Hi guys,

I wrote a new article

Hashing to elliptic curves y^2 = x^3 + b provided that b is a quadratic residue.pdf (244.9 KB)

In my opinion, this is the most useful result for applied cryptography I have ever obtained. Please read its abstract:

Let \mathbb{F}_{!q}

be a finite field and E_b!: y_0^2 = x_0^3 + b

be an ordinary elliptic \mathbb{F}_{!q}

-curve of j

-invariant 0

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

. In particular, this condition is fulfilled for the curve BLS12-381 and for one of sextic twists of the curve BW6-761 (in both cases b=4

). These curves are very popular in pairing-based cryptography. The article provides an efficient constant-time hashing h!: \mathbb{F}*{!q} \to E_b(\mathbb{F}*{!q})

of an absolutely new type for which at worst #\mathrm{Im}(h) \approx q/6

. The main idea of our hashing consists in extracting in \mathbb{F}_{!q}

a cubic root instead of a square root as in the well known (universal) SWU hashing and in its simplified analogue. Besides, the new hashing can be implemented without quadratic and cubic residuosity tests (as well as without inversions) in \mathbb{F}_{!q}

. Thus in addition to the protection against timing attacks, h

is much more efficient than the SWU hashing, which generally requires to perform two quadratic residuosity tests in \mathbb{F}_{!q}

. For instance, in the case of BW6-761 this allows to avoid at least approximately 2 !\cdot! 761 \approx 1500

field multiplications.

In your opinion, is this a useful result ? Please let me know in order to collaborate if any of companies or startups wants to use my hashing in its products. In this case I can implement it in one of programming languages.

Best regards.