ZoKrates 0.8.8
Jacob Eberhardt, Thibaut Schaeffer, Stefan Deml, Darko Macesic
Supports generation of zkSNARKs from high level language code including Smart Contracts for proof verification on the
Ethereum Blockchain.
'I know that I show nothing!'

USAGE:
zokrates [FLAGS] <SUBCOMMAND>

FLAGS:
-h, --help Prints help information
-V, --version Prints version information
--verbose Verbose mode

SUBCOMMANDS:
check Checks a program for errors
compile Compiles into a runnable constraint system
compute-witness Calculates a witness for a given constraint system
export-verifier Exports a verifier as Solidity smart contract
generate-proof Calculates a proof for a given constraint system and witness
generate-smtlib2 Outputs the constraint system in the SMTLib2 format
help Prints this message or the help of the given subcommand(s)
inspect Inspects a compiled program
mpc Multi-party computation (MPC) protocol
nova Nova IVC
print-proof Prints proof in the chosen format
profile Profiles a compiled program, indicating which parts of the source yield the most constraints
setup Performs a trusted setup for a given constraint system
universal-setup Performs the universal phase of a trusted setup
verify Verifies a given proof with the given verification key

zokrates-compile
Compiles into a runnable constraint system

USAGE:
    zokrates compile [FLAGS] [OPTIONS] --input <FILE>

FLAGS:
        --debug   Include logs
    -h, --help   Prints help information
    -V, --version   Prints version information
        --verbose   Verbose mode

OPTIONS:
    -s, --abi-spec <FILE>   Path of the ABI specification [default: abi.json]
    -c, --curve <curve>   Curve to be used in the compilation [default: bn128] [possible
values: bn128,
                                bls12_381, bls12_377, bw6_761, pallas, vesta]
    -i, --input <FILE>   Path of the source code
    -o, --output <FILE>   Path of the output binary [default: out]
    -r, --r1cs <FILE>   Path of the output r1cs file [default: out.r1cs]
        --stdlib-path <PATH>   Path to the standard library [env: ZOKRATES_STDLIB=] [default:
                                /home/admin/.zokrates/stdlib]

zokrates-setup
Performs a trusted setup for a given constraint system

USAGE:
    zokrates setup [FLAGS] [OPTIONS]

FLAGS:
    -h, --help   Prints help information
    -V, --version   Prints version information
        --verbose   Verbose mode

OPTIONS:
    -b, --backend <backend>   Backend to use [default: ark] [possible values: bellman, ark]
    -e, --entropy <entropy>   User provided randomness
    -i, --input <FILE>   Path of the binary [default: out]
    -p, --proving-key-path <FILE>   Path of the generated proving key file [default: proving.key]
    -s, --proving-scheme <proving-scheme>   Proving scheme to use in the setup [default: g16] [possible values: g16,
                                             gm17, marlin]
    -u, --universal-setup-path <FILE>   Path of the universal setup file for universal schemes [default:
                                             universal_setup.dat]
    -v, --verification-key-path <FILE>   Path of the generated verification key file [default: verification.key]

zokrates-compute-witness
Calculates a witness for a given constraint system

USAGE:
zokrates compute-witness [FLAGS] [OPTIONS]

FLAGS:
--abi Use ABI encoding. Arguments are expected as a JSON object as specified at zokrates.github.io/toolbox/abi.html#abi-input-format
-h, --help Prints help information
--json Write witness in a json format for debugging purposes
--stdin Read arguments from stdin
-V, --version Prints version information
--verbose Verbose mode

OPTIONS:
-s, --abi-spec <FILE> Path of the ABI specification [default: abi.json]
-a, --arguments <arguments>... Arguments for the program's main function, when not using ABI encoding. Expects a space-separated list of field elements like `-a 1 2 3`
--circom-witness <FILE> Path of the output circom witness file [default: out.wtns]
-i, --input <FILE> Path of the binary [default: out]
-o, --output <FILE> Path of the output witness file [default: witness]

zokrates-generate-proof
Calculates a proof for a given constraint system and witness

USAGE:
    zokrates generate-proof [FLAGS] [OPTIONS]

FLAGS:
    -h, --help   Prints help information
    -V, --version   Prints version information
        --verbose   Verbose mode

OPTIONS:
    -b, --backend <backend>   Backend to use [default: ark] [possible values: bellman, ark]
    -e, --entropy <entropy>   User provided randomness
    -i, --input <FILE>   Path of the binary [default: out]
    -j, --proof-path <FILE>   Path of the JSON proof file [default: proof.json]
    -p, --proving-key-path <FILE>   Path of the proving key file [default: proving.key]
    -s, --proving-scheme <FILE>   Proving scheme to use to generate the proof [default: g16]
[possible values: g16,
                                gm17, marlin]
    -w, --witness <FILE>   Path of the witness file [default: witness]

zokrates-export-verifier
Exports a verifier as Solidity smart contract

USAGE:
    zokrates export-verifier [FLAGS] [OPTIONS]

FLAGS:
    -h, --help   Prints help information
    -V, --version   Prints version information
        --verbose   Verbose mode

OPTIONS:
    -i, --input <FILE>   Path of the verification key [default: verification.key]
    -o, --output <FILE>   Path of the output file [default: verifier.sol]

zokrates-verify
Verifies a given proof with the given verification key

USAGE:
    zokrates verify [FLAGS] [OPTIONS]

FLAGS:
    -h, --help   Prints help information
    -V, --version   Prints version information
        --verbose   Verbose mode

OPTIONS:
    -b, --backend <backend>   Backend to use [default: ark] [possible values: bellman, ark]
    -j, --proof-path <FILE>   Path of the JSON proof file [default: proof.json]
    -v, --verification-key-path <FILE>   Path of the generated verification key file [default: verification.key]