Efficient verifiable hash–to–curve implementation

Vadym Fedyukovych
Infopulse
Kiev Ukraine

Lyudmila Kovalchuk
National Technical University of Ukraine
Kiev Ukraine

***Abstract*—Hash-to-curve is an essential part of a proof–of–stake protocol. We report a verifiable implementation of WB19 hash algorithm featuring low gate complexity of SNARK-based verification.**

***Index Terms*—Hash to curve, zk-SNARK gadget**

I. Introduction

Electric power is the scarce resource defining mining competition in proof-of-work systems like Bitcoin. Proof-of-stake systems suggest alternative motivation replacing access to cheap power with risk of loosing deposit in case of not following rules. Verifiable actions of block miners become important for proof-of-stake systems, in particular leader selection with BLS signature scheme and hash-to-curve algorithm. We refer to Ouroboros [1] protocol design report for full description.

An efficient hash algorithm was introduced by Wahby–Boneh[1] [2], and it's implementation was discussed by Mercer[2].

Gadget is a C++ class in libsnark parlance that hides complexity of the circuit to be verified, providing only a high-level interface like assign the witness and generate R1CS constraints. We refer to accompanying paper[3] for general description of SNARKs and Groth16 proof system.

We report a libsnark gadget implementation of Wahby–Boneh hash algorithm, suitable for public verification of leader selection result. In particular, we produce a compact Groth16 [3] SNARK proof that can be published on a public blockchain. We also provide a working example of using libsnark gadget to simplify software engineering. This result was supported by Horizen.

II. Boneh–Wahby hash

Consider a curve defined over a prime field $\mathbb{F}_q$:

$$y^2 = g(x) = x^3 + ax + b \tag{1}$$

Let $\xi \in \mathbb{F}_q$ be a non-residue. For a message $m \in \mathbb{F}_q$ let $u = \xi m^2$. Starting from $g(ux) = u^3 g(x)$, one would conclude that just one out of $(g(x), g(ux))$ is a square.

$$(ux)^3 + a(ux) + b = u^3(x^3 + ax + b) \tag{2}$$

$$ax(u^3 - u) + b(u^3 - 1) = 0 \tag{3}$$

$$x(u) = \frac{-b}{a}\left(1 + \frac{1}{u^2 + u}\right) \tag{4}$$

To produce a point on the elliptic curve, one would hash the message with a SNARK–friendly algorithm like Poseidon [4] producing an intermediate u, and then calculate proper x coordinate of the point with equation (4).

III. Libsnark gadgets in practice

[This section will be updated with proper source code]

IV. Efficiency analysis

WB19 hash–to–curve gadget facilitates implementation of proof–of–stake systems providing state-of-the-art efficiency.

References

[1] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," Cryptology ePrint Archive, 2016. [Online]. Available: https://eprint.iacr.org/2016/889

[2] R. S. Wahby and D. Boneh, "Fast and simple constant-time hashing to the BLS12-381 elliptic curve," Cryptology ePrint Archive, 2019. [Online]. Available: https://eprint.iacr.org/2019/403

[3] J. Groth, "On the size of pairing-based non-interactive arguments," Cryptology ePrint Archive, 2016. [Online]. Available: https://eprint.iacr.org/2016/260

[4] L. Grassi, D. Kales, D. Khovratovich, A. Roy, C. Rechberger, and M. Schofnegger, "Starkad and poseidon: New hash functions for zero knowledge proof systems," Cryptology ePrint Archive, 2019. [Online]. Available: https://eprint.iacr.org/2019/458

[1] CHES, https://ches.iacr.org/2019/program.shtml

[2] #zk0x04, https://www.youtube.com/watch?v=qWRUPzm3qPY, 18 Nov'19

[3] Private Sudoku solution verification from polynomial set representation