VDF-based MEV resilience evaluation

Paper: https://www.overleaf.com/5194187246hsjpgnzjtssr#5cf0d9

Setting

Transactions are created by the system's users, who typically operate non-specialized commercial hardware. Therefore, the evaluation of the proposed mechanisms (VDF and/or PoW) should be done on such hardware. We propose using a standard business laptop which should be expected to be used by the users of the system (e.g., below $1,500 retail price).

Research questions

1. What is the relationship between time and cost needed to compute a certain amount of PoW?
2. How long does it take to verify a VDF output that has been evaluated over x amount of time?
3. How much does it cost (in terms of energy consumption) to evaluate a VDF over x amount of time?

Research question 1

Background. The main idea of the VDF-based MEV protection is that the time needed to produce a transaction is approx. equal to the system's liveness parameter, such a MEV-attacking transaction cannot be created while censoring the honest transaction. The enhanced MEV protection makes use of a combination of VDF and PoW. Here, VDF is used to delay the production rate of a transaction, whereas PoW aims at increasing its production cost. The time needed to produce a transaction should be approx. equal to the system's liveness parameter; less time would worsen resilience against censorship attacks (which imply MEV attacks), whereas more time would worsen the system's usability. Therefore, we want to evaluate the relationship between increasing the cost and the time needed to compute PoW (on a transaction).

Client cost. For the first part, on the chosen commercial hardware, we want to measure the energy cost of running the PoW loop for x minutes. Specifically, we want to measure the kW energy consumption for: (i) x=30 sec, (ii) x=1 min, (iii) x=2 min. The evaluation can be done by either (i) running the loop for x amount of time and measuring energy consumption directly, or (ii) using the nominal energy consumption of the hardware's CPU/GPU at full utilization to estimate the expected cost.

Specialized hardware cost. For the second part, we want to measure the number of hashes that the chosen hardware can perform for the values of x given above. Then, we want to evaluate the cost of performing the same amount of hashes on various specialized hardware units, e.g., various ASICs. This can be done by using the ASIC's nominal energy consumption per hash.

Hash function comparison. Third, we want to perform the same evaluations as the first and second parts for different hash functions, specifically: (i) SHA-256, (ii) ETHash, (iii) Equihash.

Repeated evaluation. For all of the above evaluations, we need repeated executions in order to estimate the average cost in each case. We propose repeating each evaluation 50 times and computing the mean and the median values in each case.

Research questions 2 and 3

Background. There are two candidate VDFs (Wesolowski's and Pietrzak's), which have been implemented in various open source projects.[1] For research question 2, we want to evaluate the time relationship between computing ("evaluating") the VDF on a given value (e.g., a transaction) and verifying the VDF's output (e.g., proof). In our proposed mechanism, the evaluation of the VDF is done on a transaction by the system's users, while the verification of the VDF proof is done by the block creators. Therefore, the goal is to evaluate the time overhead that verifying the VDF imposes on block creators. Additionally, for research question 3, we want to measure the energy cost that the system's users need to pay in order to create a transaction.

Measurement. The input value should be a PoEM transaction. For the given input, we want to evaluate the VDF over a time that corresponds to the following number of confirmations on PoEM:[2] (i) 1, (ii) 10, (iii) 40, (iv) 70.[3] We want to measure two values: (i) for RQ2, the verification time for each computed VDF evaluation; (ii) for RQ3, the energy cost of computing each VDF evaluation.

Repeated evaluation. For each number of confirmations, we want to repeat the evaluation 50 times and compute the mean and the median values of each case's verification time and computation cost.

[1] See https://docs.google.com/document/d/1Tq3-9y2krlD5e66ErsZLph2W61rl8pVbulgKgnQxON4/edit for a short list of them.
[2] Confirmations here refer to a chain of blocks, e.g., 10 confirmations is equivalent to the creation of a chain of 10 blocks.
[3] These values are typically used for confirmations on Kraken: https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times