

Updating zkSNARK Public Parameters

Vitalik Buterin
vitalik@ethereum.org

Christian Reitwießner
chris@ethereum.org

Abstract

1 GGPR

Remember that in the QSP we are given polynomials $v_0, \dots, v_m, w_0, \dots, w_m$, a target polynomial t (of degree at most d) and a binary input string u . The prover finds $a_1, \dots, a_m, b_1, \dots, b_m$ (that are somewhat restricted depending on u) and a polynomial h such that

$$th = (v_0 + a_1v_1 + \dots + a_mv_m)(w_0 + b_1w_1 + \dots + b_mw_m).$$

In the previous section, we already explained how the common reference string (CRS) is set up. We choose secret numbers s and α and publish

$$E(s^0), E(s^1), \dots, E(s^d) \quad \text{and} \quad E(\alpha s^0), E(\alpha s^1), \dots, E(\alpha s^d)$$

Because we do not have a single polynomial, but sets of polynomials that are fixed for the problem, we also publish the evaluated polynomials right away:

- $E(t(s)), E(\alpha t(s)),$
- $E(v_0(s)), \dots, E(v_m(s)), E(\alpha v_0(s)), \dots, E(\alpha v_m(s)),$
- $E(w_0(s)), \dots, E(w_m(s)), E(\alpha w_0(s)), \dots, E(\alpha w_m(s)),$

and we need further secret numbers β_v, β_w, γ (they will be used to verify that those polynomials were evaluated and not some arbitrary polynomials) and publish

- $E(\gamma), E(\beta_v\gamma), E(\beta_w\gamma),$
- $E(\beta_vv_1(s)), \dots, E(\beta_vv_m(s))$
- $E(\beta_wv_1(s)), \dots, E(\beta_wv_m(s))$
- $E(\beta_vt(s)), E(\beta_wt(s))$

This is the full common reference string. In practical implementations, some elements of the CRS are not needed, but that would complicate the presentation.