Circuit

From a user perspective, a circuit is like a program, which can be written in a programming language like Go. However, internally a circuit is a constraint system; that is, a list of constraints which have an algebraic form.

ForGroth16 , a constraint looks like(∑ i a i x i) (∑ i b i y i) = ∑ i c i z i (\sum_ia_ix_i)(\sum_ib_iy_i)=\sum_ic_iz_i (∑ i

a i x i) (∑ i

b i y i)

= ∑ i

c i z i wherea , b , c a,b,c a ,

b ,

c are constants andx , y , z x,y,z x ,

y ,

z are variables which depend on the secret inputs known by a prover.

Translating a circuit, written withgnark API to such a constraint system is called the "arithmetization" of a circuit.

An important point is that every component of a constraint (variables, inputs and constants) live inF p \mathbb{F}_p F p , a finite field of characteristicp p p . To write a circuit which contains a reasonable number of constraints, it is important to work on the fieldF p \mathbb{F}_p F p , so that the field in which the circuits variables live is the same as the field on which the constraint system reasons.

On the other hand, a circuit reasoning on variables which live inF r \mathbb{F}_r F r wherer ≠ p r\neq p r

?

p , has a high number of constraints because of the algebraic constraints needed to emulate the arithmetic modulor r r on a field of characteristicp p p .

Finally, the number of constraints in a circuit is limited; you cannot write arbitrarily large circuits. For example, using Groth16 on BN254, you cannot exceed ~250 M 250M 250 M constraints. Edit this page Last updatedonMar 2, 2023 byaybehrouz Previous zk-SNARK Next Proving schemes and curves