THE MATHEMATICS OF SNARKS: POLYNOMIALS, FINITE FIELDS, ELLIPTIC CURVES AND ERROR CORRECTING CODES

CARLA RÀFOLS

Abstract. In the last few years we have seen how SNARKs (Succinct Non-Interactive Arguments of Knowledge) have moved from the theory of complexity and cryptography to applied cryptography and engineering. We will review the applications of SNARKs and the mathematical ingredients that have made this possible.

Introduction

The PCP theorem [1, 5], that states that every language in NP can be probabilistically verified with a constant number of randomized checks to a very long proof, is a very beautiful and fundamental theorem in the theory of complexity. The "very long proof" is a redundant encoding of the witness, the piece of information owned by the prover that allows to decide membership in the language efficiently.

At an intuitive level, one way to interpret the PCP theorem is as saying that probabilistic verification of many computations can be done much more efficiently than redoing the entire computation. Therefore, in principle, it implies efficient solutions to the following cryptographic problems:

(1) Delegation of Computation: a single powerful party (the prover) can compute a proof that a certain computation is correct that is super efficient to verify, so that the verifier can be a resource-constrained device that could not do the computation on its own. More generally, this means that expensive computations need to be carried out only once, and posterior verifications can be done in a very cheap way by any party.

(2) Zero-Knowledge Proofs (ZKPs) [7]: with some additional tweaks, a proof that asserts that certain computation is correct can be usually turned into a ZKP, that is, a proof without any additional information leakage. This means that the verifier only learns that the claim of the prover is correct and does not gain any knowledge of the prover's secret information (the witness). At a high level, this means that ZKPs allow to prove that complex computations over secret data are correct in a way that is both cheap and secure (privacy compliant) to verify.

Although these applications of the theorem were studied from relatively early on [9, 10], it is not so simple to construct concretely efficient and scalable solutions to any of these problems in practice. Thus, for quite some time, these applications of the PCP theorem were considered less of a success and did not receive as much as attention as others, like the study of hardness of approximation [5].

The situation has changed in the last decade due to several breakthroughs [8, 6] that have allowed to construct truly efficient *Succinct Non-Interactive Arguments of Knowledge*

(SNARKs), that is, short proofs of statements that are efficient to create and superefficient to verify. Succinctness implies that the proof is much shorter than the witness (in fact, it is usually of constant size) so even very complex statements (encoded as very large arithmetic circuits over some finite field) are very efficient to verify. Starting from the work of [2], there are also many ongoing engineering efforts (e.g. [3, 4]) to use SNARKs in real-world applications. The long term vision is that SNARKs are a fundamental tool not only for privacy (because of the zero-knowledge property) but also for scalability in decentralized settings where nodes need to agree on a common state, since a single short (or succinct) proof can replace the complex history that would otherwise need to be stored by any party who wants to verify it.

This talk will deal with mathematical aspects of practical SNARKs and explain the connection to their main building blocks: finite fields, error correcting codes, polynomials, Number Theoretic Fourier Transforms or cycles of elliptic curves.

In particular, we will explain in some detail how to express arithmetic circuit satisfiability as a polynomial divisibility problem, which, starting from the work of Gentry et al. [6], is one of the main tools to achieve succinctness. We will review joint work with Arantxa Zapico [11], in which we construct very efficient SNARKs that have a property, universal and updatable setup, that is very desirable in practice since it avoids the need for a trusted third party in the setup stage.

References

[1] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; a new characterization of NP. *FOCS*, (1992), 2-13, 10.1109/SFCS.1992.267824.

[2] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. *IEEE S&P*, (2014), 459-474, 10.1109/SP.2014.36.

[3] Eli Ben-Sasson and Iddo Bentov and Yinon Horesh and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol. ePrint Arch.* (2018), 46

[4] Joseph Bonneau, Izaak Meckler, Vanishree Rao, and Evan Shapiro. Coda: Decentralized Cryptocurrency at Scale. *IACR Cryptol. ePrint Arch.* (2020), 352

[5] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra and Mario Szegedy. Approximating clique is almost NP-complete. *FOCS*, (1991), 2-12, 10.1109/SFCS.1991.185341

[6] Rosario Gennaro, Craig Gentry, Bryan Parno and Mariana Raykova. Quadratic Span Programs and Succinct NIZKs without PCPs. *EUROCRYPT*, (2013), 626–645, 10.1007/978-3-642-38348-9_37

[7] Shafi Goldwasser, Silvio Micali and Charles Rackoff. The knowledge complexity of interactive proof-systems. *STOC*, (1985), 10.1145/22145.22178

[8] Jens Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. *ASIACRYPT*, (2010), 321-340, 10.1007/978-3-642-17373-8_19

[9] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. *STOC*, (1992), 723–732, 10.1145/129712.129782

[10] Silvio Micali. Computationally Sound Proofs. *SIAM J. Comput.*, 30 (4), (2000), 1253–1298, 10.1137/S0097539795284959

[11] Carla Ràfols and Arantxa Zapico. An Algebraic Framework for Universal and Updatable SNARKs. *CRYPTO*, (2021), 774–804, 10.1007/978-3-030-84242-0_27.

Carla Ràfols; Universitat Pompeu Fabra, Department of Information and Communication Technologies

Email address: `carla.rafols@upf.edu`