Verifiable Delay Functions for Permissionless Replicated Systems

Goals: Design a permissionless agreement protocol based on verifiable delay functions.

Tools: Logic, algorithmic reasoning, programming

Prerequisites: basic knowledge of distributed algorithms and cryptographic tools, basic concurrent programming skills, curiosity and persistence

The prominent *blockchain* technology aims at implementing a public "ledger": a decentralized consistent history of transactions proposed by an *open* set of participating processes, with no static membership. This problem can be seen as an instance of fault-tolerant *state-machine replication* [19], prominent examples of which are the *crash-tolerant* Paxos protocol by Lamport [16] and the BFT (*Byzantine* fault-tolerant) system by Castro and Liskov [5]. These systems use instances of *consensus* protocols in order to ensure that users get consistent views of the system evolution.

Principal downside of classical consensus protocols are lack of scalability and the need for a fixed or properly reconfigurable set of participants out of which only a bounded fraction (up to one third) can be faulty. This can be hard to ensure in an *open* (also called *permissionless* system, where an arbitrary fraction of participants can be controlled by the adversary [9]. Prominent blockchain protocols [18,20] achieve (nondeterministic) consistency by assuming that (1) the system is synchronous, (2) participants can use asymmetric cryptography, and (3) the adversary can control at most a minority (in practice, a minor fraction) of computing power.

Intuitively, these assumptions are used to overcome the folklore CAP theorem [3, 10] stating that no system can combine Consistency, Availability, and Partition-Tolerance. In particular, these protocols avoid partitioning by enforcing the *proof of work* (PoW) mechanism requiring that a participant must solve a time-consuming cryptographic puzzle before updating the ledger. The resulting protocols are notoriously slow and energy-demanding. More recent blockchain prototypes propose to obviate the energy demands via using *proof-of-stake* [1, 13]. However, these solutions are subject to multiple attacks enabling forks, such as the "nothing-at-stake attack", where many miners are incentivized to extend every branch in a potential fork, and the "long range attack", where a long alternative branch is held privately by the adversary in order to overtake the longest chain. More recent proposals [7] resort to synchronous networks, rely on expensive cryptographic assumptions, and/or impose restrictions on honest players. An immediate question is whether these costs and assumptions are unavoidable.

To mitigate these attacks and, possibly, improve performance, one may invest into *verifiable delay functions* (VDFs) [2], a recently proposed cryptographic mechanism based on puzzles that can only be solved *sequentially*, but without wasting energy on meaningless intensive computations. There is an evidence that VDFs can be used to replace PoW in *Nakamoto consensus* [17], however we still do not understand how this mechanism can be used in more general contexts.

This project intends to explore the potential of using VDFs in permissionless implementations of replicated services, not only consensus-based [6,16], but also weaker abstractions: reliable broadcast [12], lattice agreement [14, 15] and cryptocurrencies [8, 11].

Milestones

1. Study the literature on permissionless computing [4,7,18] and modern cryptographic tools [2].
2. Define a range of permissionless system model that allows the use of VDFs for consistent data replication: synchrony assumptions, failure patterns, the use of trusted cryptographic setup.
3. Implement a replicated state machine and weaker abstractions in the resulting models and study their performance.

Contact

Duong Hieu Phan
`https://www.di.ens.fr/users/phan/`
`hieu.phan@telecom-paris.fr`
INFRES, Télécom ParisTech, Institut Polytechnique Paris

Petr Kuznetsov
`http://www.infres.enst.fr/~kuznetso/`
`petr.kuznetsov@telecom-paristech.fr`
INFRES, Télécom ParisTech, Institut Polytechnique Paris

References

[1] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies without proof of work. In *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, pages 142–157, 2016.

[2] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch. Verifiable delay functions. In H. Shacham and A. Boldyreva, editors, *CRYPTO*, volume 10991 of *Lecture Notes in Computer Science*, pages 757–788. Springer, 2018. `https://eprint.iacr.org/2018/601.pdf`.

[3] E. A. Brewer. Towards robust distributed systems (abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 7–, 2000.

[4] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X. Zhang. Combining GHOST and casper. *CoRR*, abs/2003.03052, 2020.

[5] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *OSDI: Symposium on Operating Systems Design and Implementation*. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, Feb. 1999.

[6] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, Nov. 2002.

[7] J. Chen and S. Micali. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.*, 777:155–183, 2019.

[8] D. Collins, R. Guerraoui, J. Komatovic, P. Kuznetsov, M. Monti, M. Pavlovic, Y. A. Pignolet, D. Seredinschi, A. Tonkikh, and A. Xygkis. Online payments by merely broadcasting messages. In *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 - July 2, 2020*, pages 26–38. IEEE, 2020.

[9] J. R. Douceur. The sybil attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, pages 251–260, 2002.

[10] S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[11] R. Guerraoui, P. Kuznetsov, M. Monti, M. Pavlovic, and D. Seredinschi. The consensus number of a cryptocurrency. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019.*, pages 307–316, 2019.

[12] R. Guerraoui, P. Kuznetsov, M. Monti, M. Pavlovic, and D. Seredinschi. Scalable byzantine reliable broadcast. In *DISC 2019*, 2019. To appear, TR: https://arxiv.org/abs/1908.01738.

[13] A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 357–388, 2017.

[14] P. Kuznetsov, T. Rieutord, and S. Tucci Piergiovanni. Reconfigurable lattice agreement and applications. In P. Felber, R. Friedman, S. Gilbert, and A. Miller, editors, *23rd International Conference on Principles of Distributed Systems, OPODIS 2019, December 17-19, 2019, Neuchâtel, Switzerland*, volume 153 of *LIPIcs*, pages 31:1–31:17, 2019.

[15] P. Kuznetsov and A. Tonkikh. Asynchronous reconfiguration with byzantine failures. In H. Attiya, editor, *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*, volume 179 of *LIPIcs*, pages 27:1–27:17, 2020.

[16] L. Lamport. The Part-Time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

[17] J. Long and R. Wei. Nakamoto consensus with verifiable delay puzzle. *CoRR*, abs/1908.06394, 2019.

[18] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, May 2009. `https://bitcoin.org/bitcoin.pdf`.

[19] F. B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4):299–319, Dec. 1990.

[20] G. Wood. Ethereum: A secure decentralized generalized transaction ledger. White paper, 2015.