In this post, I share my paper VTBC on privatizing the volume and timing of blockchain transactions (with an implementation using Ethereum). Full paper can be found here. It has been accepted to ICCCN 2023 in July.

Problem:

Existing privacy-preserving blockchain solutions can maintain confidentiality and anonymity; however, they cannot privatize the volume and timing of transactions for an application. This is problematic for volume-dependent or time-dependent applications, such as a Dutch auction where everything is priced at $10 on day 1, $5 on day 2, etc (time-dependent), and there are only 10 items for sale (volume-dependent).

Such an auction cannot be implemented currently on blockchains in a privacy-preserving manner because volume and timing metadata for an applications' transactions is always leaked due to core blockchain architecture. This means it will always leak information like number of sales, bids, the sellers' revenue, etc. which all may want or need to be privatized in many situations.

Or, think of a grading policy for student assignments which is time-dependent and volume-dependent. For example, students can submit late for a 10% penalty and/or submit multiple times for a 10% penalty. Currently, the public volume and timing metadata can be used to deduce information about the students' grades, even if all submissions are anonymous and confidential.

[

image

2146×574 47.2 KB

](https://ethresear.ch/uploads/default/original/2X/e/eea3f4e0817915bf2c4b0a6f2911120fe8a59e74.png)

Solution

: The solution proposed in this paper is to build on top of existing privacy-preserving solutions (zkSNARKs, the Hawk paper's model) and create applications which support decoy, no-op transactions. Decoy transactions are simply no-op transactions that do not contribute to the outcome of the application but are used to obfuscate the overall volume and timing dataset because they are indistinguishable from "real" transactions.

For example, if we have a student exam deadline where exams can be late or on-time, students can obfuscate the volume and timing of their submission by submitting one real and one decoy submission on either side of the deadline. The grading function will take in both submissions but never leak which one was real and which was fake.

For enforcing adequate obfuscation of the volume and timing metadata, we show that applications can define K time periods that correspond to all possible outcomes and enforce that all users must submit >1 transaction during each of the K time periods, or else, they are "disqualified". If transactions are submitted outside the time period, those transactions are ignored. These rules help to maintain that sufficient noise is added to not leak any useful information to adversaries.

In the paper, we propose a solution based on the Hawk multi-party privacy-preserving blockchain application model which uses a minimally trusted manager to help facilitate the application. The manager is trusted for maintaining privacy; however, they are not trusted for correctness of execution

. They are not to be equated with a trusted third party. The correctness of execution can be publicly verified by anyone to be fair and honest (due to the properties of zkSNARKs and using the blockchain as the trusted timekeeper).

Results:

We evaluated our method via an Ethereum private blockchain and tested with up to N=128 inputs / transactions. We found that our proposed method is implementable and deployable on a blockchain such as Ethereum but can add significant overhead (especially as N or the number of decoy transactions increases). Libraries (contracts) can exceed 160 KB in size, and transactions can exceed 12m gas (30m limit per block).

We believe that, over time, our approach will continue to become more scalable and reasonable for a public blockchain like Ethereum (as zkSNARKs and blockchain scalability continue to improve). For now, our solution is suitable to private or permissioned blockchain environments, where resources are not as scarce.

Feel free to ask any questions below!