

Research Statement

Amitanand S. Aiyer

Research interests I am interested in modeling real-world situations and developing provably correct algorithms for addressing practical problems. In this regard, I have been working closely with theory and systems researchers to develop algorithms and abstractions for various problems in distributed systems. I started working in distributed algorithms as an undergraduate student in the theory group at IIT-Madras, and have then been working in the systems group at UT-Austin. My research focuses on developing techniques to improve the resilience and availability of distributed register implementations [2, 3, 4], and to improve the throughput of replicated state machines [5].

Distributed registers Registers are used to abstract the guarantees provided by storage systems, by specifying the set of values that the system is allowed to return under different scenarios. Different register implementations offer different trade-offs between consistency and performance for any given set of operating conditions. Depending on the desired consistency requirements and workload, applications can use different register implementations.

It is well-known that under conditions involving a network partition, good consistency and high availability may not be attained simultaneously [20]. Early approaches have opted for better consistency, by being unavailable during network partitions; while more recent approaches, seek to provide only a probabilistic guarantee that clients read the latest value, and choose high availability at the cost of possibly returning arbitrarily old values. Neither of these approaches are applicable to systems that cannot tolerate arbitrarily old values, but still desire high availability. For such systems, we have introduced the notion of k -atomic semantics which ensures that the register always returns one of the k latest written values. I have designed protocols to implement k -atomic registers and shown that, when writes are infrequent, these protocols can be used to achieve a higher availability than existing non-probabilistic approaches [2, 3].

For systems requiring stronger freshness guarantees, I have also designed protocols for implementing an atomic register in a bounded wait-free manner, allowing clients to complete their operations in a finite number of steps regardless of the progress made by other clients. The best known protocol to provide such guarantees could only tolerate up to $n/4$ Byzantine faults [10]. I have designed a protocol that improves upon these results to tolerate up to $n/3$ Byzantine faults, which is optimal [4].

In order to apply my skills to practical systems, I have also been actively seeking research internships at various industrial laboratories. These internship opportunities have helped me better understand production systems and to identify useful notions for describing them. During one such internship at HP-Labs, we observed that existing notions of consistency semantics, which only focus on the worst case guarantees provided by the system, are insufficient to articulate the guarantees provided by these systems effectively. Large scale storage systems that are used to support online businesses in production are required to be highly available and may consequently provide weak consistency guarantees during certain conditions; however, under normal operating conditions they do provide a much stronger guarantee which is not captured by existing notions [7, 15, 18]. For example, the Dynamo [18] system, which supports the shopping-cart application on amazon.com, returns the latest written value for read operations as long as there are only a few failures. However, if there are more than a certain number of failures, the system goes through a hinted hand-off mechanism to store values on a different set of replicas; when this happens, the system may also return older values. The guarantees provided by these systems cannot be compared by studying the worst case alone, and I have worked with researchers at HP-Labs to formalise the concept of consistability for comparing systems that provide consistency and performance guarantees which vary under different operating conditions [7].

Replicated state machines Replicated state machines (RSM) provide an abstraction of a single reliable node that is capable of performing any deterministic operation. RSM protocols implement this abstraction over a set of unreliable machines by using Byzantine fault-tolerance techniques that coordinate replicas to agree on performing all the operations in the same order. Ensuring that the performance overhead is acceptable in practice requires that these implementations use message authentication codes (MACs) instead of digital signatures [1, 12, 16, 21]. While digital signatures provide stronger properties, they are 2-3 orders of magnitude slower to compute and verify than MACs.

Recently however, it has been shown that these MAC-based implementations are vulnerable to certain advanced attacks that can render their performance unusable in the event of failures. To address this problem, researchers have reverted back to using digital signatures, but doing so decreases the throughput that can be achieved [14]. I have instead been looking at techniques which can use MACs to provide the same properties as digital signatures. I have shown that if fewer than $n/3$ processes are faulty, one can use the concept of matrix-signatures, which uses a matrix of $n \times n$ MACs, to provide all the properties required of digital signatures [5]. This construction has been used in building the RSM protocol for the UpRight system, to ensure that it handles such attacks and provides good performance [13].

Prior work I have also worked on designing and implementing a replicated state machine protocol for cooperative services. Unlike systems that are centrally managed, machines participating in a cooperative service may have different goals and objectives. It may, thus, be inappropriate for systems to assume that all non-faulty processes will follow the protocol correctly. We model cooperative distributed systems as consisting of three kinds of processes (BAR model): Byzantine processes that may behave unpredictably, altruistic processes that will always follow the protocol, and finally, rational processes that will follow the protocol unless they gain by deviating from it. Using this model, we have designed a RSM protocol for cooperative systems and used it to build a prototype peer-to-peer storage system [6].

During my undergraduate program, I have also designed protocols for implementing the distributed consensus primitive under the synchronous model. It is well-known that if processes can only communicate over point-to-point channels, distributed consensus protocols can tolerate up to $n/3$ Byzantine faults [22]. However, in the presence of a broadcast channel up to $n/2$ Byzantine faults can be tolerated [9]. I have studied the possibility of using multicast channels, to a subset of processes, for tolerating more than $n/3$ faults. Using the non-threshold model, which provides more detailed information on the set of processes that can fail simultaneously, I have characterised the conditions when consensus is possible and designed algorithms for achieving it [8].

