This is a follow-up to an earlier research post.

For Eth 2.0 we need a VDF $F(x)=y$

so that

- there exists a succinct proof $\pi(x,y)$

that can be found only for x,y

as above.

- there exists a lower bound on the latency of F

and π

, which can be matched closely on existing hardware.

The current plan is to use an RSA-based VDF where $F=g^{2^t}\bmod{N}$

is a sequence of t

squarings modulo an RSA modulus N

with unknown factorization. The proof π

is constant size, and can be computed in $O(t/\log t)$

time. In practice computing π

can be parallelized to make its latency be negligible compared to the actual computation of y

.

There is an STARK-based construction alternative to an RSA-based VDF. It works as follows:

- The VDF function F

is a MiMC-like construction whose iterations are represented as low-degree polynomials. Our round function is$(X,Y) \leftarrow ((X+Y)^{(p-1)/3},2Y);$

where X,Y

are $n>64$

-bit elements of some field F_p

. Note that the doubling of Y

is crucial; otherwise precomputation attacks allow to invert a VDF with a little amortized cost.

- An alternative VeeDo by StarkWare doesalike
- $(X,Y) \leftarrow (X^{1/3},Y^{1/3})$
- $(X,Y) \leftarrow M\cdot (X,Y) +C_r$

,

where M

is a matrix and C_r

are round constants. It seems to be more expensive to compute with the same latency.

- $(X,Y) \leftarrow (X^{1/3},Y^{1/3})$
- $(X,Y) \leftarrow M\cdot (X,Y) +C_r$

,

where M

is a matrix and C_r

are round constants. It seems to be more expensive to compute with the same latency.

- The proof is a STARK proof of correctness that y=F(x)

where x

and y

are public inputs. The STARK prover benefits from the fact that F

is invertible and has low degree (3) in the reverse direction, which makes the prover efficient.

The obvious benefit of STARK-based VDS is that its prover is post-quantum and does not need a trusted setup. However, the disadvantage is that the prover is more expensive compared to the RSA prover, i.e. makes O(t\log^a t)

operations compared to O(t/\log t)

of the RSA one. As a result, the prover running time becomes the dominant term in a VDF run, and, given it is parallelizable, the VDF latency becomes more volatile and we may miss the second requirement to a good VDF.

We thus face the following questions:

1. Do we have to extend the construction for bigger state/smaller field to further increase the precomputation protection ?

2. Is the resulting VDF fully post-quantum, i.e. is F

secure as a post-quantum hash? Do precomputation attacks have quantum speed up beyond Grover's search algorithm with square-root complexity?

1. Should we consider a VDF proof as a part of the VDF output formally, work with only a single latency parameter and optimize it alone?

2. What would the minimal hardware that would make the prover latency close to the lower bound? Is it much bigger than that for RSA?