I want to be able to verify signatures created by smart meters. Unfortunately, their signature algorithm doesn't use curve ALT_BN_128 but BrainpoolP256r1. I wrote this verification algorithm that requires an inversion, an addition of two curve points, and two scalar multiplications of curve points. The problem is especially the gas cost of the latter.

I wrote a contract for these operations where the scalar multiplication is implemented via a simple double-and-add algorithm. Unfortunately, a scalar multiplication using this method costs between 84'000 and 19'000'000 gas. Originally, I didn't want to write my own contract for that but the only Solidity implementation I was able to find was ECops.sol

by orbs-network

on Github. Scalar multiplication using this contract is much cheaper (about 623'000 gas), but then there's the minor disadvantage that the results are wrong.

Is there a better solidity implementation out there where I can just plug the right curve parameters in?

I have now posted this question on Ethereum SE too. I can't post a link to it because I'm only allowed two use two per post. But I used the same title, so if you google "stackexchange Cheap EC Operations on Unsupported Curves", it should pop up.