Survey of ZK-SNARK

YUNCONG ZHANG*, Shanghai Jiao Tong University

ACM Reference Format:
Yuncong Zhang. 2020. Survey of ZK-SNARK. 1, 1 (June 2020), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Zero-Knowledge Succinct Non-interactive ARgument of Knowledge (zkSNARK) [1] enables verifying computation outputs without knowing the inputs and faster than the original computation. Currently, zkSNARKs are under active research, particularly due to their applications in blockchains [2, 3], where zkSNARKs facilitate creating confidential transactions that conceal part or all of the transaction details. Recent years have seen an explosion of zkSNARK implementations enjoying different properties, including constant-size proofs [4–8], universal or trustless setups [9–14], and post-quantum security [12, 13].

However, the rapid development of zkSNARK poses considerable challenges for researchers to keep up with the state-of-the-art. Existing zkSNARK implementations rely on a large and growing number of underlying tools. It is increasingly complicated to understand the complete toolset and uncover its full potential to construct more efficient and secure zkSNARKs. It is also tricky to assess and compare existing schemes due to the high-dimensionality of measurement metrics, including efficiency, security, and functionality. Existing studies trying to overview this field are oversimplifying and never tried to illustrate existing concrete implementations [15–17]. A comprehensive survey of literature in zkSNARKs can serve as an anchor of knowledge in this field of research, provide an overview of the most significant ideas behind current implementations, and inspire new perspectives to understand zkSNARKs.

In this paper, we present a survey that overviews the current status of zkSNARKs. First, we discuss the concepts that are necessary to understand the related literature. Then we recall the history of zkSNARKs and examine the motivations and insights behind each major contribution. Finally, we propose a framework for classifying and evaluating the zkSNARK implementations in terms of efficiency, functionality, expressiveness, infrastructure, building blocks, and security assumptions. Using this framework, we reveal potential ways to leverage the existing tools and patterns to construct zkSNARKs with improved properties.

2 PRELIMINARIES

2.1 zkSNARK

A zkSNARK is called *preprocessing* if the Setup algorithm takes the circuit C as input. Fig. 1 illustrates how preprocessing zkSNARK works.

Author's address: Yuncong Zhang, yczhangsjtu@163.com, Shanghai Jiao Tong University, Dongchuan Rd. 800, Minhang, Shanghai, 200240.

XXXX-XXXX/2020/6-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

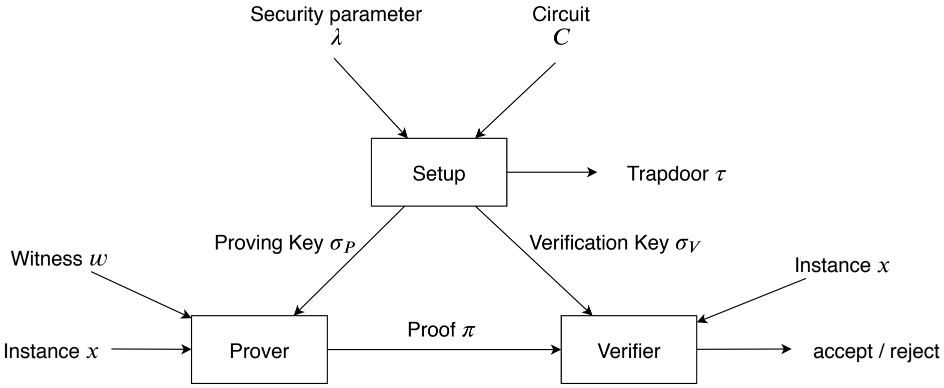

Fig. 1. Preprocessing zkSNARK

The Setup **algorithm takes as inputs both the security parameter λ and a circuit C, and outputs common reference strings, i.e. the proving key σ_P and the verification key σ_V, to the prover and verifier respectively.** Setup **also outputs a trapdoor τ which enables simulating proofs without witness. The** Prove **algorithm takes as inputs an instance x with corresponding witness w, together with proving key σ_P, and outputs the proof π as validation of x. The** Verify **algorithm verifies the correctness of x with proof π and verification key σ_V, and decides if to accept or reject.**

3 EFFICIENCY

In evaluating the efficiency of zkSNAKRs, the properties we care most about are proving time, verification time, proof sizes, and common reference string sizes. Fig. 2 summarizes the asymptotic complexities of current zkSNARK implementations.

REFERENCES

[1] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 326–349. ACM, 2012.

[2] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 459–474. IEEE Computer Society, 2014.

[3] Shifeng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II*, volume 10493 of *Lecture Notes in Computer Science*, pages 456–474. Springer, 2017.

[4] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326. Springer, 2016.

[5] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 626–645. Springer, 2013.

[6] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for C: verifying program executions succinctly and in zero knowledge. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2013.

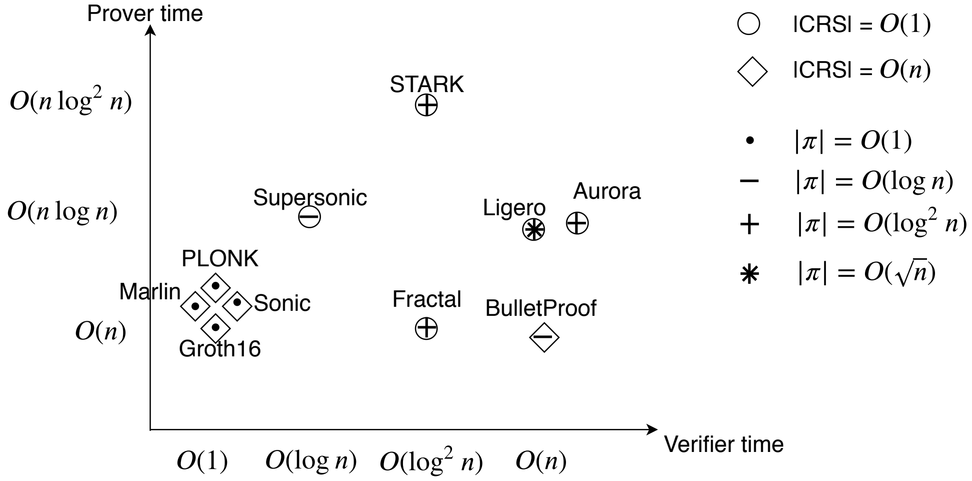

Fig. 2. Asymptotic complexities of zkSNARK implementations

The big-O notation here hides the security parameter λ. The n denotes the statement size. For circuit-based zkSNARKs, n is the circuit size. For statements with succinct representation, e.g. STARK, n is the length of execution trace, i.e. the size of the circuit representing the unrolled computation.

[7] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 238–252. IEEE Computer Society, 2013.

[8] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive arguments for a von neumann architecture. *IACR Cryptol. ePrint Arch.*, 2013:879, 2013.

[9] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-snarks. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 698–728. Springer, 2018.

[10] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings. *IACR Cryptol. ePrint Arch.*, 2019:99, 2019.

[11] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from DARK compilers. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 677–706. Springer, 2020.

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

[13] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128. Springer, 2019.

[14] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2087–2104. ACM, 2017.

[15] E. Tromer. D. Benarroch, L. T. A. N. BrandaĬČo. Zkproof community reference, 2019. https://zkproof.org.

[16] Anca Nitulescu. A gentle introduction to snarks. 2019.

[17] Michael Walfish and Andrew J. Blumberg. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, 2015.