Social networks to mitigate sybil attacks Currently, I am studying the problem of using social networks to mitigate the sybil attack in distributed systems. Sybil attacks allow the adversary to control a large fraction of processes in a distributed system by having many sybil identities join the system [19]. While centralized admission control and resource limiting have been considered as an effective means to prevent sybil attacks, these techniques may also discourage honest users from joining the system. Recently, researchers have proposed the idea of using social networks to bound the number of sybil identities accepted into the system. The number of sybil identities accepted can be bounded only if the number of links between honest nodes in the system to those that are by the adversary is small [24, 23, 17].

It is, however, unclear if such an assumption can hold in practice. Studies have shown that it is relatively easy to create online friendships with strangers and to impersonate others in various online social networks [11]. I am working with social networking researchers from the University of Rome to characterise the conditions under which an adversary may succeed in creating many edges to the honest nodes and the conditions under which it cannot.

References

- [1] Michael Abd-El-Malek, Gregory R. Ganger, Garth R. Goodson, Michael K. Reiter, and Jay J. Wylie. Fault-scalable Byzantine fault-tolerant services. In *Proceedings of the 20th ACM Symposium on Oper-*

- ating Systems Principles*, pages 59–74. ACM Press, 2005.
- [2] **Amitanand S Aiyer**, Lorenzo Alvisi, and Rida A. Bazzi. On the availability of non-strict quorum systems. In *Proceedings of the 19th International Symposium on Distributed Computing*, pages 48–62. Springer-Verlag, 2005.
 - [3] **Amitanand S Aiyer**, Lorenzo Alvisi, and Rida A. Bazzi. Byzantine and multi-writer k-quorums. In *Proceedings of the 20th International Symposium on Distributed Computing*, pages 443–458. Springer-Verlag, 2006.
 - [4] **Amitanand S Aiyer**, Lorenzo Alvisi, and Rida A. Bazzi. Bounded wait-free implementation of optimally resilient Byzantine storage without (unproven) cryptographic assumptions. In *Proceedings of the 21st International Symposium on Distributed Computing*, pages 443–458. Springer-Verlag, 2007.
 - [5] **Amitanand S Aiyer**, Lorenzo Alvisi, Rida A. Bazzi, and Allen Clement. Matrix Signatures: From MACs to Digital Signatures. In *22nd International Symposium on Distributed Computing*, pages 16–31. Springer-Verlag, 2008.
 - [6] **Amitanand S Aiyer**, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. BAR fault tolerance for cooperative services. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles*, pages 45–58, New York, NY, USA, 2005. ACM.
 - [7] **Amitanand S Aiyer**, Eric Anderson, Xiaozhou Li, Mehul A. Shah, and Jay J. Wylie. Consistability: Describing usually consistent systems. In *Proceedings of the 4th Workshop on Hot Topics in Syetms Dependability*, December 2008.
 - [8] **Amitanand S Aiyer**, Sanketh Indarapu, Srinathan Kannan, Vinod Vaikuntanathan, and C. Pandu Rangan. Distributed consensus in the presence of sectional faults. In *Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing*, pages 202–210, New York, NY, USA, 2003. ACM.
 - [9] O. Babaoglu and R. Drummond. Streets of byzantium: Network architectures for fast reliable broadcasts. *IEEE Trans. Softw. Eng.*, 11(6):546–554, 1985.
 - [10] Rida A. Bazzi and Yin Ding. Bounded wait-free f-resilient atomic Byzantine data storage systems for an unbounded number of clients. In *Proceedings of the 20th International Symposium on Distributed Computing*, pages 299–313. Springer-Verlag, 2006.
 - [11] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web*, pages 551–560, New York, NY, USA, 2009. ACM.
 - [12] Miguel Castro and Barbara Liskov. Practical Byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, pages 173–186. USENIX Association, 1999.
 - [13] Allen Clement, Manos Kapritsos, Sangmin Lee, Yang Wang, Lorenzo Alvisi, Mike Dahlin, and Taylor Riche. Upright cluster services. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles*, pages 277–290, New York, NY, USA, 2009. ACM.
 - [14] Allen Clement, Mirco Marchetti, Edmund Wong, Lorenzo Alvisi, and Mike Dahlin. Making Byzantine fault tolerant systems tolerate Byzantine faults. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, pages 153–168, April 2009.
 - [15] Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, and Ramana Yerneni. PNUTS: Yahoo!’s hosted data serving platform. *Proc. VLDB Endow.*, 1(2):1277–1288, 2008.

- [16] James Cowling, Daniel Myers, Barbara Liskov, Rodrigo Rodrigues, and Liuba Shrira. HQ replication: A hybrid quorum protocol for Byzantine fault tolerance. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, pages 177–190, Berkeley, CA, USA, November 2006. USENIX Association.
- [17] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of 16th Annual Network and Distributed Systems Security Symposium*, February 2009.
- [18] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: Amazon’s highly available key-value store. *SIGOPS Operating Systems Review*, 41(6):205–220, 2007.
- [19] J Douceur. The sybil attack. In *In: Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002). Volume 2429 of LNCS*. Springer, 2002.
- [20] Seth Gilbert and Nancy Lynch. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.
- [21] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. Zyzzyva: Speculative byzantine fault tolerance. *SIGOPS Operating Systems Review*, 41(6):45–58, 2007.
- [22] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.
- [23] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17. Citeseer, 2008.
- [24] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM ’06*, pages 267–278, New York, NY, USA, 2006. ACM.