There are a lot of work done for improving proof generation. However, for use cases besides anonymous transactions(which we can use zcash sprout/sapling parameters), there seems no solution for trusted setup. Users have to perform their own trusted setup and let others to trust them, which is very inconvenient.

So here I propose to use Intel SGX to alleviate the problem, even though that it cannot completely solve the problem.

Steps:

1, Users use python/js/c/rust to describe their trusted setup procedure for specific use case, which includes procedures to remove all toxic waste. The program is open to everyone and can be audited.

2, SGX hardware takes the logic, uses embedded random source as seeds to generate proving key and verifying key, then delete the toxic waste, with a quote generated.

Alternatively, one can use the SGX secrets when manufactured but unknown to Intel, to deterministicly derive all random numbers for trusted setup.

3, The quote is transmitted to Intel Attestation Service (IAS), which verifies the report and produces a final report with Intel's certificate chain.

4, Everybody can verify the final report and be conviced that the verifying key is rightly generated and all toxic waste is deleted.

Problems & possible solutions:

1, Intel may deny legal quotes. It can be made indistinguishable to Intel whether or not it's a trusted setup program.

2, Intel may collude with some SGX users to store toxic waste or use fake SGX. It can be avoided by allowing free participation and randomly selecting SGX providers.

3, Intel may manipulate randomness in SGX with some trapdoor. It can be alleviated with tricks in 2.

References:

1, Costan V, Devadas S. Intel SGX Explained[J]. IACR Cryptology ePrint Archive, 2016, 2016(086): 1-118.