Prover efficient delegation of computations.

Cryptology, exact computations, outsourcing.

Contacts:	Jean-Guillaume.Dumas@univ-grenoble-alpes.fr, tel: 0 457 421 732.
Laboratoire:	Jean Kuntzmann (LJK), IMAG - CS 40700 700 avenue centrale, 38058 Grenoble. ljk.imag.fr
Earnings:	Standard by the LJK.

In an emerging computing paradigm, computational capabilities, from processing power to storage capacities, are offered to users over communication networks as a service.

This new paradigm holds enormous promise for increasing the utility of computationally weak devices. A natural approach is for weak devices to delegate expensive tasks, such as storing a large file or running a complex computation, to more powerful entities (say servers) connected to the same network. While the delegation approach seems promising, it raises an immediate concern: when and how can a weak device verify that a computational task was completed correctly? This practically motivated question touches on foundational questions in **cryptography, coding theory, complexity theory, proofs and algorithms**.

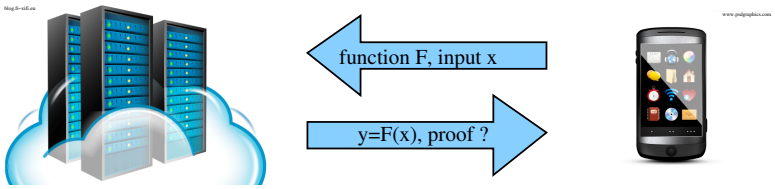

Figure 1: Verifying the computation, should take less time than computing it

More generally, the question of verifying a result at a lower cost (time, memory) than that of recomputing it, is of paramount importance. Another example of application is for the extension of the trust about results computed via probabilistic or approximate algorithms. There the idea is to gain confidence into the correctness, but only at a cost negligible when compared to that of the computation.

Major ideas to realize this are to use sum check protocols [7, 6, 9, 1], **homomorphic encryption** or **pairings** [8, 5, 3]. In the last 5 years, progress was made towards generic protocols with Verifier efficiency, but quite often with a very large overhead for the overall computational cost [10].

The goal of this thesis is instead to have a look at dedicated protocols (for instance **linear algebra** or **polynomial computations** [2, 4]) in order to combine probabilistic methods and verifiable computing. First results shows promising efficiency and it now seems time to build upon those, in order to extend the practicality of many other building blocks.

References

[1] E. Ben-Sasson, A. Chiesa, A. Gabizon, M. Riabzev, and N. Spooner. Interactive oracle proofs with constant rate and query complexity. In I. Chatzigiannakis, P. Indyk, F. Kuhn,

and A. Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 40:1–40:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[2] J.-G. Dumas and E. Kaltofen. Essentially optimal interactive certificates in linear algebra. In K. Nabeshima, editor, *ISSAC'2014, Proceedings of the 2014 ACM International Symposium on Symbolic and Algebraic Computation, Boston, USA*. ACM Press, New York, July 2014.

[3] J.-G. Dumas and V. Zucca. Prover efficient public verification of dense or sparse/structured matrix-vector multiplication. In J. Pieprzyk and S. Suriadi, editors, *ACISP 2017, 22nd Australasian Conference on Information Security and Privacy*, volume 10343, pages 115–134. Springer, July 2017.

[4] K. Elkhiyaoui, M. Önen, M. Azraoui, and R. Molva. Efficient techniques for publicly verifiable delegation of computation. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 119–128, New York, NY, USA, 2016. ACM.

[5] D. Fiore, C. Fournet, E. Ghosh, M. Kohlweiss, O. Ohrimenko, and B. Parno. Hash first, argue later: Adaptive verifiable computations on outsourced data. In *ACM Conference on Computer and Communications Security (CCS)*. ACM, Oct. 2016.

[6] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: interactive proofs for muggles. In C. Dwork, editor, *STOC'2008, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada*, pages 113–122. ACM Press, May 2008.

[7] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, Oct. 1992.

[8] B. Parno, J. Howell, C. Gentry, and M. Raykova. Pinocchio: Nearly practical verifiable computation. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 238–252, Washington, DC, USA, 2013. IEEE Computer Society.

[9] J. Thaler. Time-optimal interactive proofs for circuit evaluation. In R. Canetti and J. Garay, editors, *Advances in Cryptology - CRYPTO'13*, pages 71–89. Springer Berlin Heidelberg, 2013.

[10] M. Walfish and A. J. Blumberg. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, Jan. 2015.

See also: Microsoft Research. Verifiable Computing:
www.microsoft.com/en-us/research/project/verifiable-computing