Compositional Reasoning about Randomized Distributed Protocols

Internship proposal (2024 – 2025)

Supervision: Constantin Enea

LIX, CNRS, Ecole Polytechnique

cenea@lix.polytechnique.fr

Research context. Abstraction is key to the design and verification of large, complicated software. In particular, distributed systems are designed as a modular composition of various kinds of protocols that implement common abstractions such as consensus (reaching agreement on some value or a sequence of values), leader election (electing a leader in a network of nodes), secret sharing (sharing a value among a set of nodes without disclosing it to an adversary), etc. Designing abstractions that compose correctly with randomization, or with programs that should not leak information, is very challenging.

Modeling abstractions and implementations thereof as labeled transition systems (LTSs), a classic refinement relation (between an implementation and a corresponding abstraction) is trace inclusion: the set of traces of the implementation is included in the set of traces of the abstraction. While refinement preserves safety and liveness properties, i.e., properties of individual traces, it does *not* preserve hyperproperties [17], which are properties of sets of traces [5]. In the context of randomized protocols or programs that should not leak information, many interesting properties can not be expressed as safety/liveness properties, but as hyperproperties. Notable examples are security properties such as *noninterference* [8], stipulating that commands executed by users with high clearance have no effect on system behavior observed by users with low clearance. Other examples are quantitative properties like bounds on the *probability distribution* of events, e.g., the mean response time over sets of executions.

Therefore, a number of works in the literature have proposed various ways of strengthening the relationship between implementations and abstractions, e.g.,

- for concurrent objects, Golab et al. have introduced the notion of strong linearizability [9] which was later proved to be equivalent to the existence of forward simulations [2, 7]
- for cryptographic protocols, the notions of simulatability [10, 11, 3] and universal compositionality [4, 1, 12] have been introduced.

Objectives. In this internship, we plan to investigate this issue for randomized distributed protocols that solve classic agreement problems like consensus. These protocols are typically built from various components, see [13, 6] for instance, that compose in different ways. We will explore abstraction frameworks that make it possible to vary the tradeoff between the simplicity of the abstraction and the complexity of proving it correct. These abstractions will be defined in an automata theoretic framework based on Labeled Transition Systems [16] and their probabilistic extensions [14, 18, 15].

Skills. During the internship, we will rely on knowledge of concepts in the area of automated formal verification, algorithmic reasoning, concurrency theory, and automata theory.

References

[1] Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *J. Cryptol.*, 30(1):58–151, 2017.

[2] Hagit Attiya and Constantin Enea. Putting strong linearizability in context: Preserving hyperproperties in programsthat use concurrent objects. In *DISC*, pages 2:1–2:17, 2019.

[3] Michael Backes, Jörn Müller-Quade, and Dominique Unruh. On the necessity of rewinding in secure multiparty computation. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 157–173. Springer, 2007.

[4] Ran Canetti, Alley Stoughton, and Mayank Varia. Easyuc: Using easycrypt to mechanize proofs of universally composable security. In *32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019*, pages 167–183. IEEE, 2019.

[5] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010.

[6] Luciano Freitas de Souza, Petr Kuznetsov, and Andrei Tonkikh. Distributed randomness from approximate agreement. In Christian Scheideler, editor, *36th International Symposium on Distributed Computing, DISC 2022, October 25-27, 2022, Augusta, Georgia, USA*, volume 246 of *LIPIcs*, pages 24:1–24:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[7] Brijesh Dongol, Gerhard Schellhorn, and Heike Wehrheim. Weak progressive forward simulation is necessary and sufficient for strong observational refinement. In Bartek Klin, Slawomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory, CONCUR 2022, September 12-16, 2022, Warsaw, Poland*, volume 243 of *LIPIcs*, pages 31:1–31:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[8] Joseph A. Goguen and José Meseguer. Security policies and security models. In *1982 IEEE Symposium on Security and Privacy, Oakland, CA, USA, April 26-28, 1982*, pages 11–20. IEEE Computer Society, 1982.

[9] Wojciech Golab, Lisa Higham, and Philipp Woelfel. Linearizable implementations do not suffice for randomized distributed computation. In *STOC*, page 373–382, 2011.

[10] Dennis Hofheinz and Dominique Unruh. Comparing two notions of simulatability. In Joe Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 86–103. Springer, 2005.

[11] Dennis Hofheinz and Dominique Unruh. Simulatable security and polynomially bounded concurrent composability. In *2006 IEEE Symposium on Security and Privacy (S&P 2006), 21-24 May 2006, Berkeley, California, USA*, pages 169–183. IEEE Computer Society, 2006.

[12] Yehuda Lindell. How to simulate it - A tutorial on the simulation proof technique. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 277–346. Springer International Publishing, 2017.

[13] Julian Loss and Tal Moran. Combining asynchronous and synchronous byzantine agreement: The best of both worlds. *IACR Cryptol. ePrint Arch.*, page 235, 2018.

[14] Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. Compositionality for probabilistic automata. In Roberto M. Amadio and Denis Lugiez, editors, *CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings*, volume 2761 of *Lecture Notes in Computer Science*, pages 204–222. Springer, 2003.

[15] Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. Observing branching structure through probabilistic contexts. *SIAM J. Comput.*, 37(4):977–1013, 2007.

[16] Nancy A. Lynch and Frits W. Vaandrager. Forward and backward simulations: I. untimed systems. *Inf. Comput.*, 121(2):214–233, 1995.

[17] John McLean. A general theory of composition for trace sets closed under selective interleaving functions. In *1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, May 16-18, 1994*, pages 79–93. IEEE Computer Society, 1994.

[18] Roberto Segala. Probability and nondeterminism in operational models of concurrency. In Christel Baier and Holger Hermanns, editors, *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Bonn, Germany, August 27-30, 2006, Proceedings*, volume 4137 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2006.