TPDP, WPES and CCS talks

Sebastian Meiser
University College London, United Kingdom, e-mail: s.meiser@ucl.ac.uk

October 15, 2018

Abstract

Summaries of talks I attended.

1 Invited talk TPDP: Composition, Verification, and Differential Privacy

Speaker: Justin Hsu

After introducing differential privacy in general, we see a few well-known composition results. Post-processing is seen as an instantiation of sequential composition of an (ε, δ)-DP mechanism with a $(0, 0)$-DP mechanism, which is kind of nice. Similarly, local DP is presented as an instantiation of parallel composition.

1.1 Formally ferifying privacy

Our goals here are twofold: we want to have *dynamic* verificatoin, i.e.,raise errors if DP is violated and *static* verification, i.e., checking DP on all possible inputs. As a side-note, we want to simplify verification by using composition results; composition allows verification techniques to have much better automation.

The Fuzz family of languages allows for describing each type with a metric and, using group privacy ($(\varepsilon, 0)$-DP for $\Delta_f = 1$ implies $(k \cdot \varepsilon, 0)$-DP for $\Delta_f = k$). The description allows for an immediate static analysis that can use both sequential and parallel composition. These languages, so far, cannot deal well with ADP, but the speaker is currently working on that.

Privacy as an appriximate coupling Idea: verify privacy by game hops. We somehow relate pairs of sampling instructions to show properties of a pair of inputs, but I'm not sure how exactly we do that. This static-analysis technique seems to be related to how Tim and me modeled differential indistinguishability, i.e., as a proprety of a pair of programs/machines. They, so far, cannot automatically generate proofs, but can check them.

1.2 Advanced composition

They analyze the old "advanced composition theorem" by Dwork, Rothblum and Vadhan. The speaker says that the analysis is more complicated, since the composition theorem needs to be applied "as a block", instead of on small units of the program. The speaker then mentions that novel approaches (e.g., by Mironov on RDP) makes formal verification easier, as it allows to analyze a program step-by-step. I think that using our privacy buckets approach, such an analysis could be improved even more and made flexible as well.