Tendermint: Consensus without mining.

[16] Leslie Lamport et al. 2001. Paxos made simple. *ACM Sigact News* 32, 4 (2001), 18–25.

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

[18] Sujaya Maiyya, Danny Hyun Bum Cho, Divyakant Agrawal, and Amr El Abbadi. 2020. Fides: Managing Data on Untrusted Infrastructure. In *40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE.

[19] C Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, and Peter Schwarz. 1992. ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems (TODS)* 17, 1 (1992), 94–162.

[20] JP Morgan. 2016. Quorum whitepaper. *En línia]. Available: https://github. com/jpmorganchase/quorumdocs/blob/master/Quorum% 20Whitepaper% 20v0* 1 (2016).

[21] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[22] Arvind Narayanan and Jeremy Clark. 2017. Bitcoin's academic pedigree. *Commun. ACM* 60, 12 (2017), 36–45.

[23] Tier Nolan. 2013. Alt chains and atomic transfers. https://bitcointalk.org/index.php?topic=193281.msg2224949/msg2224949.

[24] Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 2 (1978), 120–126.

[25] Pingcheng Ruan, Dumitrel Loghin, Quang-Trung Ta, Meihui Zhang, Gang Chen, and Beng Chin Ooi. 2020. A Transactional Perspective on Execute-order-validate Blockchains. In *SIGMOD International Conference on Management of Data*. ACM, 543–557.

[26] Ankur Sharma, Felix Martin Schuhknecht, Divya Agrawal, and Jens Dittrich. 2019. Blurring the lines between blockchains and database systems: the case of hyperledger fabric. In *SIGMOD International Conference on Management of Data*. ACM, 105–122.

[27] Gerhard Weikum and Gottfried Vossen. 2001. *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*. Elsevier.

[28] Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. 2020. Atomic commitment across blockchains. *Proceedings of the VLDB Endowment* 13, 9 (2020), 1319–1331.