Prove schemes and curves

gnark supports two proving schemesGroth16 andPlonK . These schemes can be instantiated with any of the following elliptic curves:BN254 ,BLS12-381 ,BLS12-377 ,BLS24-315 ,BW6-633 orBW6-761 .

An ID is supplied tognark to choose the proving scheme and the instantiating curve.

Choosing a proving system

Quick system guide Groth16 PlonK trusted1 setup circuit-specific universal ⋆⋆ proof length ⋆⋆⋆ ⋆ prover work ⋆⋆ ⋆ verifier work ⋆⋆ ⋆ Groth16 is best suited when an application needs to generate many proofs for the same circuit (for instance a single logic computation) and performance is critical, while PlonK is best suited when it needs to handle many different circuits (for example different arbitrary business logics) with reasonably fast performance.

Groth16

Groth16 is a circuit-specific preprocessing general-purpose zk-SNARK construction. It has become a de-facto standard used in several blockchain projects due to the constant size of its proof, and its appealing verifier time. On the downside, Groth16 needs a circuit-specific trusted setup for its preprocessing phase.

info We recommend this shortexplaination of Groth16 .

Some projects that use Groth16 include ZCash, Loopring, Hermez, Celo, and Filecoin.

PlonK

PlonK is a universal preprocessing general-purpose zk-SNARK construction.

It's a proving scheme with a preprocessing phase that can be updated, and has a short and constant verification time. On the downside, PlonK proofs are bigger and slower to generate compared to Groth16.

info For more information, we recommend thisPlonk paper , andthis Plonk article .

Some projects that use PlonK include Aztec, ZKSync, and Dusk. note PlonK comes in different version according to the chosen polynomial commitment scheme. For example:

- KZG
- Pedersen-Bulletproofs
- FRI-based
- DARK

There are also versions for the prover/verifier tradeoff. For example "fast-prover-but-slow-verifier" or "slow-prover-but-fast-verifier" settings.

There are also different optimizations. For example:

- TurboPlonK
- ,
- Plookup
-).

Currently,gnark supports PlonK with KZG polynomial commitment.

Choosing an elliptic curve

Both Groth16 and PlonK (with KZG scheme) need to be instantiated with an elliptic curve.gnark supports six elliptic curves: BN254, BLS12-381, BLS12-377, BW6-761, BLS24-315, and BW6-633. All these curves are defined over a finite fieldF p \mathbb{F}_p F p and have an equation of the formy 2 = x 3 + b y^2=x^3+b y 2

= x 3

+ b (b ∈ F p b\in \mathbb{F}_p b

∈ F p).

To work with Groth16 and PlonK, the curves must:

- Be secure, for proof soundness
- Be pairing-friendly, for proof verification

- Have a highly 2-adic subgroup order, for efficient proof generation.

info BN254 is used in Ethereum 1.x, BLS12-381 in Ethereum 2.0, ZCash Sapling, Algorand, Dfinity, Chia, and Filecoin, and BLS12-377/BW6-761 in Celo, Aleo and EY.

BN254 and BLS12-381 curves

For applications that target Ethereum 1.x mainnet, BN254 is the only supported curve. EIPs for other curves exist but are not integrated yet:

- EIP-2539
- EIP-2537
- EIP-3026
- .

For applications that target Ethereum 2.0, use BLS12-381.

For platform-agnostic applications, the choice requires a tradeoff between performance (BN254) and security (BLS12-381). We recommend choosing BLS12-381 as it is more secure, still fast enough to be practical, but slower than BN254.

BLS12-377 and BW6-761 curves

Applications that require one-layer proof composition (a proof of proofs) cannot use BN254 or BLS12-381 as they are quite inefficient for this purpose.

In fact, such an application needs a pair (E 1 , E 2 E_1, E_2 E 1 ,

E 2) of elliptic curves that:

- Are secure, for proof soundness
- Are pairing-friendly, for proof verification
- Have a highly 2-adic subgroup order, for efficient proof generation.
- E
- 2
- E_2
- E
- 2
-
- has a subgroup order equal toE
- 1
- E_1
- E
- 1
-
- 's field characteristic, for efficient proof composition.

BLS12-377 and BW6-761 curves satisfy these conditions, while having fast implementations.

note GivenE 1 E_1 E 1 must have a highly 2-adic field characteristic, BLS12-381 cannot be used. info Some benchmarks and comparisons of third-parties implementations againstgnark-crypto .

Some applications that use one-layer proof composition include ZEXE, Celo, Aleo, and Zecale.

BLS24-315 and BW6-633 curves

In Groth16, elliptic curve operations take place in three different groups:G 1 G_1 G 1 ,G 2 G_2 G 2 andG T G_T G T , whereas in PlonK (with KZG) operations take place only inG 1 G_1 G 1 andG T G_T G T . While BN254, BLS12-381 and BLS12-377 are optimized for all the three groups, BLS24-315 is better optimized forG 1 G_1 G 1 only while still competitively optimized forG T G_T G T . Moreover, it comes in a 2-chain setting with BW6-633 to enable PlonK one-layer proof composition efficiently.

In summary, (BLS24-315, BW6-633) is a pair of elliptic curves that:

- Are secure, for proof soundness.
- Are pairing-friendly, for proof verification.
- Are optimized for KZG-based SNARKs (for example, PlonK).
- Have a highly 2-adic subgroup order, for efficient proof generation.
- For efficient proof composition, BW6-633 has a subgroup order equal to BLS24-315's field characteristic.Edit this page Last updatedonMar 2, 2023 byaybehrouz Previous Circuits and constraint systems Next Tutorials
- Choosing a proving system