A Fully Anonymous e-Voting Protocol Employing Universal zk-SNARKs and Smart Contracts

Aritra Banerjee

ADAPT Centre, School of Computer Science and Statistics,
Trinity College Dublin, Dublin, Ireland,
abanerje@tcd.ie

Abstract. The idea of smart contracts has been around for a long time. The introduction of Ethereum has taken the concept of smart contracts to new heights because of its integration with Blockchain technology. As a result, the applications of smart contracts have also surged in areas such as e-Voting, Insurance, Crowdfunding, etc. In this paper, we aim to present the construction of a "Fully Anonymous e-Voting" protocol using the concepts of zkHawk and Zcash. zkHawk is a novel smart contract protocol designed during this Ph.D. that improves upon the Hawk protocol by solving the underlying anonymity problem of a trusted manager. We will leverage the concept of zk-SNARKs in Zcash to carry out the voting phase of the election and the zkHawk smart contract protocol to tally the results of the election. The voting phase employing Zcash will be initially designed with Non-Universal zk-SNARKs and improved upon with Universal zk-SNARKs.

Keywords: Zcash, e-Voting, Blockchain, Hawk, Smart Contracts, zk-SNARKs

1 Introduction

e-Voting is a classic example of a smart contract evaluation [1]. Advancements in Blockchain have made e-Voting more private and decentralized [2,3]. There have been some recent developments using the Ethereum smart contracts [4] for private boardroom voting. Related works have shown leveraging a multi-party computation (MPC) program for the tallying phase in the e-Voting procedure [5,6,7]. Smart contracts are generally used during the "tallying" procedure in e-Voting. The design of zkHawk [8] facilitates off-chain Smart contract execution that ensures privacy. Therefore, zkHawk can be used in e-Voting for such cryptocurrency blockchains which do not support Smart contracts like Zcash [9] or Monero [10,11] instead of Ethereum [12]. The construction of zkHawk was inspired by Hawk [13] but it added in a feature whereby there was no need to have the trust assumption of a manager. Instead, an MPC program would exe-

cute the smart contract while maintaining the zero-sum constraint[1]. The smart contract execution is done off-chain between the participants by running the MPC and the blockchain is just used to validate the transactions and execute contract closure[2].

1.1 Problem Statement

The research goal of this Ph.D. is to improve upon the underlying concerns of Privacy and Scalability [14,2] in e-Voting using zkHawk [8] and Zcash [9] protocols. Zcash e-Voting was first suggested in 2017 by Pavel et al. [14] where each voter is given a Zcash wallet from which they can send one coin to a candidate that is counted as one vote for the candidate, and then use the JoinSplit transaction [15] of Zcash to tally the votes and declare the results. The problem with this approach is the massive number of zk-SNARK computations that will occur if there are a large number of voters. To solve this problem, we suggest leveraging the zkHawk protocol in the "tallying" phase such that there is a significant number of zk-SNARK computations while maintaining the privacy and anonymity we get from Zcash.
Assumption: The voters are registered, and they have been invited to vote.
Goal: To develop a fully anonymized e-Voting protocol that uses Zcash tokens for e-Voting and leverages zkHawk for tallying the votes and declaring the results.

2 Proposal

We now present an overview of the novel Universal zkHawk e-Voting protocol.

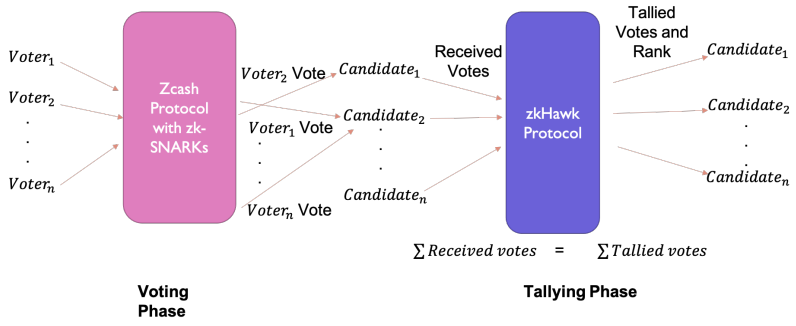

Fig. 1. An Overview of the suggested zkHawk e-Voting protocol

[1] The sum of the input balance of a smart contract is equal to the sum of the output balance
[2] Contract closure signifies that a smart contract has closed and all the payouts have been completed

