Hi guys,

I am very sorry for spamming. However I wrotea new article, which is a remarkable improvement ofmy previous topic. In my opinion, this is the most useful result for blockchain I have ever obtained. Please read the abstract:

Let \mathbb{F}_{!q}

be a finite field and E_b!: y^2 = x^3 + b

be an ordinary (i.e., non-supersingular) elliptic curve (of j

-invariant 0

) such that \sqrt{b} \in \mathbb{F}_{!q}

and q \not\equiv 1 \: (\mathrm{mod} \ 27)

. For example, these conditions are fulfilled for the group \mathbb{G}_1

of the curves BLS12-381 (b=4

) and BLS12-377 (b=1

) and for the group \mathbb{G}_2

of the curve BW6-761 (b=4

). The curves mentioned are a de facto standard in the real world pairing-based cryptography at the moment. This article provides a new constant-time hash function H!: {0,1}^* \to E_b(\mathbb{F}_{!q})

indifferentiable from a random oracle. Its main advantage is the fact that H

computes only one exponentiation in \mathbb{F}_{!q}

. In comparison, the previous fastest constant-time indifferentiable hash functions to E_b(\mathbb{F}_{!q})

compute two exponentiations in \mathbb{F}_{!q}

. In particular, applying H

to the widely used BLS multi-signature with m

different messages, the verifier should perform only m

exponentiations rather than 2m

ones during the hashing phase.

For your taste, is this an important achievement ? Please let me know about a collaboration if one of companies or startups wants to use my hash function in its products. In the near future, I will also try to generalize this hash function to the more difficult case \sqrt{b} \not\in \mathbb{F}_{!q}

in order to be applicable to all pairing-friendly curves.

Best regards.