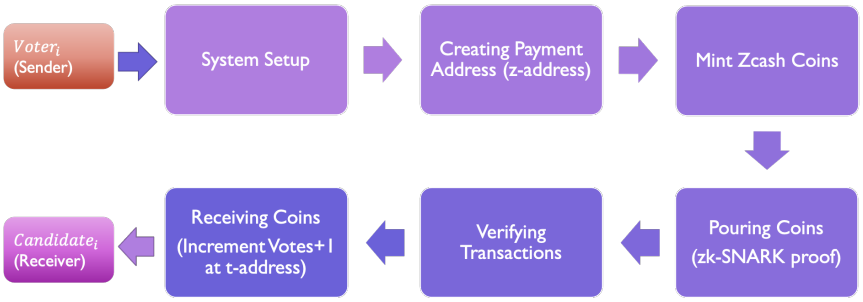

Fig. 2. Step 1: The Voting Phase with zk-SNARKs and Zcash

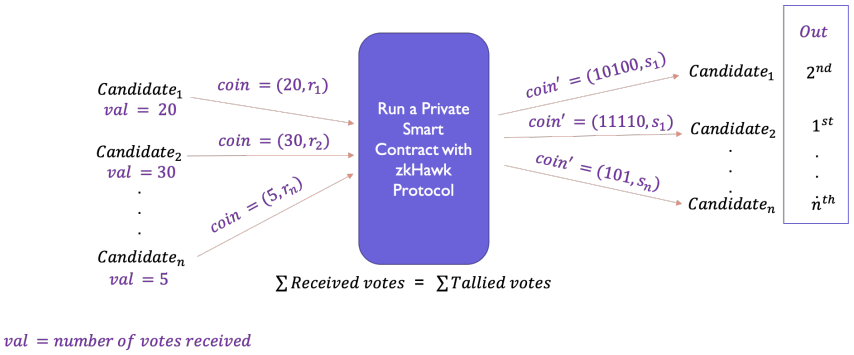

Fig. 3. Step 2: The Tallying Phase with zkHawk

2.1 Computational Advantage

This section discusses exactly how much we can improve computationally when using zkHawk e-Voting instead of Zcash JoinSplit e-Voting. The privacy remains the same but the number of zk-SNARK [16,17,18] computations drastically reduces when using zkHawk e-Voting. Let the number of zk-SNARK computations in vanilla Zcash e-Voting be E and the number of computations in zkHawk e-Voting be F, then mathematically speaking:
if,

$$\begin{aligned} Voters &= x \\ Candidates &= y \end{aligned} \tag{1}$$

then,

$$\begin{aligned} E &= x * (y + 1) \\ F &= x \end{aligned} \tag{2}$$

Suppose, there are 500 Voters and 10 Candidates, then using Equation (1) and (2) instead of 5500 zk-SNARK computations (E) only 500 zk-SNARK com-

putations (F) have to be performed, i.e. this protocol reduces the number of zk-SNARK computations by $1/10^{th}$ as compared to vanilla Zcash e-Voting.

3 Evaluation Plan

Currently, the fastest and the smallest known zk-SNARK is the Groth16 [17] algorithm which is used in Zcash. But this algorithm is Non-Universal and hence the trusted setup is always tied to one specific circuit e.g. one-to-one money transfer. Zcash recently performed a zk-SNARK trusted setup ceremony in 2018 for a one-to-one direct money transfer relation i.e., you can transfer money from A to B, B to C, etc. But, for Zcash to support smart contracts or other relations each time a zk-SNARK trusted setup ceremony has to be performed which is computationally very expensive. Hence, the concept of Universal zk-SNARKs [19,20,21,22,23,24] is introduced. Universal zk-SNARKs imply that all circuits (i.e., relations) can be validated using just one trusted setup. The CRS (Common Reference String) [25] generated in Universal zk-SNARKs is "updatable" and "unending" (also called SRS) [26]. The SRS can support one-to-one token transfer, smart contracts, and other relations without needing to create a separate trusted setup for each relation. And since the SRS is Updatable, anytime any information is leaked any honest party can go to the SRS and update the parameters. We will first design the protocol using Non-Universal zk-SNARKs for the Voting phase, then move on to Universal zk-SNARKs for its feasibility and computational inexpensive nature. Contemporary Universal zk-SNARKs that will be explored during this Ph.D.:

- **Sonic** (2019) [19] is the first Universal zk-SNARK introduced which uses the universal and updatable SRS [26].
- **Plonk** (2019) [20] is an improvement on Sonic which has a significantly lower prover time than Sonic (around 5 times better)
- **Marlin** (2020) [21] is another improvement on Sonic with 10 times better prover time and 4 times better verification time.
- **Mirage** (2020) [22] was suggested by the authors of Hawk and this protocol has linear Universal circuit operations instead of quasi-linear universal circuits. It is built on the Groth16 [17] protocol instead of Sonic.
- **Lunar** (2020) [23] used polynomial holographic IOPs[3] (a new variant of IOPs) to give a very small proof size and prover time.

4 Conclusion

The first step of this Ph.D. was to design a fully anonymous off-chain Smart Contract protocol inspired by Hawk [13] but would omit the requirement to trust a manager. This construction was achieved as designed in [8] and is necessary for the next step of the Ph.D. After successfully designing zkHawk, our next step

[3] Interactive Oracle Proofs [27]

is to leverage this concept and design an e-Voting protocol that is fast, private, and scalable. This novel protocol also uses the concepts of zk-SNARKs (Non-Universal and Universal) and Zcash. Furthermore, we will provide an elaborate construction of the designed protocol in our future work and implement the algorithm for bench-marking against other e-Voting protocols in terms of privacy and computational complexity.

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland grants 13/RC/2106 (ADAPT) and 17/SP/5447 (FinTech Fusion). This work was also supported in part by Science Foundation Ireland grant 13/RC/2094 (Lero). I would also like to take the opportunity to thank my supervisor Dr. Hitesh Tewari for his flawless guidance in this PhD. A special acknowledgement is due for Dr. Michael Clear, a postdoc in Trinity working closely with Hitesh and me for his guidance.

References

1. Tso, R., Liu, Z.Y., Hsiao, J.H.: Distributed e-voting and e-bidding systems based on smart contract. Electronics 8(4), 422 (2019)
2. Hjálmarsson, F., Hreiđarsson, G.K., Hamdaqa, M., Hjálmtýsson, G.: Blockchain-based e-voting system. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp. 983–986. IEEE (2018)
3. Kshetri, N., Voas, J.: Blockchain-enabled e-voting. IEEE Software 35(4), 95–99 (2018)
4. McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. In: International Conference on Financial Cryptography and Data Security. pp. 357–375. Springer (2017)
5. Khader, D., Smyth, B., Ryan, P., Hao, F.: A fair and robust voting system by broadcast. Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft fur Informatik (GI) pp. 285–299 (2012)
6. Küsters, R., Liedtke, J., Müller, J., Rausch, D., Vogt, A.: Ordinos: a verifiable tally-hiding e-voting system. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 216–235. IEEE (2020)
7. Cortier, V., Gaudry, P., Yang, Q.: A toolbox for verifiable tally-hiding e-voting systems. Cryptology ePrint Archive, Report 2021/491 (2021), https://eprint.iacr.org/2021/491
8. Banerjee, A., Clear, M., Tewari, H.: zkhawk: Practical private smart contracts from mpc-based hawk. arXiv preprint arXiv:2104.09180 (2021)
9. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474. IEEE (2014)
10. Van Saberhagen, N.: Cryptonote v 2.0 (2013)
11. Noether, S.: Ring signature confidential transactions for monero. IACR Cryptol. ePrint Arch. 2015, 1098 (2015)
12. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper 3(37) (2014)

13. Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE symposium on security and privacy (SP). pp. 839–858. IEEE (2016)
14. Tarasov, P., Tewari, H.: The future of e-voting. IADIS International Journal on Computer Science & Information Systems 12(2) (2017)
15. Hopwood, D., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification. GitHub: San Francisco, CA, USA (2016)
16. Banerjee, A., Clear, M., Tewari, H.: Demystifying the role of zk-snarks in zcash. In: 2020 IEEE Conference on Application, Information and Network Security (AINS). pp. 12–19. IEEE (2020)
17. Groth, J.: On the size of pairing-based non-interactive arguments. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 305–326. Springer (2016)
18. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 321–340. Springer (2010)
19. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2111–2128 (2019)
20. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. IACR Cryptol. ePrint Arch. 2019, 953 (2019)
21. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.: Marlin: Preprocessing zksnarks with universal and updatable srs. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–768. Springer (2020)
22. Kosba, A., Papadopoulos, D., Papamanthou, C., Song, D.: {MIRAGE}: Succinct arguments for randomized algorithms with applications to universal zk-snarks. In: 29th {USENIX} Security Symposium ({USENIX} Security 20). pp. 2129–2146 (2020)
23. Campanelli, M., Faonio, A., Fiore, D., Querol, A., Rodrıguez, H.: Lunar: a toolbox for more efficient universal and updatable zksnarks and commit-and-prove extensions. ERINT IACR, Report 1069(2020), 101 (2020)
24. Ràfols, C., Zapico, A.: An algebraic framework for universal and updatable snarks. Cryptology ePrint Archive, Report 2021/590 (2021), https://eprint.iacr.org/2021/590
25. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications. In: Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 329–349. ACM (2019)
26. Groth, J., Kohlweiss, M., Maller, M., Meiklejohn, S., Miers, I.: Updatable and universal common reference strings with applications to zk-snarks. In: Annual International Cryptology Conference. pp. 698–728. Springer (2018)
27. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive oracle proofs. In: Theory of Cryptography Conference. pp. 31–60. Springer (2016)